

Neural network controller for Active Demand-Side Management with PV energy in the residential sector

E. Matallanas , M. Castillo-Cagigal , A. Gutiérrez , F. Monasterio-Huelin , E. Caamaño-Martín
D. Masa , J. Jiménez-Leube

A B S T R A C T

In this paper, we describe the development of a control system for Demand-Side Management in the residential sector with Distributed Generation. The electrical system under study incorporates local PV energy generation, an electricity storage system, connection to the grid and a home automation system. The distributed control system is composed of two modules: a scheduler and a coordinator, both implemented with neural networks. The control system enhances the local energy performance, scheduling the tasks demanded by the user and maximizing the use of local generation.

1. Introduction

Nowadays, there is an increasing electricity demand and an increasing cost of the raw materials. It is necessary to do a better use of the electricity through proper management. Governments are passing laws to improve this management by means of Demand-Side Management (DSM). Demand-Side Management has been identified as one of the main strategies to be promoted in order to guarantee security of electrical energy supply in the European Union [1]. However, there is not a commonly accepted definition for the term Demand-Side Management (DSM). In this paper, DSM is defined as the actions that influence the way consumers use electricity in order to achieve savings and higher efficiency in energy use [2].

The combination of DSM with an automatic control of the household demand leads to a new concept called Active Demand-Side Management (ADSM) [3,4]. ADSM allows to modify the demand profile in order to reduce the stress of the electrical system, maximize consumption when the resources are available and decrease congestion situations. There are several benefits of the

ADSM, such as reduction of losses and load shedding in the grid, reduction in energy bills, demand curve smoothing or reduction in the production cost [5,6]. Moreover, there are three main strategies to implement ADSM [7]: (i) peak clipping, (ii) valley filling and (iii) load shifting. However, as stated in Ref. [5], ADSM have several challenges to overcome such as the lack of ICT infrastructure, inappropriate market structure or lack of incentives. Furthermore, ADSM has to guarantee the comfort of the users and their preferences without changing their behavior.

ADSM can be implemented based on different criteria, such as the price of the energy, maximization of the self-consumption or limiting the maximum power between others. The term self-consumption on distributed generation electric networks focuses on the usage of the own generated energy, while the energy provided by the grid remains an optional generator or consumer [3]. This paper focuses on ADSM with a load shifting strategy to maximize self-consumption.

Different techniques to implement DSM have been proposed in the literature [5]: load shifting, direct-load control, load limiters, interruptible loads, frequency regulation, time-of-use pricing or demand bidding between others. However, most of these proposals have not yet been implemented in real environments. Moreover, typically these studies focus on macroscopic impacts without addressing their implementation at the lower level. Notwithstanding, some researches focusing on designing real implementations can be found. In Ref. [8], the authors implement a load control

frequency and guarantee that the fluctuations of the load frequency converge to a range. In Ref. [9] a direct-load controller based on programmable logic is described, while in Ref. [10] the authors make use of a fuzzy logic controller. Some approaches try to maximize self-consumption by managing the flows of energy in the industrial sector [11,12]. While others focus on the residential sector [3,13]. In the work presented by Ref. [14], measurement data of seven households in Belgium have been used to acquire relationships to dimension storage packages for grid connected PV panel installations. A methodology for the evaluation of PV array orientation, ADSM and storage to improve load matching has been presented in Ref. [15]. The method was applied in simulations to high-latitude data from detached houses and apartments in Sweden to observe their impact. Furthermore, Artificial Neural Networks (ANNs) have been extensively applied in energy systems [12,16,17]. ANNs have characteristics of optimization, generalization ability, adaptability, a legacy of information processing, failure tolerance and low power consumption [18]. In Ref. [19] the author presents several applications of ANNs in energy problems, such as modelling and designing a solar steam generating plant, estimation of a parabolic-trough collector's intercept factor and local concentration ratio, modelling and performance prediction of solar water-heating systems, between others.

In this paper we implement a distributed ADSM controller based on ANNs to maximize self-consumption in the residential sector. The distributed control system presented is made up of several ANNs located at the different appliances in a house provided with local PV energy generation. The appliances self-organize in a distributed way and a coordinator corrects their outputs in order to enhance self-consumption. It is expected that our system would schedule household tasks for the next day, coordinating the user preferences and the predicted generated electricity, so that self-consumption could be maximized. Thus, the system acts in an almost transparent way to the user and it is in charge of activating the household tasks when the PV generator produces its maximum of energy, leading to an increase of the energy efficiency.

The remainder of this paper is as follows. Section 2 describes the electrical energy system. In Section 3, the implementation of the neural controller and its training is described. Section 4 presents the postevaluation of the neural controller implemented. Finally, Section 5 concludes this paper.

2. System under study

The ADSM system has been developed in a real solar house named "Magic Box" (see Fig. 1). The house integrates sustainable elements based on renewable energies, self-sufficiency energetic methods, bioclimatic architecture and recycled construction materials [20]. It includes PV generation, electricity storage through

batteries, a set of automated appliances and a connection to the grid [21].

The PV installation consists of single-crystalline PV generators distributed in four south-oriented surfaces with different inclinations. The energy is collected in six arrays with a total nominal power of 7.2 kW_p. In addition, the electrical system embodies a battery energy storage system of 36 kWh. In a grid-connected installation, batteries are used to improve the electrical behavior by controlling the maximum consumed and generated power at different hours or ensuring the electrical supply when a grid breakdown occurs [22–24]. In this case of study, the battery system stores the excess of PV generation and supplies it to the loads when there is not enough local generation. This operation decreases the exported and imported electricity from the grid.

"Magic Box" includes typical electrical appliances of a highly electrified home: washing machine, dryer, dishwasher, refrigerator, cooking appliances, lighting, computers and entertainment appliances. The appliances are integrated in a home automation system, which allows them to be monitored and controlled by a remote system [3]. Some appliances involve an instantaneous use because of the user demand (e.g. lights, TVs, computers) while others can be time-shifted. For this reason, we have defined two types of appliances based on their operation mode:

- *Deferrable loads*, as the demand that can be displaced along the day. In this case, the user set up time limits between which task has to be carried out. E.g.: running the washing machine between 10:00 h and 16:00 h.
- *Non Deferrable loads*, as the demand that is not controllable. It represents the instantaneous appliances, like lights or TVs or the continuous consumption, like the fridge or control computers.

All the elements of the electrical system are connected to an AC bus (see Fig. 2). This connection allows the energy exchange between the different devices without an explicit hierarchy; therefore, it increases the system scalability. Notice that in our system, the battery stores only the PV energy excess and it gives its stored energy solely to the loads. Therefore, there are no energy exchanges between the battery storage system and the grid.

Based on this schematic flow of energy, the ADSM system tries to reduce the consumption of electricity from the grid and to maximize the consumption from the PV generator and the battery system.

3. System implementation

As aforementioned, the main objective of the ADSM system is to maximize self-consumption. Therefore, the ADSM system must



Fig. 1. (a) Birds eye view and (b) south frontage of "Magic Box".

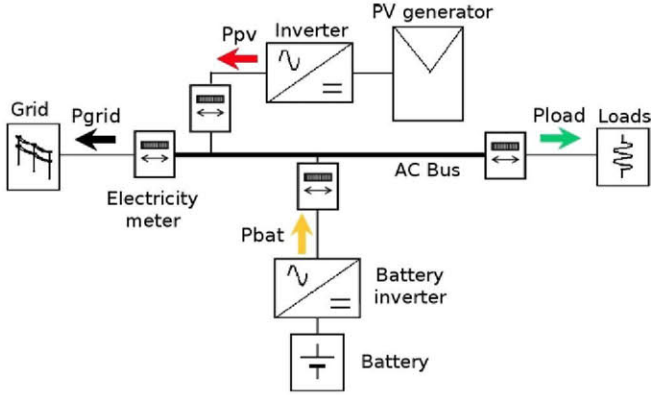


Fig. 2. Topology of the electrical system.

schedule different appliance tasks commanded by the user throughout the day. Only deferrable appliances can be controlled by the system. However, deferrable and non deferrable appliances are taken into account to analyze and validate the system.

To allow the ADSM system to create the scheduling, the user must provide a list with the appliances to be executed within the next 24 h. This list consists of the name of the appliance together with the program variables and the time limits. A summary of these variables is shown in Table 1. Note that the user is involved in the ADSM system and he offers flexibility by giving time limits to the appliances which must be executed.

Before starting, the ADSM system must obtain the tasks that the user wants to execute and the generation forecast. Once all the information is ready, the system starts scheduling the different tasks according to a specified energy criterion, maximizing self-consumption in our case. Note that the ADSM system does not reduce the energy consumption, but moves the tasks throughout the day favoring the use of the generated energy.

3.1. System inputs

- *User information.* The user establishes the tasks he wants to carry out in the next 24 h. A task consists of selecting the appliance, its program to execute and the temporal interval in which the user would like to do the task, $T_k = \{A_k, \Delta t_k\}$, where T_k is the k th task to schedule, A_k is the appliance for the k th task, which contains the variables of the program (temperature, spin revolutions, etc.) and its duration, and $\Delta t_k = (t_{k,start}, t_{k,end})$ is the temporal interval of the k th task within which the user wants to execute it. A_k is composed of the consumed power ($P_{A,k}$), and the duration of the program ($\Delta t_{A,k}$), $A_k = \{P_{A,k}, \Delta t_{A,k}\}$. These data are located at each appliance. An example of the information provided by the user is shown Fig. 3.
- *Generation forecast.* It is a forecast of the expected generated power for the next 24 h, (P_{pv}). The forecast consists of a vector of 24 values, one for each hour of the day. $P_{pv,h}$ is the h th forecasted hour of generated power [25].

Table 1
Appliances variables supplied by the user to the control system for its scheduling on the next 24 h.

Appliance	Variable 1	Variable 2
Washing machine	Temperature	Spin revolutions
Dryer	Spin revolutions	Not used
Dishwasher	Washing parameter	Not used
Oven	Temperature	Cooking time
Hood	Fan speed	Light intensity
Refrigerator	Temperature	Not used
Freezer	Temperature	Not used
Air conditioning	Temperature	Cooling time

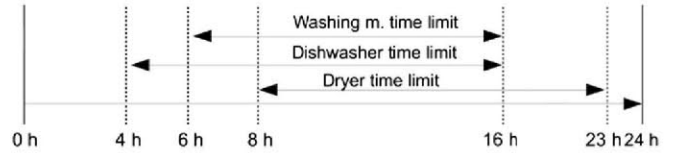


Fig. 3. Example of user time constrain for three different tasks.

3.2. System architecture

The architecture of the ADSM system consists of three modules (see Fig. 4):

- *Scheduler.* It is responsible for setting the time to execute the task T_k taking into account the user constraints (Δt_k), and the generation forecast ($P_{pv,0-23}$). It is a distributed layer; a neural controller is present in each appliance. Each appliance controller receives information about its corresponding task to be executed. Each controller outputs the time at which the task has been scheduled. Therefore, the output of this layer is a vector of scheduled times (t_s).
- *Coordinator.* It receives information from all the controllers of the scheduling layer. It checks, and modifies if necessary, that the tasks to be carried out do not overlap. Therefore, it requires the vector of scheduled times (t_s), and the time duration of each task ($\Delta t_{A,k}$) as inputs. The output of this layer is a vector consisting of the coordinated times (t_c).
- *Actuator.* It performs the communication between the coordination layer and the physical appliances. The actuator starts the tasks at the time indicated by the coordinator (t_c).

3.2.1. System controller

The controller consists of several Multilayer Perceptrons (MLPs) as ANNs [18]. Each MLP have three layers: input, hidden and output layer. The coordinator consists of one MLP, while the scheduler is composed of as many MLPs as tasks to schedule. Therefore, each appliance schedules the time at which the task must be executed. However, because of this modularity, the scheduling time of different appliances could overlap in the time domain and the instant power demand could be outside. For this reason, it is necessary the coordinator checks, and modifies if necessary, that the tasks do not overlap.

In this paper, we use a Genetic Algorithm (GA) to adjust the free parameters of the MLPs mapped into a genotype [26]. These free parameters are: (i) input gains ($g_{in,i}$), (ii) neural weights (w_{ij}) and (iii) bias (θ_i). The GA is used because it has the potential to produce a global minimum of the search space and thereby avoid local minima. It consists of evolving a population of genotypes according to a fitness function. In our controllers, the population consists of 100 genotypes. Each genotype is a vector of genes in which the gains, weights and bias are mapped and whose length (g_i) is expressed in the following equation.

$$g_L = G_{in} + \Theta_{hi} + \Theta_{out} + W_{hi} + W_{out} \quad (1)$$

where G_{in} is the number of gains in the input layer, Θ_{hi} is the number of bias of the hidden layer, Θ_{out} is the number of bias in the output layer, W_{hi} is the number of synaptic weights in the hidden layer, and W_{out} is the number of synaptic weights in the output layer.

The gene values are chosen in the range [0, 1]. Genotypes of the first generation are generated randomly. The following generations are produced by a combination of selection with elitism, crossover and mutation. For each new generation, the five individuals with the highest fitness values (called "elite") are retained unchanged. The remainder of the new generation is generated by fitness-proportional selection from the individuals of the previous generation. In addition, the new genotypes are subjected to a simple crossover,

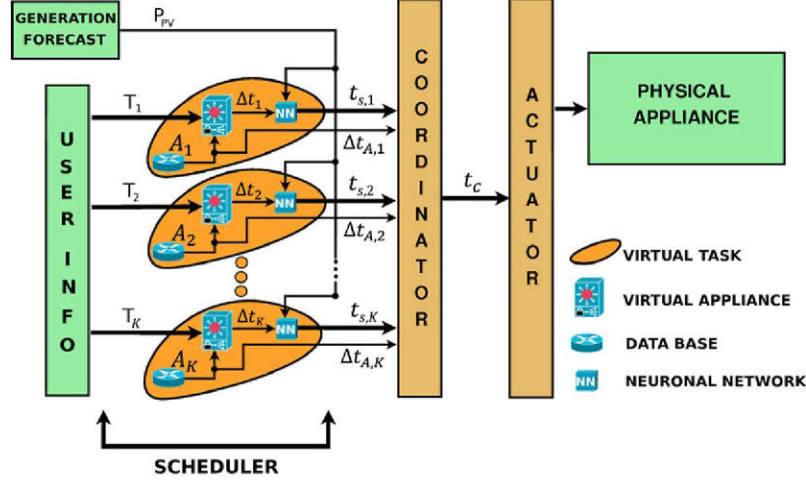


Fig. 4. Control architecture of the ADSM system.

in which two individuals exchange their genes in one point from a random position. Finally, a mutation process occurs. Every gene of every genotype changes its value randomly with a probability of $p = 0.05$. The fitness values are assigned to each individual according to a fitness function built to the specific optimization problem, one to tune the parameters of the ANNs of the scheduler and another one for the ANN of the coordinator. Finally the components of the genotype vector are mapped to produce the MLP parameters within the following ranges: input gains $g_{in,i} \in [0, 1]$, weights $w_{ij} \in [-5, 5]$ and biases $\theta_i \in [-5, 5]$.

3.2.2. Scheduler

The scheduler selects the time at which the tasks required by the user must be executed, maximizing self-consumption and giving priority to the user. Therefore, the inputs of the scheduler are the time limits of the tasks selected by the user (Δt_k) and the generation forecast (P_{PV}). The scheduler controllers, once the inputs are established, start scheduling each task independently (see Fig. 5).

The architecture of the scheduler, as explained in Section 3.2.1, consists of as many homogeneous MLPs as tasks established by the user. One scheduler MLP is shown in Fig. 6. Each MLP is composed of 26 neurons in the input layer ($x_{in} = \{t_{k,start}, t_{k,end}, P_{PV,0}, \dots, P_{PV,23}\}$), 13 neurons in the hidden layer and 1 neuron in the output layer ($y_{40} = t_{s,k}$). Each neuron is governed by the following equation.

$$y_i = \begin{cases} g_{in,i} \cdot x_{in,i} & i \in \{1, \dots, 26\} \\ \sigma \left(\sum_{j=1}^{N_{in}} w_{ij} \cdot y_j + \theta_i \right) & i \in \{27, \dots, 40\} \end{cases} \quad (2)$$

$$\text{with } \sigma(x) = \frac{1}{1 + e^{-x}}$$

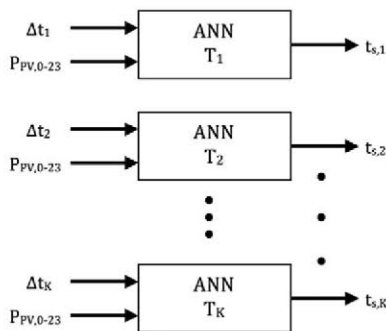


Fig. 5. Scheduler architecture.

where N_{in} is the number of inputs of the neuron and y_j are the outputs of the neurons of the previous layer.

As aforementioned, the free parameters of the MLP are mapped into a genotype. In this case, the number of genes of each genotype is $g_L = 26 + 13 + 1 + (26 \cdot 13) + (13 \cdot 1) = 391$ genes. The fitness function of the GA is built to achieve the objective of maximizing self-consumption. Therefore, the fitness function evaluates how close is the MLP output ($t_{s,k}$), from the maximum of the forecasted generated power within the time interval provided by the user (Δt_k), for K tasks and L forecasted generation power profiles. The mathematical expression of the fitness function can be observed in the following equation.

$$FF_{k,l}^s [t_{s,k}, P_{PV}^l] = \begin{cases} 0 & \text{if } t_{s,k} \notin \Delta t_k \\ g(t_{s,k}, P_{PV}^l) & \text{if } t_{s,k} \in \Delta t_k \end{cases}$$

$$\text{with } g(t_{s,k}, P_{PV}^l) = \begin{cases} 0 & \text{if } t_{s,k} < \gamma_{k,l} - 2 \\ \frac{P_{PV}^l(t_{s,k})}{\Gamma_{k,l}} - 0.5 & \text{if } \gamma_{k,l} - 2 \geq t_{s,k} < \gamma_{k,l} - 1 \\ \frac{P_{PV}^l(t_{s,k})}{\Gamma_{k,l}} - 0.25 & \text{if } \gamma_{k,l} - 1 \geq t_{s,k} < \gamma_{k,l} \\ \frac{P_{PV}^l(t_{s,k})}{\Gamma_{k,l}} = 1 & \text{if } t_{s,k} = \gamma_{k,l} \\ \frac{P_{PV}^l(t_{s,k})}{\Gamma_{k,l}} - 0.25 & \text{if } \gamma_{k,l} < t_{s,k} \leq \gamma_{k,l} + 1 \\ \frac{P_{PV}^l(t_{s,k})}{\Gamma_{k,l}} - 0.5 & \text{if } \gamma_{k,l} + 1 < t_{s,k} \leq \gamma_{k,l} + 2 \\ 0 & \text{if } t_{s,k} > \gamma_{k,l} + 2 \end{cases} \quad (3)$$

where $\Gamma_{k,l}$ is the maximum value of the forecasted generated power inside the time interval provided by the user (Δt_k) for the l^{th} profile, $\gamma_{k,l}$ is the time at which $\Gamma_{k,l}$ occurs, and $P_{PV}^l(t_{s,k})$ is the l^{th} forecasted generation power value for $t_{s,k}$. Remember that, $t_{s,k}$ is the scheduling time of the k^{th} task and Δt_k is the temporal interval of the user for the k^{th} task.

Finally, the fitness value for one individual (FF_{ind}^s) is the arithmetic mean of all the situations (K tasks and L forecasted generation power profiles) in which the genotype was evaluated (see Eq. (4)).

$$FF_{ind}^s = \frac{\sum_{k=1}^K \sum_{l=1}^L FF_{k,l}^s}{K \cdot L} \quad (4)$$

The dataset for the evolution of the scheduler MLPs was composed of $L = 142$ forecasted generation power profiles of the year 2009, ($l \in \{0, \dots, 141\}$), and the time preferences of the user for each task where intervals of $\Delta t_k = 4$ h which vary along the day in steps of

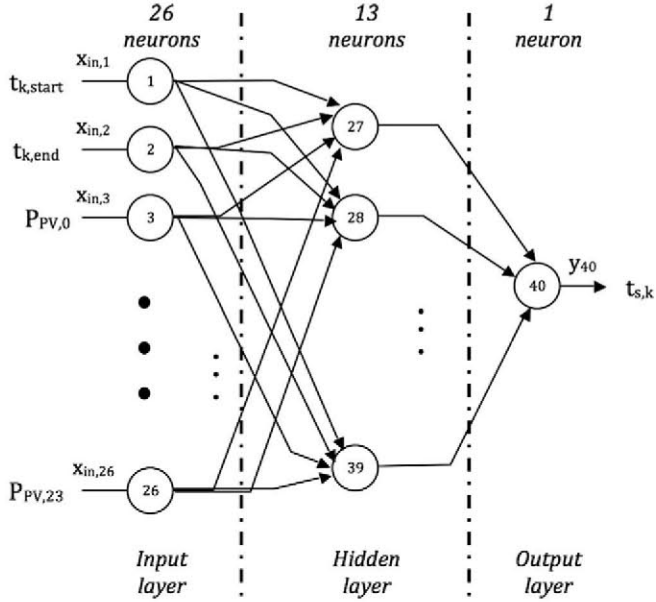


Fig. 6. Structure of one ANN of the scheduler.

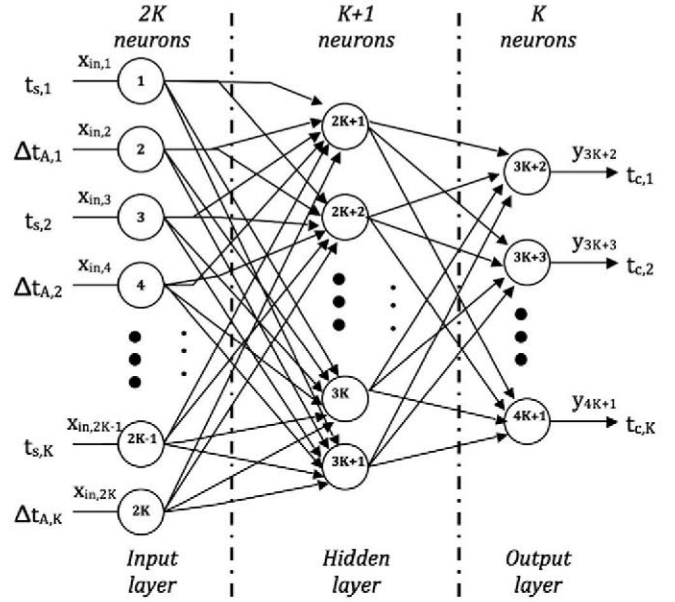


Fig. 7. ANN structure of the coordinator.

2 h ($k \in \{0, \dots, 11\}$) were selected. The evolution was carried out on different steps. Firstly, the system was evolved with 10 profiles of the 142 available. Once it reached a steady-state value, we stopped the evolution. Then, the evolution was restarted with 30 profiles for the individuals of the last generation evolved with 10 profiles. After reaching again a plateau, we stopped the evolution. We repeated the same process with 50, 90 and 142 profiles. For the different steps of the evolution, we only changed the number of forecasted generation power profiles, maintaining the time preferences of the user explained before. The results of the evolution are shown in Table 2. The maximum number of generations selected was 10,000, but the algorithm reached a steady-state before accomplishing the total number of generations. In addition, the fitness value, to which the GA converges, is a compromise value for all the forecasted power profiles and all the temporal intervals defined by the user. The best individual of the last generation evolved with 142 profiles was set as the parameters of the scheduler MLPs.

3.2.3. Coordinator

As aforementioned, the aim of the coordinator is that tasks do not overlap during their execution. The coordinator will try to fit for the closest scheduled times with no overlap. The inputs for the coordinator are the scheduled times (t_s) and the duration of each task, ($\Delta t_{A,k}$). The output of the coordinator is a vector of time references at which the selected tasks must be executed (t_c).

The coordinator is composed of one MLP whose number of neurons depends on the number of tasks. So that, the input layer has $2 \cdot K$ neurons, with $x_{in} = \{t_{s,1}, \Delta t_{A,1}, t_{s,2}, \Delta t_{A,2}, \dots, t_{s,K}, \Delta t_{A,K}\}$, the hidden layer has $K+1$ neurons and the output layer has K neurons, where K is the number of tasks (see Fig. 7). The ANN is governed by Eq. (5) and its outputs ($y_i = t_{c,j}; i \in \{3K+2, \dots, 4K+1\}, j \in \{1, \dots, K\}$) create a vector of coordinated times (t_c).

Table 2
Results of evolution for a scheduler ANN.

N^p profiles	Best fitness	Generations
10	0.9091740680	2189
30	0.9172684079	996
50	0.9027720441	287
90	0.8171219180	2189
142	0.7026183810	847

$$y_i = \begin{cases} g_{in,i} \cdot x_{in,i} & i \in \{1, \dots, 2K\} \\ \sigma \left(\sum_{j=1}^{N_{in}} w_{ij} \cdot I_j \cdot y_j + \theta_i \right) & i \in \{2K+1, \dots, 4K+1\} \end{cases} \quad (5)$$

$$\text{with } \sigma(x) = \frac{1}{1 + e^{-x}}$$

where N_{in} is the number of inputs of the neuron, y_j is the output of the previous layer neurons, and $I_j \in \{0, 1\}$ is an inhibitor signal explained hereafter.

The inhibitor signal is introduced in the ANN to give modularity to this network. Modularity is very convenient when the number of tasks to be executed is unknown. However, an ANN with an undefined structure cannot be evolved with the characteristics of the GA presented in this paper. Thus, the coordinator is implemented to handle seven tasks ($K=7$), which is the maximum number of deferrable appliances in "Magic Box" (see Section 2). In the case that not all the seven tasks are activated, the remaining entries are zero. The inhibitors have been established between the hidden layer and output layer to inhibit the connection of the entries which are zero. Therefore, the neurons in the hidden layer, assigned to the tasks that are not going to be coordinated, do not interfere in the calculation of the coordinated time for the active tasks.

As previously explained, the length of the coordinator genotype is $g_L(K) = 2 \cdot K + (K+1) + (2 \cdot K \cdot (K+1)) + ((K+1) \cdot K) = 3 \cdot K^2 + 7 \cdot K + 1$. In the evolution of the coordinator, the ANN is evolved with $K=7$ tasks; therefore, the length of the genotype is $g_L(7) = 3 \cdot 7^2 + 7 \cdot 7 + 1 = 197$ genes.

The evaluation of each genotype is made according to a fitness function developed to fulfill the objective of spreading the tasks in order not to overlap in the time domain. The fitness function for this evolution consists of two functions: one function (FF_1^c) evaluates the overlapping between the coordinated outputs, while the second one (FF_2^c) measures the similarity between the ANN outputs (coordinated times t_c) and its inputs (scheduled times t_s). The mathematical expression of each function is shown in Eqs. (6) and (7).

$$FF_1^c = \begin{cases} \frac{1}{\delta} \cdot |t_{f1}| & \text{if } 0 \leq |t_{f1}| < \frac{\delta}{4} \\ \frac{3}{\delta} \cdot |t_{f1}| - \frac{1}{2} & \text{if } \frac{\delta}{4} \leq |t_{f1}| < \frac{\delta}{2} \\ 1 & \text{if } \frac{\delta}{2} \leq |t_{f1}| \end{cases} \quad (6)$$

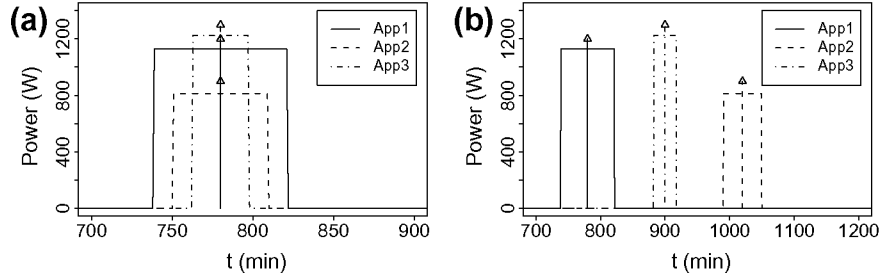


Fig. 8. (a) Example of three overlapped tasks provided by the scheduler. (b) Result of the coordination layer for the same tasks.

where δ is the maximum program duration of all the tasks, ($\delta = \max(\Delta t_{A,k}); k \in \{1, \dots, K\}$), and $|t_{j1}|$ is the absolute value of the difference between the outputs of the ANN ($|t_{j1}| = |t_{c,k} - t_{c,j}|$, $j \in \{1, \dots, K\}; j \neq k$).

$$FF_{2\ k,m}^c = \begin{cases} -\frac{3}{2\delta} \cdot |t_{j2}| + 1 & \text{if } 0 \leq |t_{j2}| < \frac{\delta}{2} \\ -\frac{1}{2\delta} \cdot |t_{j2}| + \frac{1}{2} & \text{if } \frac{\delta}{2} \leq |t_{j2}| < \delta \\ 0 & \text{if } \delta \leq |t_{j2}| \end{cases} \quad (7)$$

where $|t_{j2}|$ is the absolute value of the difference between a task output and its corresponding input ($|t_{j2}| = |t_{s,k} - t_{c,k}|$, $k \in \{0, \dots, K\}$).

The final fitness value of one individual (FF_{ind}^c) is presented in the following equation.

$$FF_{ind}^c = \frac{1}{M} \cdot \sum_{m=1}^M \left[\left(\frac{2}{K \cdot (K-1)} \cdot \sum_{k=1}^{K-1} FF_{1\ k,m}^c \right) \cdot \left(\frac{1}{K} \cdot \sum_{k=1}^{K-1} FF_{2\ k,m}^c \right) \right] \quad (8)$$

where K is the number of tasks and M is the number of cases of the evolution explained hereafter.

The dataset of the coordinator during the evolution consists of the duration of seven tasks, one for each controllable appliance of "Magic Box" ($\Delta t_A = \{\Delta t_{A,1}, \Delta t_{A,2}, \dots, \Delta t_{A,7}\}$), which remains constant during the entire evolution process, and the scheduled times ($t_s = \{t_{s,1}, t_{s,2}, \dots, t_{s,7}\}$) whose values for the evolution vary from 10 a.m. to 8 p.m, selected as the temporary interval of sunny hours. Therefore, M is the number of combinations of the seven scheduled times along the 10 h, that is $M = 10^7$ combinations.

As can be deduced, the coordinator evolution evaluates if the outputs of the ANN meet the goal of not being overlapped and resembling as much as possible to the inputs. In this way, the best fitness value, which has been achieved for 10,000 generations, was 0.38. Notice that the best fitness value reached is far from the maximum fitness value ($\max\{FF_{ind}^c\} = 1$). This is because of the geometric mean of $FF_{1\ k,m}^c$ and $FF_{2\ k,m}^c$ shown in Eq. (8). However, we observe that in average each of them achieves a value of 0.62, being a compromise of all the $M = 10^7$ combinations defined in the evolution. Fig. 8 shows an example of the separation of the tasks for a selected case of the evolution.

4. Postevaluation

In this section, we evaluate the output of the scheduler, the output of the coordinator and the result of the complete ADSM control system. The dataset used to evaluate the system consist of forecasted generated power profiles of the year 2009 not used in the evolution (223 profiles), and three tasks (T_1, T_2 and T_3) whose time intervals ($\Delta t_1, \Delta t_2$ and Δt_3) defined by the user are composed of 4 h which varies along the day in steps of 2 h. The tasks defined are:

- Task 1: a washing machine with program 1, whose variables are 90 °C and 1200 r.p.m. (A_1), and executed for all the time intervals defined previously ($T_1 = \{A_1, \Delta t_1\}$).

Table 3

Results of the evaluation of the scheduling layer.

	$t_{s,k} \approx \gamma_{k,l}$	$ t_{s,k} - \gamma_{k,l} \leq 1h$	$ t_{s,k} - \gamma_{k,l} > 1h$
Scheduler	87%	10%	3%

Table 4

Results of the evaluation of the coordination layer.

	Right	$\Delta t_k \leq 10min$	$\Delta t_k > 10min$
Coordinator	89%	7%	4%

- Task 2: a dryer with program 3, whose variable is 1200 r.p.m. (A_2), and executed for all the time intervals defined previously ($T_2 = \{A_2, \Delta t_2\}$).
- Task 3: a dishwasher with program 5 (A_3), and executed for all the time intervals defined previously ($T_3 = \{A_3, \Delta t_3\}$).

For the aforementioned dataset, the scheduler situates the execution time for each of the three appliances inside the time constraints of the user in all the cases. However, it locates the execution time at the time when the maximum of the forecasted generation power profile occurs in 87% of the cases. In 10% of the cases it was deviated less than one hour from the maximum and in the remaining 3%, it was deviated in more than one hour (see Table 3).

The dataset used for the validation of the coordinator was composed of the duration of the three tasks described above ($\Delta t_{A,1}, \Delta t_{A,2}$ and $\Delta t_{A,3}$), and the vector of the scheduled times obtained from the scheduler. The results of the evaluation are as follows: the tasks were spread in the time domain complying with the temporary interval constraints set by the user, in 89% of the cases. Only in 7% of the cases the tasks were deviated from the temporary interval in less than 10 min and in the remaining 4%, $t_{c,k}$ was deviated from the temporary interval in more than 10 min (see Table 4).

Finally the verification of the whole system was made to explore the combination of the two modules. The system was successful in both the scheduler and coordinator modules in an 85% of the situations. The remaining 15% had failures in the scheduler, the coordinator or both (see Table 5).

Table 5

Results of the evaluation of the whole system.

	Right scheduling/right coordination	Failure scheduling/failure coordination
Whole system	85%	15%

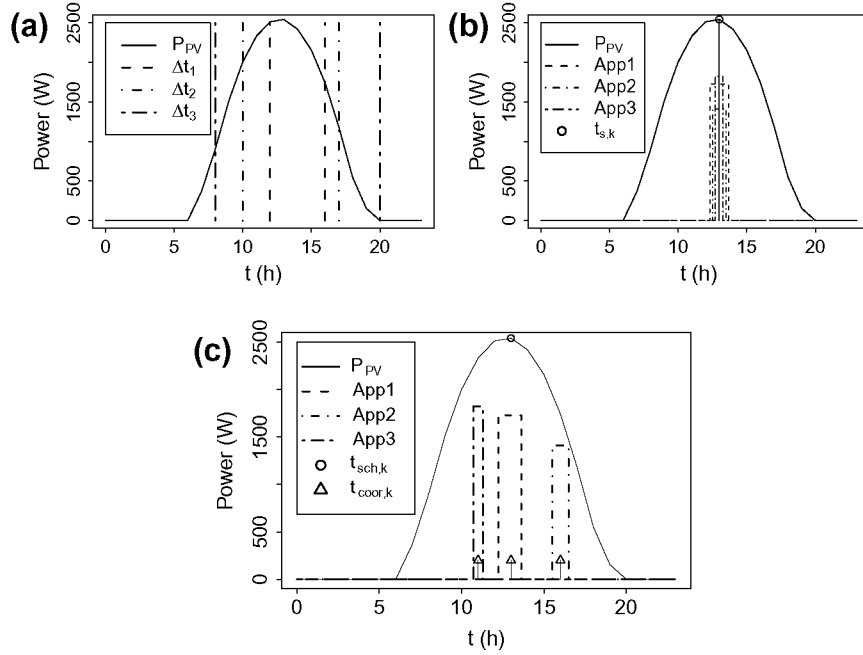


Fig. 9. Example of the whole process of the ADSM system: (a) inputs for the ADSM system, (b) response of the scheduler and (c) final response of the ADSM system.

To better understand the behavior of the system, an example is described in what follows. Let us assume the tasks proposed by the user are the following:

- Task 1: a washing machine with program 1, whose variables are 90 °C and 1200 r.p.m. (A_1), with an execution time interval (Δt_1) defined between 12 a.m. and 4 p.m. ($T_1 = \{A_1, \Delta t_1\}$).
- Task 2: a dryer with program 3, whose variable is 1200 r.p.m. (A_2), with an execution time interval (Δt_2) defined between 10 a.m. and 7 p.m. ($T_2 = \{A_2, \Delta t_2\}$).
- Task 3: a dishwasher with program 5 (A_3), with an execution time interval (Δt_3) defined between 8 a.m. and 8 p.m. ($T_3 = \{A_3, \Delta t_3\}$).

In Fig. 9a, the time intervals and PV profile inputs to the neural controller can be observed. In Fig. 9b, the response of the scheduler is shown. In this case the scheduled times of the three tasks are the same because the time intervals provided by the user are wide open. Therefore, the scheduler controllers select the same time for the task execution (close to the maximum of the forecasted generation power profile). Finally, Fig. 9c shows the result after coordination. The coordinator has separated the scheduling time of the three tasks to avoid overlapping. The coordinated times are the times in which the tasks are going to be executed the next day. Note that the three tasks are under the PV generation curve, so that self-consumption is guaranteed.

5. Conclusion

We have developed a control system using ANNs tuned by GAs to implement an ADSM system for the residential sector. Results show that ANNs are able to implement an ADSM system that meets the user requirements and schedules the tasks for the next day to improve the electrical local behavior. This concept is related to the idea of the self-consumption of the local energy. The implemented ADSM system, with a strategy of load shifting, maximizes the consumption of PV generated energy increasing self-consumption.

The ADSM system inherits the properties of the applied algorithms. So that, the system is robust against failures because of

the distributed flow of information inside the ANNs. We have shown that, with a small amount of information, the system is capable of establishing a plan of action to start controllable appliances in a distributed way.

In the execution of the neural controllers, we have shown that the tasks overlap in the time domain because of the distributed architecture of the scheduler. However, this problem was solved with a coordinator which splits the overlapped tasks. The evaluation of the whole system was made taking into account that (i) the tasks cannot overlap, (ii) must be inside the user interval and (iii) should be near the forecasted maximum power value. This objective was accomplished in 85% of the evaluated cases. Therefore we have shown that it is possible to use distributed algorithms to build ADSM systems, but also that the energy efficiency can be raised maximizing the self-consumption of the local energy.

For a future perspective, we observe that ADSM techniques give the user the possibility to control his energy behavior because of the feedback information. This information is obtained through the constant monitoring of the energy flow of the electrical system. With this monitoring, the user could reduce the rates of energy consumption at home [27]. The benefits for the users are not only related to the understanding of its consumption behavior, but also they obtain economical benefits by not consuming energy from the grid. These techniques also benefit the grid, reducing the transportation losses and load shedding because of the self-consumption of local energy. Moreover, the use of these techniques will play an important role in the future smart-grids, helping to guarantee the energy supply and reducing the raw materials imports [28].

Acknowledgments

M. Castillo is sponsored by the Spanish Ministry of Education with a Ph.D. grant (FPU-2010). This work has been financed by the Spanish Ministry of Education and Science (Plan Nacional I+D+I 2004-2007) within the framework of the project *Gestión de la Demanda Eléctrica Doméstica con Energía Solar Fotovoltaica* (ENE2007-66135). The authors want to thank the companies SMA, for the PV system components and Siemens, for the automated appliances and the Escuela Técnica Superior de Ingenieros

de Telecomunicación of the UPM for their support in the construction and maintenance of “Magic Box”.

References

- [1] European Commission. Towards a EU strategy for the security of energy supply. COM 321; 2002.
- [2] Pérez JJ, Sánchez IJ, Pardo M. La gestión de la demanda de electricidad. Madrid (España): Ed. Fundación Alternativas; 2005.
- [3] Castillo-Cagigal M, Gutiérrez A, Monasterio-Huelin F, Caamaño-Martín E, Masa-Bote D, Jiménez-Leube J. A semi-distributed electric demand-side management system with PV generation for self-consumption enhancement. *Energy Convers Manage* 2011;52(7):2659–66.
- [4] Castillo-Cagigal M, Caamaño-Martín E, Masa-Bote D, Gutiérrez A, et al. PV self-consumption optimization with storage and Active DSM for the residential sector. *Solar Energy* 2011;85(9):2338–48.
- [5] Strbac G. Demand side management: benefits and challenges. *Energy Policy* 2008;36(12):4419–26.
- [6] Papagiannis G, Dagoumas A, Lettas N, Dokopoulos P. Economic and environmental impacts from the implementation of an intelligent demand side management system at the European level. *Energy Policy* 2008;36(1):163–80.
- [7] Gellings CW, Chamberlin JH. Demand-side management: concepts and methods. 2nd ed. USA: Fairmont Pr; 1993.
- [8] Zribi M, Al-Rashed M, Alrifai M. Adaptive decentralized load frequency control of multi-area power systems. *Int J Electr Power Energy Syst* 2005;27(8):575–83.
- [9] El-Amin I, Al-Ali A, Suhail M. Direct load control using a programmable logic controller. *Electr Power Syst Res* 1999;52(3):211–6.
- [10] Goel L, Wu Q, Wang P. Fuzzy logic-based direct load control of air conditioning loads considering nodal reliability characteristics in restructured power systems. *Electr Power Syst Res* 2010;80(1):98–107.
- [11] Middelberg A, Zhang J, Xia X. An optimal control model for load shifting with application in the energy management of a colliery. *Appl Energy* 2009;86(7–8):1266–73.
- [12] Babu PR, Divya VPS, Srikanth P, Singh BD, Varun K. Neural network and dsm techniques applied to a industrial consumer a case study. In: 5th IEEE international conference on compatibility in power electronics. Gdansk (Poland): IEEE; 2007. p. 1–4.
- [13] Chaabene M, Ammar MB, Elhajjaji A. Fuzzy approach for optimal energy-management of a domestic photovoltaic panel. *Appl Energy* 2007;84(10):992–1001.
- [14] Mulder G, De Ridder F, Six D. Electricity storage for grid-connected household dwellings with pv panels. *Solar Energy* 2010;84(7):1284–93.
- [15] Widén J, Wckelgard E, Lund PD. Options for improving the load matching capability of distributed photovoltaics: methodology and application to high-latitude data. *Solar Energy* 2009;83(11):1953–66.
- [16] Wong S, Wan KK, Lam TN. Artificial neural networks for energy analysis of office buildings with daylighting. *Appl Energy* 2010;87(2):551–7.
- [17] Fadare D. Modelling of solar energy potential in nigeria using an artificial neural network model. *Appl Energy* 2009;86(9):1410–22.
- [18] Haykin S. *Neuronal networks and learning machines*. 3rd ed. New Jersey (USA): Pearson International; 2009.
- [19] Kalogirou SA. Applications of artificial neural-networks for energy systems. *Appl Energy* 2000;67(1–2):17–35.
- [20] Gutiérrez A, Jiménez-Leube J, Magdalena L. A distributed sensor network for the control of a bioclimatic house in Spain. *Sensors* 2009;9(10):8197–214.
- [21] Castillo-Cagigal M, Caamaño-Martín E, Gutiérrez A, Masa-Bote D, Monasterio F, Porro J, et al. Self-consumption of PV electricity with active demand side management: the GeDELLOS-PV system. In: Proceedings of the 25th European photovoltaic solar energy conference and proceedings of the 5th world photovoltaic solar energy. Munich (Germany): WIP-Renewable Energies; 2010. p. 4866–70.
- [22] Aditya S, Das D. Battery energy storage for load frequency control of an interconnected power system. *Electr Power Syst Res* 2001;58(3):179–85.
- [23] Wagner R. Large lead/acid batteries for frequency regulation, load levelling and solar power application. *J Power Sources* 1997;67(1–2):163–72.
- [24] Wagner R, Schroeder M, StephanBlome T, Handschin E. A multifunctional energy-storage system with high-power lead-acid batteries. *J Power Sources* 1999;78(1–2):156–63.
- [25] Masa-Bote D, Caamaño-Martín E. Forecast of energy production for PV systems 24 hours ahead. In: Proceedings of the 25th European photovoltaic solar energy conference and proceedings of the 5th world photovoltaic solar energy. Munich (Germany): WIP-Renewable Energies; 2010. p. 4813–9.
- [26] Golberg DE. *Genetic algorithms in search. Optimization and machine learning*. Indiana (USA): Addison Wesley; 2006.
- [27] Wood G, Newborough M. Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design. *Energy Build* 2003;35(8):821–41.
- [28] Vojdani A. Smart integration. *IEEE Power Energy Mag* 2008;6(6):71–9.