Cogn Comput (2013) 5:264–277
DOI 10.1007/s12559-012-9191-y

# Adaptive Dynamic Programming Algorithm for Renewable Energy Scheduling and Battery Management

**Matteo Boaro · Danilo Fuselli · Francesco De Angelis · Derong Liu · Qinglai Wei · Francesco Piazza**

Received: 28 April 2012 / Accepted: 10 September 2012 / Published online: 26 September 2012


**Abstract** The employment of intelligent energy management systems likely allows reducing consumptions and thus saving money for consumers. The residential load demand must be met, and some advantages can be obtained if specific optimization policies are taken. With an efficient use of renewable sources and power imported from the grid, an intelligent and adaptive system which manages the battery is able to satisfy the load demand and minimize the entire energy cost related to the scenario under study. In this paper, an adaptive dynamic programming–based algorithm is presented to face dynamic situations, in which some conditions of the environment or habits of customer may vary with time, especially using renewable energy. Based on the idea of smart grid, we propose an intelligent management scheme for renewable resources combined with battery implemented with a faster and simpler scheme of dynamic programming, by considering only one critic network and some optimization policies in order to satisfy the load demand. Since this kind of problem is suitable to avoid the training of an action network, the training loop among the two neural networks is deleted and the training process is greatly simplified. Computer simulations confirm the effectiveness of this self-learning design in a typical residential scenario.

M. Boaro · D. Fuselli · F. D. Angelis · F. Piazza
Dipartimento di Ingegneria dell'Informazione, University Politecnica delle Marche, 60131 Ancona, Italy
e-mail: upf@deit.univpm.it

D. Liu (✉) · Q. Wei
State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
e-mail: derong.liu@ia.ac.cn

**Keywords** Adaptive dynamic programming · Approximate dynamic programming · Neural networks · Energy scheduling · Battery management

## Introduction

Over the years, the need to develop intelligent control and management system in residential environment has greatly increased. The distribution of energy on a wide area is considered as one of the main aspects for an efficient growth of intelligent power grids, as well as the presence of renewable sources and dedicated system storages, from which stored energy can be used to meet the load demand in different operative scenarios. For all these reasons, smart grids have been gaining an increasing importance; thus, many researches have been carried out to improve the features related to it [1–7]. Computational intelligence (CI) in smart grids can be implemented in many different ways. The simplest and most straightforward strategies are predefined rule-based [8–10]: a set of *if–then* rules are initially established for a specific scenario, and each predefined strategy is taken while the automatic system is computing. Unfortunately, this technique can assume an heuristic behaviour, and for this reason, it cannot give an optimal solution, though it may have a very intuitive and immediate form and structure. A larger number of more complex optimization techniques have been applied to solve this problem, such as dynamic programming [11, 12], linear programming [13] and nonlinear programming [14]: these techniques try to reduce either computation time or memory requirements through the minimization of a given specific function that must satisfy some constraints or conditions of the problem. Recently, methodologies including fuzzy optimization and genetic algorithms [15, 16], simulated annealing method

[17] and particle swarm optimization [18, 19] have been realized to deal with the operation cost of hybrid energy systems with storage systems. Generally, all these approaches can provide reasonable solutions, but their behaviour is static, so these systems cannot react in dynamic and time-variant situations and conditions.

In the last decades, intelligent control has attracted more and more attentions [20–24]. Adaptive dynamic programming (ADP) is an effective approach to solve optimal control problems. ADP was proposed by Werbos in 1991 to solve optimal control problems forward-in-time [25]. The idea of ADP is to approximate the optimal performance index function and the optimal controller by using function approximation structures [26]. In recent years, ADP and related research have gained much attention from researchers [27–29]. In [25], ADP approaches were classified into four main schemes: heuristic dynamic programming (HDP), action-dependent HDP (ADHDP), also known as Q-learning, dual heuristic dynamic programming (DHP) and action-dependent DHP (ADDHP). In [30], two more ADP are introduced: globalized DHP (GDHP) and ADGDHP.

Adaptive dynamic programming has grown in the power sector, and also some applications have been realized for generators and grid management [31–39]. Smart grids are very powerful whenever some policies are integrated in the system, so the controller is able to optimize important aspects of the problem through specific parameters.

In this work, based on [40], we develop an operational scheme with self-learning ability in a more complex scenario, so that an optimized management of renewable resources in the residential environment is obtained, according to the system configuration and users' demand. The proposed scheme based on ADP is able to improve own performances through learning from environment. The aim is to apply an intelligent optimization to manage a battery, in order to meet the load demand and save money, considering the load profile, the cost rate for electricity and, differently from [40], renewable energy, that can derive from photovoltaic panels, wind turbines or some other devices. So the best solution in terms of costs is not obtained using a static or memoryless technique, but using only a neural network as a function approximation in the implementation of ADP.

The main scheme reported in this paper is based on the procedure explained in [40], but we propose a more complex scheme for a scenario in which also renewable sources are considered in the critic network training, and an intelligent controller provides a correct energy scheduling with minimized costs. In this case, the automatic system must take different decisions, because now the optimization criteria depend also on renewable energy profile, so the power flows considered by the controller cover all the possibilities in a real case, as represented in Fig. 1.

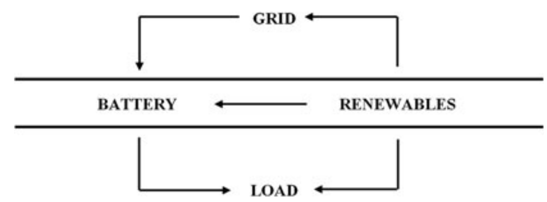


**Fig. 1** Power flows

The renewable sources can make the following three actions:

- Meet the load;
- Charge the battery;
- Be sold to the grid.

The battery makes the following three actions:

- Charged from the grid;
- Charged from renewable energy;
- Discharged to meet the load.

Considering all these possibilities, the controller can take a great number of different decisionscompared with [40] for finding the best energy scheduling for the specific profile. By training the network on a typical cost rate and load profile (established by the customer or some automatic scheduling system), remarkable results can be obtained for the specific scenario and renewable resources involved, and a comparison without a optimal controller (baseline approach) provides clearly the effectiveness of this intelligent system.

The paper is organized as follows: the analytical models used for the different renewable energy sources are shown in Sect. "Resource Modelling" and the theoretical aspects of ADP are briefly discussed in Sect. "Adaptive Dynamic Programming". The procedure based on action-dependent heuristic dynamic programming (ADHDP) scheme is reported in Sect. "Action-Dependent HDP". Finally, Sect. "Computer Simulations" deals with the conducted computer simulations and Sect. Conclusions draws the work conclusions.

## Resource Modelling

Regarding solar energy, the estimated total solar radiation varies from a certain period to another and also depends on position of the sun in the sky. A typical solar panel characteristic of an entire day [41] is shown in Fig. 2.

The power output (in Watts) for a photovoltaic (PV) panel at a given time can be expressed as [42]:

$$P = \mathrm{GHI} \cdot \eta_{pv} \cdot A_{pv} \quad (1)$$

where GHI is the global horizontal irradiance in Wh/m$^2$ received on a horizontal surface $\eta_{pv}$ is the efficiency of the PV included in the range [0, 1], and $A_{pv}$ is the total area of the PV panel in $m^2$.

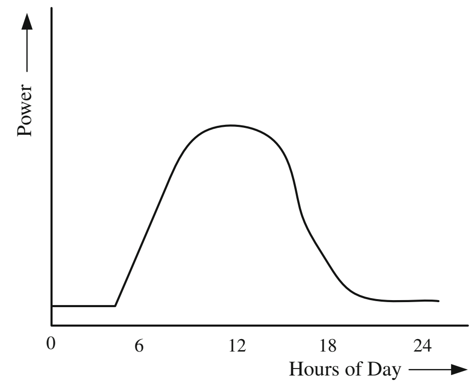


**Fig. 2** A typical solar panel characteristic

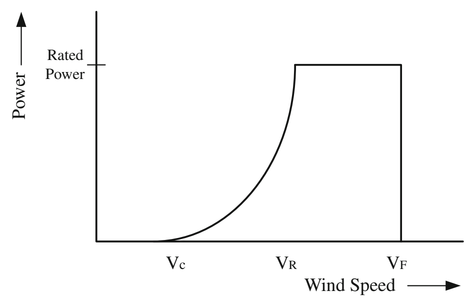


**Fig. 3** A typical wind turbine characteristic

Moreover, the energy obtained from a wind turbine depends mainly on the velocity of wind across the turbine. A typical wind turbine characteristic is shown in Fig. 3.

Total power from the wind turbine is given by

$$P = P_{\mathrm{w}} \cdot A_{\mathrm{w}} \cdot \eta_{\mathrm{w}} \tag{2}$$

where $P_{\mathrm{w}}$ is the power output from the wind turbine (WT) in W/m$^2$, $A_{\mathrm{w}}$ is the total rotor swept area, and $\eta_{\mathrm{w}}$ is the efficiency of the wind turbine included in the range [0,1]. The power output from a WT at any given time is the following [43]:

$$P_{\mathrm{w}} = \begin{cases} 0, & v \le V_C \\ av^3 - bP_R, & V_C < v < V_R \\ P_R, & V_R < v < V_F \\ 0, & v > V_F \end{cases}$$

where $a = P_R/(V_R^3 - V_C^3)$, $v$ is the wind speed in m/s at a given time $t$, $b = V_C^3/(V_R^3 - V_C^3)$, $P_R$ is the rated power, $V_C$ is the cut-in velocity in m/s, $V_F$ is the cut-out velocity in m/s, and $V_R$ is the rated velocity of the turbine in m/s.

In practice, a variety of storage systems exist, and each of them has own characteristics. Regarding a battery model, here follows the maximum energy storage capacity used [44]:

$$E_{\mathrm{b}} = A_{\mathrm{b}} \cdot V_{\mathrm{b}} \tag{3}$$

where $A_{\mathrm{b}}$ is the current-hour (Ah) rating of the battery, and $V_{\mathrm{b}}$ (in Volts) is the maximum voltage of the battery when it is fully charged (100 % state of charge).

## Adaptive Dynamic Programming

Based on Bellman's principle of optimality [45], the dynamic programming approach is used to find optimal sequence actions to solve complex and nonlinear optimization problems. The discrete-time nonlinear system is given by

$$x(t+1) = F[x(t), u(t), t] \tag{4}$$

where $x \in R^n$ is the state vector of the system, $u \in R^m$ represents the control action, and $F$ is the transition function from the current state $x(t)$ to the next state $x(t+1)$ under the control action $u(t)$ at time $t$. The system can be associated with the performance cost:

$$J[x(t), t] = \sum_{k=t}^{\infty} \gamma^{k-t} U[x(k), u(k), k] \tag{5}$$

where $U$ is the utility function, and $\gamma$ is the discount factor with $0 < \gamma \le 1$. Dynamic programming is able to find a sequence of control actions $u(k)$, $k = t, t+1, \ldots$ so that the performance cost in (5) is minimized. According to Bellman, the optimal cost is equal to

$$J^*[x(t), t] = \min_{u_k}(U[x(t), u(t), t] + \gamma J^*[x(t+1), t+1]) \tag{6}$$

The optimal control $u^*(t)$ at time $t$ is the $u(t)$ that provides the minimum cost:

$$u^*(t) = \arg\min_{u_k}(U[x(t), u(t), t] + \gamma J^*[x(t+1), t+1]) \tag{7}$$

ADP can be realized in many different ways, depending on the implementation of the optimal policy routine related to values that approximate the problem solutions. Generally speaking, three different ADP design families can be identified: heuristic dynamic programming (HDP), dual heuristic programming (DHP) and globalized dual heuristic dynamic programming (GDHP). In this work, the used ADP scheme refers to ADHDP, in which the use of a model network is not explicitly required. In the next section, the procedure employed will be explained in detail.

## Action-Dependent HDP

### ADHDP Analytical Issues

For this problem, the ADHDP is considered, because it does not require an explicit model network. The ADHDP scheme is shown in Fig. 4, and in this case, the critic network is trained by minimizing the following error over time:

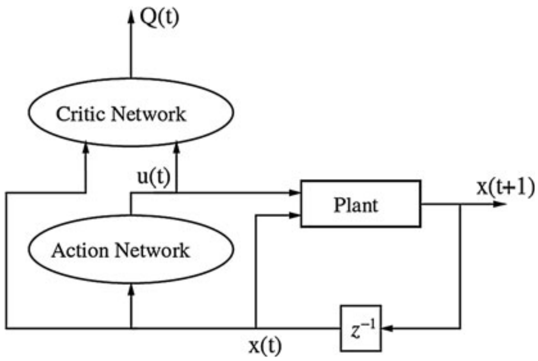


**Fig. 4** ADHDP scheme

$$\|E_q\| = \sum_t E_q(t) = \sum_t \left[Q(t-1) - U(t) - \gamma Q(t)\right]^2 \tag{8}$$

where $Q(t)$ represents the critic network output at time $t$, and its input–output relationship is given by:

$$Q(t) = Q[x(t), u(t), t, W_c] \tag{9}$$

where $x(t)$ is the state of the system at time $t$, $u(t)$ represents the control action at time $t$, and $W_c$ is the weights matrix of the critic network that we are considering.

When $E_q(t) = 0$ for all time $t$, (8) implies that:

$$\begin{aligned} Q(t-1) &= U(t) + \gamma Q(t) \\ &= U(t) + \gamma\left[U(t+1) + \gamma Q(t+1)\right] \\ &= \cdots \\ &= \sum_{k=t}^{\infty} \gamma^{k-t} U(k). \end{aligned} \tag{10}$$

Comparing (5) and (10), we obtain $Q(t-1) = J[x(t), t]$. According to (8), we will use the forward-in-time approach because we consider the $Q$ cost function. So the critic network is trained with the following mapping:

$$\left\{ \begin{array}{c} x(t-1) \\ u(t-1) \end{array} \right\} \longrightarrow \{Q(t-1)\} \tag{11}$$

where $x(t-1)$ and $u(t-1)$ are respectively the state and the control at time $(t-1)$, and $Q(t-1)$ is the related output of the critic network at time $(t-1)$. To approximate dynamic programming solutions, we can consider $\|E_q\| \cong 0$ in order to obtain:

$$Q(t-1) \approx U(t) + \gamma Q(t) \quad \forall t \tag{12}$$

So the critic network is rightly trained using $U$ and $Q$ at time $t$ as follows:

$$\left\{ \begin{array}{c} x(t-1) \\ u(t-1) \end{array} \right\} \longrightarrow \{U(t) + \gamma Q(t)\} \tag{13}$$

So the input and the target can be computed, and the critic network is trained with "Levenberg–Marquardt backpropagation" algorithm. Once this training is completed, the action network has to be trained with the objective of minimizing the critic network output $Q(t)$.

## Self-Learning Procedure Based on ADHDP

The classic ADHDP scheme can be simplified in this case, because only a few actions can be taken by the controller: the battery is limited to a ternary choice (charge, discharge or idle, according to some rates related to the discrete battery model) related to the scenario configuration that occurs at a specific time. In fact, the major priority is given to availability of renewable energy, and in relationship with this aspect, the other controls of the battery are chosen to find the best solution in terms of costs that satisfy the load demand. Thus, an optimized energy scheduling is achieved based on ADHDP procedure proposed in this work if the training of the critic network relies on certain parameters that derive from a "2-step structure": at first, the renewable energy use is evaluated as in Fig. 5, and then, actions for the battery are taken, as reported in Fig. 6. As shown in Fig. 5, the renewables management follows a specific hierarchy; in fact, renewable resources have this priority policy:

1. they immediately meet partially or entirely the load demand;
2. the possible amount in surplus charges the battery if it is not full;
3. all the eventual remaining quantity is sold to grid in order to avoid waste.

Although actions with available renewable energies provide different policies for the battery management, there are three actions that can be taken on the battery (which is connected to the grid): the storage system can be idle, can be charged or can be discharged to use the energy previously stored. Due to this reason, action network is not considered, because just three actions must be managed, and only a critic network remains within the scheme shown in the Fig. 4.

As represented in Fig. 6, after the scheduling of renewable energies, whenever power demand occurs, the critic network verifies which is the action that involves the smallest output value, so the most convenient action is chosen. So, when the network is trained correctly, we have to check which is the more convenient action at that specific time $t$ between three options: *charge* with $u(t) = -1$, *discharge* with $u(t) = 1$ or *idle* with $u(t) = 0$. For the chosen criteria, like in [40], the battery is discharged only if its stored energy is able to satisfy all the load demand; otherwise, the action $u(t) = -1$ is discarded from the available ones.

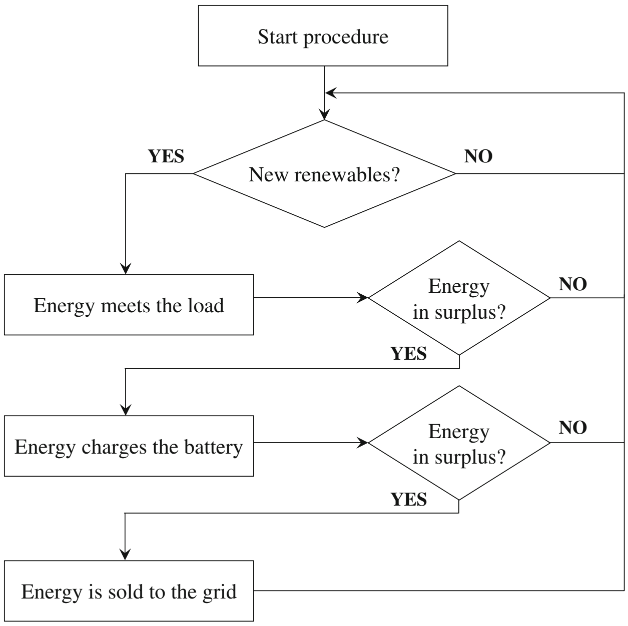


**Fig. 5** *Block diagram* of renewable energy management

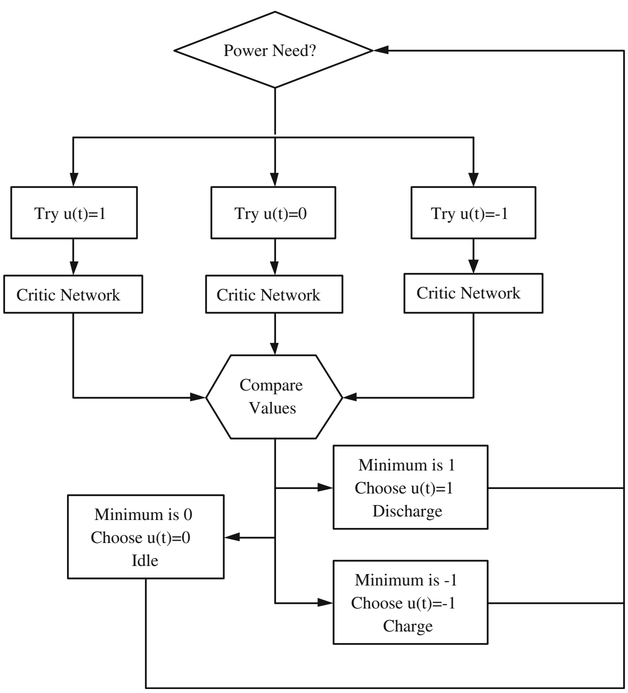


**Fig. 6** ADHDP algorithm *block diagram* for each temporal slot

## Optimization Procedure: Analytical Aspects

The controller is made optimal if the critic network is trained correctly, so specific policies must be taken in order to train the neural network in the right way, whenever a new scenario is considered. To make the controller optimal, it is necessary to train it on all the possible scenario configurations that can occur, as a combination of all the different parameters that form the state $x(t)$ for the automatic system. Knowing the following profiles:

- electricity cost profile,
- the load demand (given by the customer by some scheduling procedure or automatically thanks to passed training),
- the level of the battery,
- the available renewable energy,

it is possible to train the network on this specific set of data. Due to this, the highest possible number of data sets to train the critic network in a general and optimal way must be considered: that is why this controller is capable of learning from the environment over the time. As initial general training, it is a good idea to repeat the entire procedure (described in this subsection) for many iterations, because in this way the randomness of the situations is solved covering a great number of cases.

As shown in Fig. 7, the critic network training procedure consists of the following steps:

1. Collecting data. The action $u(t)$ is chosen randomly for each time $t$ because the best value is unknown a priori, and the state is characterized by the normalized values of the cost rate $C(t)$, the load profile $L(t)$, the battery level $E_b(t)$ and the renewable energy $R(t)$ at each time $t$. According to the first part of (13), we can write:

$$\left\{\begin{array}{c} x(t-1) \\ u(t-1) \end{array}\right\} \longleftrightarrow \left\{\begin{array}{c} C(t-1) \\ L(t-1) \\ E_b(t-1) \\ R(t-1) \\ u(t-1) \end{array}\right\} \tag{14}$$

2. Target computing. The utility function $U(t)$ and the critic network output $Q(t)$ are calculated:

$$T(t) = U(t) + \gamma Q(t) \tag{15}$$

in order to obtain the target of (13) that must be used for the training.

3. Network training. The critic network is trained with the “Levenberg–Marquardt backpropagation” [46] algorithm: given as input the state $x(t)$ and the action $u(t)$ and the target $T(t)$ as output, weights matrix of the neural network is changed to reach the global transfer function of the network that ensures the mapping of (13).
4. Changes in the scenario. If there are some changes in the environment, the critic network must be re-trained to response in an optimal way to the new scenario, so the entire training procedure is repeated, and new weights for the neurons in the network are found.

It is evident that the training success depends on how the target is chosen: in fact, according to it, the problem can be

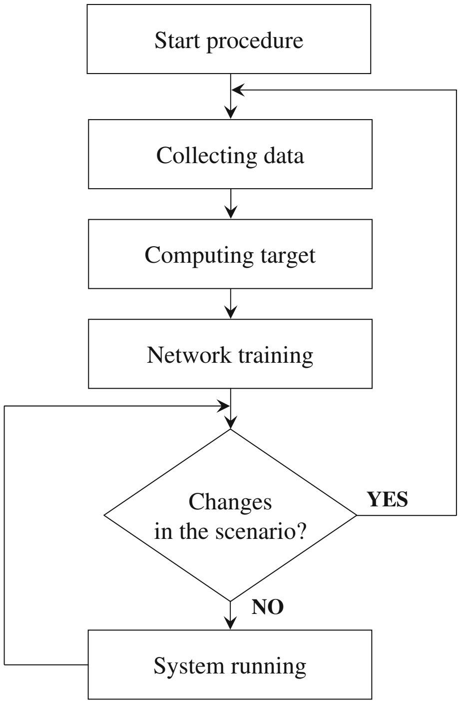


**Fig. 7** *Block diagram* for the ADHDP training procedure

optimized by minimizing some aspects only if the target is appropriately defined. The main goal of this work is to minimize the costs in terms of imported and exported energy towards the grid, so the utility function (included in the target) to be minimized is:

$$U(t) = C(t)P_g(t) \quad \forall t \tag{16}$$

where $C(t)$ is the cost rate for electricity, and $P_g(t)$ is the power acquired from the grid. In this way, the controller minimizes the power acquired from the grid, related also to the electricity cost at time $t$. We can write an energy balance equation to relate all the power flows, with the main purpose of meeting the load demand:

$$P_L(t) = P_g(t) \pm P_b(t) \pm P_r(t) \quad \forall t \tag{17}$$

where $P_L(t)$ is the load demand at time $t$, $P_g(t)$ is always the power acquired from the grid at time $t$, $P_b(t)$ is the charged $(-)$/discharged $(+)$ amount of energy for the battery at time $t$, and $P_r(t)$ is the renewable energy that meets the load $(+)$ or that is sold to the grid $(-)$ at time $t$.

The computation of the utility function is different for all the situations that scenario can provide, so it is necessary to split the procedure in two subproblems in order to find the right value for each case. In the following description, $R(t)$ is the total renewable energy at time $t$, $R_b(t)$ is the power from the renewable sources to the battery at time $t$, $R_g(t)$ is the renewables sold to the grid at time $t$, $BL(t)$ is the battery level at time $t$, $BL_{MIN}$ is the minimum battery capacity, and $BL_{MAX}$ is the maximum capacity of the battery. In general, any model of the battery can be included in this procedure, but the following constraints always remain and they must be satisfied:

$$BL_{MIN} \leq BL(t) \leq BL_{MAX} \quad \forall t \tag{18}$$

$$B_{rateMIN} \leq B_{rate}(t) \leq B_{rateMAX} \quad \forall t \tag{19}$$

where $B_{rateMIN}$ and $B_{rateMAX}$ are respectively the smallest and the greatest rate for the battery model (the rate is called $A_b$ in Sect. "Resource Modelling").

**Case 1**: $R(t) \leq P_L(t)$

In this case, available renewables are smaller than the load demand, so all the renewable energy can meet partially the load, so there is power flow neither from renewables to the battery nor from renewables to the grid.

$$R_b(t) = 0 \tag{20}$$

$$R_g(t) = 0 \tag{21}$$

So, combining (16) and (17), the following utility function is obtained:

$$U(t) = C(t)[P_L(t) - R_L(t)] \pm P_b(t)C(t) \tag{22}$$

where in this case $R_L(t) = P_r(t)$, because all the available renewable energy is used to meet partially the load demand, and the case study $-P_r(t)$ from the second member of (17) is discarded, because renewables cannot be sold to the grid in this situation. The equation (22) can be seen as the equivalent of:

$$U(t) = C(t)P_L{'}(t) \pm P_b(t)C(t) \tag{23}$$

where $P_L{'}(t)$ is a smaller load in a similar problem in which there are no renewables. Now, the critic network computes iteratively the output in order to find that control (charge, discharge or idle) that implies the smallest cost, and after this, the critic network is trained with the just found target.

**Case 2**: $R(t) > P_L(t)$

In this case, renewables are greater than the load demand, so they must be managed with more criteria than in Case 1. At first, the load demand is fully met with renewable energy, and battery can be charged with the surplus energy. At this point, we must distinguish other two cases to manage renewable energy.

**Case 2a**: $BL(t) + R(t) - P_L(t) \geq BL_{MAX}$

The renewables are enough to charge the battery fully, but obviously, the maximum capacity of the battery cannot be exceeded because of (18), so the remaining part is sold to the grid (otherwise, this energy could be wasted). Thus, the different flows of renewable energy to the battery and to the grid are computed as follows:

$$R_b(t) = BL_{MAX}(t) - BL(t) \tag{24}$$

$$R_g(t) = R(t) - P_L(t) \quad (25)$$

In this case, the utility function used for the training target is simply:

$$U(t) = P_b(t)C(t) \quad (26)$$

So, the actions for the battery in relation to the grid are only two: battery can be idle or it can be charged by importing energy from the grid, only if it is convenient for the controller from a pure monetary perspective.

**Case 2b**: $BL(t) + R(t) - P_L(t) < BL_{MAX}$

In this case, battery can be charged by all remaining available renewable energy, and there is no power flow to the grid:

$$R_b(t) = R(t) - P_L(t) \quad (27)$$

$$R_g(t) = 0 \quad (28)$$

Anyway, the utility function also in this case is given by:

$$U(t) = P_b(t)C(t) \quad (29)$$

Different from Case 1 and like Case 2a, when $R(t) > P_L(t)$, the battery can be idle or can be charged from the grid (only if it is convenient for the low cost), but it cannot discharge to meet the load (already satisfied with renewable energy).

In summary, a remarkable training for the critic network schedules the energy in order to use renewables at first to meet the load, then to charge the battery and finally to sell the energy to the grid (if it is convenient for the optimal controller in monetary terms): this behaviour is a symptom of a proper training of the critic neural network, especially when the network is trained on long-period profiles.

Because of the random actions collection, the entire procedure is repeated a certain number of times to find the critic network weights that involve a smaller value in (16), which means a smaller cost to solve the load demand problem. For each iteration training, the weights of the critic network are initialized pseudorandomly as scalar values drawn from a uniform distribution on the [ − 0.01, 0.01] interval. The greater the number of iterations, the greater the possibilities to find out a better solution to the problem; the greater the number of hidden neurons, the smaller the time necessary to have an acceptable solution (however, the results do not seem to be affected by this aspect). Obviously, during the iterations, each time the computed cost reduces, the critic network weights are stored and considered as the solution of the problem.

## Computer Simulations

In this section, four simulations are reported, and each of them refers to a different scenario in terms of available renewable resources. Specifically, one city from the West Coast of United States (San Francisco) and one city of the East Coast (Boston) are chosen, both for winter (first week of January) and summer (first week of August) period, in order to test the optimal controller in different cases, and the performances of this procedure are evaluated for many different configurations.

**Table 1** Wind turbine parameters

| $\eta_w$ | $P_R$ (W) | $V_C$ (m/s) | $V_R$ (m/s) | $V_F$ (m/s) | Rotor radius (m) |
|---|---|---|---|---|---|
| 40 % | 850 | 3 | 15 | 35 | 1.5 |

**Table 2** Battery parameters

| $\eta_b$ | $BL_0$ (kWh) | $BL_{MIN}$ (kWh) | $BL_{MAX}$ (kWh) | $Ch_{rate}/Dh_{rate}$ (kW) |
|---|---|---|---|---|
| 100 % | 80 | 20 | 100 | 16 |

Furthermore, a comparison with a baseline approach has been made in order to highlight the advantages of the method proposed in this paper. In the case, without optimization, the problem is solved with a static and simple *if–then* structure: checks on the use of renewable energy remain unchanged, but obviously a neural network training is not considered, and an optimal controller with memory misses. In the baseline approach, checks and decisions are merely taken, but the ADHDP procedure derives from the same conditions important parameters, and with them the critic neural network is trained on 1-week profiles: so the trained critic network can find the best solution on a long time period, unlike the baseline approach which cannot base own current decisions on the future.

In general, there are some aspects that are in common with all these simulations: we suppose that three houses share in the own system a series of photovoltaic panels, one wind turbine and a battery managed by the controller, while (18) and (19) are satisfied.

Each house is equipped with a photovoltaic panel which covers an area of 13 $m^2$ with an efficiency of 20 %. The wind turbine parameters are shown in Table 1: $\eta_w$ is the efficiency, $P_R$ is the rated power, $V_C$ is the cut-in velocity in *m/s*, $V_F$ is the cut-out velocity in *m/s*, and $V_R$ is the rated velocity of the turbine in *m/s*.

The battery parameters are reported in Table 2: $\eta_b$ is the battery efficiency (for simplicity equal to 100 %), $BL_0$ is the initial level (in *kWh*), $BL_{MAX}$ is the maximum capacity (in *kWh*), and $Ch_{rate}/Dh_{rate}$ refers to the charge/discharge rate (in *kW*). Even if the network training is based on forecasted (or historical) data about renewable energy deriving from any technology, in the simulations of this section, we consider the global horizontal irradiance and wind speed, respectively for solar and eolic historical data taken from [47].

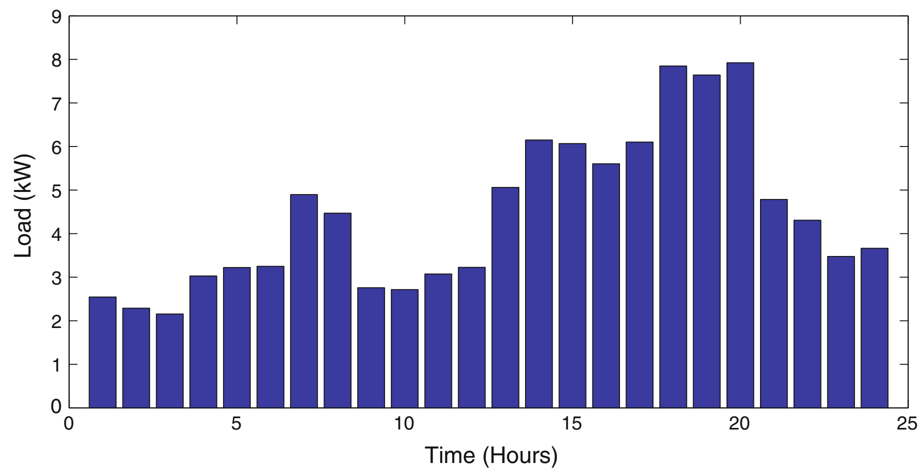


**Fig. 8** A typical residential 24-h load profile

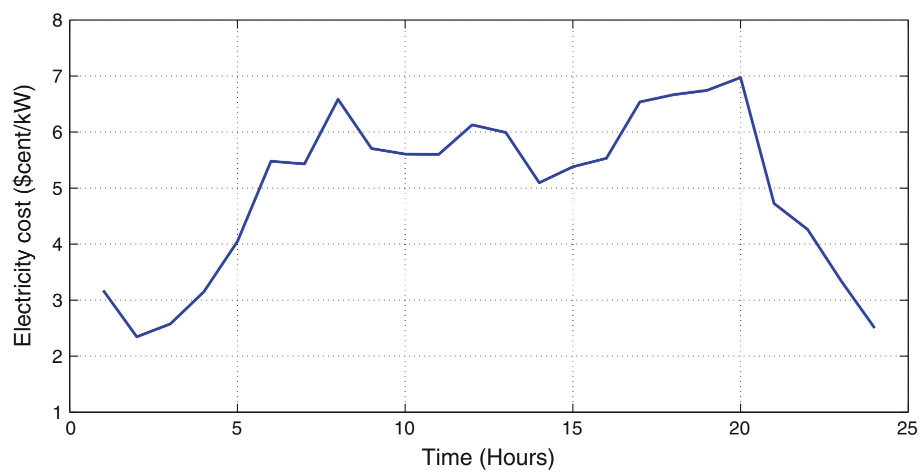


**Fig. 9** A typical daily real-time pricing

Like [40], we consider a typical residential load profile over a week [48]: in Fig. 8, only 24 h are plotted for simplicity, even if the controller considers 168 (24 h per day $\times$ 7 days) samples during its training. The daily load profile is divided into 24-h periods to represent each hour of the day, so the resolution which has been adopted is 1-h step, but it can be greatly increased in the optimization procedure, especially when the training data have a great granularity. Anyway for simplicity, we use a 24-h period each day in this work, like [40]. Furthermore, this choice allows to consider easily the energy consumption: in fact all the values plotted in Fig. 8 refer to the energy required in 1 h (kWh).

The real-time pricing in a residential environment is related to a load management policy, because it is used to shift the electricity use from peaks to lighter load hours to improve power system efficiency. The price of electricity varies from hour to hour in relation to the wholesale price of the market. Typically, the price changes as the demand for electricity changes, so higher is the use of energy imported from the grid and higher is the hourly price. In general, we have a small price peak in the morning and an higher peak during the evening, when the related demand is high. A typical daily real-time pricing, taken from [48], is plotted in Fig. 9. For our simulations, we supposed a selling price equal to half the purchasing cost.

San Francisco: Winter

In this simulation, the scenario in San Francisco during the winter (first week of January) is considered, thanks to the data of renewable energy deriving from the sun and the wind taken from [47]. As one might expect, there is a greater presence of renewable energy in the central hours of each day, and a small negligible portion of renewables during the night in the week, as shown in Fig. 10. Nevertheless, the renewable energy profile in this case has approximately a periodic behaviour during the week, except for some spikes related to the eolic energy (in the third and in the seventh day) that can be distinguished from the solar energy. After the training, the critic network is tested on the same training profiles in order to view which is the best sequence of actions chosen by the controller to minimize the cost. A greater number of hidden neurons implies a smaller computation time without affecting the

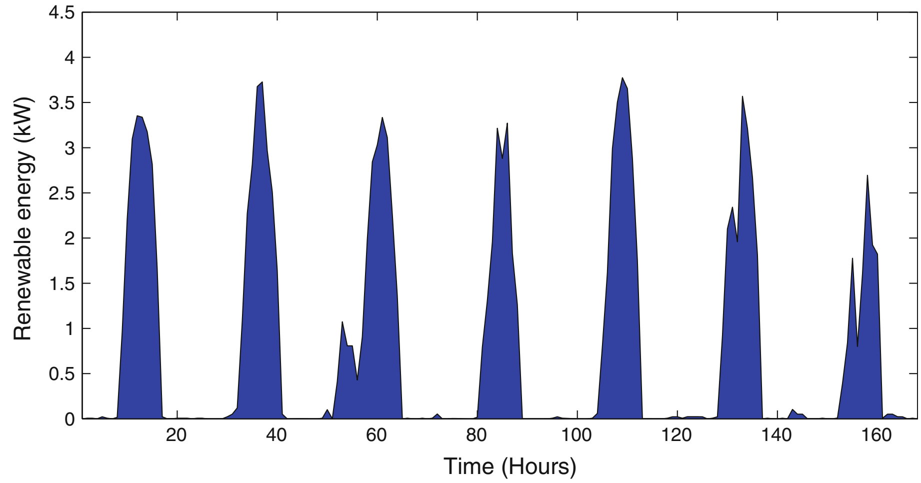


Fig. 10 Available renewable energy (San Francisco, first week of January)

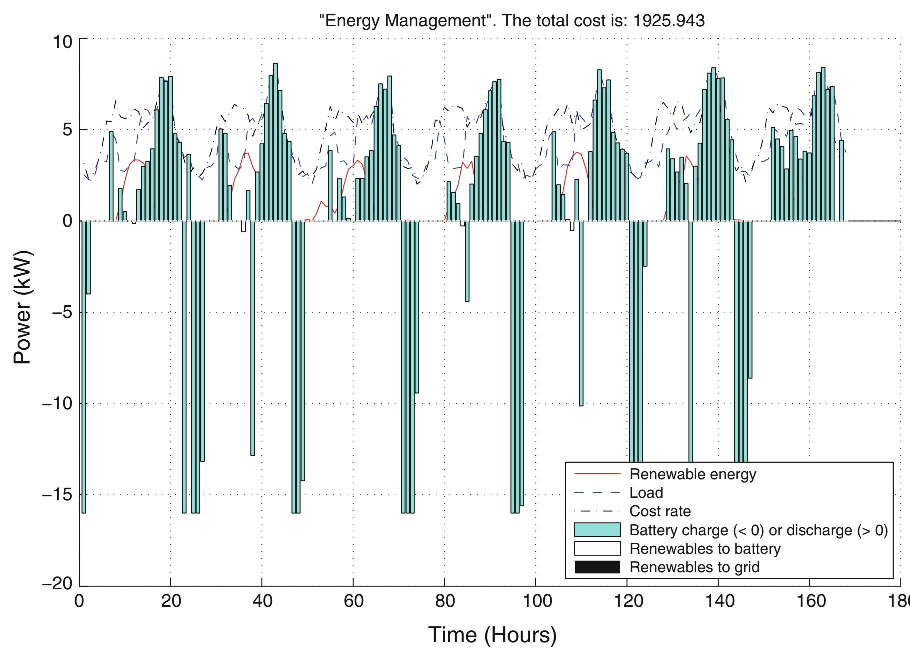


Fig. 11 Optimal scheduling of the battery (San Francisco, first week of January)

solution found. The results are plotted in Fig. 11, and the computed total cost is only 19.26 $ in the week period, unlike a baseline approach, where the total cost is 29.40 $. A proper use of renewable energy shows that the cost can be decreased consistently (in this case, a reduction of 10.14 $ is obtained in one week) when a controller is trained correctly with the profiles of the scenario. So in this case, we have an improvement of 34.5 % in terms of costs.

It is clear that the intelligent system takes actions in an optimal way trying to minimize the product expressed in (16): during the day, the battery meets the load demand (when the load and the electricity cost are high), and during the night, battery is charged from the grid because the cost profile is low. Unfortunately, in this scenario, there are no enough renewables neither to charge consistently the battery nor to export them to the grid. However, the optimal controller reaches a cheap solution, because the battery is always managed with correct criteria and policies.

San Francisco: Summer

Obviously, in this case, the available renewable energy deriving from the sun and the wind are much more than in previous case; in fact, there are higher peak levels of energy that form a periodic profile, as shown in Fig. 12.

For this simulation, the computed total cost is 7.89 $ and the baseline approach provides a cost of 10.63 $, so in this case 2.5 $ is saved and a cost reduction of 25.8 % is achieved. As expected, great amount of renewable energy in August provides a major cost reduction for the San

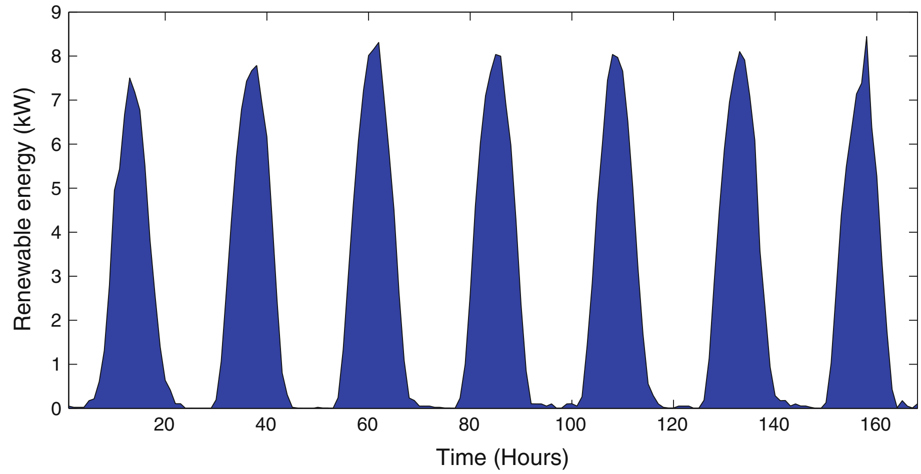


**Fig. 12** Available renewable energy (San Francisco, first week of August)

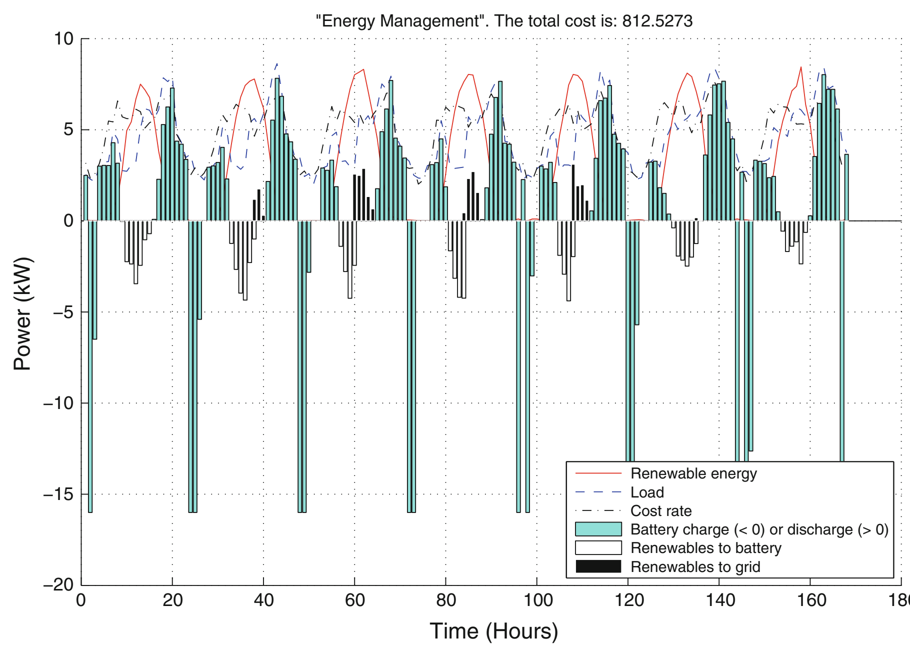


**Fig. 13** Optimal scheduling of the battery (San Francisco, first week of August)

Francisco scenario, due to the exploitation of these resources. In fact, renewables are enough to meet all the load demand during their energy peaks, and they are able also to charge fully the battery: in this favourable case for the customer, there is also surplus renewable energy that is sold, so a further reduction in money is achieved.

Boston: Winter

For the first week of January in Boston, there are no many renewables, and their behaviour is quite different from one day to another (due to both eolic and solar energy), and for this reason, we expect a worse performance in terms of costs, with respect to the case reported in Sect. "San Francisco: Winter".

In fact, in this simulation, the total cost provided by the optimal controller is 21.95 $ while the baseline approach gives a global cost of 29.68 $, so 7.73 $ is saved, and a cost reduction of 26 % is obtained for the Boston scenario during the first week of January.

Boston: Summer

Obviously, during the summer, there is much more renewable energy (deriving from the sun) compared with the winter: in this case, we have higher energy peaks during the day, even if there is a decreasing behaviour in the first part of the week in terms of energy, due to weather, as shown in Figs. 13, 14, 15, and 16.

Anyway, renewables are enough to meet the load in the middle part of the day and to charge the battery (especially in the first part of the week).

Furthermore, some energy in surplus is sold to the grid, and for all these reasons, optimal controller can provide a

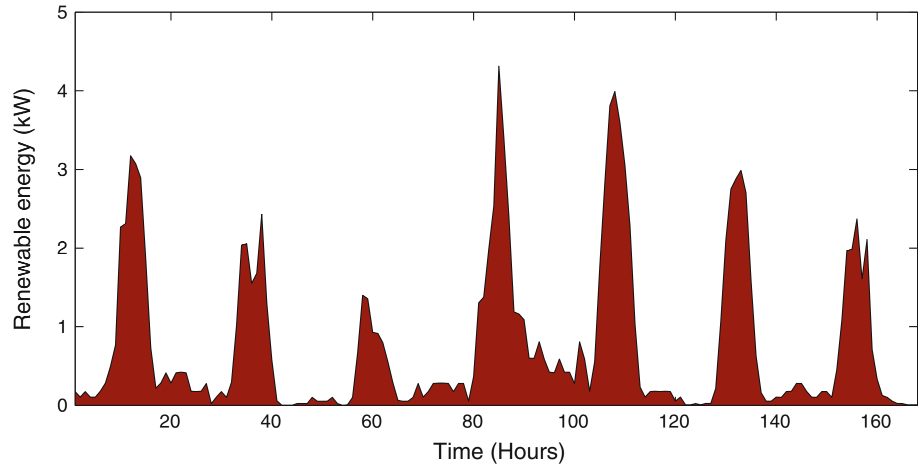


**Fig. 14** Available renewable energy (Boston, first week of January)

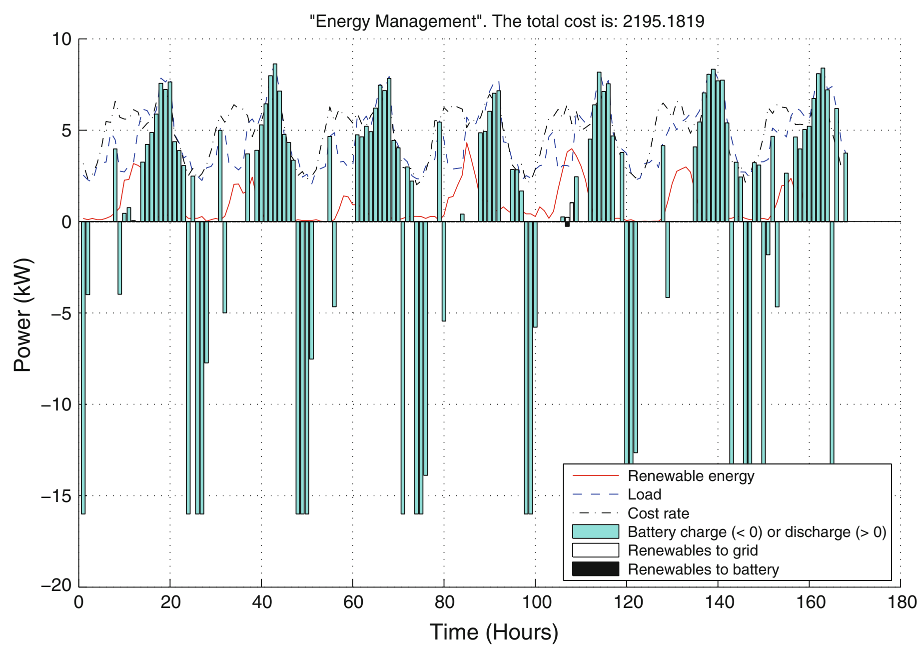


**Fig. 15** Optimal scheduling of the battery (Boston, first week of January)

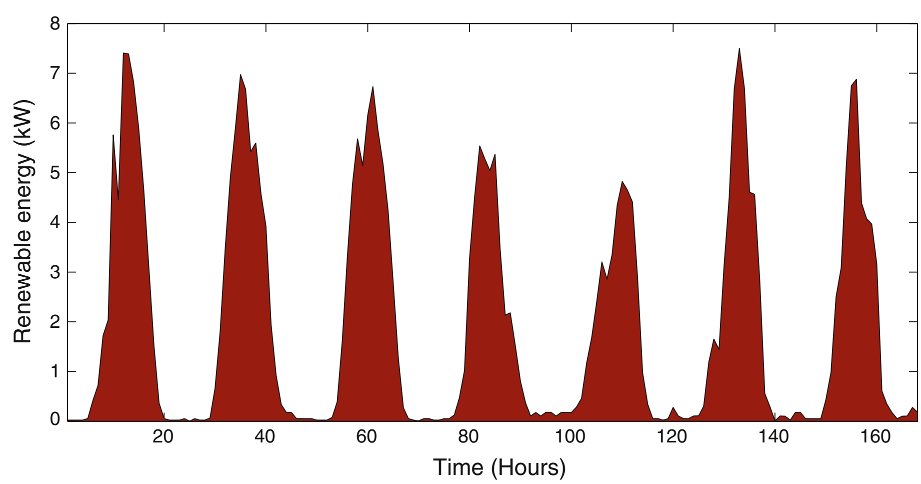


**Fig. 16** Available renewable energy (Boston, first week of August)

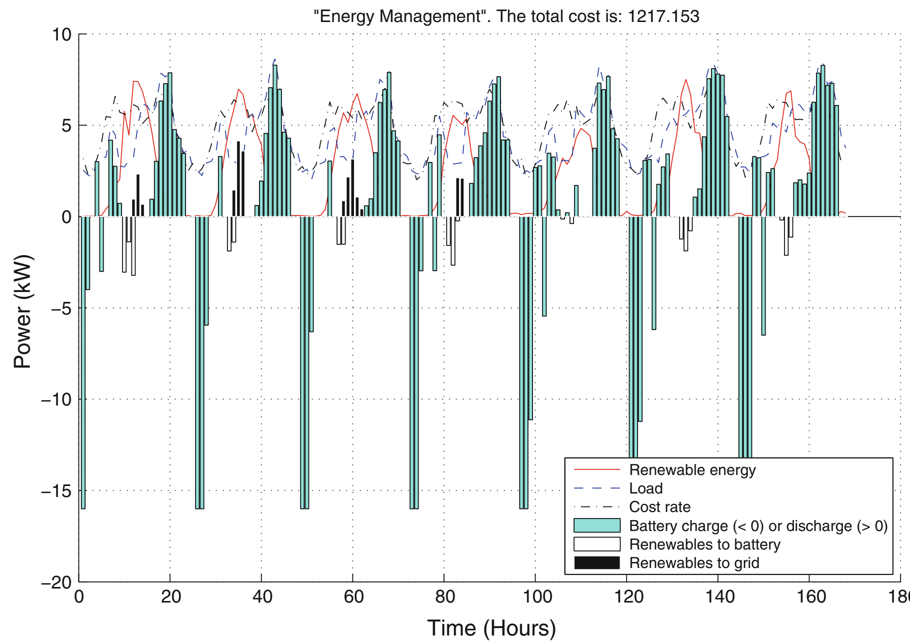


**Fig. 17** Optimal scheduling of the battery (Boston, first week of August)

**Table 3** Comparison of scenarios results

| City | Period | Optimal controller ($) | Nonoptimal controller ($) | Money saving (%) |
|---|---|---|---|---|
| San Francisco | January | 19.26 | 29.40 | 34.5 |
| | August | 7.89 | 10.63 | 25.8 |
| Boston | January | 21.95 | 29.68 | 26.0 |
| | August | 12.17 | 18.37 | 33.8 |

total cost of 12.17 $ lower than the cost of baseline approach which is 18.37 $, so 6.2 $ is saved, and a cost reduction of 33.8 % is achieved for the Boston scenario during the first week of August (Fig. 17).

In Table 3, all the results related to different scenarios are reported for a comparison. As expected, for all the simulations, the optimal controller gives better results with respect to the baseline approach, both for January and August, offering a consistent money saving in all examined cases.

## Conclusions

A self-learning procedure has been realized in order to give a controller ability to manage in an optimal way the energy, minimizing the costs for a certain horizon. In addition to [40], in this work, a more complex system has been realized, because renewable sources are considered in the optimization scheme and the controller manages the battery according to the renewable profiles, and different power flows are provided. This system is flexible because different critic networks in terms of hidden neurons can be designed. The training algorithm for the critic network is robust and suitable to operate in different scenarios, because also renewable energy is considered in the system differently from [40]: if some modifications of the profiles happen, the neural network training starts in order to meet the new requirements.

Furthermore, a new training could be avoided if some profiles and related weights matrices have been previously stored: in fact, it is possible to check which set of saved profiles is the nearest to the new set of scenario profiles. If a parameter sets provide a global error smaller than a certain threshold, then the related weights matrix is chosen as the best for the critic network to solve the optimization problem.

Performed simulations have shown that the optimal controller based on the self-learning scheme gives better results compared with the baseline approach: in fact, the optimal controller takes decisions for the best energy scheduling basing its criteria on the future, instead of the nonoptimal controller that provides a solution only considering the current time. In this work, an optimal controller which also includes renewable energy for its training has been realized, and the results confirm the effectiveness and the robustness of this automatic system. Future efforts will aim to extend the horizon of the training in order to have a behaviour for the controller averaged on a longer period, until the forecasted data are reliable. Moreover, the same optimization scheme could be applied to a residential area, where the global load demand from different groups of houses must be met, according to the minimization criteria to reduce cost and optimize the renewable energy that can be exchanged from point to point in the smart grid.

**Acknowledgments** This work was supported in part by the National Natural Science Foundation of China under Grants 60904037, 60921061, and 61034002, in part by Beijing Natural Science Foundation under Grant 4102061, and in part by China Postdoctoral Science Foundation under Grant 201104162.

## References

1. Amin SM, Wollenberg BF. Toward a smart grid: power delivery for the 21st century. IEEE Power Energy Mag 2005;3(5):34–41.
2. Garrity TF. Getting smart. IEEE Power Energy Mag 2008;6(2): 38–45.
3. Meliopoulos APS, Cokkinides G, Huang RK, Farantatos E, Choi S, Lee Y, Yu X. Smart grid technologies for autonomous operation and control. IEEE Trans Smart Grid 2011;2(1):1–10.
4. Klein KM, Springer PL, Black WZ. Real-time ampacity and ground clearance software for integration into smart grid technology. IEEE Trans Power Deliv 2011;25(3):1768–1777.
5. Li F, Qiao W, Sun H, Wan H, Wang J, Xia Y, Xu Z, Zhang P. Smart transmission grid: vision and framework. IEEE Trans Smart Grid 2010;1(2):168–177.
6. Roncero JR. Integration is key to smart grid management. In: Proceeding of the IET-CIRED seminar smartgrids for distribution, 2008. p. 1–4.
7. Sheble GB. Smart grid millionaire. IEEE Power Energy Mag 2008;6(1):22–28.
8. Belvedere B, Bianchi M, Borghetti A, Paolone M. A microcontroller-based automatic scheduling system for residential microgrids. In: Proceeding of IEEE Bucharest power technology, 2009. pp. 1–6.
9. Chiang SJ, Chang KT, Yen CY. Residential photovoltaic energy storage system. IEEE Trans Industr Electron 1998;45(3): 385–394.
10. Momoh JA, Wang Y, Eddy-Posey F. Optimal power dispatch of photovoltaic system with random load. In: Proceeding of IEEE Power Engineering Society, 2004. p. 1939–1945.
11. Maly DK, Kwan KS. Optimal battery energy storage system (BESS) charge scheduling with dynamic programming, In: Proceeding of IEEE science, measurement and technology, 1995. p. 453–458.
12. Bakirtzis AG, Dokopoulos PS. Short term generation scheduling in a small autonomous system with unconventional energy system. IEEE Trans Power Syst 1988;3(3):1230–1236.
13. Corrigan PM, Heydt GT. Optimized dispatch of a residential solar energy system. In: Proceeding of the North American power symposium, 2007. p. 183–188.
14. Rupanagunta P, Baughman ML, Jones JW. Scheduling of cool storage using non-linear programming techniques. IEEE Trans Power Syst 1995;10(3):1279–1285.
15. Liang RH, Liao JH. A fuzzy-optimization approach for generation scheduling with wind and solar energy systems. IEEE Trans Power Syst 2007;22(4): 1665–1674.
16. Cau TDH, Kaye RJ, Jones JW. Multiple distributed energy storage scheduling using constructive evolutionary programming. In: Proceeding of the IEEE international conference on power industry computer applications, 2001. p. 402–407.
17. Fung CC, Ho SCY, Nayar CV. Optimization of a hybrid energy system using simulated annealing technique. In: Proceeding of IEEE international conference on computer communication control power engineering, 1993. p. 173–191.
18. Pedrasa MAA, Spooner TD, MacGilland IF. Scheduling of demand side resources using binary particle swarm optimization. IEEE Trans Power Syst 2009;24(30):1173–1181.
19. Lee TY. Operating schedule of battery energy storage system in a time-of-use rate industrial user with wind turbine generators: a multipass iteration particle swarm optimization approach. IEEE Trans Energy Convers 2007;22(3):774–782.
20. Zhang H, Quan Y. Modeling, identification and control of a class of nonlinear system. IEEE Trans Fuzzy Syst 2001;9(2): 349–354.
21. Cui L, Zhang H, Zhang X, Luo Y. Data-based adaptive critic design for discrete-time zero-sum games using output feedback. In: IEEE symposium on adaptive dynamic programming and reinforcement learning, Paris, France, 2011. p. 190–195.
22. Zhang X, Zhang H, Wang X, Luo Y. A new iteration approach to solve a class of finite-horizon continuous-time nonaffine nonlinear zero-sum game. Int J Innov Comput Inf Control 2011;7(2):597–608.
23. Ikonen E, Najim K. Multiple model-based control using finite controlled markov chains. Cogn Comput 2009;1(3):234–243.
24. Bishop JM, Caines S. Creativity and autonomy in swarm intelligence systems. Cogn Comput. 2012; in press. doi:10.1007/s12559-012-9130-y.
25. Werbos PJ. A menu of designs for reinforcement learning over time. In: Miller WT, Sutton RS, Werbos PJ (eds). Neural Networks for Control. Cambridge, MA: MIT Press; 1991. p. 67–95.
26. Wang F, Zhang H, Liu D. Adaptive dynamic programming: an introduction. Comput Intell Mag 2009;4(2):39–47.
27. Wang F, Jin N, Liu D, Wei Q. Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with $\epsilon$-error bound. IEEE Trans Neural Netw 2011;22(1): 24–36.
28. Wei Q, Zhang H, Dai J. Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions. Neurocomputing 2009;72(7–9), 1839–1848.
29. Wei Q, Liu D. An iterative $\epsilon$-optimal control scheme for a class of discrete-time nonlinear systems with unfixed initial state. Neural Networks, 2012; in press, doi:10.1016/j.neunet.2012.02.027.
30. Prokhorov DV, Wunsch DC. Adaptive critic designs. IEEE Trans Neural Netw 1997;8(5):997–1007.
31. Barto AG. Reinforcement learning and adaptive critic methods. In: White DA, Sofge DA, editors. Handbook of intelligent control: neural, fuzzy and adaptive approaches. New York: Van Nostrand Reinhold; 1992.
32. Bertsekas DP, Tsitsiklis JN. Neuro-dynamic programming. MA: Athena Scientific; 1996.
33. Liu D, Xiong X, Zhang Y. Action-dependent adaptive critic designs. In: Proceeding of the INNS-IEEE international joint conference on neural network, 2001. p. 990–995.
34. Murray JJ, Cox CJ, Lendari GG, Saeks R, Adaptive dynamic programming. IEEE Trans Syst Man Cybern C Appl Rev 2002;32(2):140–153.
35. Si J, Wang YT. On-line learning control by association and reinforcement. IEEE Trans Neural Netw 2001;12(2):264–276.
36. Balakrishnan SN, Biega V. Adaptive-critic-based neural networks for aircraft optimal control. In: Proceeding of the second international conference on computer modeling and simulation. 2010. p. 2288–2293.
37. Sutton RS, Barto AG. Reinforcement learning: an introduction. MA: MIT Press; 1988.
38. Werbos PJ. Advanced forecasting methods for global crisis warning and models of intelligence. In: Proceeding of the American control conference, 2008. p. 25–38.
39. Werbos PJ. Approximate dynamicprogramming for realtime control and neural modeling. Handbook of intelligent control: neural, fuzzy and adaptive approaches. New York: Van Nostrand Reinhold; 1992.

40. Huang T, Liu D. Residential energy system control and management using adaptive dynamic programming. In: Proceeding of IEEE international joint conference on neural networks (IJCNN), 2011. p. 119–124.
41. Miceli R, La Cascia D, Di Stefano A, Fiscelli G, Giaconia C. Impact of novel energy management actions on household appliances for money savings and CO2 emissions reduction. In: Proceeding of the EVER09 conference, Principato di Monaco, Monaco, 2009. p. 1–5.
42. Markvart T. Solar electricity. 2nd ed. New York, NY: Wiley; 2000.
43. Chedid R, Akiki H, Rahman S. A decision support technique for the design of hybrid solar-wind power systems. IEEE Trans Energy Convers 1998;13(1):76–83.
44. Linden D, Reddy TB. Handbook of batteries. 3rd ed. New York, NY: McGraw-Hill; 2002.
45. Bellman RE. Dynamic programming. New Jersey: Princeton University Press; 1957.
46. Hagan MT, Menhaj MB. Training feedforward networks with the Marquardt algorithm. IEEE Trans Neural Netw 1994;5(6): 989–993.
47. National renewable energy laboratory (NREL) of U.S. Department of Energy: http://www.nrel.gov/rredc/, Office of Energy Efficiency and Renewable Energy, operated by the Alliance for Sustainable Energy, LLC.
48. ComEd, USA. Accessed 16 May 2010. [Online] Available: http://www.thewattspot.com/.