A Reinforcement Learning-Based Decision System For Electricity Pricing Plan Selection by Smart Grid End Users

Lu, Tianguang ; Chen, Xinyu; Mcelroy, Michael B.; Nielsen, Chris P. ; Wu, Qiuwei; Ai, Qian

Published in:
IEEE Transactions on Smart Grid

Link to article, DOI:
10.1109/TSG.2020.3027728

Publication date:
2021

Document Version
Peer reviewed version

A Reinforcement Learning-Based Decision System For Electricity Pricing Plan Selection by Smart Grid End Users

Tianguang Lu, *Member, IEEE,* Xinyu Chen, *Member, IEEE,* Michael B. McElroy, Chris P. Nielsen, Qiuwei Wu, *Senior Member, IEEE* and Qian Ai, *Senior Member, IEEE*

***Abstract*—With the development of deregulated retail power markets, it is possible for end users equipped with smart meters and controllers to optimize their consumption cost portfolios by choosing various pricing plans from different retail electricity companies. This paper proposes a reinforcement learning-based decision system for assisting the selection of electricity pricing plans, which can minimize the electricity payment and consumption dissatisfaction for individual smart grid end user. The decision problem is modeled as a transition probability-free Markov decision process (MDP) with improved state framework. The proposed problem is solved using a Kernel approximator-integrated batch Q-learning algorithm, where some modifications of sampling and data representation are made to improve the computational and prediction performance. The proposed algorithm can extract the hidden features behind the time-varying pricing plans from a continuous high-dimensional state space. Case studies are based on data from real-world historical pricing plans and the optimal decision policy is learned without a priori information about the market environment. Results of several experiments demonstrate that the proposed decision model can construct a precise predictive policy for individual user, effectively reducing their cost and energy consumption dissatisfaction.**

***Index Terms*—Smart grid end user, decision system, electricity market, value-based Q learning, demand response**

NOMENCLATURE

Indices and Sets

t	Time index of month
n	Index of week in a month
i	Electricity retail plan (ERP) index
l	Transition sample index
k	Iteration index
$\mathcal{T}$	Set of t
$\mathcal{I}$	Set of i
$\mathcal{F}$	Set of transition samples
$\mathcal{S}/\mathcal{A}$	Set of states/actions
$\mathcal{C}^{tdu}/\mathcal{C}_i/\mathcal{C}_i^{ad}$	Set of $c^{tdu}/c_i/c_i^{ad}$

Variables

e_t^c	End user's monthly energy consumption at time t
e_t^d	End user's monthly cumulative energy demand at time t
$c_{i,t}^e$	Energy unit rate of ERP i at time t
c_t^{tdu}	Transmission and distribution utility (TDU) unit rate at time t
s_t	System state at time t
a_t	Action taken at time t
r_t	Reward received at time t
s_t'	Next state reached from state s_t by taking action a_t
π	Decision policy
m_t	Time-dependent mark variable for time t

Parameters

Δt	Length of time t
c_{fix}^{tdu}	TDU fixed rate
c_{add}^{tdu}	Additional TDU fee during peak demand
p^{tdu}	Power threshold of peak demand
c_i^{ad}	System administration fixed rate of ERP i
c^b	Monthly base charge
β	Sensitivity factor of user's disuitility
α	Load shift factor
e_t^{new}	Newly generated monthly energy demand at time t
$e_{t,n}^{new}$	Newly generated weekly energy demand in week n of month t
e_t^{rw}	Real-world monthly energy consumption of individual end user at time t
λ	Weight factor of user's satisfaction and billing cost
e^{max}	Upper limit of monthly energy consumption
h	Time step length of observation memory
ρ	Smoothness factor of Kernel approximator
c_t^{aia}	Monthly billing cost caused by the proposed artificial intelligence agent (AIA) at time t
c_t^{hum}	Real-world average monthly billing cost at time t
c_t^{opt}	Monthly optimized billing cost at time t

Corresponding authors: Xinyu Chen and Michael B. McElroy

T. Lu is with Harvard University, Cambridge, MA 02138, USA and also with Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: tlu@seas.harvard.edu)

X. Chen is with the School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan 430074, China (xchen@seas.harvard.edu)

M. B. McElroy, and Chris P. Nielsen are with Harvard University, Cambridge, MA 02138, USA (e-mail: xchen@seas.harvard.edu; mbm@seas.harvard.edu; nielsen2@fas.harvard.edu)

Q. Wu is with the Centre for Electric Power and Energy, Department of Electrical Engineering, Technical University of Denmark, Lyngby, 2800, Denmark (e-mail: qw@elektro.dtu.dk)

Q. Ai is with Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: aiqian@sjtu.edu.cn).

I. Introduction

With the rapid development of deregulated retail electricity markets and smart grid technologies, utility companies are able to compete equally in the retail market, offering different time-varying pricing plans [1]. This gives the end users, such as residential and small-scale industrial/commercial users, opportunities to choose and change freely among various pricing plans to reduce their electricity costs . Currently, several competitive retail markets have already been established. For example, the Public Utility of Commission of Texas in the U.S., has established the "Power-to-Choose" platform for different retailers to sell their electricity retail plans to the end users [2]. "Reliant" is another platform in Texas, where different companies can trade their residential or business utility service with more than 1.5 million Texans [3]. In these examples, an end user chooses and purchases a time-varying pricing plan during a trading window. The chosen plan is not allowed to be changed until the next trading window, i.e., the next month. The end user then pays the utility bill according to the disclosed prices after one month of consumption and decides to continue this plan or change to another plan. However, individual end users are confronted with several unprecedented challenges when making these decisions [4]. First, the relationship between previous and future plans is hard to understand due to many uncertainties, which mainly come from the complexity of retailer market behavior. Second, the variety of pricing plans offered by diverse companies is hard to compare and analyze.

To resolve the above-mentioned challenges, an end-to-end decision system is proposed for assisting the individual utility users in making decisions on their retail pricing plans. This work is inspired by several existing decision systems in other fields, where data mining technologies are usually leveraged to improve the prediction and economic performance [5]. In health informatics, a clinical decision support system is designed to provide physicians and other health professionals with clinical decision support based on a big dataset of patients [6]. In a specific example of transportation informatics, the Canadian National Railway system, which tests its equipment on a regular basis using a decision support system, manages to decrease the incidence of derailments [7]. In addition to resolving the challenges, this work is driven by the following additional motivations. One is that in most cases, a user will not devote much time going though a large amount of periodically changing pricing plans and will just select a plan without enough decision support, which often results in higher payments and lower satisfaction rates. The other is that this decision making process may be considered as a potential indicator of demand response.

Recently, there has been growing interest in adopting reinforcement learning algorithms to the smart grid, but no actual adoption in end utility user decision making has occurred. In [8], a fitted Q-iteration-based demand response of thermostatically controlled loads was proposed to reduce the electricity cost, where a convolutional neural network was used as a function approximator to capture the underlying features of non-observable mass temperature states. Reference [9] developed a dynamic pricing and energy consumption scheduling model for utility service providers. In particular, an approximate state-based reinforcement learning tool was designed with virtual experience to allow the provider to learn a cost-efficient strategy. The authors presented a batch reinforcement learning-based charging approach in [10] for plug-in electric vehicles to optimize daily charging. An optimal charging policy was trained by the presented learning algorithm from a batch of transition samples using a Bayesian neural network to predict the electricity prices. A Q-learning-based heuristic charging strategy for a electric vehicle fleet in a day-ahead electricity market was studied in [11], where the unknown charging flexibility of electric vehicles was learned using a batch mode Q-learning algorithm to precisely predict a consumption plan. When applied to the proposed system, most existing reinforcement learning methods have some level of prediction and computation issues due to three main reasons. First, the training data set is sequential, where samples between consecutive time sections are correlated and non-independent identical distributed. Second, the interaction between the end user and the retail electricity market is so complicated that the training data set is hard to sample from the environment. Third, the state space of the market environment is large, which increases the computational cost.

To deal with the above-mentioned limitations, we formulate a modified model-free Q-learning algorithm to cater to the proposed pricing-plan decision problem and improve the prediction and computational performance. To avoid data distortion of the time-related pricing plans, a historic observation-based state vector is introduced by adding previous price information as an indicator of price fluctuation. To efficiently sample the training data, the user's consumption satisfaction and cost are modeled as an optimization problem, whose solution can be obtained easily as a part of the training data sample. To reduce the computational burden, the dimensionality of the state space is reduced using data processing methods. Moreover, the transition sample set is enlarged by additional tuples to increase the convergence speed of the algorithm and improve the precision of decision prediction. Case studies from a practical retail electricity market platform prove the effectiveness of the proposed methodology for making accurate prediction of future pricing and proper selection of plans.

The key contributions of this paper are threefold:

1) An innovative electricity-pricing-plan decision support system for smart grid end users is proposed to for application in a promising retail market environment where multiple retailers offer different market-indexed time-varying pricing plans.

2) A modified reinforcement learning algorithm combined with sampling and data processing methods is proposed to cater to the proposed complex decision problem and improve the prediction and computational performance.

3) Detailed comparisons and analyses are conducted based on a real-world dataset to explore features of decision-making behavior and the proposed batch Q learning algorithm.

The rest of this paper is organized as follows. Section II introduces the proposed decision system architecture. The Markov decision process problem formulation is presented in Section III. The proposed reinforcement learning algorithm is explained in Section IV. Section V analyzes numerical results, followed by concluding remarks in Section VI.

II. Retail Pricing Plan Decision System

We assume that a retail pricing plan decision system is able to select the optimal electricity retail plan (ERP) over the next-consumption period for individual electricity user based on current environment information. The selected plan minimizes the user's future dissatisfaction and cost. The user can adjust several settings in the decision system to determine a cost-saving plan that best fits the user's consumption patterns and satisfaction.

As an artificial intelligence agent (AIA), the decision system achieves a decision policy π through reinforcement learning (RL) and uses this policy to trade with multiple retail electricity companies. RL can model a complex system where an agent needs to imitate human brains to make decisions (e.g., choose

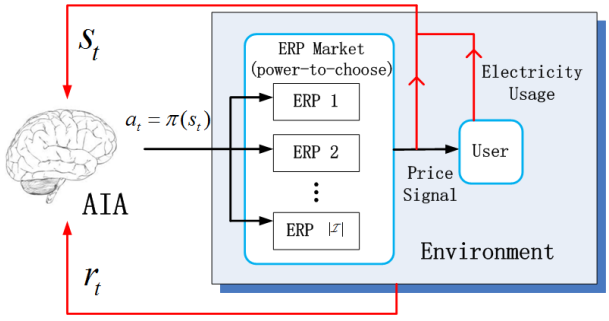


Fig. 1. Decision support system and its interaction with the environment.

an ERP) based on interactions with a complicated environment (e.g., dependently time-changing market of electricity retailers and energy consumptions). Fig. 1 illustrates the interactions among the AIA, individual end user, and retail market, where the red arrows represent different feedbacks from the environment. In each operation period $t \in \mathcal{T} \equiv \{t : t = 1 \dots T\}$, the AIA executes an action a_t, which is to select an ERP from the retail market. Based on the selected ERP, the corresponding retailer charges the end user an electricity bill, which affects the end user's electricity consumption. The environment information, such as service charges of ERPs and consumption patterns of the end user, is recorded in the system state s_t as one of the feedbacks on the AIA's action. The other feedback is the reward r_t, which is indicated by the cost and user's disutility. The decision policy of the AIA is learned by interacting with the environment to maximize the future reward.

The modeling of the AIA's decision-making and learning process is detailed in Sections III and IV, respectively.

A. Configuration of Electricity Retail Plan

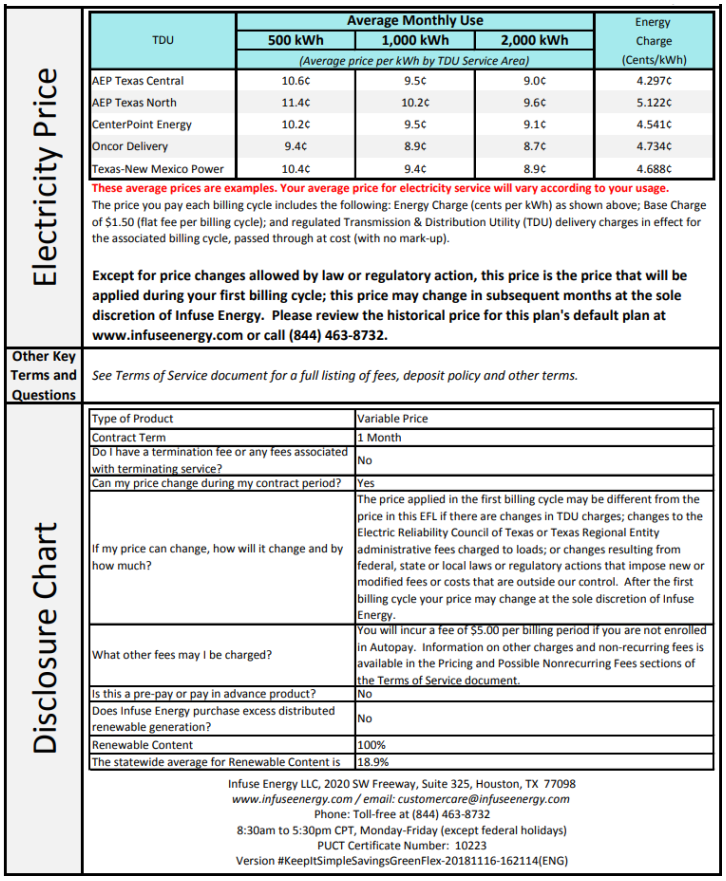
The choices of ERPs by end users are heavily determined by the plan configurations. Currently more than 1000 retail plans from different retail electricity companies of Texas are published on the "Power-to-Choose" website platform [12]. A customer can access detailed information on each plan available in its region on the platform by inputting its zip code.

A typical time-varying ERP of "Power-to-Choose" is shown in Fig. 2, the configuration of which is characterized by the following features.

1) Transmission and distribution utility (TDU) charges: The retailers charge TDU fees that are included in ERPs, on behalf of the six transmission/distribution companies providing transmission and distribution services in Texas. TDU charges can be calculated by the following equation:

$$\begin{cases} \text{if} \quad e_t^c/\Delta t > p^{tdu}: \\ \psi_t^{tdu}(e_t^c) = c_{fix}^{tdu} + c_t^{tdu} e_t^c + c_{add}^{tdu}(e_t^c/\Delta t - p^{tdu}) \\ \text{if} \quad e_t^c/\Delta t \le p^{tdu}: \\ \psi_t^{tdu}(e_t^c) = c_{fix}^{tdu} + c_t^{tdu} e_t^c \end{cases} \tag{1}$$

where c_t^{tdu} is determined by delivery and maintenance service provided by utility companies. The third item in the above equation represents the additional TDU fee caused by a harder dispatch work of the grid during peak demand. When a customer's power demand measured in kilowatts exceeds the threshold of p^{tdu}, the additional TDU fee is charged. More segments can be used to describe different TDU charges based on multiple intervals of energy consumption (in Fig. 2, for example, 0-500 kWh, 500-1000 kWh, and 1000-2000 kWh). In the case study, all users' energy consumptions are in the same interval, and therefore equation (1) is practical enough to build model for the case study.



Electricity Price

TDU	Average Monthly Use 500 kWh (Average price per kWh by TDU Service Area)	1,000 kWh	2,000 kWh	Energy Charge (Cents/kWh)
AEP Texas Central	10.6¢	9.5¢	9.0¢	4.297¢
AEP Texas North	11.4¢	10.2¢	9.6¢	5.122¢
CenterPoint Energy	10.2¢	9.5¢	9.1¢	4.541¢
Oncor Delivery	9.4¢	8.9¢	8.7¢	4.734¢
Texas-New Mexico Power	10.4¢	9.4¢	8.9¢	4.688¢

These average prices are examples. Your average price for electricity service will vary according to your usage.

The price you pay each billing cycle includes the following: Energy Charge (cents per kWh) as shown above; Base Charge of $1.50 (flat fee per billing cycle); and regulated Transmission & Distribution Utility (TDU) delivery charges in effect for the associated billing cycle, passed through at cost (with no mark-up).

Except for price changes allowed by law or regulatory action, this price is the price that will be applied during your first billing cycle; this price may change in subsequent months at the sole discretion of Infuse Energy. Please review the historical price for this plan's default plan at www.infuseenergy.com or call (844) 463-8732.

Other Key Terms and Questions

See Terms of Service document for a full listing of fees, deposit policy and other terms.

Disclosure Chart

Type of Product	Variable Price
Contract Term	1 Month
Do I have a termination fee or any fees associated with terminating service?	No
Can my price change during my contract period?	Yes
If my price can change, how will it change and by how much?	The price applied in the first billing cycle may be different from the price in this EFL if there are changes in TDU charges; changes to the Electric Reliability Council of Texas or Texas Regional Entity administrative fees charged to loads; or changes resulting from federal, state or local laws or regulatory actions that impose new or modified fees or costs that are outside our control. After the first billing cycle your price may change at the sole discretion of Infuse Energy.
What other fees may I be charged?	You will incur a fee of $5.00 per billing period if you are not enrolled in Autopay. Information on other charges and non-recurring fees is available in the Pricing and Possible Nonrecurring Fees sections of the Terms of Service document.
Is this a pre-pay or pay in advance product?	No
Does Infuse Energy purchase excess distributed renewable generation?	No
Renewable Content	100%
The statewide average for Renewable Content is	18.9%

Infuse Energy LLC, 2020 SW Freeway, Suite 325, Houston, TX 77098
www.infuseenergy.com / email: customercare@infuseenergy.com
Phone: Toll-free at (844) 463-8732
8:30am to 5:30pm CPT, Monday-Friday (except federal holidays)
PUCT Certificate Number: 10223
Version #KeepItSimpleSavingsGreenFlex-20181116-162114(ENG)

Fig. 2. A sample of the ERP in "Power-to-Choose" disclosed on 11/6/2018.

2) System administration fee: Electric Reliability Council of Texas charges this fee with an annual fixed rate of c_i^{ad}, which is usually in the range of [0.05, 0.08] $/kWh.

3) Energy charge: As the main body of the ERP, energy charges have two categories of daily rates and three types of monthly rates. For daily rates, the time-of-use (TOU) rate differs in peak and non-peak hours, while the non-TOU rate is fixed during the day. Each TOU-based plan and non-TOU-based plan can integrate the following three monthly rates: (a) a variable rate, which is decided by electricity retailers according to the retail market; (b) a fixed rate, which remains constant over the contract term; (c) an index rate, which is determined by the last settled price of the natural gas. For (a) and (c), most ERPs allow a contract term of at least one month, which means once an end user chooses a variable or indexed plan, the user is required to keep this plan for at least one month before it can change to another plan.

Among the above energy charges, (a) and (c) are the focus of this paper and they use the variable of $c_{i,t}^e$. The implication of renewable content (i.e., renewable penetration) on $c_{i,t}^e$ is complicated, including many factors such as tariff, company policy, etc.

4) Base charge and minimum usage charge: Base charge is a flat fee per billing cycle, and the minimum usage charge is applied when the user's energy consumption for a month is less than the contracted minimum consumption threshold. We assume that this threshold is satisfied in any contract term.

According to the above configuration, the monthly cost of ERP i in month t can be calculated below:

$$\psi_{i,t}^{erp}(e_t^c) = \psi_t^{tdu}(e_t^c) + c_{i,t}^e e_t^c + c_i^{ad} e_t^c + c^b \tag{2}$$

B. User's Energy Consumption Model

The growing implementation of advanced metering infrastructures (AMI) and smart controllers makes it possible to measure and manage an end user's energy consumption based on the user's preference and the price signal [13]. Modeling the user's energy consumption helps the AIA to learn a better decision policy.

In each month, the user has a cumulative energy demand $e_t^d \in \mathcal{E}$. This demand includes both the electricity the user wants to consume in month t and unsatisfied demands in previous months. When the user actually consumes e_t^c in month t, a part of e_t^d is offset by e_t^c, while the residual, $e_t^d - e_t^c$, results in user dissatisfaction, which is defined by a disutility function $\varphi : \mathbb{R}_+ \to \mathbb{R}$. Here a novel disutility function is introduced as follows to represent most users' monthly consumption preference [14]–[16]:

$$\varphi(e_t^c) = e^{\beta\left(1-e_t^c/e_t^d\right)} - 1, \beta > 0 \tag{3}$$

where β reflects the sensitivity of the user's disutility to energy consumption e_t^c. When consuming the same e_t^c, the user with a smaller β has a higher disutility. Specifically, when $e_t^c < e_t^d$, $\varphi > 0$, i.e., the user is dissatisfied with the unsatisfied cumulative demand; the value of the disutility function increases more and more dramatically as e_t^c decreases. When $e_t^c > e_t^d$, $\varphi < 0$, i.e., the user is satisfied with the extra energy consumption; as e_t^c increases, the negative disutility value converges to a saturation level of satisfaction.

Note that using the proposed disutility function, precise prediction and cost reduction (detailed in Section V) can be obtained, which demonstrates the effectiveness of the function.

Traditional user consumption is not flexible on a monthly basis. Smart meters and controllers can shift power demand from one month to the other with monthly rescheduling [17], [18]. To allow future smart rescheduling from those meters and controllers, here we propose a load shift model. The fixed load is included in current consumption e_t^c, while the shiftable one at time t can be carried forward to $t+1$, which is defined as $\alpha(e_t^d - e_t^c)$, where $0 \leq \alpha \leq 1$ (as detailed in Section V.A, when $\alpha = 0$, there is no monthly flexibility). In each month, there is a new energy demand e_t^{new} from the user, and the cumulative energy demand e_t^d can be iterated according to the following expressions:

$$e_{t+1}^d = \alpha(e_t^d - e_t^c) + e_{t+1}^{new}, e_1^d = e_1^{new} \tag{4}$$

Note that in our case study, the data is obtained from traditional energy users and therefore, α is close to zero.

The parameterized user consumption model can be applied to the smart controller, which can interact with the environment. The AIA can then collect data from the interaction and updates its decision policy based on the collected data.

III. Problem Formulation

The Markov decision process (MDP) is a mathematical formulation for modeling decision making in uncertain situations [19]. The AIA's decision-making problem is formulated as an MDP without transition probabilities, which is defined by a series of iteration steps $\mathcal{T} \equiv \{t : t = 1 \dots T\}$, a set of κ-dimensional states $\mathcal{S} \subset \mathbb{R}^\kappa$ that represent the information of the environment, a set of actions $\mathcal{A} \subset \mathbb{R}$ for each state, and a real-valued reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. The objective of the MDP is to build an optimal decision policy $\pi^* : \mathcal{S} \to \mathcal{A}$ that maximizes the expected T-steps reward for any s_t.

An efficient method for estimating the policy π is to use a state-action value function called a Q function, which is the accumulated reward at start point (s, a) following π. The optimal Q function is the one with the maximum Q value over all policies:

$$Q^*(s, a) = \max_\pi Q^\pi(s, a) \tag{5}$$

Given the optimal Q value for each $s - a$ pair, π^* is obtained as follows:

$$\pi^*(s) = \arg\max_{a \in \mathcal{A}} Q^*(s, a) \tag{6}$$

The following three subsections present the formulations of the state, action and reward.

A. State Vector Description

The state vector consists of three parts: time-dependent component, reactive component, and exogenous component [20].

1) Time-dependent component: Since variables of the environment vary at different time periods, variables associated with time variation are employed here. To describe the time dependency, the variable $m_t \in \{1, 2, ..., 12\} \equiv \mathcal{M}$ is introduced to indicate the current month t of a year:

$$m_t = mod(t, 12) \qquad \forall t \in \mathcal{T} \tag{7}$$

By adding this variable, the monthly features for behavior of the user and retailers can be captured by the Q-learning of the AIA.

2) Reactive component: The reactive part of the state vector is measured locally and affected by action a_t. The variable e_t^c, recorded locally in the AIM, is influenced by the ERP and the user's consumption preference. e_t^c is introduced therefore to describe the reactive component. The process of the interaction between e_t^c and the AIA's action, and the determination of e_t^c are detailed in Section IV.

3) Exogenous component: The exogenous part of the state vector can not be affected by a_t and can be obtained from the configuration of ERPs. According to equations (1)-(2), ERP i in month t can be identified by its state tuple $s_{i,t}^{erp}$:

$$s_{i,t}^{erp} = (c_{fix}^{tdu}, c_t^{tdu}, c_{add}^{tdu}, p^{tdu}, c_i^{ad}, c^b, c_{i,t}^e) \tag{8}$$

Therefore, the exogenous state information of the retail market is given by:

$$s_t^{ex} = (s_{1,t}^{erp}, s_{2,t}^{erp}, ..., s_{|\mathcal{I}|,t}^{erp}) \in \mathcal{S}^{ex} \tag{9}$$

where:

$$\begin{aligned} &\mathcal{S}^{ex} = \textstyle\prod_{i \in \mathcal{I}} \mathcal{C}^{tdu} \times \mathcal{C}_i \times \mathcal{C}_i^{ad} \\ &c_t^{tdu} \in \mathcal{C}^{tdu}, c_{i,t}^e \in \mathcal{C}_i, c_i^{ad} \in \mathcal{C}_i^{ad} \end{aligned} \tag{10}$$

To sum up, the state of the environment including the retail electricity market and the user at time t is the combination of the above three components:

$$s_t = (m_t, e_t^c, s_{i,t}^{ex}) \in \mathcal{S} \tag{11}$$

where $\mathcal{S} = \prod_{i \in \mathcal{I}} \mathcal{M} \times \mathcal{E} \times \mathcal{C}^{tdu} \times \mathcal{C}_i \times \mathcal{C}_i^{ad}$.

B. Action Description

In an MDP model, the action must be confined to a finite set of possible actions. The AIA is designed to execute an action a_t at each time from a finite set $\mathcal{A} \equiv \{a_1, a_2, ..., a_{|\mathcal{I}|}\}$ of selection functions. The action is to select an ERP from various ERPs offered by different service providers in the retail electricity market.

To design the selection function $a_i : \mathbb{R}_+^{|\mathcal{I}|} \to \mathbb{R}_+$, two real-valued vectors are proposed. One is the combination of the monthly cost of each ERP:

$$\boldsymbol{\psi} = [\psi_{1,t}^{erp}, \psi_{2,t}^{erp}, ..., \psi_{|\mathcal{I}|,t}^{erp}] \tag{12}$$

The other vector is a unit vector with the length of $|\mathcal{I}|$, and this vector only contains one nonzero component:

$$\boldsymbol{e}_i = [0_1, ..., 0_{i-1}, 1_i, 0_{i+1}, ..., 0_{|\mathcal{I}|}] \tag{13}$$

where the subscript of $\boldsymbol{e}$ denotes the location of 1 in the vector.

Therefore, the selection function is given by:

$$a_i(\boldsymbol{\psi}) = \boldsymbol{\psi}\boldsymbol{e}_i^\top \qquad \forall i \in \mathcal{I} \tag{14}$$

The value of the selection function is the monthly cost of ERP i at time t.

C. Reward Description

The reward r is an important indicator for guiding the learning of the decision policy. The possibility of taking an action a_i from a policy $\pi(s)$ is higher if the learning algorithm receives a higher reward value. To better describe the cost feedback information of the environment, a weight parameter is integrated into the proposed reward function:

$$r_t(s_t, a_t) = (\lambda - 1)\varphi(e^c_{t+1}) - \lambda a_t(e^c_{t+1}), 0 \le \lambda \le 1 \tag{15}$$

where $a_t \in (a_i(\boldsymbol{\psi}))_{j=1}^{|\mathcal{I}|}$ and λ is the weight parameter of the user's consumption satisfaction and the electricity bill cost. The AIA places more emphasis on the user's consumption satisfaction with a smaller λ, while more emphasis is placed on the cost with a larger λ. This parameter can be adjusted by the user according to its consumption preference.

IV. Algorithm Description

A conventional MDP requires a probability distribution for state transitions, in order to learn the optimal decision policy. However, the underlying probability distribution is hard to achieve for end users. To solve the proposed MDP problem with unknown transition probabilities, a model-free RL approach in the AIA is developed to learn the optimal decision policy from a batch of transition tuples, which are obtained by interacting with the environment.

The proposed model-free RL method is based on a batch Q-learning algorithm, which leverages a kernel regression function to estimate the Q function $\widehat{Q}(s_t, a_t)$ that implies the optimal decision policy based on a batch of tuples (s_t, a_t, r_t, s'_t), where $s'_t = s_{t+1}$. In each tuple, based on the observed state s_t and the executed action a_t, the next state s'_t is reached with the associated reward r_t. The tuples comprise the following transition samples:

$$\mathcal{F} = \{(s_l, a_l, r_l, s'_l)\,|l = 1, ..., |\mathcal{F}|\,\} \tag{16}$$

where the subscript symbol is different from the one in a single tuples since l denotes the sample index in the set $\mathcal{F}$.

In the following subsections, the approach to obtaining the reactive component of the state vector when interacting with the environment is described to construct the transition samples. Moreover, several data processing methods are applied to the reactive and exogenous components of the state vector in order to reduce the computational cost of the proposed algorithm and increase the prediction accuracy.

A. Sampling From Environment

In the formulation of a tuple $q_t = (s_t, a_t, r_t, s'_t)$ at time t, the proposed Q-learning algorithm takes an action a_t, i.e., chooses an ERP, when the system is in state s_t. As a reactor, the agent then determine the electricity e^c_{t+1} under this ERP to maximize the current reward r_t and the expected future rewards.

To determine e^c_{t+1}, a period of δ months in the future is considered, during which the price information of the ERPs is known. Note that the future information can be obtained since the proposed algorithm uses the training data set to learn the decision policy.

For the end user, knowing the ERP information for future δ months, the energy consumption e^c_{t+1} is determined by the following optimization problem:

$$e^c_{t+1} = \arg\max \sum_{j=t}^{t+\delta} \gamma^{j-t} r_j(s_j, a_j) \tag{17}$$

$$s.t. \quad 0 \le e^c_j \le e^{\max} \tag{18}$$

$$(4) \tag{19}$$

where $e^{\max}$ is determined by physical limitations of the appliances.

To formulate the transition sample set $\mathcal{F}$, for each time period t, the reactive component e^c_{t+1} in tuple q_t is obtained by solving the optimization problem (17)-(19) with the state s_t and action $a_t = \pi^{(k)}(s_t)$ under current policy $\pi^{(k)}$. Note that different from the way users decide on their consumption in a real environment, the proposed sampling is done with optimization. The reason is that the purpose of sampling is to use the best result (s'_t, r_t) obtained by executing action a_t to train the agent. In this way, the trained agent can make better decisions than real users [15], as indicated in Section V.

B. Data Processing on State Vector

When implementing the proposed batch Q-learning algorithm in the electricity pricing plan decision system, two challenges exist. 1) The dimension of the state space is so high that the algorithm occupies a large memory capacity to store the state-action pairs and requires a long-time convergence. 2) The prediction error for price signals exists in the decision policy, which increases the cost and user dissatisfaction.

To handle the above challenges, three data processing methods are proposed. The first is to reduce the computational cost, and the latter two are to increase the prediction accuracy.

1) Dimensionality Reduction for State Vector: For the exogenous state tuple $s^{erp}_{i,t}$, since c^{tdu}_{fix}, c^{tdu}_{add}, c^b, and p^{tdu} are time-invariant and hard for the end user to gain, they can be eliminated. In equation (2), since $c^e_{i,t}e^c_t + c^{ad}_i e^c_t = (c^e_{i,t} + c^{ad}_i)e^c_t$, the state variables $c^e_{i,t}$ and c^{ad}_i can be replaced by one averaged variable: $(c^e_{i,t} + c^{ad}_i)/2$.

For the time-varying state variable c^{tdu}_t and reactive component e^c_t, they can be discretized into a finite number of cost and energy levels in $\mathcal{D}^{tdu}$ and $\mathcal{D}^e$ by using two quantization operation functions, $d_1(\cdot)$ and $d_2(\cdot)$, respectively.

After dimensionality reduction, the recast state of the environment is expressed as follows:

$$s^{erp}_{i,t} = (d_1(c^{tdu}_t), (c^e_{i,t} + c^{ad}_i)/2) \tag{20}$$

$$s_t = (m_t, d_2(e^c_t), s^{ex}_{i,t}) \in \mathcal{S} \tag{21}$$

where $\mathcal{S} = \prod_{i\in\mathcal{I}} \mathcal{M} \times \mathcal{D}^e \times \mathcal{D}^{tdu} \times \mathcal{C}_i$. The recast state reduces the dimensionality from $\left|\prod_{i\in\mathcal{I}} \mathcal{M} \times \mathcal{E} \times \mathcal{C}^{tdu} \times \mathcal{C}_i \times \mathcal{C}^{ad}_i\right|$ to $\left|\prod_{i\in\mathcal{I}} \mathcal{M} \times \mathcal{D}^e \times \mathcal{D}^{tdu} \times \mathcal{C}_i\right|$, and also reduces the memory complexity from $O(\prod_{i\in\mathcal{I}} |\mathcal{M}|\,|\mathcal{E}|\,\left|\mathcal{C}^{tdu}\right|\,|\mathcal{C}_i|\,\left|\mathcal{C}^{ad}_i\right|\,|\mathcal{A}|)$ to $O(\prod_{i\in\mathcal{I}} |\mathcal{M}|\,|\mathcal{D}^e|\,\left|\mathcal{D}^{tdu}\right|\,|\mathcal{C}_i|\,|\mathcal{A}|)$.

2) Observation Memory of Exogenous State Vector: In order to improve the accuracy and learning efficiency of the AIA, additional information can be included in the training set $\mathcal{F}$. An observation record of historical energy charges $(c^e_{i,j})^{t-1}_{j=t-h+1}$ is added to $s^{erp}_{i,t} \in \mathbb{R}^{h+1}$:

$$s^{erp}_{i,t} = (c^e_{i,t-h+1}, ..., c^e_{i,t-1}, d_1(c^{tdu}_t), (c^e_{i,t} + c^{ad}_i)/2) \tag{22}$$

The concatenated observation record acts as a price fluctuation indicator, which presents the price difference and market trend over previous months.

3) Additional Sample Knowledge: Although the end user can only change an ERP once a month, the price information (i.e., $c^e_{i,t}$ and c^{tdu}_t) of ERPs, in practice, is updated daily or weekly. With a large number of historical transaction records and the improvement we make in this subsection to further enlarge the sample tuples, there are enough training data to let the agent learn an optimal policy. Take weekly-updated price information for example, it can enlarge the transition sample set $\mathcal{F}$ by adding three other types of tuples q_{t_1}, q_{t_2} and q_{t_3} in each month, where $t_n, 0 < n \leq 3, n \in \mathbb{Z}$ means the data is sampled n week(s) after the beginning of month t. We explain how to construct these tuples and integrate them into $\mathcal{F}$ by taking q_{t_n} as an example.

In the additional tuple q_{t_n}, s_{t_n} is defined as $(m_t, e^c_{t_n}, s^{ex}_{i,t_n})$, where s^{ex}_{i,t_n} is the price information updated n week(s) after the beginning of month t. According to the problem (17)-(19), in addition to available data (i.e., s_{t_n} and $a_{t_n} = \pi^{(k)}(s_{t_n})$), $e^c_{t_n}$ is determined by $e^{new}_{t_n}$, which can be calculated by the sum of $e^{new}_{t,n}$:

$$e^{new}_{t_n} = e^{new}_{t-1,n+1} + ... + e^{new}_{t-1,4} + e^{new}_{t,1} + ... + e^{new}_{t,n} \tag{23}$$

The additional tuples construct a subset of transition samples $\mathcal{F}_n$:

$$\mathcal{F}_n = \{q_{1_n}, ..., q_{t_n}\}, n = 1, 2, 3 \tag{24}$$

which are concatenated in sequence along with the original sample set $\mathcal{F}_0$ to construct $\mathcal{F}$:

$$\begin{aligned} \mathcal{F} &= \textstyle\bigcup_{n=0}^{3} \mathcal{F}_n \\ &= \{q_l \,|l = 1, ..., |\mathcal{F}_0|, |\mathcal{F}_0| + 1, ..., |\mathcal{F}|\} \end{aligned} \tag{25}$$

The enlarged batch sample space can not only enable the proposed algorithm to learn more information during each iteration, but also improve the convergence speed.

C. Kernel Approximator-Based Batch Q-learning Algorithm

Combined with the above-mentioned sampling method and data processing approaches, the proposed batch Q-learning algorithm is outlined in **Algorithm 1**, where steps 2-11 consist of the interaction stage. Steps 12-14 represent the learning stage where the policy π is estimated by iterating the Q function with the information from the interaction stage:

$$Q^{\pi}(s, a) = \underset{\omega \sim p_\omega(\cdot|s)}{E} [r(s, a, \omega) + \gamma J^{\pi}(f(s, a, \omega))] \tag{26}$$

where ω denotes a random process under probability distribution $p_\omega(\cdot\,|s)$. This random process is based on s and controlled by a. Once the Q function is converged after iterations, the optimal policy π^* is learned through equations (5)-(6).

A Kernel approximator is proposed to match the data with the estimated Q function $\widehat{Q}(s_l, a_l)$, and the approximator function is defined as follows:

$$\tau(s_l, s) = \frac{\upsilon\left(\frac{\|s_l - s\|}{\rho}\right)}{\sum_{s_j \in \mathcal{F}} \upsilon\left(\frac{\|s_j - s\|}{\rho}\right)} \tag{27}$$

Algorithm 1: Kernel Approximator-Based Batch Q-learning Algorithm Combined With Sampling And Data Processing Methods

Input: $k \leftarrow 0$, $\mathcal{F} = \{(s_l, a_l, r_l, s'_l)\,|l = 1, ..., |\mathcal{F}_0|, ..., |\mathcal{F}_1| + |\mathcal{F}_2| + |\mathcal{F}_3|, ..., |\mathcal{F}|\}$, where $e^c_{l+1} = 0$ and $r_l = 0$, h, δ, γ, ϵ, α, β, $e^{new}_{t,n}$, $\widehat{Q}^{(0)}(s_l, a_l) = 0$;

Output: $\widehat{Q}(s_l, a_l)$;

1 **while** $||\sum_{l=1}^{|\mathcal{F}|} \sum_{a_l \in \mathcal{A}} (\widehat{Q}^{(k+1)}(s_l, a_l) - \widehat{Q}^{(k)}(s_l, a_l))|| > \epsilon$ **do**
2 **for** $\mathcal{F}$ **do**
3 Generate a random number g, $g \in \mathbb{R}$ and $0 < g \leq 1$;
4 **if** $g > 1/k^{0.7}$ **then**
5 $a_l = \arg\max_{a_l \in \mathcal{A}} \widehat{Q}^{(k)}(s_l, a_l)$
6 **else**
7 a_l is taken radomly from $\mathcal{A}$;
8 **end**
9 **end**
10 Obtain e^c_{l+1} and r_l by solving (17)-(19);
11 **end**
12 **for** $s_l \in \mathcal{F}, l = 1, ..., |\mathcal{F}|$ **do**
13 Update the Q function by using Kernel approximator: $\widehat{Q}^{(k+1)}(s_l, a_l) \leftarrow \sum_{(s'_l, r_l) \in \mathcal{F}} \tau(s'_l, s_l)[r_l + \gamma \max_{a_l \in \mathcal{A}} \widehat{Q}^{(k)}(s_l, a_l)]$;
14 **end**
15 $k \leftarrow k + 1$;
16 **end**
17 return **Output**;

where ρ is to adjust the smoothness of the approximator. $\tau(s_l, s)$ acts as a weighting operator that varies according to $\mathcal{F}$, s, and Kernel function $\upsilon(\cdot)$. Based on the initial value of $\widehat{Q}(s_l, a_l)$, each iteration k of the proposed algorithm finally leads to solving the exact Bellman equation with guaranteed convergence [21].

After the AIA learns the optimal decision policy derived from $\widehat{Q}(s_l, a_l)$ by using the proposed algorithm, the Q function of a new state $s^{new} \notin \mathcal{F}$ is described as follows:

$$\hat{Q}(s^{new}, a_l) = \sum_{(s'_l, r_l) \in \mathcal{F}} \tau(s'_l, s^{new})[r_l + \gamma \max_{a_l \in \mathcal{A}} \hat{Q}(s_l, a_l)] \tag{28}$$

V. Numerical Results

The proposed ERP decision system is tested employing a real-world training data set from the "Power-to-Choose" platform. All calculations are performed on a personal computer with a 3.4-GHz Intel Core i7 processor and 16 GB of RAM using Python. The coding package uses TRFL, which is based on TensorFlow.

A. Decision Performance With Two Benchmarks: Prescient Ideal Optimization and Common User Decision

To test the effectiveness of the proposed decision system, the training data set is based on the electricity usage and transaction records of 521 utility end users in a region in north Texas. In that region, 5 ERPs offered by 3 retailers are available on "Power-to-Choose". The length of the training set is 5 years. The resolution for electricity and pricing information is on a weekly basis. The training dataset includes two parts. One is the service charges of ERPs; the other is the new weekly energy demand, which can be obtained from the load profile of this region. Fig. 3 (a) depicts a sample training dataset.

Parameters of the user's energy consumption model, in practice, can be obtained and adjusted by individual end users according to their consumption preferences. Instead of specifying the parameters of α and β in equation (3) and (4) exogenously, we search every combination of α and β and cross-validate with actual

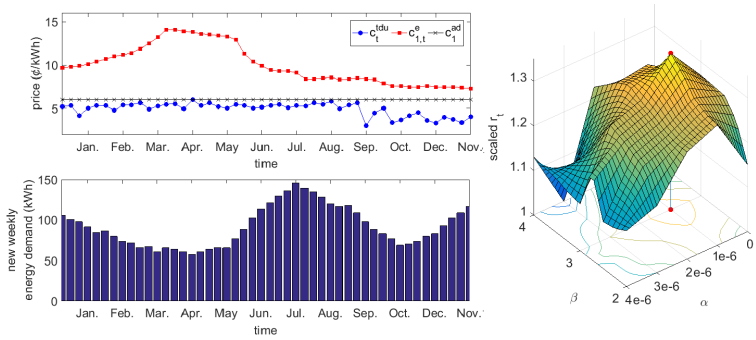


Fig. 3. (a) A sample dataset of price information of ERP 1 and new energy demand during 2015. (b) Scaled system reward with respect to different values of α and β

data from "Power-to-Choose". For every combination of α and β, the system reward r_t is calculated. The optimal combination of α and β, indicated by the highest reward r_t, can lead to the best possible performance of AIA. As illustrated in Fig 3 (b), the parameter combination (α=1e-6, β=3) results in the best prediction performance to maximize the system reward, and the small value of α indicates the inflexibility of present user consumption with few smart equipments. For parameters of the proposed batch Q learning algorithm, $h = 4$, $\gamma = 0.85$, $\delta = 6$, $\rho = 0.8$, $|\mathcal{D}^{tdu}| = 3$, $|\mathcal{D}^{e}| = 25$ and the Kernel function is given by [22]:

$$\upsilon(s) = \frac{1}{\sqrt{2\pi}\rho} e^{\frac{-\|s_l - s\|_1^2}{2\rho^2}} \tag{29}$$

Two benchmarks are presented to compare with the performance of the proposed AIA: 1) The real-world average monthly electricity bill cost over the 521 utility end users and 2) the optimized monthly electricity bill cost ψ_t^{erp*} calculated by:

$$\psi_t^{erp*} = \max\{\psi_{i,t}^{erp}(e_t^{rw}) | i \in \mathcal{I}\} \tag{30}$$

where it is assumed that the future price signal of the ERPs has been known and the perfect decision can be made to select the most cost-saving ERP for every month.

During the test, after each test time period, the training dataset is updated with the newest information of the test subset in the interaction stage, and the learning stage is executed again in the next test time period.

The results of the proposed AIA based on a one-year test dataset are presented in Fig. 4 (a), and Fig. 4 (b) presents the switching frequencies of ERP plans for actural users in the year. Monthly average costs for different ERPs are indicated by different curves in Fig. 4 (a), and the monthly selected ERPs by the AIA are given in the table above the plot. Once the proposed AIA replaces the current ERP with a new one in the table, a new dotted line appears to connect the point on the curve with the new ERP. A new dotted line appears most of the time, when the curve with the lowest average price changes, and this line connects to the lowest point in the current month. This means that the proposed AIA can predict the most cost-saving ERP of the next month almost every time.

During the whole year, the proposed AIA changes ERPs 4 times to search for the optimal ERP, whereas approximately 2/3 of actual users do not change their ERPs at all, as indicated in Fig. 4 (b). Insufficient switching of ERPs by actual end users is a primary reason for the increase of cost, since a single ERP cannot guarantee the lowest average price throughout the year.

The average costs for end users employing the proposed AIA are compared with actual users' costs and optimized costs (aforementioned benchmark 2) in Fig. 5. As indicated, the proposed AIA reduces the average monthly cost by 19.8% and the annual cost by 19.9%, respectively, when comparing to real-world average costs. In addition, the cost of the proposed AIA is a little higher than the optimized cost. This is because the latter results from the best decision with perfect information. However, their performances are very close, and compared to the proposed AIA, the optimal average monthly cost is only 3% less.

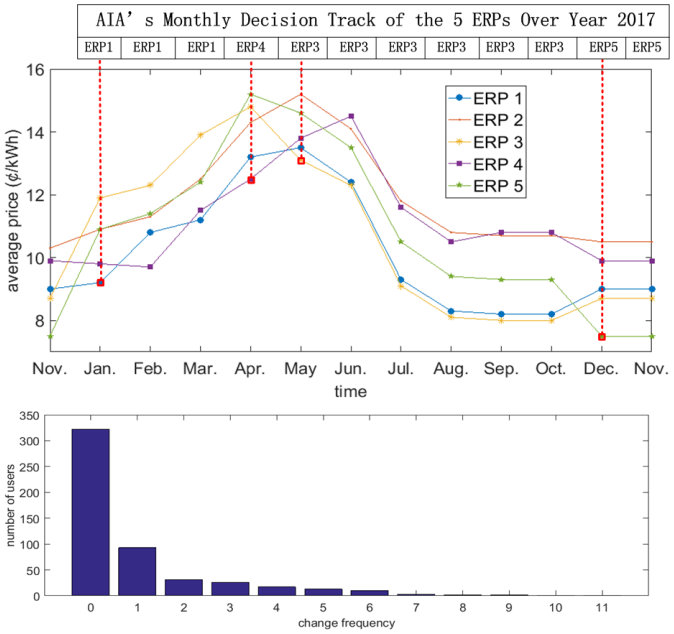


Fig. 4. (a) Test dataset of 5 ERPs in 2017 with decision track of the proposed AIA. (b) Distribution of ERP change frequency for real-world users during 2017

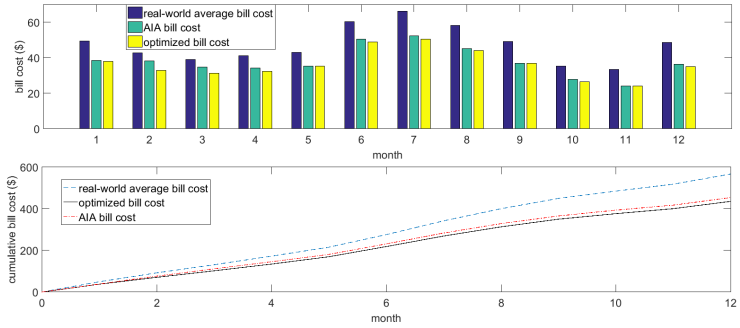


Fig. 5. Test results averaged over 50 experiments: (a) Monthly chosen ERP cost in 2017 of the end user under different cases. (b) Comparison of cumulative costs in 2017

B. Prediction and Computation Performance Under Different Settings

Using the same test dataset in the above subsection, different settings of the proposed AIA are made to analyze for their effect on the prediction and computation performance. Two indices are defined as follows to measure the cost prediction performance of the proposed AIA:

$$M_1 = \frac{c_t^{aia} - c_t^{hum}}{c_t^{opt} - c_t^{hum}}, M_2 = \frac{\sum_{t\in\mathcal{T}}(c_t^{aia} - c_t^{hum})}{\sum_{t\in\mathcal{T}}(c_t^{opt} - c_t^{hum})} \tag{31}$$

which means that the index M equals 1 if the proposed AIA achieves the same performance as the optimized cost, and is 0 if the proposed AIA achieves the same performance as the actual users.

1) Length of Future Time Periods of the Sampling Model: To analyze the effect of parameter δ, results under different values of δ are presented in Fig 6.

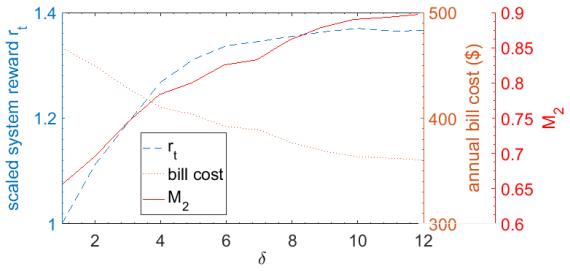


Fig. 6. Effect of δ on r_t and annual cost.

Fig 6 shows that as the time window δ increases from 0 to 12 months, M_2 and scaled r_t rise and the annual cost decreases, since e_t^c is determined by both the current and expected rewards. With a larger δ, the AIA can see the future expected rewards further to schedule e_t^c and this yields additional cost reductions. The marginal improvements decrease drastically when δ is higher than 6, and after that level the performance of the AIA model is influenced mainly by other factors.

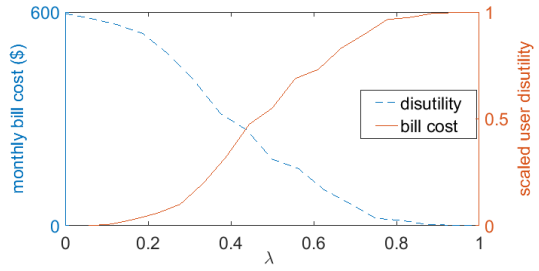


Fig. 7. Effect of λ on monthly cost and scaled user disutility.

2) Weight Parameter λ: Varying λ from 0 to 1, the effect of λ on the system reward r_t is depicted in Fig. 7. As λ increases, the AIA lowers the user's energy consumption to reduce the monthly ERP cost, and the user's disutility rises. For example, when $\lambda = 0$, the AIA only aims to minimize the user's disutility. Hence, the energy consumption is increased to $e^{\max}$ and the monthly ERP cost is high. When $\lambda = 1$, the AIA aims only to minimize the monthly cost, which leads to a high disutility. The end user can adjust λ according to its preference in order to influence the AIA's learning stage.

3) Observation Memory: In Fig. 8, the prediction performance of the proposed observation memory-based AIA is compared to that without memory. It can be seen that the observation memory-based AIA outperforms the one without memory, since the former contains more learning information about price signal variation. The average value of M_1 over the test period is 0.86 for the observation memory-based AIA and 0.72 for that without observation memory, indicating an improvement of 19.4%.

4) Enlarged Training Dataset: Fig. 9 illustrates that with the proposed additional sample tuples, the AIA obtains better prediction performance than with the original transition sample set. Compared to the original transition sample set, the AIA with additional sample tuples increases average M_1 by 40.9%.

5) Computation Performance: Table I presents the effect of the recast state space and enlarged dataset on computation performance averaged over 50 experiments. With the enlarged dataset, there is a 41% decrease of the iteration number, since the AIA learns more in each iteration according to step 13 in **Algorithm 1**. With the dimensionality-reduced state vector, there is a 32% decrease of the training time.

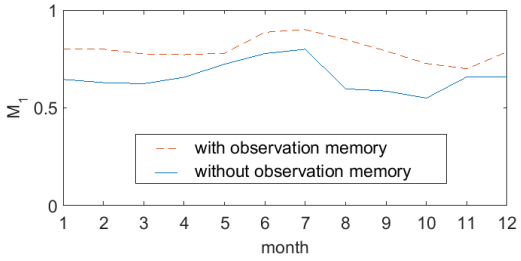


Fig. 8. Effect of observation memory on prediction performance.

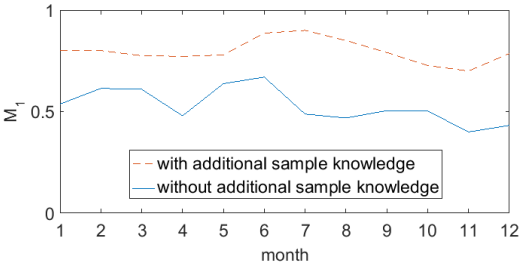


Fig. 9. Effect of enlarged training dataset on prediction performance.

C. Comparisons Under Different Methods and Larger System

The proposed AIA is compared with the following learning methods to validate its advantages on ERP selection: 1) inverse optimization (IO) [23]; 2) random forest (RF) [24]; 3) support vector machine (SVM) [25]; 4) deep Q-network (DQN) [26]; and 5) asynchronous advantage actor critic (A3C) [27].

Fig. 10 (a) and the second row of Table II illustrate prediction performances using the same real-world dataset with 5 ERPs in the above subsections. It can be seen that the proposed AIA outperforms other learning methods both in overall prediction accuracy (M_2) and that for each month (M_1), since **Algorithm 1** is designed to cater more to the proposed problem. In addition, costs for end users employing RF and IO are even more than actual users costs ($M_2/M_1 < 0$), indicating their inefficiency on ERP selection.

In practice, the maximum number of qualified ERPs an end user can choose from is 13. To further test the performance of the proposed AIA in a larger system (i.e., a future expanded market), we collect transaction records of 100 ERPs from 17 regions and assume an electricity usage dataset to construct an environment with 100 ERPs. The simulation is formulated on a workstation

TABLE I
COMPUTATION PERFORMANCE UNDER DIFFERENT CASES

	Training time (s)	Iteration number
Proposed AIA	907	108
Without enlarged dataset	314	183
Without recast state space	1348	113

TABLE II
OVERALL PREDICTION ACCURACY (M_2)

Dataset	AIA	SVM	DQN	A3C	RF	IO
5 ERPs	0.83	0.51	0.02	0.22	-0.3	-1.36
100 ERPs	0.81	0.23	-0.2	0.16	-1.6	-2.78

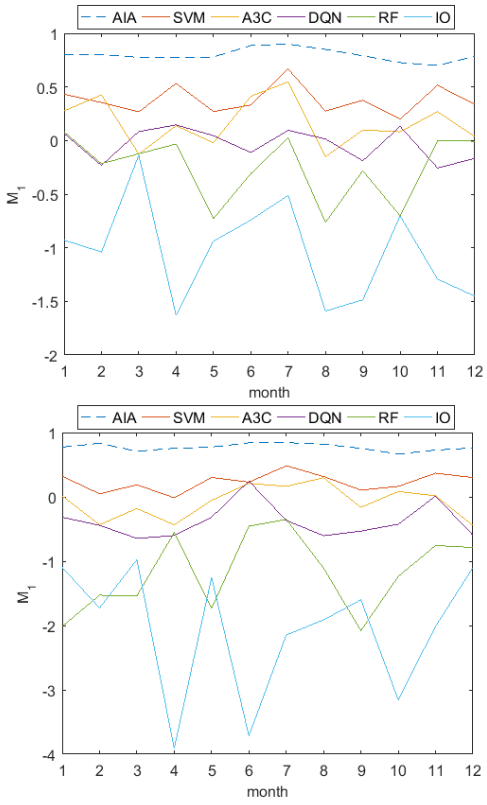


Fig. 10. Prediction comparisons under different datasets: (a) Real-world dataset used in Section V. A (b) A simulation dataset including 100 ERPs with assumed user data

equipped with a core i7 processor and GTX1080 GPU. User parameters are set as α=1e-6,β=2.7, with the same other settings as 5 ERP-based experiment has.

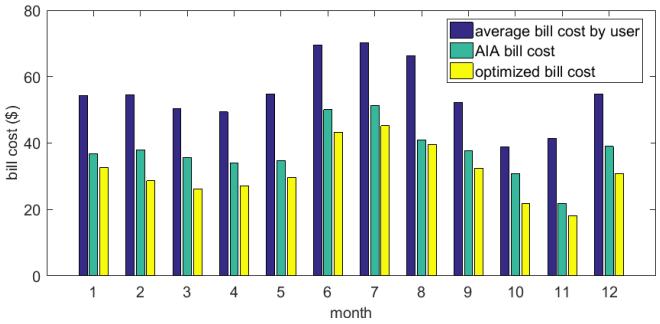


Fig. 11. Monthly chosen ERP cost in a market with 100 ERPs under different cases

As indicated in Fig. 11, the proposed algorithm (AIA) reduces the total cost by 35.8% for 100-ERPs market. The costs resulting from the proposed algorithm for both 5-and-100 ERPs markets are very close to the optimal costs. This means that as a prediction method, the proposed algorithm is efficient enough to learn an optimized policy, to accurately forecast the future information and to obtain cost-effective results.

Prediction results of AIA and methods 1) - 5) are compared in Fig. 10 (b) and the third row of Table II using the same hyperparameters. Similar to the results of 5 ERPs, AIA outperforms the compared methods. In contrast to methods 1) - 5), theres little difference between AIAs prediction performance (M_2/M_1) in the system with 5 ERPs and that with 100 ERPs. This indicates that AIA has higher adaptability to different ERP market environments.

TABLE III
COMPUTATION PERFORMANCE OF DIFFERENT METHODS FOR LARGER SYSTEM

	Training time (s)	Iteration number
Proposed AIA	1486	119
DQN	6403	296
A3C	4921	263

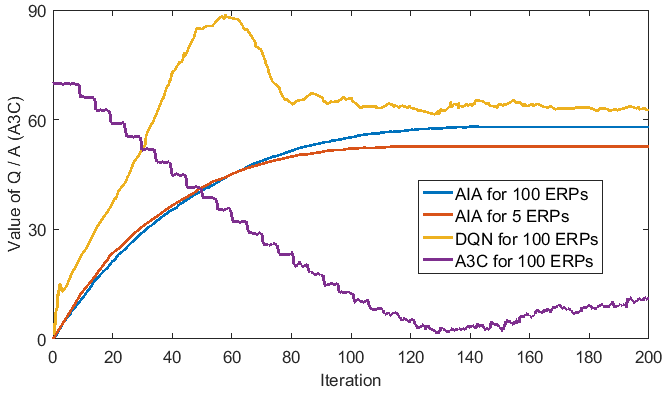


Fig. 12. Iteration process for different learning methods

The computational efficiencies of two reinforcement learning methods, DQN and A3C, are compared with the proposed AIA in Table III and Fig. 12, averaged over 50 experiments. Table III shows that the training time and iteration number of AIA are 63.7% and 54.8% less than those of A3C. Compared to DQN, the advantage of AIA is more obvious: a 76.8% decrease in training time and a 59.8% decrease in iteration number. From 5-ERPs market in Table I to 100-ERPs market in Table III, the training time and iteration number of AIA only increase by 61.8% and 10.2%, indicating its high adaptability to different ERP market environments.

Catering to the proposed pricing-plan decision problem, the Kernel approximator-based algorithm has advantages on computation and prediction. Different from the proposed algorithm, deep reinforcement learning, such as DQN and A3C, has a multilayer structure, which results in more computational costs. As shown in Table III, DQN and A3C spend more time to train the model. In addition, DQN and A3C cannot properly handle discontinuous data. Therefore, they take more iterations to converge and have lower values of M_1, as indicated in Table III and Fig. 10.

Fig. 12 illustrates the value of Q and A (A3C) functions in terms of the iteration. Both the proposed AIAs for 5 ERPs and 100 ERPs smoothly aim for the optimized value whereas the other two methods show some oscillations and difficulties. This computational stability allows the proposed algorithm to learn correct behavior in much complicated environments.

VI. CONCLUSIONS

This paper proposes a RL-based decision system for an individual smart grid user to select the optimal electricity pricing plan offered by retailers. The decision-making process is formulated as an MDP without transition probability. In the MDP, the pricing information of ERPs and the user's energy consumption are both modeled and integrated into the state vector with a time-dependent component. A novel reward function is developed to determine the optimal energy consumption considering future rewards including the payment cost and user satisfaction. The action is to choose an ERP according to the decision policy under the current state.

The proposed MDP is solved using a model-free batch Q learning algorithm with a Kernel appoximator to estimate the Q value of state-action pairs. The tuple sample set is enlarged and the state structure is modified to improve the computational and prediction performance of the proposed algorithm.

Experimental results using real-world data indicate that the prediction performance of the proposed decision system is much better than actual human users and close to perfect decision making. A main reason for the substantial increase of actual users' payment is that they rarely change an ERP over a whole year, while the proposed decision system can select the most cost-saving ERP in each month through precise prediction. Also, the user's energy consumption and the performance of different ERPs are time-variant, and the changes in information for different months play an important role in making the precise prediction. In addition, sensitivity analyses of different parameters illustrate that a larger δ, observation memory, and enlarged training dataset can improve the prediction performance. The data processing method can reduce the computational burden.

REFERENCES

[1] E. G. Kardakos, C. K. Simoglou, and A. G. Bakirtzis, "Short-term electricity market simulation for pool-based multi-period auctions," *IEEE Transactions on Power Systems*, vol. 28, no. 3, pp. 2526–2535, Aug 2013.

[2] F. Luo, G. Ranzi, X. Wang, and Z. Y. Dong, "Social information filtering based electricity retail plan recommender system for smart grid end users," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2017.

[3] M. van Berlekom, C. Sjöland, and E. Widman, "Nyckeln till a–en studie av kreditbetyg på den amerikanska marknaden före och efter finanskrisen 2007-2008," 2014.

[4] B. Luo, D. Liu, and H. N. Wu, "Adaptive constrained optimal control design for data-based nonlinear discrete-time systems with critic-only structure," *IEEE Transactions on Neural Networks and Learning Systems*, vol. PP, no. 99, pp. 1–13, 2017.

[5] C. D. Charalambous and N. U. Ahmed, "Team optimality conditions of distributed stochastic differential decision systems with decentralized noisy information structures," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 708–723, Feb 2017.

[6] M. Ziba, "Service-oriented medical system for supporting decisions with missing and imbalanced data," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 5, pp. 1533–1540, Sept 2014.

[7] M. Qin, N. Zhang, and Z. Wang, "Research on bus travel decision system based on generalized cost calculation," in *2016 International Conference on Logistics, Informatics and Service Sciences (LISS)*, July 2016, pp. 1–4.

[8] B. J. Claessens, P. Vrancx, and F. Ruelens, "Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load control," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2016.

[9] B. G. Kim, Y. Zhang, M. van der Schaar, and J. W. Lee, "Dynamic pricing and energy consumption scheduling with reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2187–2198, Sept 2016.

[10] A. Chi, J. Lundn, and V. Koivunen, "Reinforcement learning-based plug-in electric vehicle charging with forecasted price," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 3674–3684, May 2017.

[11] S. Vandael, B. Claessens, D. Ernst, T. Holvoet, and G. Deconinck, "Reinforcement learning of heuristic ev fleet charging in a day-ahead electricity market," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1795–1805, July 2015.

[12] C.-H. Tsai and Y.-L. Tsai, "Competitive retail electricity market under continuous price regulation," *Energy Policy*, vol. 114, pp. 274–287, 2018.

[13] T. Zhan, S. Chen, C. Kao, C. Kuo, J. Chen, and C. Lin, "Non-technical loss and power blackout detection under advanced metering infrastructure using a cooperative game based inference mechanism," *IET Generation, Transmission Distribution*, vol. 10, no. 4, pp. 873–882, 2016.

[14] S. Wang, S. Bi, and Y. A. Zhang, "Demand response management for profit maximizing energy loads in real-time electricity market," *IEEE Transactions on Power Systems*, vol. 33, no. 6, pp. 6387–6396, Nov 2018.

[15] Z. Liu, C. Zhang, M. Dong, B. Gu, Y. Ji, and Y. Tanaka, "Markov-decision-process-assisted consumer scheduling in a networked smart grid," *IEEE Access*, vol. 5, pp. 2448–2458, 2017.

[16] D. Hurley, P. Peterson, and M. Whited, "Demand response as a power system resource," *Synapse Energy Economics Inc*, 2013.

[17] M. Zouai, O. Kazar, B. Haba, and H. Saouli, "Smart house simulation based multi-agent system and internet of things," in *2017 International Conference on Mathematics and Information Technology (ICMIT)*, Dec 2017, pp. 201–203.

[18] T. Alquthami and A. P. S. Meliopoulos, "Smart house management and control without customer inconvenience," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2553–2562, July 2018.

[19] S. Doltsinis, P. Ferreira, and N. Lohse, "An mdp model-based reinforcement learning approach for production station ramp-up optimization: Q-learning analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 9, pp. 1125–1138, Sept 2014.

[20] M. Onderwater, S. Bhulai, and R. van der Mei, "Value function discovery in markov decision processes with evolutionary algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 9, pp. 1190–1201, Sept 2016.

[21] R. E. Bellman and S. E. Dreyfus, *Applied dynamic programming*. Princeton university press, 2015, vol. 2050.

[22] D. Ormoneit and Ś. Sen, "Kernel-based reinforcement learning," *Machine learning*, vol. 49, no. 2-3, pp. 161–178, 2002.

[23] K. Ghobadi, T. Lee, H. Mahmoudzadeh, and D. Terekhov, "Robust inverse optimization," *Operations Research Letters*, vol. 46, no. 3, pp. 339–344, 2018.

[24] T. G. Pavey, N. D. Gilson, S. R. Gomersall, B. Clark, and S. G. Trost, "Field evaluation of a random forest activity classifier for wrist-worn accelerometer data," *Journal of science and medicine in sport*, vol. 20, no. 1, pp. 75–80, 2017.

[25] B. Manavalan and J. Lee, "Svmqa: Support–vector-machine-based protein single-model quality assessment," *Bioinformatics*, vol. 33, no. 16, pp. 2496–2503, 2017.

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[27] S. Yang, B. Yang, H.-S. Wong, and Z. Kang, "Cooperative traffic signal control using multi-step return and off-policy asynchronous advantage actor-critic graph algorithm," *Knowledge-Based Systems*, vol. 183, p. 104855, 2019.