A win-win algorithm for aggregated residential energy management: resource optimisation and user acceptance learning

Original
A win-win algorithm for aggregated residential energy management: resource optimisation and user acceptance learning / De Vizia, Claudia; Patti, Edoardo; Macii, Enrico; Bottaccioli, Lorenzo. - (2020). ((Intervento presentato al convegno 2020 IEEE International Conference on Environment and Electrical Engineering (EEEIC 2020) tenutosi a Madrid, Spain nel 09-12 June 2020.

Availability:
This version is available at: 11583/2837815 since: 2020-07-01T10:41:38Z

Publisher:
IEEE

Published
DOI:

(Article begins on next page)

A win-win algorithm for aggregated residential energy management: resource optimisation and user acceptance learning

Claudia De Vizia*‡, Edoardo Patti*‡, Enrico Macii¶ and Lorenzo Bottaccioli*‡
*Dept. of Control and Computer Engineering, Politecnico di Torino, Turin, Italy.
¶Interuniversity Dept. of Regional and Urban Studies, Politecnico di Torino, Turin, Italy.
‡Energy Center Lab, Turin, Italy.
Email:{name.surname}@polito.it

***Abstract*—This paper proposes a solution based on Multi Agent System to study a residential Demand Side Management (DSM) program with a centralised approach. It focuses on minimising the cost considering different energy sources, such as photovoltaic panels and energy storage system, while optimally scheduling the appliances that can be shifted in time. The cost minimisation is formulated as a Mixed Integer Linear Programming (MILP) problem. The optimal allocation of the shiftable loads takes into account the modelled users' preferences that are learnt by means of an algorithm based on an explore-exploit strategy. From the results, it emerges that a win-win situation could be achieved if user preference are considered.These benefits include savings and users' satisfaction.**

***Index Terms*—Agent Based Modelling, Energy Aggregator, Demand Side Management, Multi Agent System, MILP, Explore-Exploit, User Preference**

NOMENCLATURE

Indices	
i	appliances, $i \in \{1,2,3\}$
j	households (users), $j \in \{1,2,..,N\}$
t	time slots, $t \in \{1,2..,96\}$
f_i	interval "allowed" for i, $f_i \in \{\mathrm{low_i},..,\mathrm{upper_i}\}$
Data	
δ	time interval duration (h) $\rightarrow$ 1/4
$c\_dis\_bat_t$	discharging price at time t [€/ kWh]
c_from_t	market price at time t (buying) [€/ kWh]
c_pv_t	PV price at time t [€/ kWh]
c_to_t	market price at time t (selling) [€/ kWh]
$capacity$	battery capacity [kWh]
$Cmax/Dmax$	max charge/discharge rate [kW]
E_bat_init	battery initial condition [kWh]
eff	battery efficiency [%]
$L\_shift_{i,j}^{f,t}$	cycle matrix for each appliance, for each user
min_charge	min stored energy [kWh]
Not_shift_t	power needed by not shiftable i at time t [kW]
PV_t	PV generated at time t [kW]
$request_{i,j}$	there is/not a request for appliance i from user j (1/0)
Decision Variables	
$\mathrm{E\_bat_t} \in \mathbb{R}$	amount of energy in the battery at time t
$\mathrm{P\_from_t} \in \mathbb{R}$	amount of power from the grid at time t
$\mathrm{P\_to_t} \in \mathbb{R}$	amount of power given to the grid at time t
$\mathrm{PC\_bat_t} \in \mathbb{R}$	charging power of the battery at time t
$\mathrm{PD\_bat_t} \in \mathbb{R}$	discharging power of the battery at time t
$\mathrm{PD\_on_t/PC\_on_t}$	binary variable that indicates if the battery is discharging or charging at time t
$\mathrm{x_f^{ij}} \in \{0,1\}$	binary variable that selects the day load profile of the appliance i of the customer j

I. INTRODUCTION

The growing presence of smart grids favours the exploitation of Renewable Energy Sources (RES), reducing energy costs and CO_2 emissions. Unfortunately, RES are unpredictable since they depend on weather conditions and can cause large energy imbalances. Hence, there is the need of finding mechanisms that handle this problem on the demand side. A possible way to address this issue is to manage customers' electric resources in optimal ways [1]. This approach takes the name of Demand Side Management (DSM).

In this paper, we focus on DSM in the residential context, facing a centralised approach since it may give greater benefits in terms of RES usage [2]. Thus in the proposed work, we identified three main actors: i) the *Aggregator*, which manages also RES and Storage systems, ii) *Users*, which participate to the DSM program and iii) *Energy Market*, which gives information on day-ahead prices.

Due to the large number of interactions between independent entities, it has been decided to adopt a framework that couples a co-simulator platform, MOSAIK [3], and a Multi-Agent System (MAS) called AIOMAS [4]. MOSAIK allows us to change each single model (e.g. photovoltaic panels (PV) and loads) effortlessly, following a modular approach.

Prior existing works, i.e., [5]–[7], studied the optimal scheduling of appliances and/or cost reduction from different sources with different methods. Among all, one of the most used is Linear Programming (LP). We likewise formulate the problem applying the Mixed Integer Linear Programming (MILP). However, prior articles that use LP usually consider the time slots allowed for the shift fixed, communicated by the user or depending on the trade-off between comfort and savings. Instead, our proposal is to mix MILP formulation with an algorithm that learns these allowed time slots, thus the preferences of the simulated users, without any prior knowledge relying only on the answers of users. The acceptance or refusal of the proposed shift will increase the knowledge of users' preferences. The idea is to give some control to the user, exploiting the double-flow of information in order to have a system more oriented towards the user.

Thus, the overall problem (appliances scheduling and cost minimisation) has been divided into two sub-problems. On the one hand, we learn the modelled users' preference using our algorithm based on an explore-exploit mechanism. On the other hand, we optimise the cost of different energy sources. They are combined in such a way that the output of the former is the input of the latter.

The remainder of this paper is divided into five sections. Section II reviews literature solutions related to this topic. Section III presents the adopted methodology introducing the interacting agents in our MAS, providing the formulation of the MILP problem and the algorithm for learning the time slots preferred by users. Section IV presents our experimental results. Finally, Section V discusses concluding remarks and future works.

II. Related works

Many works that focus on load shifting and/or cost minimisation from different energy sources in the residential context have been published. We decided to focus on those that also consider user preference. For instance, [5]–[9] focus only on load scheduling without taking into account batteries and RES.

In [5], the results are shown for one single day with different time slot lengths (3/5/10 min). Different tariffs of different countries are compared. The user can set up time preferences.

In [6], the time preferences are set to a fixed value. The authors consider six dwellings and six types of appliances. Not all dwellings have all the types of appliances to compare different types of users.

In [7], for each appliance, each of the 250 consumers communicates the level of preference for each time period in which the appliance may be turned on. Preferences are not communicated daily, but only if they change. Authors exploited a Mixed Integer Programming to formulate the problem.

In [8], the formulation simultaneously minimises electricity costs and maximise user convenience, properly weighted. The user can choose between three levels of priority. Level 1 (highest priority) is given to the time interval in which the appliance is preferable to run, Level 3 to the lowest priority time slots. Similarly, [9] maintains the trade-off between cost and satisfaction. It uses iterative learning to set parameters in the objective function. Thus to the best of our knowledge, w.r.t. previous works, [9] is the only literature solution that learns from user answers. The trade-off parameter will influence the amount of appliances to shift proposing an algorithm based on linear programming relaxation technique.

Focusing on those studies that focus on both load shifting and cost minimisation from different energy sources using MILP, different works have been proposed in literature ([10]–[13]). For example, [10] takes into account PV systems and different appliances in an individual home and the user - i.e., the inhabitant - decides the preferred time window for each operation. In [11], also the Energy Storage System (ESS) has been considered in a smart building with 30 houses. The authors jointly minimise electricity cost and CO_2 emissions ensuring that appliances end their tasks within the interval defined by the user.

In [12] wind turbine and combined heat and power are included for a single household. It uses Artificial Neural Networks to predict solar/wind production and daily energy demand. The user can choose the objective of the optimisation among CO_2 reduction, cost saving and user comfort. In [13], ESS and PV have been used too. Moreover, it uses a MAS to represent the selected entities. It states that for simplicity, the appliance usage-times set by users are equal and fixed for everyone, which is, in our view, its main limitation.

Unlike the others, [14] conducts a survey on 427 subjects and it categorises the user on the basis of the preference obtaining different profiles. New customers do not fill any form. A profile, among the determined previous ones, is assigned based on the rating provided by new customers during a testing period. Based on the profile, two different algorithms may allocate the load in an optimal time slot. The considered maximum shift is ± 3 hours.

To sum up, many works, i.e., [5], [10], [11], [13], consider users' preferences through some constraints in the MILP problem. Instead, others, i.e., [8], [9], include the consumer dissatisfaction in the objective function. Thus, weighting properly economical gain and discomfort. The main contribution of this paper w.r.t. literature solutions consists in adopting a new way to consider the modelled user preferences when dealing with the MILP formulation since no information about each user's preference is available in advance. The users' preferences are taken into account in the dynamic constraints. It appears that only [9] sets preferences dynamically. Differently from us, it considers the preferences in the objective function and does not consider RES and ESS. Moreover, the following simplifications that have been done in various of literature solutions have been changed or removed, taking advantage of the proposed dynamic framework. Indeed, with respect to [10], [11], [13] that suppose that the load is turned on exactly once per day, we consider day by day if there is a request from each user to use that appliance. Many works perform the optimisation from one individual day, we simulate an entire year to evaluate how cost and, especially, users' acceptance may evolve over the time. The considered works that included a battery [11]–[13] force the State Of Charge (SOC) at the end of the day to take a fixed value, usually equal to the SOC at t=1. In our solution, we remove this constraint.

III. Methodology

This section presents the proposed MAS framework by introducing both agents and their interactions.

We identified three main agents that are reported in the following. The *User Agent* simulates the energetic behaviour of a house and its inhabitants. Each *User Agent* is set with some appliances. Some of such appliances, and thus their loads, cannot be shifted over time, while others can be according to *User Agent*'s consensus. Different *User Agents* have different tolerance to shift. The *Aggregator Agent* optimally shifts the appliances taking into account both ESS and PV, which are considered as shared resources. It also learns *User Agent*' tolerance thanks to the proposed Acceptance Learning Algorithm (ALA). The *Market Agent* is a very simple agent that daily provides the *Aggregator Agent* with the day-ahead market price.

The interactions among these agents start from the *Market Agent* that sends day-ahead price information to the *Aggregator Agent* for the day after. For the same day, the *Aggregator Agent* receives also the forecast for both PV production and load consumption of *Users Agents*. Then, the *Aggregator Agent* performs a first iteration optimisation without shifting the appliances from the user's desired time-slot. This represents the worst case. Then in a second iteration, the optimisation is performed shifting loads in time-slots identified by the ϵ-greedy algorithm, which is the core of the *Aggregator Agent*. After this second iteration, each *User Agent* evaluates the proposed time-slot for the day after. If the *User* agrees, appliances will be turned on following the time-schedule provided by the *Aggregator Agent*; otherwise, they will be switched on according to *User*'s preferences.

In a third iteration, the *Aggregator Agent* computes a cost optimisation by considering the shifted loads according to *Users*' responses. This iteration determines both the actual cost and the actions the *Aggregator Agent* will take in the day after, such as how to manage the energy stored in the ESS.

The rest of this section will provide more details on the engine that drives the behaviours of both *User* and *Aggregator Agents*. The engine of the *Market Agent* is very simple, indeed it just periodically send information on day-ahead prices.

A. User Agent Engine

The daily aggregated and disaggregated - i.e. individual appliance - load profiles for each *User Agent* are computed following the methodology in [15]. Appliances are divided into two sub-classes: i) *Shiftable* and ii) *Non-Shiftable*.

Shiftable Appliances can be shifted (e.g. washing machines and dishwashers). The delay in their utilisation creates a certain amount of discomfort. Depending on the delay the *User Agent* may refuse or not the proposed shift. If the *User Agent* accepts, it will receive an economic reward.

Non-Shiftable Appliances cannot be postponed (e.g. TVs and lights) since they would create too much discomfort to the *User Agent*. To model the behaviour of a *User Agent* and its tolerance, the following assumptions have been considered:

Assumption 1. Each *User Agent* is characterised by a *preference* more or less in favour of the DSM program. According to [16], this is strictly related to the different levels of willingness of the *User Agent* to save up at the expense of comfort. This *preference* is modelled with a coefficient between -1 (the *User Agent* does not like the DSM program preferring the comfort) and 1 (the *User Agent* likes the DSM program); 0 stands for a *User Agent* with a neutral opinion. The *preference* is translated into the *opinion coefficient* in range [0,1] in Equation 1.

$$opinion_{coeff} = \frac{preference - (-1)}{2} \tag{1}$$

The *opinion coefficient* or equivalently the *preference* can rise if the *User Agent* likes the proposed shift and consequently its savings, otherwise it decreases (Equation 2).

$$answer = \begin{cases} yes : preference = preference + q \\ no : preference = preference - q \end{cases} \tag{2}$$

where q is an arbitrary quantity that assumes the same value in both cases (e.g. 0.02). Thus, if the *User Agent* answers positively, its *preference* increases (i.e., it is more in favour of using the program) and vice versa.

It follows the implicit assumption that if a rational user participates to the DSM program, it answers positively to the aggregator's request if the proposed shift does not bother him too much. Thus, it is saving money and it seems reasonable to suppose that it will be quite satisfied. Otherwise, it will be quite disappointed.

Technically speaking, we used the opinion coefficient as a way to visualise how well the DSM program is performing (i.e. if the proposed shifts are accepted) and how the performances evolve over time.

Assumption 2. For the selected appliances, it is supposed that a real user does not act and does not answer completely random, i.e. it is influenced by its habits, by its perception and by what causes discomfort to it.

Moreover, it is supposed that if the user receives a reward for shifting the usage of an appliance of a quite small amount (and it has decided to participate to the DSM program), the discomfort created to the user would be so low that it would almost certainly answer positively. In the same manner, a very large shift may create (for any reason) so much discomfort to some users that they would never accept it.

Thus, as done in other works, including [17], [18], a disutility function has been used to indicate the dissatisfaction due to the introduced delay from the desired user time-slot. The more the load is shifted, the more the dissatisfaction function increases in value. It is an idealised function, but as emerges from the survey [14], it is not so distant from reality. We formulated it as the square difference between the desired and the proposed time normalised following Equation 3.

$$Dissat(t_{prop}) = \left(\frac{t_{des} - t_{prop}}{96}\right)^2 \tag{3}$$

The maximum tolerance to the shift is modelled as a threshold *T* w.r.t. the dissatisfaction function. This threshold is the highest value for which the *User Agent* gives a positive answer. After that value (which corresponds to a certain amount of delay), the *User Agent* refuses the proposal made by the *Aggregator*. This threshold has not a fixed value, it can increase or decrease a little depending on whether the *User Agent* likes or not the proposed shifts (see Equation 4). In any case, it is supposed that the user will accept at least one hour of shift since it is participating to the DSM program.

if shift $>$ *1 hour:*

$$answer = \begin{cases} yes & if\, Dissat(t_{prop}) \leq T + preference/100 \\ no & if\, Dissat(t_{prop}) > T + preference/100 \end{cases} \quad (4)$$

It has been supposed that the *opinion* (or *preference*) - which is linked to the performance of the program - has a role in the answer of the *user*. This choice simulates the fact that the *user* behaviour can be influenced by the *user*'s experience. Indeed, there are some examples in experimental projects, e.g. [19], where the users changed their habits after having gained knowledge of their energy consumption and after having understood that they can save up.

Thus, when the *User Agent* sees that the mechanism is rewarding it, it will be more prone to make a little effort. Those who allow for a little shift will increase more the allowed shift (e.g. to contribute/ have more possibility to save money). The same reasoning can be done in the opposite case: the mechanism is not working and it is less willing to contribute.

B. Aggregator Agent Engine

The *Aggregator Agent* performs the optimisations and learns the acceptance of users. It finds the best match among users' profiles and energy sources by managing both shiftable loads and batteries. In this work, batteries are considered as a single virtual battery with a capacity equal to the sum of all capacities.

Every simulated day, the *Aggregator* receives as input the PV production forecast for the upcoming 24 hours exploiting the simulator presented in [20].

The optimisation problem is formulated as a MILP that minimises cost considering PV production, ESS and energy price, as expressed in Equation 5. The energy surplus can be sold to the grid. Thus, the objective function is given as follows:

$$\begin{aligned} \min \quad & \sum_{t=1}^{96} \delta * [c\_pv_t\, PV_t + c\_from_t\, P\_from_t \\ & +c\_dis\_bat_t\, PD\_bat_t - c\_to_t\, P\_to_t] \end{aligned} \quad (5)$$

The power balance, the users' preferences and the ESS limits are taken into account in the constraints expressed by Equations 6-15.

$\diamondsuit$ *ESS Constraints:*

$$PC\_bat_t \leqslant PC\_on * C_{max} \forall t \quad (6)$$

$$PD\_bat_t \leqslant PD\_on * D_{max} \forall t \quad (7)$$

$$E\_bat_t \leqslant capacity\ \forall t \quad (8)$$

$$E\_bat_t \geqslant minCharge\ \forall t \quad (9)$$

$$E\_bat_{t=1} = E\_bat\_init \quad (10)$$

$$\begin{aligned} E\_bat_t \quad &= E\_bat_{t-1} + \delta * eff * PC\_bat_t \\ & -PD\_bat_t * \delta/eff\ \forall t > 0 \end{aligned} \quad (11)$$

$$PC\_on_t + PD\_on_t \leqslant 1\ \forall t \quad (12)$$

$$P\_to_t \leqslant M * (1 - PD\_on_t)\ \forall t \quad (13)$$

The battery has a specific charge/discharge rate that cannot be exceeded (Equations 6-7, respectively).

The energy stored in the battery cannot exceed the maximum capacity and cannot be lower than the minimum charge (Equations 8-9).

The energy stored at t=1, i.e., beginning of a new day, must be equal to the energy stored at t=96, i.e., end of the day, of the previous day (Equation 10). The battery must follow the considered model (Equation 11).

The battery cannot be charged and discharged at the same time (Equation 12). We decided to use the battery for self-consumption (Equation 13).

$\diamondsuit$ *Balance Constraint:*

$$\begin{aligned} & Not\_shift_t + \sum_{i=1}^{3} \sum_{j=1}^{N} \sum_{f=1}^{M} x_f^{ij} L\_shift_{ij} = \\ & = P\_pv_t + P\_from_t - PC\_bat_t + PD\_bat_t - P\_to_t\ \forall t \end{aligned} \quad (14)$$

Power balance must be respected (Equation 14), where L_shift_{ij} is a cycle matrix - actually it is coded as a dictionary - that contains the possible allocation of the consumption vector of the appliance.

$\diamondsuit$ *User request:*

$$\sum_{f=low}^{up} x_f^{ij} = request_{ij}\ \forall i, j \quad (15)$$

Low and *Up* in Equation 15 are referred to the relative appliance and the relative user. If the user does not make a request, the sum of all the binary variables is zero and the appliance remains off. *Low* and *Up* change thanks to the ALA, which goes through the following steps:

Preaction: For the selection of the preaction, a decreasing ϵ-greedy algorithm is used. Thus, for a fraction ϵ of the requests (*Explore*) the optimisation problem receives in input a vector containing time slots in between a number randomly large (e.g. ± 3 hours). Otherwise, the vector in between the shift (*action*) that gives the major reward (*Exploit*) is chosen as input.

Action: The optimisation is performed and the shift proposed is the *action* that is evaluated.

User evaluation: Each *User Agent* communicates to the *Aggregator Agent* its positive or negative answer according to its own threshold.

Update: If the *User Agent* does not accept the request, the *action* is penalised with a really small negative reward. If the *User Agent* accepts the request, that *action* receives a reward *R* proportional to the introduced delay from the desired time slot in such a way that a bigger time shift is rewarded more. Each answer does not count in an equal way. New answers are weighted more w.r.t. previous, since the *User Agent* may change a little bit its opinion. Thus, information on the chosen *action* is updated according to Equation 16.

$$Q_{n+1} = Q_n + \alpha(R_n - Q_n) \quad (16)$$

where Q$_n$ is the estimated value after it has been selected n-1 times, α is a constant step-size parameter and R$_n$ is the n^{th} reward [21].

IV. Case study and results

The proposed solution has been tested by simulating an entire year with 15 min time steps, considering 1011 individual households.

Market price varies according to the Italian day-ahead market price [22] for 2013, with the addition of taxes, system and network charges. We supposed each *User Agent* - i.e., each virtual house - is equipped with a 1 kW photovoltaic system. The Levelized Cost of Energy (LCOE), which takes into account the costs derived from the installation and maintenance, has been set to 0.13 €/kWh according to [23]. The LCOE of the ESS has been set to 0.12 €/kWh, while the photovoltaic surplus is plausibly sold to the grid for 0.1 €/kWh. Each *User Agent* has up to two different shiftable appliances (washing machine and dishwasher), each of them characterised by its own consumption profile.

At the beginning of the simulation, each *User Agent*'s opinion is picked from a normal distribution truncated to the range [-0.8,0.8] (μ=0, σ=1/3). We excluded the extreme values because we supposed that - since the program is new - the *User Agents* do not have too strong preferences (i.e. they have to try it to decide).

Different *User Agents* have different levels of acceptance.

In [14], real people had to rate the level of annoyance between 1-5 (minimum - maximum level of annoyance respectively) w.r.t. the amount of shift up to ±3 hours from the desired appliance's start. It is possible to notice that not all users get close to the maximum level of annoyance (which can be interpreted as a refusal since "if the user is not willing to shift appliance's starting time or set temperature, the value 5 is automatically set" [14]). Therefore, we modelled also certain users that accept shifts larger than 3 hours.

More specifically, the acceptance has been modelled with a normal distribution truncated to the range [1,5] hours (μ=3, σ=1).

To understand the advantage of the proposed algorithm w.r.t. the lack of consideration of *User Agents*' preference, the percentage of affirmative answers with and without ALA is shown in Figure 1. As shown in the plot, if the proposed algorithm is applied, the acceptance rate of *User Agent* increases for the first 5 months up to about 90%. Then, it remains constant to almost 94% (with ϵ=10%). Without applying our solution, the acceptance rate fluctuates over the months between 16% and 32%.

Different cost curves are compared in Figure 2. The pink curve represents the worst case, since it is the cost obtained without any shift. Looking for minimisation only better or equal results may be obtained. The grey curve represents the best case since it shows the savings reached when the acceptance of the *User Agent* is completely ignored, which is clearly an unreal DSM program. Indeed, if the *User Agents* had to evaluate the amount of shifts used to get the grey curve, the light-blue curve would be obtained, since many *User Agents* would refuse the proposed shift. The orange curve demonstrates what can be achieved taking into account *User Agent* preference. Energy savings are lower with respect to the grey curve, but still remarkable with respect to the starting worst case and the one without ALA (light-blue curve).

Figure 3 and Figure 4 show how opinions change with and without ALA.

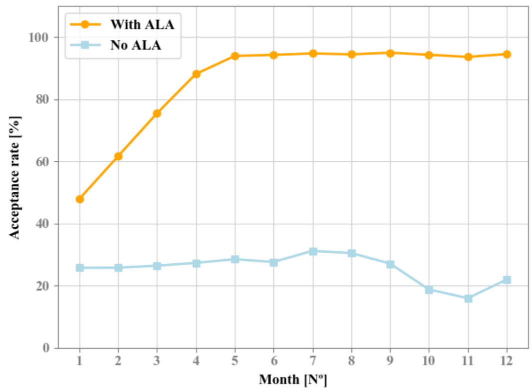

Fig. 1. Acceptance rate with and without ALA

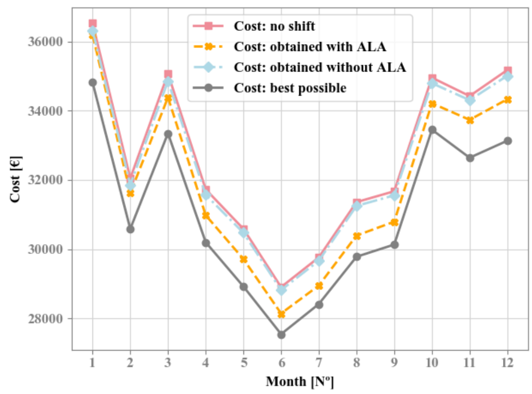

Fig. 2. Monthly cost curves

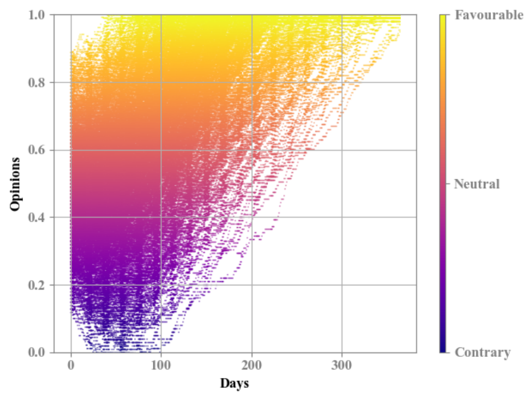

Fig. 3. Evolution of opinions with ALA

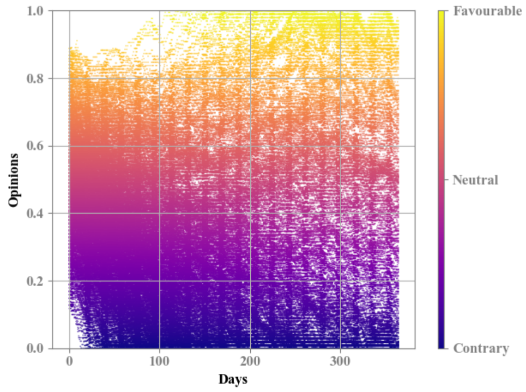

Fig. 4. Evolution of opinions without ALA

In both cases, at the beginning the majority of *User Agents* has a quite neutral opinion. When the proposed algorithm is used (see Figure 3), after an initial period of strong exploration, *User Agents* receive requests for an amount of shifts that they like. Thus, they are saving and all *User Agents*' opinions on the DSM program rapidly converge towards a completely positive one.

Instead, if the *User Agent* acceptance is not taken into account (see Figure 4), for the few who accepted the proposed shifts there is a slower convergence to a positive opinion, while for the majority, the opinions converge to negative ones.

V. Conclusion

In this paper, we addressed the problem of energy cost optimisation with a centralised approach. In particular, we illustrated a different manner to consider user's preferences. In order to learn the user acceptance, a MILP formulation and an algorithm based on an explore-exploit mechanism have been implemented. Despite the strong assumptions, ALA is the first step in a different direction, since we try to give some degree of control to the user while trying to not bother him too much, only asking a yes/no question when the user schedules to use an appliance one day in advanced. Savings and the evolution of the modelled users' opinions for the selected parameters have been shown for a period of one year.

Results demonstrate that the savings obtained with ALA are smaller than the best achievable one but w.r.t. this last case where preferences are not considered, all the *User Agents* are satisfied with the proposed shift, i.e., the proposed shift are in line with the modelled users' preference.

Therefore, it is possible to have a win-win situation for both the aggregator (i.e. profits) and each single user (i.e. monetary gain and comfort).

User opinions are fundamental since they influence user future choices, i.e., sign again the DSM contract.

In our future works, the initial dissatisfaction created to the user will be avoided and the user will be modelled taking into account upstream cognitive processes.

References

[1] A. Barbato and A. Capone, "Optimization models and methods for demand-side management of residential users: A survey," *Energies*, vol. 7, no. 9, pp. 5787–5824, 2014.
[2] J. S. Vardakas, N. Zorba, and C. V. Verikoukis, "A Survey on Demand Response Programs in Smart Grids: Pricing Methods and Optimization Algorithms," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 152–178, 2015.
[3] S. Schütte, S. Scherfke, and M. Tröschel, "Mosaik: A framework for modular simulation of active components in Smart Grids," *2011 IEEE 1st SGMS*, pp. 55–60, 2011.
[4] "Aiomas," https://aiomas.readthedocs.io/en/latest/, accessed:2020-01-24.
[5] K. C. Sou, J. Weimer, H. Sandberg, and K. H. Johansson, "Scheduling smart home appliances using mixed integer linear programming," *In Proc. of: 50th IEEE CDC*, pp. 5144–5149, 2011.
[6] Z. Bradac, V. Kaczmarczyk, and P. Fiedler, "Optimal scheduling of domestic appliances via MILP," *Energies*, vol. 8, no. 1, pp. 217–232, 2015.
[7] R. Jovanovic, A. Bousselham, and I. S. Bayram, "Residential demand response scheduling with consideration of consumer preferences," *Applied Sciences (Switzerland)*, vol. 6, no. 1, pp. 1–14, 2016.
[8] L. Yao, F. H. Hashim, and S. Sheng, "An Optimal Load Scheduling Approach Considering User Preference and Convenience Level for Smart Homes," *In Proc. of: 19th IEEE EEEIC*, 2019.
[9] B. Chai, Z. Yang, K. Gao, and T. Zhao, "Iterative learning for optimal residential load scheduling in smart grid," *Ad Hoc Networks*, vol. 41, pp. 99–111, 2016.
[10] F. A. Qayyum, M. Naeem, A. S. Khwaja, A. Anpalagan, L. Guan, and B. Venkatesh, "Appliance Scheduling Optimization in Smart Home Networks," *IEEE Access*, vol. 3, pp. 2176–2190, 2015.
[11] Z. Pooranian, J. H. Abawajy, P. Vinod, and M. Conti, "Scheduling distributed energy resource operation and daily power consumption for a smart building to optimize economic and environmental parameters," *Energies*, vol. 11, no. 6, 2018.
[12] T. Buechler, F. Pagel, T. Petitjean, M. Draz, S. Albayrak, and T. Universit, "Optimal Energy Supply Scheduling for a Single Household : Integrating Machine Learning for Power Forecasting," *In Proc. of: 2019 IEEE ISGT-Europe*, pp. 0–4, 2019.
[13] E. Amicarelli, T. Q. Tran, and S. Bacha, "Optimization algorithm for microgrids day-ahead scheduling and aggregator proposal," *In Proc. of: 17th IEEE EEEIC*, pp. 1–6, 2017.
[14] V. Pilloni, A. Floris, A. Meloni, and L. Atzori, "Smart Home Energy Management Including Renewable Sources: A QoE-Driven Approach," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 2006–2018, 2018.
[15] L. Bottaccioli, S. Di Cataldo, A. Acquaviva, and E. Patti, "Realistic Multi-Scale Modeling of Household Electricity Behaviors," *IEEE Access*, vol. 7, no. December, pp. 2467–2489, 2019.
[16] ARENA, "Demand Response ARENA Knowledge Sharing Report," *ARENA*, no. September, pp. 78–79, 2018.
[17] D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra, "Residential Demand Response Using Reinforcement Learning," *1st IEEE SmartGridComm10*, pp. 409–414, 2010.
[18] L. Gkatzikis, I. Koutsopoulos, and T. Salonidis, "The role of aggregators in smart grid demand response markets," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1247–1257, 2013.
[19] M. Hansen, "Smart grid development and households in experimental projects," Ph.D. dissertation, 2016.
[20] L. Bottaccioli, E. Patti, E. Macii, and A. Acquaviva, "GIS-based software infrastructure to model PV generation in fine-grained spatio-temporal domain," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2832–2841, 2018.
[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.
[22] "Gme," https://www.mercatoelettrico.org/It/Default.aspx.
[23] https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2019/May/IRENA_Renewable-Power-Generations-Costs-in-2018.pdf.