Batch Reinforcement Learning for Smart Home Energy Management

Heider Berlink, Anna Helena Reali Costa
Universidade de São Paulo
São Paulo, SP, Brazil
{heiderberlink, anna.reali}@usp.br

Abstract

Smart grids enhance power grids by integrating electronic equipment, communication systems and computational tools. In a smart grid, consumers can insert energy into the power grid. We propose a new energy management system (called RLbEMS) that autonomously defines a policy for selling or storing energy surplus in smart homes. This policy is achieved through Batch Reinforcement Learning with historical data about energy prices, energy generation, consumer demand and characteristics of storage systems. In practical problems, RLbEMS has learned good energy selling policies quickly and effectively. We obtained maximum gains of 20.78% and 10.64%, when compared to a *Naïve-greedy* policy, for smart homes located in Brazil and in the USA, respectively. Another important result achieved by RLbEMS was the reduction of about 30% of peak demand, a central desideratum for smart grids.

1 Introduction

Smart grids are power grids that use information and communication technology to gather data about the behavior of suppliers and consumers, so as to improve the efficiency, reliability, and sustainability of electricity production and distribution [Hammoudeh *et al.*, 2013; Hashmi *et al.*, 2011; Uluski, 2010]. In this scenario, consumers can insert energy into the power grid and participate in the energy supply management [Palensky and Dietrich, 2011]. Such new functionalities leads to a new house concept that is integrated with smart grids: the smart home, depicted in Figure 3.

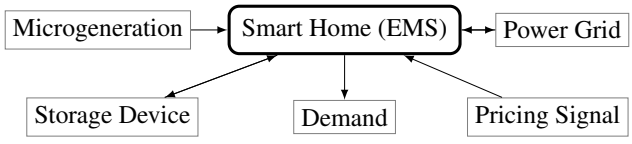

Figure 1: Components of a smart home. The Energy Management System (EMS) makes all decisions in a smart home.

Fundamental elements of smart homes are: *Microgeneration*, providing power generation through alternative sources such as wind and solar; *Energy Storage*, storing energy through devices such as rechargeable batteries or electric vehicles; *Demand Control*, regulating the use of home appliances, so as to reduce the energy bill; *Bidirectional Power Flow*, so that the smart home can acquire or insert energy into the power grid; and *Differentiated Tariff*, meaning different electricity prices for each time of the day, thus inducing the consumer to avoid the peak periods.

Because the price of energy and power generation through alternative sources are variable during the day and because the user can store energy using storage devices, the optimization of the house energy balance faces a very complex dynamics. Thus, it is necessary to develop decision-making tools that operate as Energy Management Systems (EMS), to minimize the user electricity bill or to maximize the user profit in a given period.

Indeed, in the power systems community many optimization algorithms have been developed for EMSs. Depending on the optimization objectives and constraints, appropriate techniques for modeling and for policy generation can be applied. Dusparic et al. [2013] propose a multiagent approach that uses predicted energy consumption data to manage the energy demand of a group of houses. Their strategy, called W-learning, consists of integrating independent Q-learning agents, one for each house. In contrast, O'Neill et al. [2010] present an algorithm called CAES, based on a single reinforcement learning agent that controls the home appliances to smoothen the energy usage. Their objective is to operate the appliances when the price is low and with minimum delay, reducing energy costs up to 40%. Many proposals combine different optimization techniques to provide a fast and robust solution. In [Chen *et al.*, 2013], linear programming (LP) and Monte Carlo techniques are combined in an EMS that has as output the usage schedule of a set of appliances. Their approach has as its main advantage the fast solution provided by the LP solver. Kim and Poor [2011] and Chen et al. [2012] also propose a solution that aims at operating the home appliances, which are categorized under a deadline constraint that is defined considering the particular operation of each appliance.

One of the disadvantages of these solutions is the rigid schedule for the appliances usage; this is clearly uncomfortable for the end users. Scheduling is better handled by Truong et al. [2013], who propose a system that learns the users pref-

erences, minimizing the impact on their daily habits while reducing the energy bill up to 47%. In [Mohsenian-Rad *et al.*, 2010], a distributed demand-side EMS that uses game theory (GT) is used with several users, by formulating an energy consumption scheduling game. GT is also used by Atzeni et al. [2013], which consider a day-ahead optimization process regulated by an independent central unit. Shann and Seuken [2013] promote a demand response while learning the user's preferences over time and automatically adjust home temperature in real-time as prices change.

In short, these previous proposals consider different ways of dealing with the decision-making problem. Each framework considers subsets of the subsystems depicted in Figure 3 to provide demand response, and it is difficult to pick one as benchmark.

In this paper, microgeneration and storage are controlled to supply energy, while power consumption is considered only as additional information in our model. Consequently, we avoid rigid schedules for appliance usage. We adopt a control strategy that coordinates energy generation and storage in response to prices, promoting an efficient response to demand while ensuring the comfort of the user in deciding when to use appliances.

Our main contribution is a new and flexible EMS, the Reinforcement Learning-based Energy Management System (RLbEMS), that learns an operation policy for selling or storing energy surplus in smart homes. We propose a new approach for both modeling and solving the EMS problem. The modeling of the decision-making problem as a Markov Decision Problem (MDP) offers a novel way to take energy prices into account. The solution by Batch Reinforcement Learning (BRL) is effective because of smartly uses the available historical data. RLbEMS was tested with real operational data from two different places. As these data have a strong dependence on location, we demonstrate that our proposed system can be used in situations characterized by distinct levels of uncertainty.

The remainder of this paper is structured as follows. In Section 2, we briefly describe our mathematical framework. Our proposal, the RLbEMS system, is described in Section 3. In Section 4 we experimentally evaluate and analyze the RLbEMS system and, in Section 5, we conclude and discuss future steps.

2 Theoretical Framework

An EMS is a sequential decision-making system whose aims is to maximize the smart home long-term gain. We use Markov Decision Processes as modeling tool, and produce policies using a Batch Reinforcement Learning algorithm, called Fitted Q-iteration [Ernst *et al.*, 2005a]. Our model and solution pay attention to historical data on energy prices, the current season, and energy generation, storage, and demand. The required theoretical background is given in this section.

2.1 Markov Decision Process

Sequential decision systems evolve probabilistically according to a finite and discrete set of states. At each time step the system first observes the state of the process, then chooses and executes an action that leads it to another state, and receives a reward (as shown in Figure 2). A formalism widely used for this is the Markov Decision Process (MDP) [Puterman, 1994]. Its key concept is the Markov property: every state encodes all the information needed to make the optimal decision in that state.

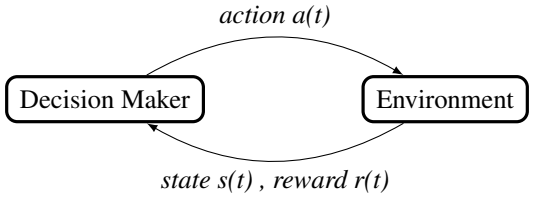

Figure 2: The decision-maker interaction with the environment. The system observes the state $s(t)$, applies the action $a(t)$, receives the reward $r(t)$, and observes the next state $s(t+1)$, restarting the process.

Formally, an MDP is a tuple $< S, A, T, R >$, where [Puterman, 1994]: S is a finite set of states; A is a finite set of possible actions; $T : S \times A \times S \rightarrow [0, 1]$ is a state transition probability function, which defines the transition probability from a state $s(t) \in S$ to a state $s(t+1) \in S$ when an action $a(t) \in A$ is applied in $s(t)$; $R : S \times A \rightarrow \Re$ is the reward function. So, for each state transition $r(t) = R(s(t), a(t))$. We define A_s as the admissible set of actions for state $s \in S$. So, if $i \in S$ is the current state, the transition from $i \in S$ to $j \in S$ in response to the application of the action $a \in A_i$ will occur with probability $T(i, a, j)$ and a reward $R(i, a)$ will be received. Solving an MDP means finding a policy π that specifies which action must be executed in each state, considering the maximization of the discounted cumulative rewards received during an infinite time horizon.

2.2 Batch Reinforcement Learning

Reinforcement Learning (RL) [Sutton and Barto, 1998] solves MDPs in problems where the agent has little or no information about the controlled system. Batch Reinforcement Learning (BRL) is a subfield of RL whose main purpose is the efficient use of collected data. Moreover, BRL tends to yield better policies, faster than classical RL algorithms such as the popular model-free on-line Q-learning algorithm [Watkins, 1989]. BRL is used in RL problems where the complete amount of learning experiences is fixed and given a priori [Ernst *et al.*, 2005a]. By taking the learning experience as a set of transitions samples from the system, the learning mechanism seeks the best possible solution out of this given batch of samples. The major benefits of this approach come from the way it uses the amount of available data and the stability of the learning process. The fast convergence and the use of a predefined set of given samples are crucial for implementing this technique in real problems, because we usually need a good solution quickly and with the lowest possible process interaction.

BRL has three specific and independent phases: Exploration, Training and Execution. In the Exploration phase, the

system acquires a batch of transition samples from the environment; in the Training phase, the system learns the best policy from this batch; and in the Execution phase the system just applies the learned policy.

The Fitted Q-iteration (FQI) algorithm [Ernst *et al.*, 2005a] is one of the most popular algorithms in BRL due to its simple implementation and its excellent results. FQI converts the "learning from interaction" scheme to a series of supervised learning problems. The FQI Training algorithm is shown in Algorithm 1. Let $F = \{(s(t), a(t), r(t), s(t+1))|t = 1, ..., n\}$ be a batch of transition samples from the environment. At each iteration, for each transition $(s(t), a(t), r(t), s(t+1))$, the value $\bar{q}^h_{s,a}$ is updated. A training set TS^h is defined, with $((s,a), \bar{q}^h_{s,a}) \in TS^h$, in which (s,a) are the inputs and $\bar{q}^h_{s,a}$ is the target of the supervised learning problem. At each iteration, a new training set is defined as the last one united to the updated values, $\bar{q}^h_{s,a}$. After that, a supervised learning algorithm is used to train a function approximator on the pattern set TS^h. At the end, the resulting function, $\bar{Q}^H$, is an approximation of the Q-function after H steps of dynamic programming. Thus, we can use a greedy policy in FQI to define the policy π. The obtained policy is the best one with respect to the fixed batch of transition samples acquired from the exploration process.

Algorithm 1 Fitted Q-Iteration Algorithm

Require: Load $F = \{(s(t), a(t), r(t), s(t+1)) \mid t = 1, ..., n\}$;
Require: Define $\bar{Q}^0(s,a) = 0$, $\forall (s,a) \in F$, and $\bar{q}^h_{s,a} \in \bar{Q}^h(s,a)$;
Require: Define H as the Horizon to be performed.
Require: Define TS^0 as an initially empty training set;
  $h = 1$;
  while $h \leq H$ **do**
    for all $(s(t), a(t), r(t), s(t+1)) \in F$ **do**
      $\bar{q}^h_{s,a} = r(t) + \gamma \max_{a \in A_{s(t+1)}} \bar{Q}^{h-1}(s(t+1), a)$;
      if $((s(t), a(t)), .) \in TS^{h-1}$ **then**
        $TS^{h-1} \leftarrow TS^{h-1} - \{((s(t), a(t)), .)\}$;
      end if
      $TS^h \leftarrow TS^{h-1} \bigcup \{((s(t), a(t)), \bar{q}^h_{s,a})\}$;
    end for
    %% Supervised Learning %%
    Use supervised learning to train a function approximator $\bar{Q}^h(s,a)$ on the training set TS^h;
    $h \leftarrow h + 1$;
  end while
  %% Obtaining Policy %%
  for $\forall s \in S$ **do**
    $\pi(s) = \arg\max_{a \in A_s} \bar{Q}^H(s,a), \forall s \in S$;
  end for

3 The RLbEMS System for Smart Homes

Here we describe the RLbEMS system, and describe how RLbEMS learns a selling policy through the FQI algorithm. Finally, we describe how RLbEMS applies the learned policy when new data from the environment are available.

3.1 RLbEMS as an MDP

We consider a single residence that has its own energy generation by a Solar PV System, that receives the pricing signal in real-time, and that has batteries to store energy whenever it is convenient, with B_{MAX} as storage capacity. RLbEMS must not sell more than the total available amount of energy (stored and surplus energy) and must respect the battery maximum capacity of storage and the battery maximum charge and discharge rate.

States

We define the MDP states using completely observable and relevant information about the problem: the energy storage level, the amount of energy generated, the energy consumed by the smart home, and data on price trends. Information on price trends consists of two indexes that represent the evolution of prices and average prices in a given time window. Considering a time window with three instants, and having prices in instants k, $k-1$, and $k-2$, we define *Price interval* $\Delta p(k) = [p(k-2), p(k-1), p(k)]$. For each $\Delta p(k)$, the price average value, $\overline{\Delta p(k)}$, is defined. We also define $\Delta_2 = p(k) - p(k-1)$, and $\Delta_1 = p(k-1) - p(k-2)$. These variables are mapped into the indexes $\overrightarrow{p(k)}_{IND}$ and $\overline{\Delta p(k)}_{IND}$ as indicated in Tables 1 and 2. The MDP state, $s(k) \in S$, is defined as an array composed of five variables: the power stored in the battery, $B(k)$; the power generated by the Solar PV System, $G(k)$; the power consumed by the smart home, $D(k)$; and the indexes of price trend, $\overrightarrow{p(k)}_{IND} \in \{1, 2, ..., 5\}$, and price average level, $\overline{\Delta p(k)}_{IND} \in \{1, 2, ..., 8\}$:

$$s(k) = [B(k), G(k), D(k), \overrightarrow{p(k)}_{IND}, \overline{\Delta p(k)}_{IND}]. \quad (1)$$

Table 1: Price trend index.

Price Trend Index	$\overrightarrow{p(k)}$	$\overrightarrow{p(k)}_{IND}$
$\Delta_1 \geq 0, \Delta_2 > 0$	$\lvert\Delta_2\rvert \geq \lvert\Delta_1\rvert$	8
$\Delta_1 \geq 0, \Delta_2 > 0$	$\lvert\Delta_2\rvert < \lvert\Delta_1\rvert$	7
$\Delta_1 < 0, \Delta_2 \geq 0$	$\lvert\Delta_2\rvert > \lvert\Delta_1\rvert$	6
$\Delta_1 < 0, \Delta_2 \geq 0$	$\lvert\Delta_2\rvert < \lvert\Delta_1\rvert$	5
$\Delta_1 \geq 0, \Delta_2 \leq 0$	$\lvert\Delta_2\rvert \leq \lvert\Delta_1\rvert$	4
$\Delta_1 \geq 0, \Delta_2 \leq 0$	$\lvert\Delta_2\rvert > \lvert\Delta_1\rvert$	3
$\Delta_1 \leq 0, \Delta_2 < 0$	$\lvert\Delta_2\rvert < \lvert\Delta_1\rvert$	2
$\Delta_1 \leq 0, \Delta_2 < 0$	$\lvert\Delta_2\rvert \geq \lvert\Delta_1\rvert$	1

Table 2: Average price index.

$\overline{\Delta p(k)}$	$\overline{\Delta p(k)}_{IND}$
$\overline{\Delta p(k)} > 120$	5
$100 < \overline{\Delta p(k)} \leq 120$	4
$80 < \overline{\Delta p(k)} \leq 100$	3
$60 < \overline{\Delta p(k)} \leq 80$	2
$\overline{\Delta p(k)} \leq 60$	1

Actions

Actions indicate when and how much of the surplus energy to sell given the system state and the current season. A *null action* means that no energy should be sold at a given time, i.e., RLbEMS must store the surplus in the battery or, if there is no surplus, RLbEMS must not act. A *null surplus* means that the energy demand is supplied by the microgeneration and by the battery, or that the EMS must acquire energy from the power grid to attend its demand. The discrete set of actions is defined as:

$$A = \left\{0, 0.5, 1, 1.5, ..., S^{MAX} + G^{MAX}\right\}, \quad (2)$$

where G^{MAX} is the maximum amount of energy generated, and S^{MAX} is the maximum amount of energy that can be charged/discharged from the battery in each discrete instant. We define the amount of available energy $C_u(k)$ as:

$$C_u(k) = B(k) + G(k) - D(k). \quad (3)$$

A subset of admissible actions is defined for each state, taking into account the value of C_u. In our model, the user consumption changes the control action to be taken, also indicating when we might sell energy to the grid. Thus, we have three cases, all complying the limit of charge/discharge of the battery, S^{MAX}:

1. If $C_u(k) \leq 0$, then the EMS should buy $C_u(k)$ energy or do nothing (if $C_u(k) = 0$), and $A_s = \{0\}$;
2. If $C_u(k) > 0$ and $C_u(k) \leq B_{MAX}$, then there is surplus and we can even store all of it; so $A_s = \{0, ..., C_u(k)\}$;
3. If $C_u(k) > 0$ and $C_u(k) > B_{MAX}$, then there is surplus and we can not store all of it; so $A_s = \{C_u(k) - B_{MAX}, ..., C_u(k)\}$.

Reward Function

We use a reward function that reflects whether energy is sold for a price that is above or below the average historical energy price for the current season. That is, the reward function conveys the average price of all available prices for that season. We adopt:

$$r(k) = R(s(k), a(k)) = a(k) \times (p(k) - \overline{p}), \quad (4)$$

where $\overline{p}$ corresponds to the average historical price, $p(k)$ is the price at the decision time and $a \in A_s$ is the amount of energy sold in state $s \in S$.

3.2 FQI in RLbEMS

BRL has three independent phases: Exploration, Training, and Execution. In the Exploration phase, RLbEMS captures historical data of energy generation, energy consumption, and energy price for each season of the year. Data are acquired synchronously, i.e., there is a measurement of each variable at each discrete instant, k. The battery energy level is initiated with a random value and, using the first measure of the historical data, the initial state, $s(0)$, is defined. For each state s, the set of admissible actions, A_s is defined, and an action $a \in A_s$ is chosen randomly and applied. After applying $a(k)$ in $s(k)$, the next state $s(k+1)$ is reached and the immediate reward $r(k)$ is calculated (Equation 4); then this experience $(s(k), a(k), r(k), s(k+1))$ is inserted in F as a new sample from the environment. This process runs until all historical data are used for each season. The main result of this step is the batch of samples, F, that is used in the Training phase to obtain the energy selling policy.

In the Training phase, RLbEMS runs the Algorithm 1 given in Section 2.2. This FQI algorithm aims at finding a function that approximates the value-action function $Q(s, a)$, $\forall s \in S$, $\forall a \in A$. Thus, the main result of this step is to produce an approximate surface $QS : S \times A \rightarrow Q$, that maps each pair state-action to its corresponding $Q(s, a)$ value. By analyzing experimental results from a benchmarking test, we chose a Radial Basis Neural Network [Park and Sandberg, 1991] to approximate the function $Q(s, a)$. Here the Algorithm 1 runs until a convergence criterion dQ is met or until a fixed value of the horizon H is reached. We define $dQ = 5\%$, which means that the convergence criterion is met if the estimated Q-value for each sample of F varies less than 5 % from one iteration to the next. We use $H = 200$.

After training, RLbEMS enters the Execution phase, where the current data determine the state of the system. Having the current state and the energy-selling policy learned in the training phase for the current season, an action is defined and the user profit is calculated, as Algorithm 2 describes. If the user has purchased energy from the power grid, the *energy bill* is acquired directly from the smart meter. Thus, the *financial gain* is calculated for each season by subtracting the energy bill from the total profit acquired with Algorithm 2, $Total_{PR}$.

Algorithm 2 The RLbEMS Execution Mode

Require: Observe the current season of the year;
Require: Load the learned policy π_{SEASON};
Require: Define $Total_{PR} = 0$.
while TRUE **do**
  Observe the current state $s(k)$;
  $a(k) \leftarrow \pi_{SEASON}(s(k))$
  Apply $a(k)$;
  Calculate the current profit: $PR(k) = a(k) \times p(k)$
  $Total_{PR} \leftarrow Total_{PR} + PR(k)$;
end while

4 Experiments and Results

Two case studies were performed to evaluate the RLbEMS performance. Given the influence of climate characteristics and the energy price model, we conducted these case studies in two different places, the USA and Brazil.

All tests compared RLbEMS-generated policies with a *Naïve-greedy* policy, which at each instant sells all energy surplus, $Sr(k)$, by the current price:

$$\begin{aligned} \textit{Naïve-Greedy Policy} : a(k) = Sr(k),\ \forall Sr(k) \geq 0, \\ \textit{with} : Sr(k) = G(k) - D(k). \end{aligned} \quad (5)$$

If $Sr(k) \leq 0$, the policy applies a *null action*, which means that the generation meets the demand or that EMS must buy energy from the grid. We also compared the RLbEMS-generated policies with a Simple Storage Policy (SSP), which

sells the energy surplus to the grid when the price is above average, and stores surplus when the price is bellow average.

To test RLbEMS, we have gathered real historical data of energy generation and price for Brazil and the USA. The energy consumption data were generated from real statistical data on the usage of home appliances, available in [PROCEL, 2014]. The energy microgeneration, the energy demand and the energy price are external variables that vary their profiles in a long period, following a typical pattern that is related to the season. This can be viewed in Figure 3 for the energy price in the USA. The same behavior is observed for the energy generation and consumption, and these data are always synchronized by day time, as stated earlier in Section 3.2.

Due to these facts, the annual historical data were divided into four sets of data, each one corresponding to one season of the year. After this, each database was divided into two subsets of data: Training Data and Validation Data. For each season, the Training Data were used in Algorithm 1 to obtain the energy selling policy π_{SEASON}, and the Validation Data were used in Algoritm 2, when we evaluated the real gains of applying the proposed algorithms. To perform our tests, we simulated the RLbEMS on-line execution using the Validation Data for Brazil and the USA. As we will see, the first one was a simplified version of the second one and was used as a preliminary test, to confirm the benefits of the proposed approach. Both smart homes, in Brazil and in the USA, generated their own energy with a Solar PV System and stored energy in a rechargeable battery. The data used in this article consider an output of a real Solar PV System. The rechargeable battery, in both cases, uses the model proposed by Atzeni et al. [2013], that considers as main parameters the charge and discharge efficiency, the maximum charging and discharging rate, rate of energy loss over time, and maximum storage capacity.

As a comparative index, we calculate the *percentage increase of the financial gain* over each season, given by the ratio of the financial gain obtained by using RLbEMS and SSP in comparison to the *Naïve-greedy* policy. We also calculate the *average peak reduction* in each case study. The grid demand has its peak reduced by RLbEMS. To calculate this, we make a daily analysis of the energy request curve of the house from the power grid perspective: for each day of operation, we compare the peak demand when using the RLbEMS or not. We record the peak reduction for each day of operation, and calculate the average reduction for each season.

4.1 Case Study in Brazil

This case study considered a smart home located in São Paulo, Brazil. The differentiated tariff implemented in Brazil is a Time-of-Use (TOU) tariff. In this case, there are three values of tariff during the day [Bueno *et al.*, 2013]. This pricing signal remains fixed during all days of the year and concentrate major values of energy price in times when there is a high demand of energy. The energy generation data were acquired from a real solar power plant in São Paulo, Brazil [Berlink *et al.*, 2014]. The solar power plant consists of ten modules, connected as a serial array. Each module generates 255W on the peak of generation, resulting in a total peak of 2.55kW. We used hourly energy generation data from August 2013 to January 2014. Given that the amount of available data is small, we chose to implement a solution with a single policy. This decision did not affect the final result, because in this case study the pricing signal was fixed and the energy generation did not vary too much during the seasons.

In this case study, we achieved a financial-gain growth of 20.78% and an average peak demand reduction of 39.78%. The results achieved show that RLbEMS is feasible for the proposed problem. In this case, the obtained policy is basically to store the surplus energy during the day when the price is low, until the maximum capacity of the battery is reached; when the battery is full, it sells the surplus at the current price. In the evening, when energy is more expensive and there is no generation, the system either uses or sells energy, with greater profitability for the user.

4.2 Case Study in the USA

This case study was performed with the real pricing signal for the District of Columbia, USA. These data correspond to the Local Marginal Price (LMP), which is a Real-Time Pricing (RTP) tariff and reflects the value of energy at a specific location at the time that it is delivered [PJM, 2015]. This price was also used to calculate the amount of money paid to the consumer when he/she sells energy to the power grid. For this test, we used the hourly pricing data of five consecutive years (2008-2013). For the simulation, we used the Solar PV System given by Chen et al. [2013]. We considered a system composed of four KD200-54 P modules from the Kyocera Solar Incorporation [Kyocera, 2015] that has 220Wp as the peak energy generation per module. The same storage device proposed by Atzeni et al. [2013] was used, and the energy consumption was generated by the same statistical data mentioned before. Results per season are shown in Table 3.

The application of the RLbEMS policy increased the financial gain by the end of the period, resulting in an annual gain of 14.51%. The reduction of the financial gain per season when compared to the brazilian case was expected given that here we used the RTP tariff and that the algorithm had to handle a step level of uncertainty. However, it is worth noting that the use of the RLbEMS system resulted in a reduction of the average peak demand of 29.38%.

Table 3: Financial gain growth for the smart home in the USA: The values represent the percentage increase of the financial gain from RLbEMS and SSP policies in comparison to a *Naïve-greedy* policy.

	Summer	Autumn	Spring	Winter
FQI	2.40%	2.73%	-1.26%	10.64%
SSP	0.42%	1.78%	0.05%	8.75%

In Figure 4, we can see the RLbEMS execution for two consecutive days in summer. Note that when the system is able to identify the price trend correctly, the policy obtained is similar to the one achieved in the first case study: RLbEMS identifies periods of low price and stores energy to use when

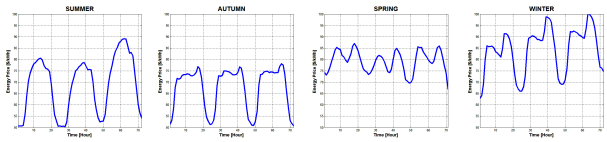

Figure 3: Price pattern for three consecutive days of each season in the USA.

there is a peak of price. However, if this peak of price is not high enough when compared to the low period, the policy loses efficiency. This happens because, when the RLbEMS stores energy, it stops selling (and profits nothing). If the price difference between these two periods is small, the sale when the price is (only slightly) higher does not justify the storage of energy surplus. In this case, always selling the energy surplus at the time it is generated (as the *Naïve-greedy* policy does), would give greater financial gain. Hence, the application of the RLbEMS policy could lead to a result that is worse than the application of the *Naïve-Greedy* policy, as in spring, for example. Comparing the obtained results with the SSP policy we can see that the RLbEMS policy reachs a higher financial gain, which means that the algorithm actually identifies a better time to sell/store the energy, mainly because it considers not only the price in the decision, but also the dynamics of generation and consumption in the house. A fact to note occurs in spring, when the application of the SSP policy resulted in a greater gain for the user.

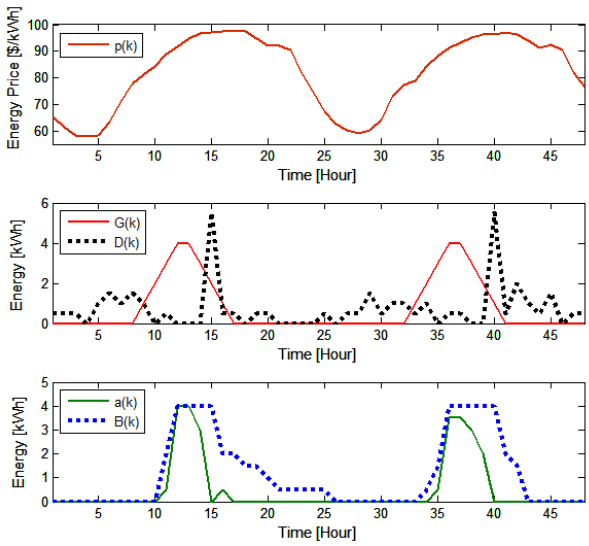

Figure 4: RLbEMS execution for two consecutive days in the USA summer: The system is able to identify lower and higher prices, operating the surplus energy in order to achieve a greater financial gain for a specific period.

5 Conclusion

In this paper we have described our efforts to develop effective decision-making tools for energy management in smart homes. The specific problem is whether to sell or store surplus energy. To solve this problem, a new energy management system was proposed, RLbEMS, which employs MDPs and BRL to maximize user financial gain. In particular, RLbEMS learns autonomously a selling policy for each season, based on batches of historical data about energy price curves, microgeneration, and storage energy. We also demonstrated, through evaluation with real world data taken from two different places, Brazil and the USA, that RLbEMS outperforms *Naïve-greedy* policy by up to 20.78% and 10.64%, respectively. Another important result achieved by our system was the reduction of the peak demand. In most cases, RLbEMS decreases peak demand by more than 30%, an exceptional result. By using RLbEMS the user does not suffer from restrictions regarding the use of appliances. We believe this degree of freedom is essential so that a solution can be accepted into the routine of end users, leading to the popularization of EMS systems.

Our future work will focus on the improvement of RLbEMS, by implementing an architecture that integrates off-line and on-line approaches through transfer learning. In this case, knowledge would be acquired by the application of BRL algorithms, and it would be transferred to an adaptive module that runs an on-line classical RL approach. Thus, the policy obtained by the BRL algorithm would be updated in real-time and the system would be completely adaptive.

Acknowledgments

We would like to thank CNPq (131239/2013-9 and 311608/2014-0) for supporting this research. We also thank the Smart Grid and Power Quality Laboratory ENERQ-CT at Universidade de São Paulo for providing the solar photovoltaic generation data used in this work.

References

[Atzeni *et al.*, 2013] I. Atzeni, L.G. Ordonez, G. Scutari, D.P. Palomar, and J.R. Fonollosa. Demand-side management via distributed energy generation and storage optimization. *Smart Grid, IEEE Transactions on*, 4(2):866–876, June 2013.

[Berlink *et al.*, 2014] H. Berlink, N. Kagan, and A. Costa. Intelligent decision-making for smart home energy management. *Journal of Intelligent & Robotic Systems*, pages 1–24, 2014.

[Bueno *et al.*, 2013] E.A.B. Bueno, W. Utubey, and R.R. Hostt. Evaluating the effect of the white tariff on a distribution expansion project in brazil. In *Innovative Smart Grid Technologies Latin America (ISGT LA), 2013 IEEE PES Conference On*, pages 1–8, April 2013.

[Chen *et al.*, 2012] Zhi Chen, Lei Wu, and Yong Fu. Real-time price-based demand response management for residential appliances via stochastic optimization and robust optimization. *Smart Grid, IEEE Transactions on*, 3(4):1822–1831, Dec 2012.

[Chen *et al.*, 2013] Xiaodao Chen, Tongquan Wei, and Shiyan Hu. Uncertainty-aware household appliance scheduling considering dynamic electricity pricing in smart home. *Smart Grid, IEEE Transactions on*, 4(2):932–941, June 2013.

[Dusparic *et al.*, 2013] I. Dusparic, C. Harris, A. Marinescu, V. Cahill, and S. Clarke. Multi-agent residential demand response based on load forecasting. In *Technologies for Sustainability (SusTech), 2013 1st IEEE Conference on*, pages 90–96, Aug 2013.

[Ernst *et al.*, 2005a] Damien Ernst, Pierre Geurts, Louis Wehenkel, and L. Littman. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6:503–556, 2005a.

[Hammoudeh *et al.*, 2013] M.A. Hammoudeh, F. Mancilla-David, J.D. Selman, and P. Papantoni-Kazakos. Communication architectures for distribution networks within the smart grid initiative. In *Green Technologies Conference, 2013 IEEE*, pages 65–70, April 2013.

[Hashmi *et al.*, 2011] M. Hashmi, S. Hanninen, and K. Maki. Survey of smart grid concepts, architectures, and technological demonstrations worldwide. In *Innovative Smart Grid Technologies (ISGT Latin America), 2011 IEEE PES Conference on*, pages 1–7, Oct 2011.

[Kim and Poor, 2011] T.T. Kim and H.V. Poor. Scheduling power consumption with price uncertainty. *Smart Grid, IEEE Transactions on*, 2(3):519–527, Sept 2011.

[Kyocera, 2015] Kyocera. Kyocera solar, data sheet of kd200-54 p series pv modules [on- line]. http://www.kyocerasolar.com/assets/001/5124.pdf, 2015. Accessed: 2015-30-04.

[Mohsenian-Rad *et al.*, 2010] A.-H. Mohsenian-Rad, V.W.S. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia. Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid. *Smart Grid, IEEE Transactions on*, 1(3):320–331, Dec 2010.

[O'Neill *et al.*, 2010] D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra. Residential demand response using reinforcement learning. In *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*, pages 409–414, Oct 2010.

[Palensky and Dietrich, 2011] P. Palensky and D. Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *Industrial Informatics, IEEE Transactions on*, 7(3):381–388, Aug 2011.

[Park and Sandberg, 1991] J. Park and I. Sandberg. Universal approximation using radial-basis-function networks. *MIT - Neural Computation*, 1991.

[PJM, 2015] PJM. Pjm monthly locational marginal pricing [on-line]. http://www.pjm.com/markets-and-operations/energy/real-time/monthlylmp.aspx, 2015. Accessed: 2015-30-04.

[PROCEL, 2014] PROCEL. Dicas de economia de energia - programa nacional de conservacao da energia eletrica. http://www.procelinfo.com.br, 2014. Accessed: 2014-08-01.

[Puterman, 1994] ML Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

[Shann and Seuken, 2013] Mike Shann and Sven Seuken. An active learning approach to home heating in the smart grid. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, IJCAI '13, pages 2892–2899. AAAI Press, 2013.

[Sutton and Barto, 1998] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[Truong *et al.*, 2013] Ngoc Cuong Truong, James McInerney, Long Tran-Thanh, Enrico Costanza, and Sarvapali D. Ramchurn. Forecasting multi-appliance usage for smart home energy management. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013.

[Uluski, 2010] R.W. Uluski. The role of advanced distribution automation in the smart grid. In *Power and Energy Society General Meeting, 2010 IEEE*, pages 1–5, July 2010.

[Watkins, 1989] Christopher John Cornish Hellaby Watkins. *Learning from Delayed Rewards*. Cambridge, UK, May 1989.