

A Win-Win Algorithm for Learning the Flexibility of Aggregated Residential Appliances

Original

A Win-Win Algorithm for Learning the Flexibility of Aggregated Residential Appliances / De Vizia, Claudia; Patti, Edoardo; Macii, Enrico; Bottaccioli, Lorenzo. - In: IEEE ACCESS. - ISSN 2169-3536. - 9:(2021), pp. 150495-150507. [10.1109/ACCESS.2021.3125247]

Availability:

This version is available at: 11583/2935372 since: 2021-11-14T22:21:01Z

Publisher:

IEEE

Published

DOI:10.1109/ACCESS.2021.3125247

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

IEEE postprint/Author's Accepted Manuscript

©2021 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collecting works, for resale or lists, or reuse of any copyrighted component of this work in other works.

(Article begins on next page)

A Win-Win Algorithm for Learning the Flexibility of Aggregated Residential Appliances

CLAUDIA DE VIZIA^{1,3}, EDOARDO PATTI^{1,3}, (Member, IEEE),
ENRICO MACII², (Fellow, IEEE) and LORENZO BOTTACCIOLI^{2,3}, (Member, IEEE)

¹Dept. of Computer and Control Engineering, Politecnico di Torino, Turin, IT. (e-mail: name.surname@polito.it)

²Interuniversity Dept. of Regional and Urban Studies and Planning, Politecnico di Torino, Turin, IT. (e-mail: name.surname@polito.it)

³Energy Center Lab, Politecnico di Torino, Turin, IT.

Corresponding author: C. De Vizia (e-mail: claudia.devizia@polito.it).

ABSTRACT In the Demand Side Management (DSM) context, residential customers have the potential for reducing costs and relieving the grid with non-thermostatic appliances. These appliances might be optimally scheduled by a central entity, taking into account user preferences. However, the user might not be able to communicate its preferences “a-priori”, leaving to the central entity the task of understanding preferences that should be learnt without causing discomfort to the user. With this premise, this study aims at exploring a DSM program that learns the acceptance of realistic simulated users to shift in time of home appliances, such as washing machines and dishwashers, analysing the benefits that arise from their inclusion. To this end, the proposed Acceptance Learning Algorithm 2.0 (ALA 2.0) minimises costs in scenarios with different energy sources and with a certain level of acceptance to shift in time, optimally scheduling the appliances according to the boundaries found by the proposed algorithm. ALA 2.0 is able to understand preferences also when modelling a behaviour of the user which is influenced by external factors not directly observable and when users make very few requests, interacting with the user in a simple way. Experimental results highlight that it is possible to understand the acceptance to the shift in time of the simulated users without any prior knowledge and without causing too much discomfort, achieving a win-win situation. As an example, more than 90% of requests were accepted in December, which is chosen as a representative month.

INDEX TERMS Demand Side Management, Energy Aggregator, Agent Based Modelling, Learn User Preference, User acceptance

Nomenclature

Abbreviations

ALA	Acceptance Learning Algorithm
DSM	Demand Side Management
ESS	Energy Storage System
MILP	Mixed Integer Linear Programming
OPT	Optimisation
PV	Photovoltaic

Indices

i	appliances, $i \in \{1, \dots, M\}$
j	households (prosumers), $j \in \{1, 2, \dots, N\}$
t	time slots, $t \in \{1, 2, \dots, T\}$
f_i	interval “allowed” for i , $f_i \in \{\text{low}_i, \dots, \text{upper}_i\}$

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]
C_{max}/D_{max}	max charge/discharge rate [kW]
E_{bat_init}	battery initial condition [kWh]
eff	battery efficiency [%]
$L_{shift_{i,j}}^{t_1}$	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

E_{bat_t}	amount of energy in the battery at time t
P_{from_t}	amount of power from the grid at time t
P_{to_t}	amount of power given to the grid at time t
PC_{bat_t}	charging power of the battery at time t
PD_{bat_t}	discharging power of the battery at time t
PD_{on_t}/PC_{on_t}	binary variable that indicates if the battery is

$x_i^{jt} \in \{0, 1\}$ discharging or charging at time t
 binary variable that selects the day load profile
 of the appliance i of the customer j

I. INTRODUCTION

Residential users are part of the energy transition the world is facing nowadays. Whilst the single consumer has a meaningless impact on the energy system, the whole residential demand could help significantly to exploit more renewable generation, relieve the grid and diminish costs.

To this end, different changes in household practices might occur, from energy conservation to temporal shift and automated appliance scheduling.

Under time-varying rates, these changes should result from the user's actions in response to the price signal. However, the user might not optimally schedule appliances or it might respond poorly: higher responsiveness has been noticed in customers who claim to know how to act to modify the electricity consumption [1].

Consequently, the task of finding the best time-slot for turning on appliances might be left to a central entity in automated demand response programs. However, when dealing with appliances such as washing machines and dishwashers, particular attention should be put into the design of the program in order not to affect the comfort of the user, leaving a certain degree of control and the possibility to override exogenous decisions to the user [2], [3].

In the attempt of understanding the effective tools for motivating user participation in automated demand response programs, [4] surveyed the intention to adopt a demand response program where the energy provider could control its customers' dishwasher to use the excess of solar energy. In the proposed program, it was guaranteed that the appliance would have automatically been started within 6 hours and that the automatic control could have been prevented twice per month.

With this background in mind, this paper proposes a DSM program similar to the one surveyed in [4] but where a realistic user can decide whether to accept or refuse the proposed shift of appliances such as washing machine and dishwasher - hereinafter referred to as "shiftable appliances" - on the basis of its preferences, which are centrally learnt by the proposed algorithm. Consequently, the proposed work tries to address three major challenges: i) finding the optimal schedule of users' appliances when preferences are not known a-priori but centrally learnt in an interactive way by proposing shifts in the use of an appliance that matches the user's preference. This minimises the users' discomfort and maximises the users' acceptance; ii) modelling of realistic user, i.e. not completely rational, including random factors that influence the acceptance of the shift proposed by the aggregator. The new model of the user makes it more difficult finding the solution; iii) testing a user-friendly DSM program to understand the possible consequences of this DSM program in the real world.

To simulate and test the DSM program, the following

actors were modelled: i) the *Aggregator*, responsible for the DSM program ii) the *Users* representing the residential customers who enrolled their appliances in the DSM program and iii) the *Market*, which informs the *Aggregator* on the day-ahead prices.

In order to centrally learn the acceptance of users, we used a Mixed Integer Linear Programming (MILP) formulation which aims at minimising costs from different energy sources coupled with an improved version of the algorithm presented in [5], i.e. the Acceptance Learning Algorithm 2.0 (ALA 2.0).

Fig. 1 shows an oversimplified and intuitive example of a user making a request to use the washing machine. In a nutshell, the shift proposed by the MILP formulation is evaluated by the user and its acceptance or refusal is used to increase the knowledge on its preferences. Depending on the answer, the appliance is turned on or off, accordingly. From a psychological point of view, this mechanism might be perceived differently from the user since i) the user is in control of its appliances and ii) from the user perspective, it is not communicating personal preferences, it is just evaluating the proposal of a central entity.

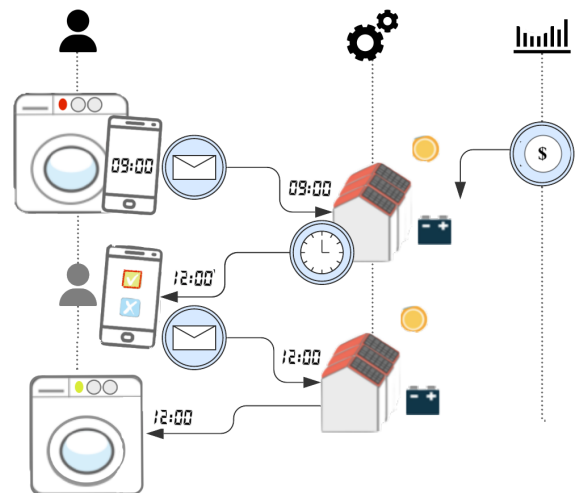


FIGURE 1: Example of the proposed DSM program

Then, we tested different scenarios increasing progressively the flexibility of the users, with and without the presence of Photovoltaic (PV) systems and Energy Storage Systems (ESS). Results obtained from the different scenarios are compared, demonstrating that the algorithm is always able to learn the acceptance to the shift in time of the simulated users.

A. RELATED WORK

Literature works propose various solutions and approaches for taking into account the user's preference, i.e. the maximum temporal shift for appliances. The discriminant factor to select literature solutions described in the following was

the way in which such studies considered and modelled the user.

In [6] and [7], the boundaries allowed for shifting the appliances are known, immutable and equal for all users. Melhem et al. [6] compare a MILP model with the proposed math-heuristic optimization algorithm under different scenarios. Mathematical models for the grid, renewable energy sources, ESS and electric vehicles are considered. In [7], costs are minimised following the known constraints on the customer's preference. Both PV systems and ESS are considered in the MILP formulation.

A mixed integer programming formulation where preferences are communicated by the customer once and updated only if limits change is proposed in [8]. The authors considered 250 users that manifest the level of preference in different periods where the appliance may be turned on.

Liu et al. [9] analyse a residential demand response program where the most representative appliances are discussed. The control strategy used for the shiftable non-thermostatic loads, e.g. the dryer, is "price naming" where the appliance is turned on when the locational marginal price drops below the desired price threshold. Annual costs of different scenarios - with and without load control, in the presence or absence of a solar farm with ESS - are compared. Instead, Manganelli et al. [10] propose a case study of an existing residential and commercial building with a microgrid and control systems, preserving users' habits and comfort. The shiftable loads considered are washing machines and dishwashers. The building energy management system indicates possible slots and their costs, then the user selects a slot.

To schedule home appliances minimizing both dissatisfaction of user and costs, [11] proposes a multi-objective DR optimization model solved through the Constrained Many-Objective Non-Dominated Sorted Genetic Algorithm. Maximum start-end times are defined by the user. RES and ESS are also considered. A multi-objective DR optimization model is also used in [12], which is solved thanks to the genetic algorithm. In the objective function, it uses a weighting factor representing the proportion of power consumption cost, based on day-ahead electricity price, and discomfort cost. The user can strike a balance between discomfort and cost through the weighting factor. Thermal loads, flexible deferrable loads (e.g. electric vehicles) and non-flexible deferrable loads are considered. Two scenarios are compared: i) the first scenario equally weighs power consumption and discomfort costs; ii) the second considers consumption cost only.

For a single household, [13] uses Q-Learning without the assumption of knowing the dis-utility function of user's dissatisfaction. The authors propose a fully automated energy management system that receives a request from the user for using an appliance. The request time does not necessarily coincide with the target time, which represents when the user prefers the request to be satisfied. Then, the energy management system schedules when to satisfy the request. The user can decide to cancel requests not completed yet.

Then, the user evaluates some completed/cancelled requests. The evaluation corresponds to the user's dissatisfaction.

In [14], a quality of experience-driven approach is used. Starting from a survey of 427 subjects, it found no correlation between appliances usage habits and users' data. Therefore, through the k-means algorithm, it obtains different profiles on the basis of the preference collected through a questionnaire. The users had to indicate the degree of annoyance from 1 to 5 - the minimum and the maximum level, respectively - for certain delay up to 3 hours. Instead, new customers answer for a short amount of time to annoyance rating questions because of task shifting. Then, one of the obtained profiles is assigned to the new customer. Two algorithms are illustrated to find the optimal time-slot for the appliances. Hakimi et al. [15] proposes a new method for certain types of controllable loads (i.e. dishwasher, washing machine and heating/cooling system) where the maximum shift was chosen considering the consumers' welfare extracted from a survey. The use of appliances is shifted to the time at which the difference between load and RES power is maximum, always taking into account the consumers' welfare.

B. CONTRIBUTION

Table 1 summarises the proposed literature review highlighting the main features of each solution and comparing them with our.

The column "*user preference*" resumes how user preferences are considered. Two main categories emerge: *known*, i.e. decided a-priori or set on the basis of the preferences of the majority, and *learned*. According to [14], "*most of the literature solutions consider the customer comfort as a set of hard constraints on appliance usage, a-priori set without profiling among different kinds of customers, which are likely to have different subjective needs. Moreover, emphasis is often put on the cost or energy optimisation, but no metrics for a-posteriori evaluation of the perceived quality is given*". As in [14], we wanted to focus on the end-user. However, we opted for a real-time evaluation of the proposed shift. Authors in [12] offer the possibility to the user to find a balance between discomfort and cost. However, the consequences of a certain value are not so immediate. Indeed, "*different people are at different stages of awareness of effective actions to take during a demand response activity. Learning to be energy adaptable and energy efficient is a journey where steps to improve results, increase ease and adjust comfort levels are taken one step at a time*" [16]. Therefore, we chose to ask an individual "yes" or "no" question which should be easily understood by users and which can guide the user in making optimal choices.

Depending on how the user is modelled, the method to solve the problem might change (refer to column "*Method*" in Table 1). Similarly to [13], we aim at learning the acceptance of the user to the shift of appliances. Nevertheless, we opt for a centralised optimisation with a view to a future collective energy community. Therefore, the main contribution of this paper w.r.t. literature solutions consists of trying to learn

TABLE 1: Comparison of proposed and literature solutions

	User preference	Method	HHs no	Time slots	RES	ESS
Melhem et al. [6]	<i>Known</i>	math-heuristic optimization algorithm	1	1 hour	✓	✓
Amicarelli et al. [7]	<i>Known</i>	MILP	4 groups of residential/small commercial users + 25 houses, each represented by an agent.	1 hour	✓	✓
Jovanovic et al. [8]	<i>Communicated by the user</i>	MIP	250	1 hour	✗	✗
Liu et al. [9]	<i>Price naming</i>	proposed algorithms	1000	1 hour	✓	✓
Manganelli et al. [10]	<i>User selection</i>	different control strategies	an existing residential and commercial building	15 min	✓	✓
da Silva et al. [11]	<i>Trade-off between cost and satisfaction</i>	Constrained Many-Objective Non-Dominated Sorted Genetic Algorithm	1x3 households	1 hour	✓	✓
Wei et al. [12]	<i>Trade-off between cost and satisfaction</i>	Genetic algorithm	1	15 min	✗	EV
Wen et al. [13]	<i>Trade-off between cost and satisfaction-learnt</i>	Q-learning	1	15 min	✗	✗
Pilloni et al. [14]	<i>Survey based-learnt</i>	2 proposed algorithms	427 real user for the survey +1000 for the simulation	30 min	✓	✗
Hakimi et al. [15]	<i>Survey based-known</i>	proposed method	Ekbatan residential complex	15 min	✓	✓
Our previous work [5]	<i>Learnt</i>	ALA 1.0	1011	15 min	✓	✓
Proposed solution	<i>Learnt</i>	ALA 2.0	3000	15 min	✓	✓

preferences in a centralised way using dynamic constraints set by the proposed ALA 2.0 in the MILP formulation for the considered 3000 users. Instead, methods such as Q-Learning suffer from the “curse-of-dimensionality” which limits the problem to few users [13].

With respect to our previous work [5], which presents Acceptance Learning Algorithm 1.0 (ALA 1.0), the main contribution lies in learning the user preferences without causing too much discomfort to all the users, i.e. we diminished the number of times in which the central entity proposes a time-slot for turning on the appliance which does not respect user preferences. Moreover, we modelled the users’ behaviour more realistically. Sometimes, even if the proposed shift meets the user preferences, the request from the aggregator is not accepted by the user to model external factors that might influence the routine of the user. These refusals will be hereinafter referred to as “random refusals”. Thus, the users are no longer modelled based on how they “should” behave, rather their behaviour is more descriptive, including also random factors not directly observable. Furthermore, we increased the number of users to 3000 to obtain more realistic simulations, comparing new scenarios with different sets of energy sources. Consequently, the performance of ALA 2.0 with the new more realistic users has been compared with ALA 1.0 presented in [5], demonstrating that ALA 2.0 learns the time-slots preferred by users without causing too much discomfort to them, even in the case of users that do not use the appliances very often or randomly reject the shifts.

II. THE FRAMEWORK

In this section, we present the proposed framework by introducing briefly the tools used, listing the models and describing the agents. For our framework, we chose Mosaik [17], a flexible and modular smart-grid co-simulation framework,

to coordinate simulators and Aiomax [18] to create the agent environment, i.e. the actors.

The simulators implement the models of the electricity consumption of each family, the PV systems generation and ESS. The electricity consumption - generated using [19] - is related to the usage of the appliance of each household. Thus, each family is always characterised by its own load profile. Instead, the PV systems from [20] and a generic model of an ESS (see the formulation in Section II-B) may or may not be included.

The agents represent the intelligence that controls the models. Three types of actors interact in the proposed scenarios: i) the *User* which decides whether to answer positively or negatively to the requests of the *Aggregator* according to its perceived discomfort, ii) the *Aggregator* which manages the DSM contracts, optimally shifting the appliances and learning the user acceptance with ALA 2.0, iii) the *Market* which communicates the day-ahead market price. The main interactions are shown in Fig. 2.

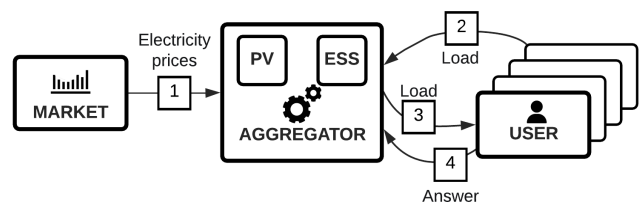


FIGURE 2: The main interactions among the agents

The simulation evolves in time steps of 15 minutes, however, two time-slots are more important than the others as described below.

i) Every (simulated) day at 21:00, the *Market* informs the *Aggregator* on the day-ahead market prices (see block "1" in Fig. 2). In the same manner, each *User* communicates the

aggregated daily load schedule as well as each shiftable appliance schedule to the *Aggregator* (see block "2" in Fig. 2). The *Aggregator* knows also the 24-hour-ahead forecast of the PV production. Using the obtained data it performs two optimisations - OPT 1 and OPT 2. OPT 1 minimises cost without shifting any appliance: it is used as a baseline to understand potential savings. Instead, OPT 2 might anticipate or postpone the shiftable appliances of a certain amount of time according to the knowledge of the *Aggregator* on user acceptance. OPT 2 is used to propose to the user the optimal action (see block "3" in Fig. 2).

ii) Every (simulated) day at 23:00, the *user* communicates its answer to the proposed shift (see block "4" in Fig. 2). This step gives more information on user preferences. Then, the optimisation, that decides the strategy for the day after, is performed using the obtained information. Therefore, costs are minimised by shifting only the appliances that have been allowed to be shifted.

The rest of this section will describe in-depth the engine for the *User* and the *Aggregator*. The *Market* simply provides the day-ahead prices. In future works, we plan to enhance it by implementing advanced market dynamics and policies.

A. THE USER AGENT

The *User* represents a realistic simplification of the real household. To model its decisions w.r.t. the delay of the shiftable appliances, first we distinguished between i) aggregated electricity consumption, containing the aggregated load profile of all not-shiftable appliances, and ii) dis-aggregated load profiles of the shiftable appliances, i.e. the washing machine and the dishwasher, using the simulator proposed in [19].

Then, we modelled the discomfort created to the user when shifting certain appliances from the desired starting time. If this delay causes dissatisfaction greater than the one tolerated, the switching-on of the appliance is not anticipated or postponed. In this case, there is no economic punishment for the user. On the contrary, if the *User* accepts the request, it receives an economic reward (direct or computed as the savings derived from the shift).

Thus, to model the behaviour and the level of acceptance of the *User*, two assumptions have been made.

Assumption 1. At the beginning of the simulation, each *User* has an opinion on the DSM program described by a coefficient in the range [0,1]. It is 0 when the user does not like the DSM program at all; 1 when the user strongly appreciates it. Accordingly, 0.5 stands for a *User* with a neutral opinion. The opinion of the *User* changes based on the shift proposed by the *Aggregator*. Indeed, the opinion highlights how well the DSM program is performing: if the proposed shifts are in line with the preferences of the user, the opinion increases; while if the *User* refuses the request of the *Aggregator*, the opinion decreases. This is formulated by Equation 1 as follows:

$$answer = \begin{cases} yes : opinion = opinion + q \\ no : opinion = opinion - q \end{cases} \quad (1)$$

where q is an arbitrary quantity equal in both cases, i.e. 0.01. Therefore, we removed the assumption, modelled in [5], through which the user acceptance is partially influenced by its experience to obtain a clear lower bound. In the previous model of the user in [5], the answer to requests was deterministic, except for shifts close to the maximum acceptance, i.e. those influenceable. Instead, with the proposed model of the user, the behaviour is more descriptive, including also random factors not directly observable, i.e. also the request for small shifts from the aggregator might be refused due to external factors that might influence the routine of the user, i.e. the "random refusal".

In case the proposed time-slot is exactly the one chosen by the *User*, the answer is implicitly affirmative and the opinion does not undergo variations. Therefore, the opinion of the *User* is a way to visualise the performances in time of the algorithm from the point of view of users' preferences. Thus at each time step, it will be visible if the proposal of the aggregator matches the user's preference since an increase in the opinion of the amount q means that the user accepts the request, while a decrease stands for a refusal.

Assumption 2. It has been supposed that a *User* acts according to its level of comfort and routine. W.r.t. our previous work [5], the *User* is not utterly rational, i.e. if the discomfort is below a certain threshold the answer may be negative. Indeed, a degree of randomness in the answer depending on external factors, i.e. deviations from the usual behaviour due to commitments, has been included. Thus, the *User* may decline usually accepted shifts. Therefore, the rational behaviour of the *User* has been modelled using a disutility function like [21], [22]. It indicates the dissatisfaction created by the delay from the desired time-slot: major shifts correspond to greater dissatisfaction. We formulated it as the square difference between desired and proposed starting time normalised following the Equation 2.

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

The threshold w.r.t. the dissatisfaction function indicates the maximum value that usually corresponds to an affirmative answer (i.e. excluding random answers). Over the threshold, the *User* does not accept the proposal made by the *Aggregator*. An example is proposed in Fig. 3.

The different simulations will consider three degrees of acceptance. The minimum level of this threshold corresponds to a 1 hour delay, according to the survey in [2]. The maximum flexibility tested reaches arbitrarily 5 hours. Additional acceptance to the shift would lead to even further savings.

B. THE AGGREGATOR AGENT

The *Aggregator* manages the DSM contracts, decreasing costs and learning the acceptance of the *Users*. The process

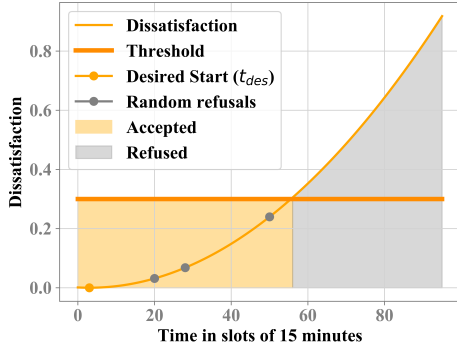


FIGURE 3: An example of the dissatisfaction curve

followed by the *Aggregator* to discover users' preference while minimising costs is described in the following and through the Algorithm 1.

1. Pre-action: This step aims at discovering the shift in time allowed by the user to be given as input to the optimisation. For the selection of the pre-action, i.e. the shift allowed, a decreasing ϵ -greedy algorithm is used. Thus, ALA 2.0 chooses as input the vector in between the shift (see the next step *Action*) that gives the major reward (*Exploit*, see lines 11-13 in Algorithm 1) for a fraction of the requests - similarly to ALA 1.0 [5]. Otherwise, a "guided" exploration (*Explore*, see lines 14-16 in Algorithm 1) is performed in order to avoid too much discomfort to the user, i.e. it starts from exploring smaller shifts, increasing the time-slots available each time the user answers positively or decreasing the time-slots allowed if it refuses the proposal of the *Aggregator*. In this version of the algorithm, the value of ϵ arbitrarily decreases each month down to the value of 0.1. It does not reach zero to keep collecting information and capture eventual variations in the behaviour of the user. With real users, ϵ should be appropriately calibrated.

2. Action: The optimisation, that may include PV systems and ESS in accordance with the designed scenario, is performed. This optimisation schedules the start of the shiftable appliances in the best allowed time-slot. This time-slot is the "action" that will be proposed to and evaluated by the user (see line 18 in Algorithm 1). The objective of the optimisation (Equation 3) is to minimise costs considering day-ahead prices (c_{from_t}) and, if installed, the cost related to PV systems (c_{pv_t}) and ESS ($c_{dis_bat_t}$). In the latter case, i.e., the presence of PV systems, the surplus may be sold to the grid at a price c_{to_t} .

$$\min \sum_{t=1}^{96} \delta * [c_{pv_t} P_{V_t} + c_{from_t} P_{from_t} + c_{dis_bat_t} P_{D_bat_t} - c_{to_t} P_{to_t}] \quad (3)$$

In this simplified scenario, we consider an individual virtual battery with a capacity equal to the sum of all capacities replacing the ESS. The following constraints are not valid if ESS is not present (Equations 4-11).

◇ *ESS Constraints:*

$$PC_{bat_t} \leq PC_{on} \cdot C_{max}, \forall t \quad (4)$$

$$PD_{bat_t} \leq PD_{on} \cdot D_{max}, \forall t \quad (5)$$

$$E_{bat_t} \leq capacity, \forall t \quad (6)$$

$$E_{bat_t} \geq minCharge, \forall t \quad (7)$$

$$E_{bat_{t=1}} = E_{bat_init} \quad (8)$$

$$E_{bat_t} = E_{bat_{t-1}} + \delta * eff * PC_{bat_t} - PD_{bat_t} * \delta / eff, \forall t > 0 \quad (9)$$

$$PC_{on_t} + PD_{on_t} \leq 1, \forall t \quad (10)$$

$$P_{to_t} \leq M \cdot (1 - PD_{on_t}), \forall t \quad (11)$$

Specific charge and discharge rates are associated with the battery since these bounds cannot be exceeded (Equations 4-5, respectively). The maximum capacity and a minimum charge characterise the battery (Equations 6-7), while the energy stored at the beginning of a new day is equal to the energy stored at the end of the previous day (Equation 8). Furthermore, the battery follows the simplified model (Equation 9). Charge and discharge cannot happen simultaneously (Equation 10). In addition, we decided to use the battery for self-consumption (Equation 11).

◇ *Balance Constraint:*

$$Not_shift_t + \sum_{i=1}^3 \sum_{j=1}^N \sum_{f=1}^M x_f^{ij} L_shift_{ij} = P_{pv_t} + P_{from_t} - PC_{bat_t} + PD_{bat_t} - P_{to_t}, \forall t \quad (12)$$

Power balance is preserved thanks to Equation 12. In case PV systems are not present, there is no possibility to sell the surplus of energy to the grid. L_shift_{ij} is a cycle matrix containing the feasible allocation of the consumption vector of the shiftable appliances, while x_f^{ij} is a binary variable that selects the day load profile of the appliance i of the customer j . All the parameters that have been presented in bold in Equation 3 and Equation 12 together with the ESS constraints are optional, thus they are removed if the scenario does not include the corresponding model.

◇ *User request:*

$$\sum_{f=low}^{up} x_f^{ij} = request_{ij}, \forall i, j \quad (13)$$

Low and *Up* limits are associated with each appliance and with each user and depends on the pre-action. When the user j decides to use appliance i , $request_{ij}$ is set to 1. Thus, the appliance must be turned on that day. On the contrary, the sum of the binary variables is zero and the domestic appliance is not turned on.

3. User evaluation: After the OPT is solved, the optimal shift is communicated to the *User*, which accepts or declines the proposal of the *Aggregator* congruently to **Assumption 2**, i.e. the threshold plus a random behaviour.

4. Update: When the *User* refuses the request, the action is penalised with a small negative reward. Otherwise, the action

is rewarded proportionally to the amount of delay from the desired start, i.e. greater time shifts are rewarded more. If the *Explore* pre-action had been chosen, the bookmark is updated based on the user answer (see lines 22-25 and 29-33 in Algorithm 1). Recent answers weight more w.r.t. previous one, in case the *User* changes its opinion. Therefore, information on the selected action is updated following Equation 14.

$$Q_{n+1} = Q_n + \alpha(R_n - Q_n) \quad (14)$$

where α is the constant step-size parameter, Q_n is the estimate for the n^{th} reward and R_n represents the n^{th} reward [23].

Algorithm 1 The Acceptance Learning Algorithm 2.0

```

1: actions ← [no_shift, 1_hour_shift, ..24_hour_shift]
2: bookmark ← 1                                ▷ only no and 1 hour shift possible
3: best_action ← 24                             ▷ all 25 actions possible
4: ε, α ← chosen values
5: Q initialised to 0 for all actions
6: for day:=1 to day:=365 do
7:   if new month and ε > 0.1 : then
8:     decrease ε
9:   end if
10:  p ← random(0, 1)
11:  if p < (1 - ε) then
12:    explore ← 0
13:    preaction ← best_action
14:  else
15:    explore ← 1
16:    preaction ← action indicated by bookmark
17:  end if
18:  action ← shift decided by the MILP problem
19:  prosumer_evaluation ← yes/no
20:  if prosumer_evaluation = yes then
21:    R ← proportional to the delay
22:    if explore then
23:      if bookmark ≠ 24 then
24:        bookmark ← bookmark + 1
25:      end if
26:    end if
27:  else
28:    R ← penalty
29:    if explore then
30:      if bookmark > 1 then
31:        bookmark ← bookmark - 1
32:      end if
33:    end if
34:  end if
35:  Q ← Q + α(R - Q)                            ▷ of the chosen action
36:  best_action ← action with highest value
37: end for

```

Thus, the difference w.r.t. ALA 1.0 (presented in our previous work [5]) is in the *Explore* preaction. For the *Explore* preaction of ALA 1.0 a random number \mathbf{r} in between 1 and 24 is generated. It corresponds to the amount of hours considered for the shift. Consequently, a vector that includes all the time-slots in between \mathbf{r} hours in advance and \mathbf{r} hours after the time-slot requested by the user is created. Depending on a random number, the time shift proposals might be not in line with user preferences for a long period, causing discomfort to the user. Instead, with ALA 2.0, at the beginning few actions, corresponding to smaller shifts, are allowed to increase the probability of asking shifts liked by the user. This allows to include also users that use appliances not very often. Then, if the user answers positively to any amount of shift during an *Explore* preaction, the number of possible actions increases. At the same time, the amount of exploration decreases with time, decreasing the mechanism

that increases the number of actions. Therefore, guiding the exploration, ALA 2.0 decreases the discomfort of the user.

III. RESULTS

In this section, first we present the experimental results for a single *User* to demonstrate how ALA 2.0 works; then, we discuss its performances when 3000 *Users* are running simulating a realistic district in a city. In both cases, we compare ALA 2.0 performances with the previous ALA 1.0 [5].

A. SINGLE USER ANALYSIS

In the following, we present the results in simulating a single *User*, which corresponds to a single family with only one shiftable appliance, i.e. the washing machine, 91 washing in the simulated year. The random refusals, arbitrarily set in this case, corresponds to the 5th, 25th, 45th, 65th, 85th request.

We propose an example where the family has only 1 hour of flexibility with ALA 1.0 and ALA 2.0, see Fig. 4. The amount

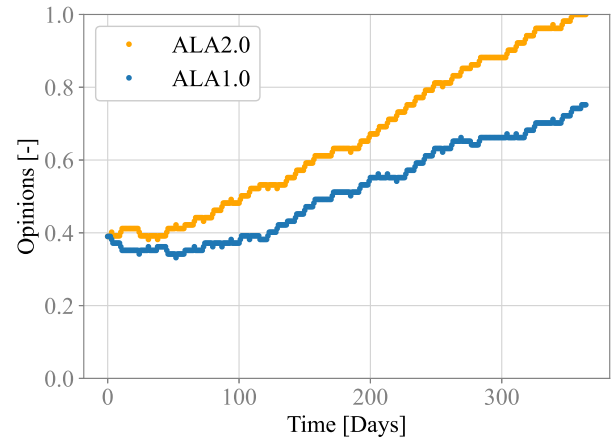


FIGURE 4: Comparison of ALA 2.0 and ALA 1.0 for a single family

of exploration is the same in both algorithms. However, since ALA 2.0 has few possible preactions among which the OPT 2 can choose, in this case, it finds faster certain proposals liked by the user (orange line in Fig. 4). Instead with ALA 1.0, the user is more annoyed (blue line in Fig. 4). The number of refusals, excluding the random ones, are 8 with ALA 2.0 and 20 with ALA 1.0. This graph allows verifying how the opinions evolve in time. For simulations with more than one user, it visually shows if all the users are accepting the proposed shifts since the overall result in percentage might hide really good results for some users and really poor for others.

ALA 1.0 performance strongly depends on the random selection of the preaction, and the action chosen, consequently. That is why, depending on the random selection, ALA 1.0 might perform very well for some users and not that much for others. Instead, ALA 2.0 should ensure the satisfaction of almost all the customers (from the point of view of requests in

line with what is liked), thanks to the guided exploration. The 3000 users, with a different number of appliances, requests and random refusals, allow to observe the general trend.

Being a trial and error algorithm, performances have also been compared with the optimal solution, i.e. the solution obtained knowing preferences a-priori. Thus, we run three optimisations, with i) the same family, ii) the same "random" refusals and iii) 1-3-5 hours of flexibility, where the allowed shifts were known, i.e. the *User* communicates the boundaries allowed for the shift representing its acceptance. Then, we compare the optimal shifts suggested by the *Aggregator* in the just mentioned scenario with the shifts suggested with ALA 2.0, i.e. where the acceptance must be learnt. The number of identical requests are 79, 60, 73 for 1 hour, 3 hours, 5 hours of flexibility, respectively.

Last but not least, we want to understand what would happen if the user changes its behaviour, e.g. the recent COVID-19 pandemic and the consequent lockdown caused a change of users' habits. Two cases are illustrated: from 1-hour acceptance to 3-hours (Scenario 1) and vice versa (Scenario 2). Starting from the ϵ equal to 0.8 in January, it decreases to 0.6 in February, then, to 0.4 in March, to 0.2 in April and from May it is set to 0.1. The reward for each action is arbitrarily set to the hours of shift multiplied by a factor 0.1 and the penalty to -0.00001. The family changes the acceptance level in May.

In the first case, at the beginning of May the estimated values of the actions, different from zeros, are shown in Table 2. Thus, the best action, i.e., the one with the highest Q value, corresponds to the user flexibility (± 1 hour). At the end of the year, the best action is ± 2 hours. The algorithm needs to continue exploring to increase the value of the ± 3 hours action, but it does not bother the user. In the second case (see Table 3), in May the best action is correctly 3 hours, while at the end of December it is 1 hour (see 1st row, column " ± 3 h" and 2nd row, column " ± 1 h", respectively in Table 3).

TABLE 2: Q values of the actions - Scenario 1

	Actions					
	± 1 h	± 2 h	± 3 h	± 4 h	± 5 h	± 6 h
1 st May	0.0998	-9.85e ⁻⁰⁶	-6.5e ⁻⁰⁶	0	0	0
31 st Dec	0.0998	0.1999	0.1049	0	-6.5e ⁻⁰⁶	0

TABLE 3: Q values of the actions - Scenario 2

	Actions					
	± 1 h	± 2 h	± 3 h	± 4 h	± 5 h	± 6 h
1 st May	0.0994	0.0699	0.1049	-6.5e ⁻⁰⁶	0	-8.7e ⁻⁰⁶
31 st Dec	0.0999	0.0085	0.0015	-8.7e ⁻⁰⁶	-6.5e ⁻⁰⁶	-8.7e ⁻⁰⁶

B. THE 3000 USERS ANALYSIS

ALA 2.0 has been tested for 3000 *Users* in presence of two different sets of energy sources: one composed by the market only (Scenario 1) and the other made up of the market, the PV systems and the ESS (Scenario 2). The simulations, which last for one year with a 15 minute time step, have been

repeated with the *Users* offering a flexibility of 1 hour, 3 hours and 5 hours, generating six case studies.

The costs (see Equation 3) or the levelised cost of energy - which includes the costs of the installation and maintenance - are i) c_{from_i} : the cost from [24] for 2013 plus taxes, system and network charges, ii) c_{pv_i} : 0.13 €/kWh from [25] (1 kW PV system per family), iii) c_{to_i} : 0.1 €/kWh and iv) $c_{dis_bat_i}$: 0.12 €/kWh (total capacity 3720 kWh). The ϵ and the values of rewards are configured according to Section III-A. Each *User* owns a washing machine and/or a dishwasher. At the beginning of each simulation, the opinion of the *User* is arbitrarily taken from a normal distribution, with mean $\mu=0.5$ and standard deviation $\sigma=1/3$, truncated to the range [0.2,0.8]. In these simulations, the uptake of the program is not taken into account, thus it has been supposed that *Users* who signed in have not an initial extreme opinion, neither positive nor negative. They will first try the DSM program and then they will evaluate it. Each time the proposed shift is below the threshold (see Section II-A), each user has the 5% of probability to answer negatively.

SCENARIO 1: Fig. 5 reports the savings obtained with ALA 2.0. The baseline, i.e., the optimisation without load shifting, is depicted with the pink line. If the level of flexibility is in between ± 1 hour, results are poor. Only around 0.22% of savings w.r.t the baseline case are obtained. It should be also pointed out that, in this first approximation, capacity constraints were not included, which with a good probability will further decrease the possibility to shift the loads. With a level of acceptance of ± 3 hours, around 0.49% can be saved. When the flexibility is ± 5 hours, results improve further. Indeed, savings reaches around 0.68%. The green line in Fig. 5 represents the best case where all the users answer positively to all the requests of the *Aggregator*, which is hard to achieve in the real world. In this case, savings reach 1.46%.

We compared the results with those obtained with ALA 1.0. From the *Aggregator* prospective the differences are almost negligible, i.e. savings with ALA 1.0 and ALA 2.0 are almost the same and trends are almost overlapped. Thus, to increase the readability of the plot, we did not report ALA 1.0 trends in Figure 5. Instead, from the *Users* point of view, the difference is remarkable as shown in Fig. 7, which reports the variations of the opinions of the 3000 *Users* in one year for both ALA 1.0 and ALA 2.0. The opinion of those who make more requests increases very fast. In the end, the opinion of the majority becomes completely favourable. No one has an opinion more contrary than the starting one, with one only exception in Fig. 7b. This represents a customer that makes very few requests. If there is a combination of few requests, random refusals and wrong guesses from ALA, this might happen. Nevertheless, considering all the other results, this represents a single case. As shown in Fig. 7, at the beginning of the simulation, ALA 1.0 causes much more discomfort to all users. Then, it discovers the flexibility for the majority of users, while for some others, especially in the case with 1 hour flexibility (see Fig. 7d), it bothers the users

too much.

To understand the advantages of ALA 2.0 w.r.t. ALA 1.0, the results for December are shown in Fig. 8. The total acceptance rate is computed as the total number of “yes” from all users divided by the total number of requests in percentage. We chose December as a significant month, since it is a month where the exploration does not change further, i.e. ϵ is set to 0.1. In December, with users with one hour of flexibility, the total acceptance rate is 90.36% with ALA 2.0 and 86.20% with ALA 1.0. However, if we consider the acceptance rate per user, with ALA 1.0, 73 users have between 0% and 10% of acceptance rate, while this decreases to 2 users only with ALA 2.0 (see Fig. 8a, 8d). These users turn on their appliances very few times and might also “randomly refuse” the requests from the aggregator. As already pointed out, from the aggregator viewpoint, there are not remarkable differences among both ALAs since users who make few requests have not a great impact on costs. Instead, from the users’ viewpoint, differences are significant as all users are treated equally and it is offered to each individual user the opportunity to participate. Thus, ALA 2.0 is fairer to users. As the acceptance rate of the user increases, the differences in performances between ALA 1.0 and ALA 2.0 decreases (Fig. 8b, 8e and Fig. 8c, 8f). For the sake of completeness also results for the whole year are shown in Fig. 9. In this case, the acceptance rate is lower because it also includes the first months of our simulated year, where ALA 2.0 is performing a strong exploration phase, thus it is learning.

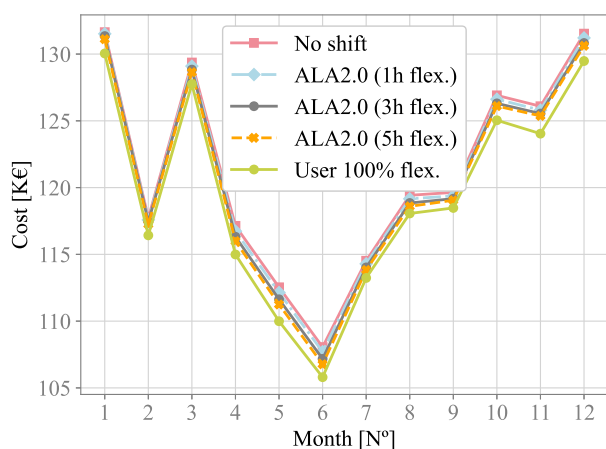


FIGURE 5: Costs in presence of the market only

SCENARIO 2: Fig. 6 reports the costs of ALA 2.0 in the scenario with PV systems and ESS. The cost of the baseline (pink line) is definitely lower than the one in Scenario 1. Savings w.r.t. the pink line of this scenario for ± 1 hour, ± 3 hours, ± 5 hours are 0.88%, 1.93%, 2.53%, respectively. Thus, the 3 hours shift scenario has higher savings than the ones of the unlikely 100% flexibility in the Scenario 1. The green line - the case where the users always accept the shift - would save 3.89%. It should be noticed that the

different curves of costs are an implicit way to obtain some indications of the performances from the point of view of the aggregator since they are decreasing with the increase of flexibility, i.e. ALA is taking advantage of the increased flexibility. Also in this scenario, from the *Aggregator Agent* viewpoint, differences in terms of savings in using ALA 1.0 or ALA 2.0 are negligible. Thus, Fig. 6 reports only the trends of costs when ALA 2.0 is applied. From the User Agents’ perspective, with both versions of ALA results improve w.r.t. the previous scenario, i.e. cases in Figures 10a to 10c w.r.t. those in Figures 7a to 7c, and cases in Figures 10d to 10f w.r.t. those in Figures 7d to 7f, yet ALA 2.0 creates much less discomfort than ALA 1.0.

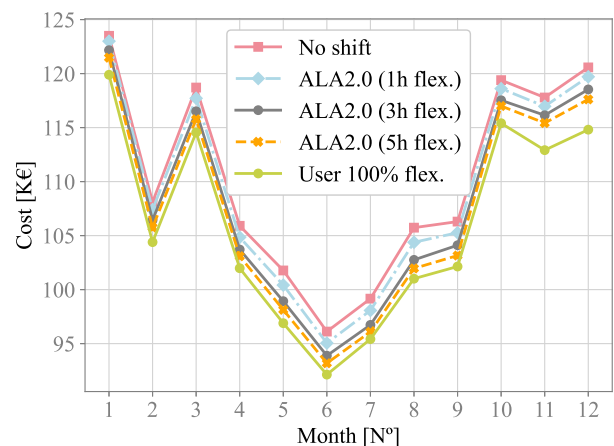


FIGURE 6: Costs in the scenario with PV system and ESS

IV. CONCLUSION

In this paper, we presented a novel framework to simulate and evaluate the impact of DSM on users. The main purpose consists on learning the acceptance of such simulated users in shifting in time the turning-on of their household appliances, without any prior knowledge and without causing too much discomfort. In order to achieve this goal, we proposed ALA 2.0, an improved version of ALA 1.0 presented in [5], which is able to learn the acceptance of the households, leaving a certain degree of control to the users. Consequently, costs were minimised and appliances were optimally scheduled considering the boundaries found thanks to ALA 2.0.

Since it was not possible to test both versions of ALA with real users, they had been modelled starting from what described in surveys in literature, e.g. [14], modelling their acceptance level and the associated discomfort. Moreover, we also included the possibility for the user to make “random refusals” because external factors may influence user habits.

Under these conditions, we demonstrated the potentiality of ALA 2.0 for a single simulated user and for 3000 users in different scenarios comparing results with ALA 1.0, analysing both the user discomfort and savings for the aggreg-

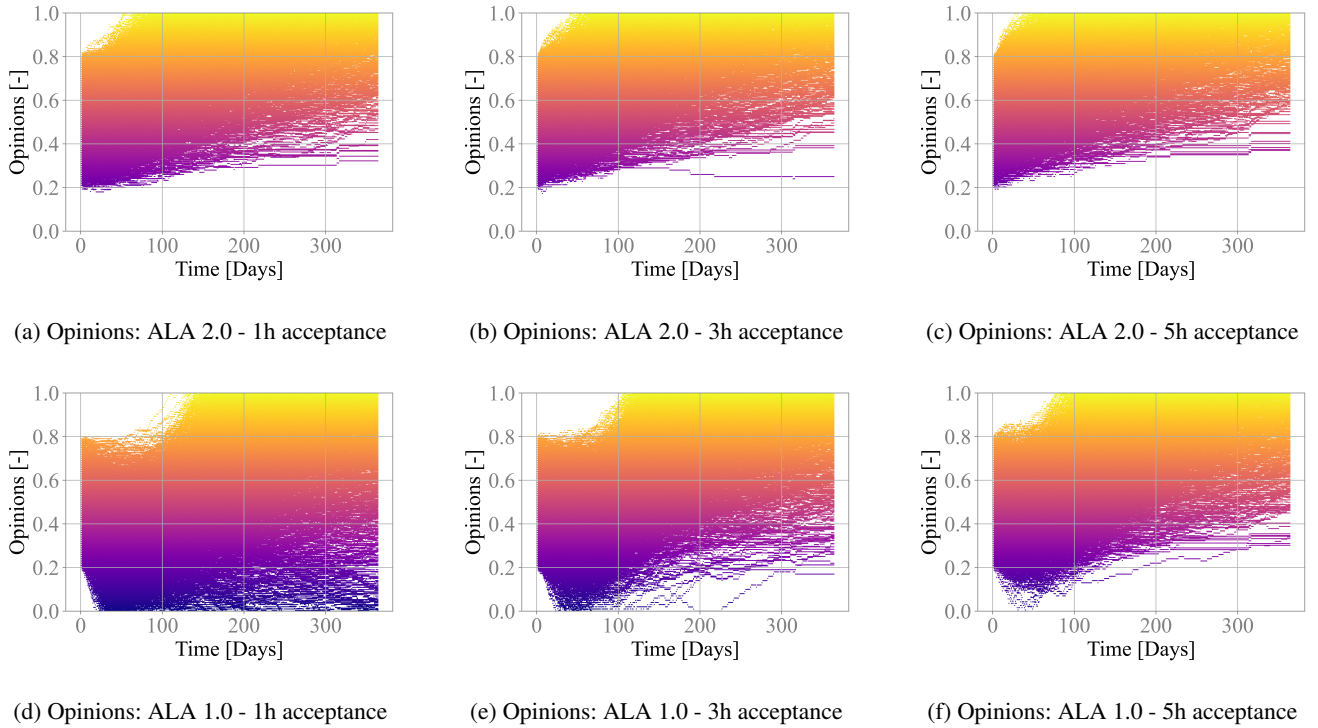


FIGURE 7: Comparison of the opinions in the case studies with the only presence of market: ALA 1.0 vs ALA 2.0.

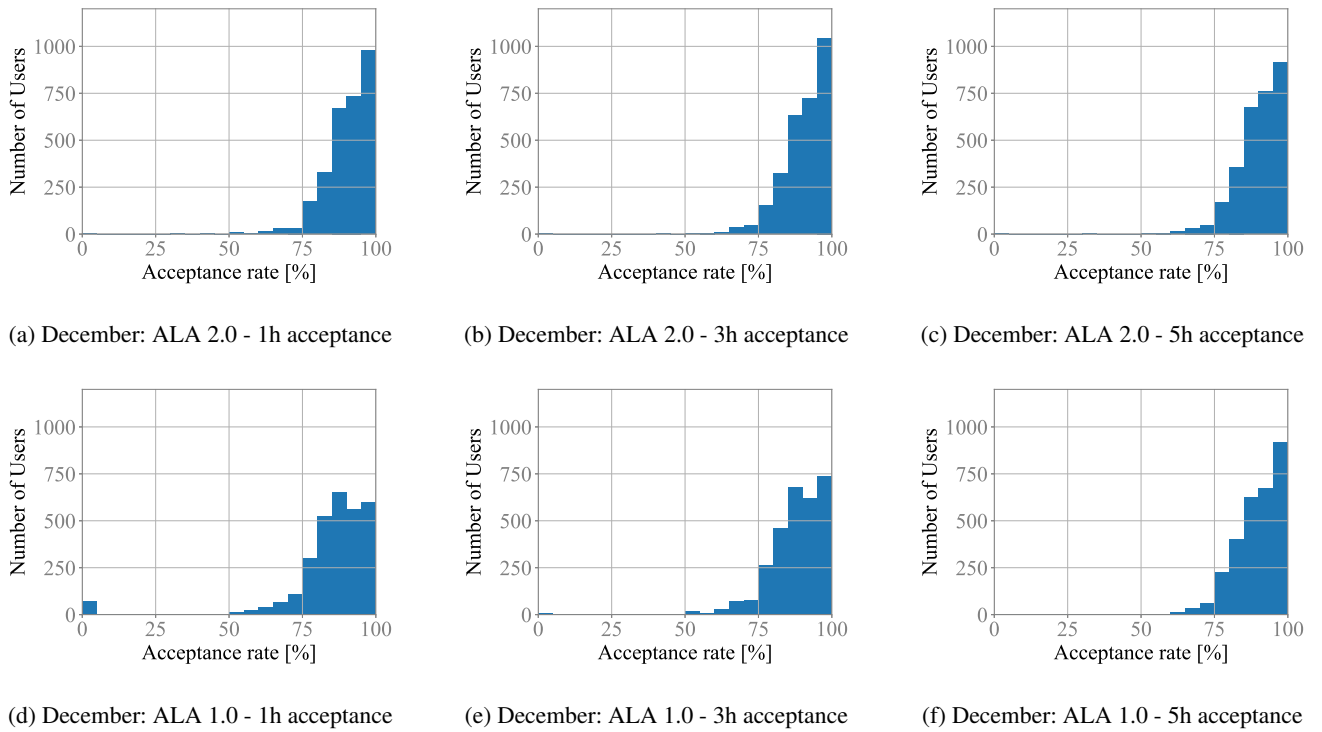


FIGURE 8: Acceptance rate in the case studies with the only presence of market - December: ALA 1.0 vs ALA 2.0.

gator considering one or more energy sources and different level of acceptance. As an example, in the "market only" scenario with 1 hour of acceptance the "Acceptance rate"

in December is 90.36% with ALA 2.0 and 86.20% with ALA 1.0. However, the performances of ALA 2.0 on the user side are definitely improved w.r.t. ALA 1.0 and almost

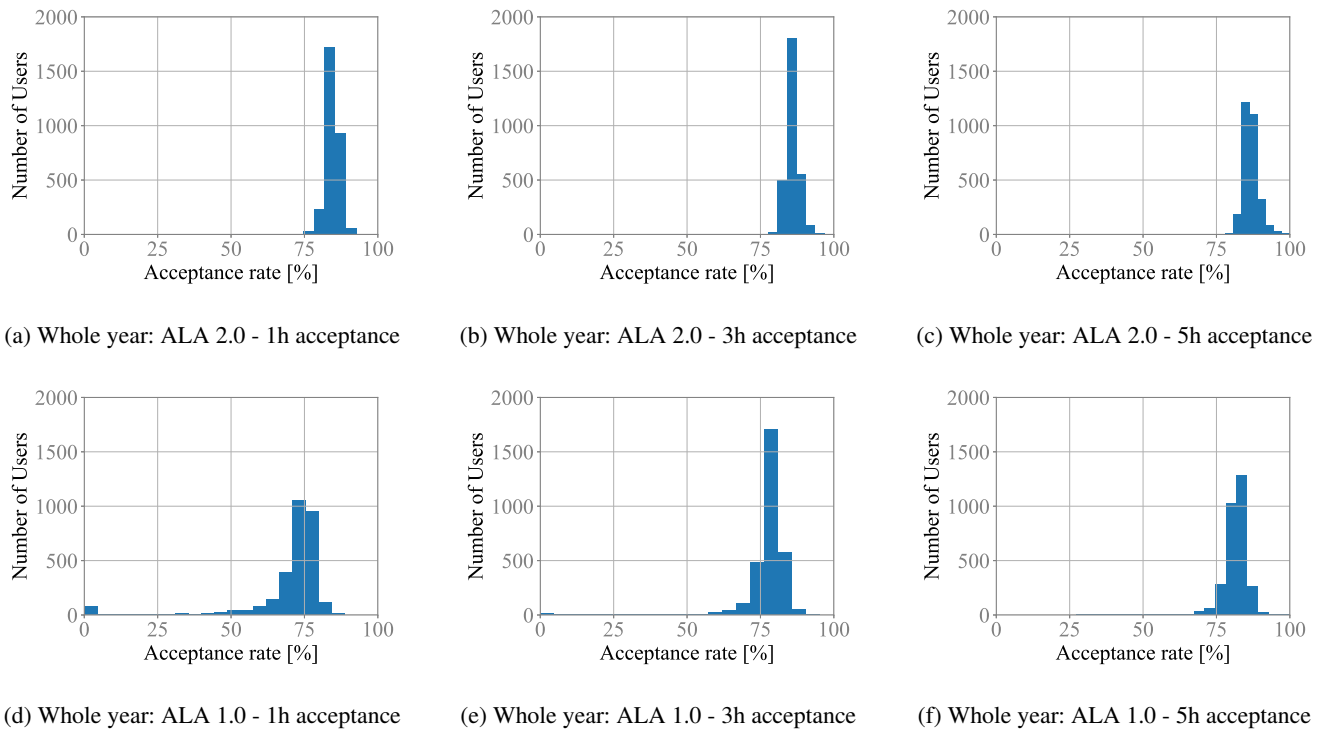


FIGURE 9: Acceptance rate in the case studies with the only presence of market - whole year: ALA 1.0 vs ALA 2.0.

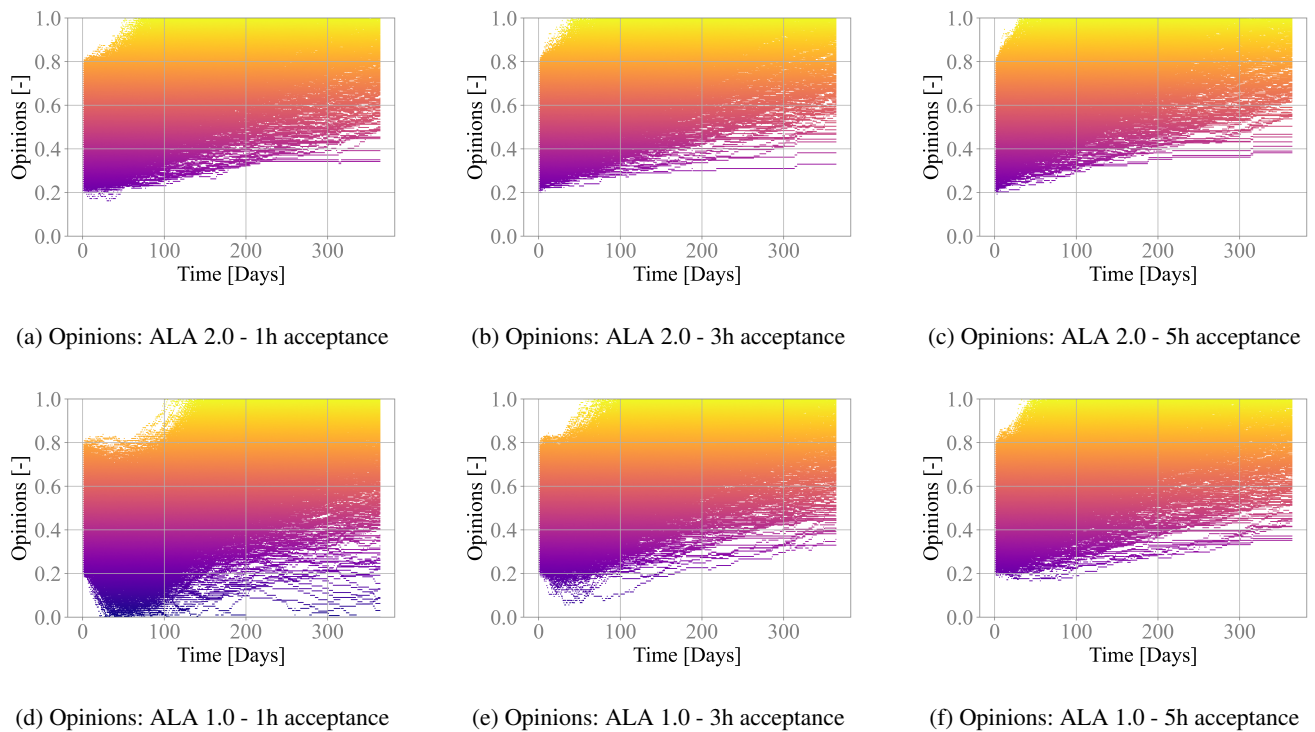


FIGURE 10: Comparison of the opinions in the case studies with market, PV systems and ESS: ALA 1.0 vs ALA 2.0.

all the simulated users have a positive opinion on the DSM program at the end, i.e. several temporal shift proposals better match user preferences. Learning every single preference has

been considered the major advantage for users. Indeed, they are treated equally and it is offered to each individual user the opportunity to participate, even if they rarely use these

appliances.

The amount of savings was strictly related to the chosen parameters and should not be interpreted in terms of absolute values, but as a way to understand when we have major benefits using this type of appliances. As future work, we plan to extend our solution to in-depth evaluate the economical impact of DSM programs. As demonstrated, a win-win situation was always obtained, but the presence of PV systems and ESS enabled to increase considerably savings, exploiting users' flexibility.

In future works, thermal loads will be included in the analysis as well.

REFERENCES

- [1] P. Guo, J. C. Lam, and V. O. Li, "Drivers of domestic electricity users' price responsiveness: A novel machine learning approach," *Applied Energy*, vol. 235, pp. 900–913, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306261918317215>
- [2] W. Mert, J. Suschek-Berger, and W. Trithart, "Consumer acceptance of smart appliances," *Smart domestic appliances in sustainable energy systems (Smart-A)*, 2008.
- [3] S. Yilmaz, X. Xu, D. Cabrera, C. Chanez, P. Cuony, and M. K. Patel, "Analysis of demand-side response preferences regarding electricity tariffs and direct load control: Key findings from a swiss survey," *Energy*, vol. 212, p. 118712, 2020.
- [4] K. Gamma, R. Mai, C. Cometta, and M. Loock, "Engaging customers in demand response programs: The role of reward and punishment in customer adoption in switzerland," *Energy Research and Social Science*, vol. 74, p. 101927, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214629621000207>
- [5] C. De Vizia, E. Patti, E. Macii, and L. Bottaccioli, "A win-win algorithm for aggregated residential energy management: resource optimisation and user acceptance learning," in *2020 IEEE EEEIC and 2020 IEEE I CPS Europe*, 2020, pp. 1–6.
- [6] F. Y. Melhem, O. Grunder, Z. Hammoudan, and N. Moubayed, "Energy management in electrical smart grid environment using robust optimization algorithm," *IEEE Trans. Ind. Appl.*, vol. 54, no. 3, pp. 2714–2726, 2018.
- [7] E. Amicarelli, T. Q. Tran, and S. Bacha, "Optimization algorithm for microgrids day-ahead scheduling and aggregator proposal," in *2017 IEEE EEEIC and 2017 IEEE I CPS Europe*, 2017, pp. 1–6.
- [8] R. Jovanovic, A. Bousseham, and I. S. Bayram, "Residential demand response scheduling with consideration of consumer preferences," *Applied Sciences*, vol. 6, no. 1, p. 16, 2016.
- [9] M. Liu, F. L. Quilumba, and W. Lee, "A collaborative design of aggregated residential appliances and renewable energy for demand response participation," *IEEE Transactions on Industry Applications*, vol. 51, no. 5, pp. 3561–3569, 2015.
- [10] M. Manganelli, G. Greco, and L. Martirano, "Design of a new architecture and simulation model for building automation toward nearly zero energy buildings," *IEEE Trans. Ind. Appl.*, vol. 55, no. 6, pp. 6999–7007, 2019.
- [11] I. R. S. da Silva, J. E. A. de Alencar, and R. de Andrade Lira Rabêlo, "A preference-based multi-objective demand response mechanism," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.
- [12] H. Wei, L. Wenbin, and H. Liangliang, "An optimization model for residential participation in demand response," in *2020 5th Asia Conference on Power and Electrical Engineering (ACPEE)*, 2020, pp. 876–880.
- [13] Z. Wen, D. O'Neill, and H. Maei, "Optimal demand response using device-based reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2312–2324, sep 2015.
- [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] S. M. Hakimi, A. Hasankhani, M. Shafie-khah, and J. P. Catalão, "Demand response method for smart microgrids considering high renewable energies penetration," *Sustainable Energy, Grids and Networks*, vol. 21, p. 100325, 2020.
- [16] Demand response customer insights report. [Accessed: 2019-09-14]. [Online]. Available: <https://arena.gov.au/assets/2018/08/demand-response-consumer-insights-report.pdf>
- [17] Mosaik's documentation. [Accessed: 2019-02-11]. [Online]. Available: <https://mosaik.readthedocs.io/en/latest/>
- [18] S. Scherfke. aiomas documentation. [Accessed: 2019-01-10]. [Online]. Available: <https://aiomas.readthedocs.io/en/latest/>
- [19] 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.
- [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] D. O'Neill, M. Levorato, A. Goldsmith, and U. Mitra, "Residential Demand Response Using Reinforcement Learning," *1st IEEE SmartGrid-Comm10*, pp. 409–414, 2010.
- [22] 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.
- [23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.
- [24] GME. Gestore mercati energetici. [Accessed: 2019-06-05]. [Online]. Available: <https://www.mercatoelettrico.org/It/Default.aspx>
- [25] IRENA, "Renewable Power Generation Costs in 2018," 2019, international Renewable Energy Agency, Abu Dhabi.



CLAUDIA DE VIZIA received the M.Sc. degree in ICT for Smart Societies at Politecnico di Torino in 2019. Since 2020, she joined the Department of Control and Computer Engineering at Politecnico di Torino as a PhD Student. Her research activities focus on the simulation of demand-side management strategies, with particular attention to user behaviour.



EDOARDO PATTI is Assistant Professor at Politecnico di Torino. He received both M.Sc. and Ph.D. degrees in Computer Engineering at Politecnico di Torino in 2010 and 2014, respectively. During the Academic Year 2014/2015, he was Academic Visiting at the University of Manchester. His research interests concern: i) Ubiquitous Computing; ii) Internet of Things; iii) Smart Systems and Cities; iv) Software architectures with particular emphasis on infrastructure for Ambient Intelligence; v) Software solutions for simulating and optimising energy systems; vi) Software solutions for energy data visualisation to increase user awareness. In the fields above, he has authored over 70 scientific publications (between 2011 and 2018).



ENRICO MACII is a Full Professor of Computer Engineering with the Politecnico di Torino, Torino, Italy. He holds a Laurea degree in electrical engineering from the Politecnico di Torino, a Laurea degree in computer science from the Università di Torino, Turin, and a PhD degree in computer engineering from the Politecnico di Torino. His research interests are in the design of electronic digital circuits and systems, with a particular emphasis on low-power consumption aspects. In the last decade, he has extended his research activities to areas such as bioinformatics, energy efficiency in buildings, districts and cities, sustainable urban mobility, clean and intelligent manufacturing. He is a Fellow of the IEEE.



LORENZO BOTTACCIOLI (M'15) is Assistant Professor of Computer Science at Interuniversity Dept. of Regional and Urban Studies and Planning, Politecnico di Torino (Italy). In 2018 received the PhD, cum laude, in Computer Engineering at Politecnico di Torino. His main research interests concern Smart Energy, Smart City and Smart Communities with focus on software solutions i) for planning, analysing and optimising smart energy system and ii) for spatial representation of energy information. He is a Member of the IEEE.

...