A Comparison between Different Optimization Techniques for Energy Scheduling in Smart Home Environment

Francesco De Angelis, Matteo Boaro, Danilo Fuselli, Stefano Squartini, and Francesco Piazza

Dipartimento di Ingegneria dell'Informazione
Università Politecnica delle Marche, Ancona, Italy

Abstract. Nowadays a correct use of energy is a crucial aspect, in fact cost and energy waste reduction are the main goals that must be achieved. To reach this objective an optimal energy management must be obtained through some techniques and optimization algorithms, in order to provide the best solution in terms of cost. In this work a comparison between different methods for energy scheduling is proposed and some analytical results are reported, in order to offer a clear overview for each technique, in terms of advantages and disadvantages. A residential scenario is considered for computer simulations, in which a system storage and renewable resources are available and exploitable to match the user load demand.

1 Introduction

The concept of smart grid faces many electrical power engineering requirements, so different solutions can be achieved for each specific application, from the generation to the customer level, where Computational Intelligence techniques can be very useful [1,2]. In this area the energy management has a main role in order to reduce costs and avoid its waste, also in a micro-grid for a residential or domestic scenario. A joint task and energy optimization framework has been already implemented [3], but several methods have been developed to accomplish efficiently only energy scheduling: linear programming techniques [4], Particle Swarm Optimization (PSO) [5], Fuzzy-Logic [6], Artificial Neural Networks [7], and also Adaptive Dynamic Programming (ADP) [8].

In this paper, the attention is focused in home environment connected to the main grid and also a photovoltaic (PV) system with a battery is considered to increase the saving. The load profile must be always satisfied managing renewable energy, battery and electrical grid in order to reduce costs. Therefore an optimal battery controller must be obtained, whose control policy is to minimize the energy cost imported from the grid managing the battery actions

B. Apolloni et al. (Eds.): *Neural Nets and Surroundings*, SIST 19, pp. 311–320.
DOI: 10.1007/978-3-642-35467-0_31

(charge/discharge) and knowing the forecasted renewable resources, load profile and energy price. In this work a comparison among six different methods, the best promising chosen from literature, for battery management is proposed: an overview for each technique is provided and also a comparison is reported, in terms of advantages and disadvantages. Some of these methods have been already presented in other papers like [9] based on Adaptive Dynamic Programming (ADP) and on Particle Swarm Optimization (PSO) technique; some other methods, based on Linear Programming (LP) and PSO, are introduced for the first time.

The system description is reported in Section 2, the analytical issues of each optimization algorithm are discussed in Section 3 and the simulated scenario is shown in Section 4. Section 5 deals with the conducted computer simulations whereas Section 6 draws the work conclusions.

2 Home Energy System Description

The proposed home model is composed of a main electrical grid, external PV array, storage system and Power Management Unity (PMU), that ensures the meeting of load demand. As reported in Fig. 1, PMU unit (energy scheduler) manages the energy flows: battery can be charged from the grid and/or from PV, moreover if necessary it can be discharged to supply the load. If there is exceeded energy from PV not usable from the system, it is sold to the main grid. In addition the battery must satisfy the following constraints:

1. The charging and discharging rate can not be exceeded.
2. Battery level must be always included between the upper and lower bound.

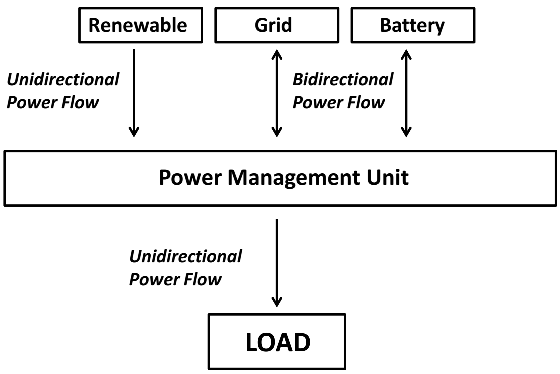

Fig. 1. Power Flows

3 Optimization Algorithms

3.1 Linear Programming Technique: LP Algorithm

The implemented algorithm is based on the "Linear Programming" (LP) paradigm. Its objective consists in maximizing or minimizing a given function, considering some constraints according to:

$$\max\ f(x) = c^T x \quad \text{or} \quad \min\ f(x) = c^T x$$

$$\text{subject to} \quad Ax \leq b \quad \text{or} \quad Ax \geq b$$

where $x \geq 0$, $x \in \mathbb{R}^{n\times 1}$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^{m\times 1}$, $c \in \mathbb{R}^{1\times n}$.

The cost function $U(t)$ used in this work with this optimization algorithm is the following:

$$U(t) = \sum_{t=1}^{T} \{ \left[L_u(t) - R_u(t) + Ch_u(t) - Dh_u(t) \right] \cdot C(t) \} \tag{1}$$

where $L_u(t)$ is the load demand at temporal slot t, $R_u(t)$ is the amount of renewables exploited at time t, Ch_u is the amount of energy used for charging the system storage, Dh_u is the discharged energy used for meeting (partially or totally) the load, $C(t)$ is the electricity cost at time t and T is the work horizon.

This kind of battery controller works in offline way, bacause it optimizes the given cost function in all the time horizon T.

The constraints of this problem, for $1 \leq t \leq T$, are reported as follows:

- Positive function: $L_u(t) - R_u(t) + Ch_u(t) - Dh_u(t) \geq 0$
- Do not exceed renewables: $R_u(t) \leq R(t)$
- Load is fixed $L_u(t) = L(t)$
- Charge and discharge limits: $Ch_u(t) \leq Ch_{rate}, \quad Dh_u(t) \leq Dh_{rate}$
- Battery level: $SL(t) = SL(t-1) + Ch_u(t) - Dh_u(t)$
- Battery level limits: $SL_{MIN} \leq SL(t) \leq SL_{MAX}$

where $R(t)$ and $L(t)$ are respectively the total renewable avaiable and load demand, while SL is the State of Charge (SoC) of the system storage.

3.2 Particle Swarm Optimization Algorithm

PSO is a technique inspired to certain social behaviors, and it is used to explore a search parameter space to find values allowing to minimize an objective function [10]. The PSO algorithm works by maintaining simultaneously various candidate solutions (particles in the swarm) in the search space. In PSO, the coordinates of each particle represent a possible solution associated with two vectors, the position x and velocity v vectors in N-dimensional search space. A swarm consists of a number i of particles "or possible solutions" that flies through the feasible solution space to find the optimal one. Each particle updates its position x_i on the basis of its own best exploration p_i, its best swarm overall experience p_g, and its previous velocity vector $v_i(k-1)$ according to (2) and (3).

$$x_i(k) = x_i(k-1) + v_i(k) \tag{2}$$

$$v_i(k) = v_i(k-1) + \rho_1 \cdot rand_1 \cdot \big[p_i - x_i(k-1)\big] + \rho_2 \cdot rand_2 \cdot \big[p_g - x_i(k-1)\big] \tag{3}$$

where ρ_1 and ρ_2 are two positive correction factors, k is the iteration step while $rand_1$ and $rand_2$ are two random numbers $[0.0, 1.0]$. The PSO algorithm can be described in general as follows:

1. For each particle, randomly initialize the position and velocity vectors with the same size as the problem dimension.
2. Measure the fitness (utility function value) of each particle and store the particle with the best fitness value (minimum utility function value).
3. Update velocity and position vectors according to (2) and (3) for each particle.
4. Repeat steps 2 and 3 until a termination criterion is satisfied.

As already done in [9] we introduce in (4) an utility function that must be minimized for each temporal slot t.

$$U(t) = \sqrt{\big\{[L(t) - R(t) + u(t)] \cdot C(t)\big\}^2 + \big\{SL_{cap} - [SL(t) + u(t)]\big\}^2} \tag{4}$$

where $u(t)$ is the optimized value of battery charge ($u(t) > 0$) or discharge ($u(t) < 0$) that must be found by the algorithm for each time t, SL_{cap} is the battery capacity and $SL(t)$ is the actual battery level. Minimizing $U(t)$ means charging the battery when renewable is high and/or when cost is low, and discharging the battery when renewable is lower than the load and/or the cost is high. Obviously $u(t)$ must satisfy the two battery constraints discussed in Section 2. If one of these constraints is not satisfied, the obtained solution $u(t)$ is not valid and must be discarded. So the function is multiplied with a penalty factor which is set to a higher value.

It is important to note that this battery controller works in online way, because the cost function is evaluated step by step without knowing the energy horizon profiles.

3.3 Extended Particle Swarm Optimization Algorithm

Similar to the scheme proposed in Section 3.2, an extended version of PSO has been realized. The operation is not online anymore, but offline in order to give an optimal solution on an extended period, for which all scenario profiles are considered in the work horizon, as well as the forecasted data about renewable energy. Differently from (4), the utility function adopted in this case does not include battery terms, and also a sum over the entire period is considered, in order to provide an optimization for the entire work horizon T.

$$U(t) = \sum_{t=1}^{T} \sqrt{\left\{[L(t) - R(t) + u(t)] \cdot C(t)\right\}^2} \tag{5}$$

Obviously $u(t)$ must satisfy the battery constraints discussed in Section 2.

3.4 Adaptive Dynamic Programming

Combining approximate dynamic programming and reinforcement learning, Werbos proposed a new optimization technique [11], whose goal is to design an optimal control policy, which can be able to minimize a given cost function called "utility function" (especially in nonlinear and noisy environments), adapting two neural networks: the Action Network and the Critic Network. The Action Network, taking the current state as input, has to drive the system to a desired one, providing a control to the latter. The Critic Network, knowing the state and the control provided by the Action Network, has to check its performances and return to the Action Network a feedback signal to reach the optimal state over time. To check Action performances, the Critic Network approximates the following Bellman equation associated with optimal control theory:

$$J(t) = \sum_{i=0}^{\infty} \gamma^i U(t+i) \tag{6}$$

where γ is the discount factor (0, 1] and $U(t)$ is the utility function.

As already implemented in [9], that was inspired by [8], an Action-Dependent Heuristic Dynamic Programming (ADHDP) model free approach is adopted for the design of an optimal controller, whose goal is to manage the battery, knowing forecasted data (Load, Price, Renewable Energy), in order to save money during an overall time-horizon. As reported in Fig. 2 the input to the Action network is the system state $x(t)$, and the output $u(t)$ is the amount of energy used to charge or discharge the battery; the input of the Critic Network consists of the current system state and the current control provided by the Action Network.

The used Critic network is composed by 15 linear neurons in input, 40 sigmoidal hidden neurons and 1 linear in output, while Action network by 4 linear neurons in input, 40 sigmoidal hidden neurons and 1 linear in output. In this study the proposed utility function $U(t)$ is reported in (7).

$$U(t) = \sqrt{\left\{\left[L(t) - R(t) + u(t)\right] \cdot C(t)\right\}^2} \tag{7}$$

where $u(t)$ is the optimized value of battery charge ($u(t) > 0$) or discharge ($u(t) < 0$) that must be found for each time t. Obviously $u(t)$ must satisfy the battery constraints discussed in Section 2 and in this case it is forced after Action Network output to respect these limits. When the utility function is minimized the control policy is optimal and the cost is the lowest. The squaring of the equation is necessary to avoid that $U(t)$ is negative.

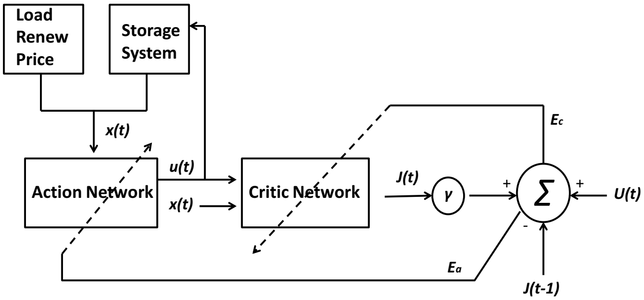

Fig. 2. ADHDP Scheme

The online training is based on the "Backpropagation algorithm": the iterative training used for both neural networks is repeated for a fixed number of epochs and explained step by step below:

1. The Action and Critic weights are initialized before the training: with random values [-1,1] or pre-trained with extended PSO.
2. Train Critic Network refreshing its weights using computing Critic error (E_c), then refresh Action Network computing Action error (E_a).
3. Evaluate the system perforance computing the total cost to minimize in the time horizon. If the cost decreases, the control policy is improving and the new action weights are the best; if not, revert to old action weights and add a small random perturbation. Then restart the training from Step 2.

Also this type of battery controller is an offline one because it optimizes the utility function during all the time horizon T, since it employs forecasted data.

3.5 Self-Learning Procedure Based on ADHDP Scheme

Like in [12] this optimization procedure is based on a simplified ADHDP scheme because only few actions can be done by the controller, in fact the battery is limited to a ternary choice (charge, discharge or idle). In this way we consider only a critic network in the scheme. If a network is trained correctly, 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. The training procedure is the following:

1. Data are collected: the action is taken randomly, the state is characterized by the cost rate, the load profile, the battery level and the renewable energy;
2. Compute U(t) and Q(t) in order to obtain the target, since the training is based on the mapping: $\{x(t-1); u(t-1)\} \longrightarrow \{U(t) + \gamma Q(t)\}$, where $x(t-1)$ and $u(t-1)$ are the previous state and control, $U(t)$ is the actual utility function, $\gamma \in (0,1]$ is a discount factor and $Q(t)$ is the actual critic network output;

3. The critic network is trained with the “Levenberg-Marquardt backpropagation” algorithm;
4. Eventually the neural network can be re-trained whenever there are consistent changes in the scenario.

The utility function that we want to minimize is:

$$U(t) = [L(t) - R(t) + u(t)] \cdot C(t) \quad (8)$$

where $u(t) = u'(t)q(t)$ is the battery charge ($u(t) > 0$) or discharge ($u(t) < 0$), $u'(t)$ is the battery action $(1, -1, 0)$ and $q(t)$ is the charging/dicharging battery quantity. Also in this case $u(t)$ must satisfy the battery constraints discussed in Section 2 and in this case it is forced to respect these limits.

Also this battery control strategy is offline and considesr the overall working horizon T.

4 Simulated Scenario

All the simulations reported in Section 5 refer to the same scenario: a system storage is supposed to be available, as well as renewable resources deriving from solar energy. In Texas, in Austin city, we consider an area of 30 m^2 covered by some photovoltaic (PV) panels, whose efficiency is 30 %, and irradiation data is taken from [13]. According to [14], the available renewable energy is computed with $P = GHI \cdot \eta_{pv} \cdot A_{pv}$, where GHI is the Global Horizontal Irradiance in Wh/m^2 received on a horizontal surface, η_{pv} is the efficiency of the PV and A_{pv} is the total area of the PV panel in m^2. The difference between the simulations is the considered time horizon: 48-h, 96-h and 168-h horizons are simulated in order to test performances of each technique for the short and long-term period. In the Tab. 1 system storage parameters used for all the simulations are reported, and since the resolution time used is one hour, kWh and kW agree so we can consider the same unit of measurement both for energy and power parameters.

Table 1. Storage system parameters (in kW)

SL_0	SL_{MIN}	SL_{MAX}	Ch_{rate}	Dh_{rate}
5	0	10	1	1

In Tab. 1 SL_0, SL_{MIN} and SL_{MAX} are respectively the initial, minimum and maximum State of Charge (SoC), while Ch_{rate} and Dh_{rate} are the maximum charge and discharge rate of the considered storage system. The efficiency η has been considered equal to 100%.

5 Computer Simulations

In this study a battery management problem is considered in order to minimize the imported energy from the main grid and increase the money saving. The cost (expressed in dollars) related to each optimization technique for three different time horizon (48-h, 96-h and 168-h) is reported in Tab. 2; while the money saving, compared to an online baseline algorithm applied in the same scenario, is given in percentage in Tab. 3. The baseline approach follows the next simple rules for each time step:

– if the load is greater than the available renewable energy, the difference is supplied discharging the battery (according with Dh_{rate} in Tab. 1). If the battery support is not enough, the needed energy to supply totally the load is imported from the main grid;
– if the available renewable energy is greater than the load demand, the surplus is used to charge the battery (according with Ch_{rate} in Tab. 1). If the battery is already full or the surplus is greater than the charging rate, the amount of energy in excess, not usable in other ways, is sold to the main grid.

Table 2. Cost comparison for energy scheduling (in $)

T	LP	ADP	$Ext\ PSO$	$Self-L\ ADP$	PSO	$Baseline$
48 h	6.46	6.48	6.49	6.63	6.97	7.12
96 h	12.80	12.84	12.86	13.08	13.74	14.09
168 h	25.08	25.16	25.22	25.75	26.04	26.79

Looking at the results reported in Tables 2-3 it is evident that the LP offline algorithm provides the best solution, due to the linear behavior of the energy scheduling problem. Furthermore it has no convergence problems and the computational cost is very low, but as mentioned this method can not work in real-time and it needs forecasted data relative to the considered horizon. Whenever a linear approximation of a nonlinear model is not valid, this linear approach cannot be used and a different method should be chosen.

Table 3. Money saving (in percentage) compared to baseline algorithm

T	LP	ADP	$Ext\ PSO$	$Self-L\ ADP$	PSO
48 h	9.27%	8.99%	8.85%	6.89%	2.10%
96 h	9.21%	8.87%	8.73%	7.17%	2.55%
168 h	6.38%	6.08%	5.86%	3.88%	2.80%

Differently, PSO is an online algorithm able to work without forecasted data, and it optimizes step by step a given utility function, with very low computational complexity. For this reason it is not possible to offer an optimal solution over a large time horizon, so the cost reduction is limited. Different is the case of Extended PSO, which gives a good solution over a considered work horizon, even if its performances degrade gradually while the horizon increases (due to the fact that the number of unknowns increases). This offline method needs forecasted data relative to all the temporal steps considered, and it has a computational cost higher than the previous mentioned techniques.

As mentioned, the Extended PSO is used to pretrain the neural networks used in ADP method. The ADP, adapting the Action and Critic weights, can improve the performances of the Extended PSO and find a better solution with an higher saving. This saving is remarkable especially in longer periods, where the ADP overcomes the performances of the Extended PSO, which finds an optimal solution over longer period because of the variables number increase.

The initial computational cost of the ADP algorithm is not very small, but it has the advantage to adapt itself quite quickly when the time horizon and the scenario change; in fact the optimization process can continue from the best weights of the neural networks stored on the previous training step, and it does not need to restart from the beginning like the other proposed methods.

Finally, the self-learning procedure based on ADHDP scheme offers a trade off between the goodness of the solution and the computational cost: slightly sacrificing the cost reduction, a much shorter time spent for the neural network training can be obtained.

6 Conclusions

A comparison between different optimization techniques for the energy scheduling in a smart home environment has been proposed, and an evaluation in monetary terms has been given in order to highlight the performances also referred to a baseline approach. Although all the shown methods are valid, the LP algorithm offers the best solution, since the energy scheduling problem is linear. Also the other techniques provide good solutions, and obviously they could solve optimization problems in more complex and nonlinear systems. PSO works fine in online configuration, but its extended offline version gives a better cost reduction. Both the two ADP procedures can be advantageous concerning cost reduction and computational cost for long-term periods. In conclusion, the choise among these different offline and online methods depends on the linearity of the problem, the computational cost and the availability of forecasted profiles.

References

1. Harley, R.G., Liang, J.: Computational Intelligence in Smart Grids. In: IEEE SSCI 2011 Symposium Series on Computational Intelligence CIASG (2011)
2. Venayagamoorthy, G.K.: Potentials and Promises of Computational Intelligence for Smart Grids. In: 2009 IEEE Power and Energy Society General Meeting, pp. 1–6 (2009)

3. De Angelis, F., Boaro, M., Fuselli, D., Squartini, S., Piazza, F., Wei, Q., Wang, D.: Optimal Task and Energy Scheduling in Dynamic Residential Scenarios. In: Wang, J., Yen, G.G., Polycarpou, M.M. (eds.) ISNN 2012, Part I. LNCS, vol. 7367, pp. 650–658. Springer, Heidelberg (2012)
4. Morais, H., Kádár, P., Faria, P., Vale, Z.A., Khodr, H.M.: Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming. Renewable Energy - An International Journal 35(1), 151–156 (2009)
5. Gudi, N., Wang, L., Devabhaktuni, V., Depuru, S.S.S.R.: A Demand-Side Management Simulation Platform Incorporating Optimal Management of Distributed Renewable Resources. In: IEEE/PES Power Systems Conference and Exposition (PSCE), pp. 1–7 (2011)
6. Liang, R.H., Liao, J.H.: A Fuzzy-Optimization Approach for Generation Scheduling with Wind and Solar Energy Systems. IEEE Transactions on Power Systems. Journals and Magazines 22(4), 1665–1674 (2007)
7. Vale, Z.A., Faria, P., Morais, H., Khodr, H.M., Silva, M., Kadar, P.: Scheduling Distributed Energy Resources in an Isolated Grid –An Artificial Neural Network Approach. In: IEEE Power and Energy Society General Meeting, pp. 1–7 (2010)
8. Welch, R.L., Venayagamoorthy, G.K.: Energy Dispatch Controllers for a Photovoltaic System. Engineering Applications of Artificial Intelligence, 249–261 (2008)
9. Fuselli, D., De Angelis, F., Boaro, M., Liu, D., Wei, Q., Squartini, S., Piazza, F.: Optimal Battery Management with ADHDP in Smart Home Environments. In: Wang, J., Yen, G.G., Polycarpou, M.M. (eds.) ISNN 2012, Part II. LNCS, vol. 7368, pp. 355–364. Springer, Heidelberg (2012)
10. Del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.C., Harley, R.G.: Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems. IEEE Transactions on Evolutionary Computation. Journals and Magazines 12(2), 171–195 (2008)
11. Werbos, P.J.: Approximate Dynamic Programming for Real-Time Control and Neural Modeling. In: Handbook of Intelligent Control (1992)
12. Huang, T., Liu, D.: Residential Energy System Control and Management using Adaptive Dynamic Programming. In: Intenational Joint Conference on Neural Networks (IJCNN), pp. 119–124 (2011)
13. 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
14. Markvart, T.: Solar Electricity, 2nd edn. Wiley, New York (2000)