A Multi-Armed Bandit Formulation for Distributed Appliances Scheduling in Smart Grids

Antimo Barbato[†], Lin Chen [‡], Fabio Martignon [‡¶] and Stefano Paris [§]
[†] DEIB
Politecnico di Milano
antimo.barbato@polimi.it
[‡] LRI
Paris-Sud University
{lin.chen,fabio.martignon}@lri.fr
[¶] Institut Universitaire de France (IUF)
[§] LIPADE
Paris-Descartes University
stefano.paris@parisdescartes.fr

***Abstract*—Game-theoretic Demand-Side Management (DSM) systems represent a promising solution to control the electrical appliances of residential consumers. Such frameworks allow indeed for the optimal management of loads without any centralized coordination since decisions are taken locally and directly by users.**
In this paper, we focus our analysis on a game-theoretic DSM framework designed to reduce the bill of a group of users. In order to converge to the equilibrium of the game, we adopt an efficient learning algorithm proposed in the literature, Exp3, along with two variants that we propose to speed up convergence. In defining these methods, we model the appliances scheduling problem as a Multi-Armed Bandit (MAB) problem, a classical formulation of decision theory. We analyze the proposed learning methods based on realistic instances in several use-case scenarios and show numerically their effectiveness in improving the performance of next generation smart grid systems.

I. Introduction

The recent evolution of power grids, with the integration of distributed generation, decentralized storage systems and communication infrastructures, is radically changing their operation and architecture. In this new scenario, consumers are playing more and more an active role, particularly in stabilizing the power grid by dynamically adjusting their demand with respect to grid and market conditions. To this end, retailers are introducing novel tariffs, such as Time-Of-Use (TOU) tariffs and Real-Time Pricing (RTP), in which prices change in response to variations in supply and demand in order to incentivize consumers to adopt more sustainable and efficient power usage habits.

Such new policies call for automatic Demand-Side Management (DSM) mechanisms which can optimally control users' power loads by modulating the power absorption of elastic appliances (e.g., heating and air conditioning systems) and scheduling the execution of shiftable devices (e.g., washing machine and dryer). In the literature, several centralized DSM mechanisms have recently been proposed [1]. All these solutions require some sort of coordination system and proper protocols to exchange all energy data and identify the best scheduling solution, which is then forwarded to all customers. However, such frameworks may not be applied in real scenarios, mainly because of the security and privacy threats arising from data gathering and exchange procedures.

DSM systems based on *decision theory* represent the solution to these problems since they allow customers (or even single appliances taking autonomous decisions, as we envision in our paper) to converge to desired operating points of the system in a *fully distributed* manner. For this reason, in this paper, we model the DSM scheduling problem as a Multi-Armed Bandit (MAB) problem, a classical and very effective formulation adopted in decision theory. The MAB problem naturally arises in contexts where agents (appliances, in our case) simultaneously attempt to acquire new knowledge about the system (the *exploration* phase) and optimize their decisions based on the acquired knowledge (the *exploitation* phase).

The approach proposed in this work is specially tailored for distributed DSM solutions designed to schedule the electric devices of residential consumers on a daily basis. In particular, in our framework, each appliance decides *autonomously* its schedule for the next day, with the objective of minimizing the expected regret with respect to the cost-optimal strategy. Appliances strategies are evaluated in terms of their corresponding energy bills that are determined based on a dynamic tariff in which prices are a function of the overall power demand of devices.

In order to identify the appliances schedule, we consider and evaluate different algorithms to find an optimal balance between the exploration and exploitation efforts. Specifically, we adopt an efficient learning algorithm proposed in the literature (named Exponential-weight algorithm for Exploration and Exploitation, Exp3 [2]), along with two variants that we design specifically for our scheduling scenario, named Exp3a and Exp3b, respectively, which are tailored to speed up the convergence to system-wide efficient equilibria. In the first variant (Exp3a), Krasnoselskij iterations are used [3], that is only a subset of appliances at each iteration updates their choice. In the second variant (Exp3b), we assume that additional information can be easily made available to appliances: after every iteration (i.e., on a daily basis) the retailer broadcasts to all players the electricity tariff applied that day, defined based on the aggregated power demand of users; such information is exploited to further speed up convergence.

We analyze the proposed algorithms based on realistic instances of the DSM game and show their effectiveness in converging to very efficient operating points in several use-case scenarios.

In summary, our paper makes the following key contributions:

- We propose a novel approach, based on the multi-armed bandit problem, to select efficient appliances scheduling

patterns in a completely distributed manner. A state-of-the-art algorithm (Exp3) is adopted, along with two variations we specifically design to speed-up convergence towards very efficient equilibria in a smart grid scenario.
- We analyze the convergence properties of the proposed learning algorithms in terms of achieved equilibrium as well as the number of iterations to reach this latter.
- We perform a thorough numerical analysis of our proposed algorithms in several realistic use-case scenarios, which demonstrate that our proposed approach is promising and very effective to improve the performance of next generation smart grid systems.

The paper is structured as follows: Section II discusses related work. Section III provides an overview of the game theoretic demand management framework that we have adopted in this work. Section IV describes the MAB-based learning algorithms that we have designed to converge to system-wide efficient equilibria. Performance assessment is illustrated and discussed in Section V. Finally, Section VI concludes this paper.

II. Related Work

Demand side management methods have been recently proposed in smart grids to properly control and schedule users' loads [4]. In particular, DSM schemes can be applied to shift the users' demand from peak to off-peak periods, therefore reducing the need for generation, transmission and distribution capacity, as well as power grids investments. At the same time, DSM can also address issues related to electric grids such as the integration of Renewable Energy Sources (RESs) which are intermittent and uncontrollable by nature, hence raising brand new problems in the demand-supply balancing process. These issues can be mitigated by DSM systems by means of properly scheduling the users' loads based on the availability of renewable energy generation [5].

In the field of DSM systems, *distributed* methods have gained increased attention. Consistent improvements of the grid efficiency can indeed be obtained only by coherently managing the energy resources of groups of users whose differences and randomness, in terms of electricity consumption needs, can be exploited to adapt the overall load demand to the grid requirements. To this end, several centralized frameworks have been proposed in the literature, aiming at controlling the electric loads of groups of collaborative customers [6], [7]. However, these solutions require a centralized controller to gather users' information and optimize their energy plans. To this end, a large volume of data must be collected and transmitted through the smart grid network, thus introducing scalability constraints, as well as novel threats to the customers' security and privacy [8]. For these reasons, distributed DSM methods have been proposed in which decisions are taken *locally* by users. In such context, game theory represents the ideal framework to design distributed DSM solutions since it permits to model and study the interactions among the independent rational players of the power grid [9]. In this case, the users' load scheduling problem is formulated as a game, where the players are the consumers and their strategies are the schedules of their electric appliances. The goal of the game is to reduce either the peak of the total demand, the overall energy costs, or the users' electricity bills [10].

DSM methods based on game theoretical frameworks have been designed to provide equilibria that improve the efficiency of the power grid from a system-wide perspective. However, converging to the game equilibria is non-trivial, and *learning algorithms* are required to enable players to reach the desired outcome [11]. Learning methods are iterative processes in which players, in turn, estimate the utility associated with their strategies based on their knowledge of the game state, and decide which strategy to play in the current iteration depending on the decision logic of the algorithm. Several learning algorithms have been proposed in the literature which differ in the learning style and in the assumptions on the interaction among players. *Regret Matching* methods [12], for example, are characterized by players who attempt to minimize their regret from using a certain strategy. These methods rely on the assumption that each player can estimate both its own utility and the utility he would have obtained by playing all other actions. On the other hand, in *Reinforcement Learning* methods [13], players attempt to maximize their utility rather than considering the regret associated with their actions. Specifically, at each iteration of the algorithms, actions leading to higher utility are associated with higher probabilities to be chosen in the next stage. *Regret Matching* and *Reinforcement Learning* methods, as well as several other algorithms, have been extensively studied in several research fields, including robotics [14] and telecommunications [15], [16]. For this reason, some of the solutions proposed in these fields can be applied to game theoretic DSM frameworks. However, security and privacy concerns could raise when applying these methods to real implementations of demand management solutions. Learning algorithms proposed in [10] and [17], for example, require each player to broadcast his appliances schedule to either the energy service provider or to other users, therefore introducing serious privacy issues [8], [18].

The algorithms proposed in this paper are related to multi-armed bandit problems which are of fundamental importance in stochastic decision theory due to their application in numerous engineering problems, such as wireless channel access, communication jamming, object tracking, and smart grids. Such problems are the most basic examples of sequential decision problems with an exploration-exploitation trade-off: this is the balance between staying with the option that gave highest payoffs in the past and exploring new options that might give higher payoffs in the future [2], [19].

A survey on multi-armed bandit problems is provided in [2], with a focus on two extreme cases in which the analysis of regret is simple and elegant: i.i.d. payoffs and adversarial payoffs. The work in [20] adopts hidden Markov models in the context of smart grids to capture the dynamics of renewable energy resources, and formulates the stochastic scheduling problem as a partially observable Markov decision process multi-armed bandit problem. In order to solve the problem, a

value iteration algorithm is used.

Differently from existing works, our paper proposes a novel MAB approach for scheduling appliances in a cost-efficient, fully-distributed way. Numerical analysis demonstrate the effectiveness of our approach both in terms of achieved results and convergence to efficient equilibria.

III. Distributed DSM: Problem Formulation and Game Model

In this paper, we consider a fully distributed demand-side management framework based on a non-cooperative game theoretical approach [21]. This framework is designed to efficiently schedule the electric appliances of a group of residential consumers, $\mathcal{H}$, over a 24-hour time period divided into a set, $\mathcal{T}$, of time slots. Each consumer $h \in \mathcal{H}$ has to schedule a set of non-interruptible appliances, $\mathcal{A}_h$, which must be executed only once during the day $\mathcal{T}$. Each appliance $a \in \mathcal{A}$, where $\mathcal{A}$ denotes the set of all appliances (i.e., $\mathcal{A} = \cup_{h\in\mathcal{H}}\mathcal{A}_h$), is characterized by a fixed load profile, l_{af}, having a duration of F_a time slots. Specifically, l_{af} represents the power consumption of a in the f-th time slot of its load profile and $f \in \mathcal{F}_a = \{1, 2.., F_a\}$. Moreover, each appliance $a \in \mathcal{A}$ can only be executed within a time window delimited by a minimum starting-time slot, ST_a, and a maximum ending-time slot, ET_a.

A real-time pricing is used to define the price of electricity at time $t \in \mathcal{T}$, c_t. Specifically, c_t is modelled as an increasing function of the total power demand, y_t, of the group of users $\mathcal{H}$ at time t:

$$c_t = c^{Anc} + c^{En} \cdot y_t \qquad \forall t \in \mathcal{T} \tag{1}$$

where c^{Anc} is the cost of ancillary services (e.g., electricity transport, distribution and dispatching, frequency regulation, power balance) and c^{En} is the slope of the cost function.

The objective of each consumer $h \in \mathcal{H}$ is to optimally schedule his appliances in order to minimize his daily bill, U_h, defined as follows:

$$U_h = \sum_{t\in\mathcal{T}} y_{ht} \cdot c_t \tag{2}$$

where y_{ht} is the power demand of consumer h at time t.

In [21], we show that if each appliance decides autonomously its scheduling in a fully distributed fashion (*single-appliance DSM*) with the goal of minimizing its bill, only a negligible increase of the consumers' bill is found with respect to the case in which each consumer schedules the whole set of his appliances (*multiple-appliance DSM*). For this reason, in this paper, we will use the *single-appliance DSM* model since it requires a less complex architecture without home servers that collect all devices information and play on behalf of the consumers.

Note that since appliances are modelled as a non-interruptible activities with fixed load profiles, defining their schedules is equivalent to deciding their start-times. For this reason, in this paper, we use the terms schedule and start-time interchangeably.

A. Distributed Single-Appliance DSM: Game Model

The appliance scheduling problem is modelled as a game $G = \{\mathcal{A}, \mathcal{I}, \mathcal{U}\}$: $\mathcal{A}$ is the set of players (i.e., appliances), $\mathcal{I} \triangleq \{\mathcal{I}_a\}_{a\in\mathcal{A}}$ is the set of strategies which correspond to the appliances schedules and $\mathcal{U} \triangleq \{U_a\}_{a\in\mathcal{A}}$ is the set of utility functions that coincide with the devices electricity bills. Specifically, the strategy of player a is $\mathcal{I}_a \triangleq \{x_{at}\}_{a\in\mathcal{A}}$, where x_{at} are binary variables defined for each device $a \in \mathcal{A}$ and for each time slot $t \in \mathcal{T}$. These variables are equal to 1 if the appliance a starts at time t and 0 otherwise. As a consequence, defining these variables is equivalent to deciding the start-time of the appliance. The possible schedules that form the strategy space $\mathcal{I}_a$ of each player a (of consumer h) must satisfy the following set of constraints:

$$\mathcal{I}_a = \Bigg\{ \vec{x}_a = \left[x_{a1}...x_{at}...x_{a|\mathcal{T}|}\right] \in \{0,1\}^{|\mathcal{T}|} :$$

$$\sum_{t=ST_a}^{ET_a-F_a+1} x_{at} = 1 \tag{3}$$

$$y_{at} = \sum_{f\in\mathcal{F}_a : f\le t} l_{af} x_{a(t-f+1)} \quad \forall a \in \mathcal{A}_h, t \in \mathcal{T} \tag{4}$$

$$\sum_{a\in\mathcal{A}_h} y_{at} \le \pi^{SL} \qquad \forall t \in \mathcal{T} \Bigg\}. \tag{5}$$

Constraints (3) guarantee that appliance a is executed only once within the interval $[ST_a, ET_a]$. Constraints (4) determine the daily consumption profiles of all the appliances of the consumer h, which depend on their schedules (i.e., $\{x_{at}\}_{a\in\mathcal{A}_h}$). Finally, constraints (5) limit the overall power consumption of consumer h, since in every time slot $t \in \mathcal{T}$ the electricity bought from the grid cannot exceed the Supply Limit (SL) defined by the retailer and denoted by π^{SL}.

Each appliance a chooses its strategy $\mathcal{I}_a$ to minimize its cost U_a. The utility function of each player, U_a, which is a function of $\mathcal{I}$, is defined as follows:

$$U_a(\mathcal{I}) = \sum_{t\in\mathcal{T}} y_{at} \cdot c_t \tag{6}$$

where y_{at}, which represents the amount of electricity demand of appliance a at time t, is a function of x_{at}.

The solution of the distributed single-appliance game is characterized by a Nash Equilibrium (NE) which is a strategy profile $\mathcal{I}^* = (\mathcal{I}_a^*, \mathcal{I}_{-a}^*)$ from which no player has an incentive to deviate unilaterally. One can prove that this game is a potential game if c_t is convex with respect to y_t. Potential games have several nice properties, such as the existence of at least one pure Nash equilibrium. Furthermore, such games have the Finite Improvement Property: any sequence of asynchronous improvement steps is finite and converges to a pure equilibrium.

IV. Distributed Learning Algorithms

In order to converge to the equilibrium of the DSM game presented in Section III, we propose three efficient learning

algorithms that enable players to reach the desired game outcome in a distributed fashion. These algorithms are executed in parallel by players, and therefore do not require any communication among them.

In the following, we first formalize the scheduling learning problem, then we describe the 3 algorithms that we have implemented to solve it.

A. Model and Problem Formulation

In the load schedule learning problem, all players, which are represented by the set of appliances $\mathcal{A}$, have to decide their schedule autonomously. Specifically, each player $a \in \mathcal{A}$ can choose any start-time s_a within its set of feasible schedules, $\mathcal{S}_a$:

$$\mathcal{S}_a = \{t \in \mathcal{T} : t \in [ST_a; ET_a - F_a + 1]\} \tag{7}$$

which represents the subset of time slots of $\mathcal{T}$ that satisfy the scheduling constraints (3).

The selection process of the devices start-time, which can be performed based on several logics as described below, has to be repeated every day within the time horizon $\mathcal{K}$. Hereafter, we denote with s_a^k the schedule chosen within set $\mathcal{S}_a$ by player $a \in \mathcal{A}$ on day $k \in \mathcal{K}$.

At the end of each day k, each player a receives a bill U_a^k from the retailer, computed as follows, based on equation (6):

$$U_a^k = \sum_{t \in \mathcal{T}} y_{at}^k \cdot c_t^k \tag{8}$$

where c_t^k and y_{at}^k are, respectively, the cost of electric energy and the power demand of each player on day k. Both c_t^k and y_{at}^k can be easily computed based on the sequence $\{s_a^k\}_{a \in \mathcal{A}}$ which is known to the retailer. Specifically,

$$y_{at}^k = \sum_{f \in \mathcal{F}_a : f \leq t} l_{af} x_{a(t-f+1)}^k \quad \forall a \in \mathcal{A}, t \in \mathcal{T} \tag{9}$$

$$c_t^k = c^{Anc} + c^{En} \sum_{a \in \mathcal{A}} y_{at}^k \quad \forall t \in \mathcal{T} \tag{10}$$

where $x_{at} = 1$ if $t = s_a^k$, 0 otherwise. It is worth noting, from equations (8) and (10), that even if every player runs the learning algorithms independently of others, each player's action affects the other ones since it modifies the electricity prices.

In choosing the devices start-time, the player's objective is to minimize its total bill, i.e. the sum of the bills received over the time horizon $\mathcal{K}$. As bills differ from schedule to schedule, the goal is to find the schedule with the lowest expected bill as early as possible, and then to keep using it on future days. To this end, the learning algorithms have to be properly designed to achieve the best trade-off between the exploration of the solutions space and the exploitation of the statistics gathered in past iterations.

In the following, we first propose the utilization of an efficient learning algorithm proposed in the literature (named Exponential-weight algorithm for Exploration and Exploitation, Exp3 [2]), and we further describe two variants we have designed specifically for our scheduling scenario, named Exp3a and Exp3b, in order to speed up the convergence to system-wide efficient equilibria. The first variant (Exp3a), makes use of Krasnoselskij iterations [3], which means that only a subset of players at each iteration updates their choice. In the second variant (Exp3b), we suppose that additional information is available to users: after every iteration, the retailer broadcasts to all players the electricity tariff applied that day, defined based on the aggregated power demand of users.

B. Exp3: Exponential-weight algorithm for Exploration and Exploitation

The scheduling learning problem of each player a can be modeled as a multi-armed bandit problem in which each possible schedule $s_a \in \mathcal{S}_a$ coincides with an arm and the reward received by the player at round k by picking a given arm corresponds to the opposite of the bill (i.e., $-U_a^k$). As a consequence, minimizing the total bill over the time horizon is equivalent to maximizing the total reward.

Since the load scheduling problem of each player can be represented as a multi-armed bandit problem, some of the solutions proposed in the literature for MAB frameworks can be efficiently applied to the DSM game. Specifically, in this work, we consider the algorithm Exp3. This method, whose pseudo code is shown in Figure 1, is a randomized algorithm in which, on each day k, the schedule of the player a is selected according to the probability distribution $p^k(s_a)$, with $s_a \in \mathcal{S}_a$, which represents the probability of choosing the schedule s_a at iteration k. In the definition of $p^k(s_a)$, γ is an exploration parameter and $w^k(s_a)$, with $s_a \in \mathcal{S}_a$, are weights that depend exponentially on the bills received in the past. This distribution, which is a mixture of uniform and exponential distributions, is designed to efficiently balance the exploration and exploitation phases of the algorithm.

After drawing a schedule s_a^k based on the distribution $p^k(\cdot)$, the player a receives a bill U_a^k and updates the weights for the next day, $w^{k+1}(\cdot)$. Note that at iteration 1 of the algorithm, $w^1(\cdot)$ are all set to one and $p^k(\cdot)$ is a uniform distribution over $\mathcal{S}_a$ since no information on the game state is available.

C. Exp3a: Exponential-weight algorithm for Exploration and Exploitation with Krasnoselskij iteration

In our work, we have observed via numerical simulations that by applying the algorithm Exp3, players may not reach a stable equilibrium state. As a consequence, players may keep switching between different schedules, in an almost cyclic manner. In order to address this issue, we propose a variant of the Exp3 algorithm, called Exp3a, based on the Krasnoselskij iteration: on each day, only a fraction $\lambda \in [0, 1]$ of players, randomly selected within set $\mathcal{A}$, are allowed to change their schedule with respect to the previous iteration.

The pseudo code of Exp3a is the same as the algorithm Exp3, except for line 8. In fact, in this case, the player selects

1: **procedure EXP3**
2: $\gamma \in (0,1]$
3: $w^1(s_a) = 1 \ \forall s_a \in \mathcal{S}_a$
4: **for** $k = 1, 2, ...$ **do**
5: **for all** $s_a \in \mathcal{S}_a$ **do**
6: $p^k(s_a) = (1-\gamma)\frac{w^k(s_a)}{\sum_{s_i \in \mathcal{S}_a} w^k(s_i)} + \frac{\gamma}{|\mathcal{S}_a|}$
7: **end for**
8: Choose s_a^k randomly accordingly to $p^k(\cdot)$
9: **for all** $s_a \in \mathcal{S}_a$ **do**
10:
$$\hat{w}^k(s_a) = \begin{cases} -U_a^k / p^k(s_a) & \text{if } s_a = s_a^k \\ 0 & \text{otherwise} \end{cases}$$
11: $w^{k+1}(s_a) = w^k(s_a) exp(\frac{\gamma \hat{w}^k(s_a)}{|\mathcal{S}_a|})$
12: **end for**
13: **end for**
14: **end procedure**

Fig. 1. Pseudo-code of algorithm Exp3

a new schedule with probability λ, otherwise it keeps using the schedule chosen on the previous day.

D. Exp3b: Exponential-weight algorithm for Exploration and Exploitation with estimated bills

In the Exp3 algorithm, at every iteration k, the distribution probability $p^k(\cdot)$ is updated based only on the actual bill received by player a due to its selection of schedule s_a^k, without considering any other information. Even if this feature of the algorithm makes it simple and easy to apply in real use-case scenarios, it may lead to low convergence rates. In fact, since no information on the other possible schedules is used in updating probability values $p^k(\cdot)$, a longer exploration phase may be required to gather statistics on all possible solutions. In order to address this issue and speed up the convergence process, we propose a variant of the Exp3 method, called Exp3b, in which probabilities $p^k(\cdot)$ of player a on day k are updated based on information on all the possible schedules $s_a \in \mathcal{S}_a$. Specifically, this algorithm relies on the assumption that after every iteration k, it is possible to estimate the bills that every player a would have paid if it had executed any other schedule within the feasible set $\mathcal{S}_a$. To this end, we suppose that at the end of each iteration, the retailer broadcasts to players the electricity tariff applied that day, c_t^k, defined based on the aggregated power demand of users.

The pseudo code of Exp3b is the same as the algorithm Exp3, except for line 10, which is rewritten as follows:

$$\hat{w}^k(s_a) = \begin{cases} -U_a^k / p^k(s_a) & \text{if } s_a = s_a^k \\ -\hat{U}_a^k / p^k(s_a) & \text{otherwise} \end{cases} \quad (11)$$

where U_a^k is the actual bill received by player a on day k by selecting the schedule s_a^k and $\hat{U}_a^k$, which is computed based on the electricity prices c_t^k broadcast by the retailer, is the bill that player a would have paid by choosing a different schedule $s_a \in \mathcal{S}_a \setminus \{s_a^k\}$. Note that $\hat{U}_a^k$ is only an estimate of the bill that player a would have paid, since by selecting a schedule different from s_a^k, the actual tariff used on day k would have been slightly different from the one broadcast by the retailer, as it is clear from the electricity price definition of equation (10).

V. NUMERICAL RESULTS

This section presents the numerical results we have obtained by testing the multi-armed bandit algorithms proposed in this paper on realistic instances of the DSM scheduling problem [22], [23]. Specifically, we first describe the experimental methodology of our tests, then we illustrate and discuss the performance achieved by the proposed algorithms.

A. Tests Methodology

In our tests, we evaluate the MAB learning algorithms over a period of 10000 days (i.e., iterations), each one represented by a set $\mathcal{T}$ of 24 one-hour time slots. In order to assess the performance of the proposed methods as the number of users increases, we vary the size of the set of consumers $\mathcal{H}$ in the range $[10, 50]$. Each of these users is connected to the grid with a power demand limit, π^{SL}, of 3 kW and has 4 shiftable electric appliances out of 11 realistically-modeled devices[1].

In order to investigate the effect of the scheduling flexibility (i.e., the size of the $[ST_a, ET_a]$ time-window) on the performance of the learning algorithms, we consider three different cases: *No Flexibility*, in which the devices scheduling is fixed and cannot be modified, *Low Flexibility* and *High Flexibility* in which, respectively, 3 and 8 different possible schedules can be selected for each device. Specifically, for each of these cases, the starting-time slot of the appliances, ST_a, is randomly selected for each consumer within the set $\mathcal{T}$ to represent a population of heterogeneous users. On the other hand, the ending-time slot, ET_a, is defined based on the value chosen for ST_a in order to guarantee the number of different possible schedules associated with the corresponding flexibility level (i.e., $ET_a = ST_a + F_a - 1$ without flexibility, $ET_a = ST_a + F_a + 1$ with low flexibility and $ET_a = ST_a + F_a + 6$ with high flexibility).

Finally, regarding the electricity tariff, we define it based on the real-time pricing currently used in Italy for large consumers. Specifically, we fix the cost of ancillary services $c^{Anc} = 50 \times 10^{-6}$ \$ and the slope of the pricing function $c^{En} = (0,11 \times 10^{-6})/|\mathcal{H}|$ $\$/kWh$.

To evaluate the performance of the proposed MAB methods, we compare the Nash Equilibrium of the DSM game with the outcome obtained by applying the distributed learning algorithms, in terms of:

- *Aggregated utility*: the overall electricity bill of the group of users, $\mathcal{H}$.
- *Peak demand*: the peak of the aggregated power demand of the group of users, $\mathcal{H}$.

[1]Namely, *shiftable* devices: washing machine, dishwasher, boiler, vacuum cleaner; *fixed* devices: refrigerator, purifier, lights, microwave oven, oven, TV, iron

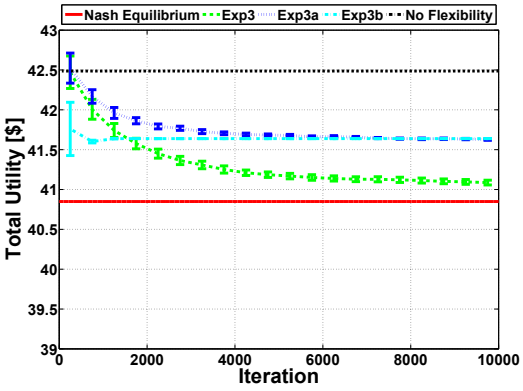
- *Fairness*: we measure the fairness of the DSM outcome in terms of sharing of the energy bill among users, based on the *Jain's Fairness Index (JFI)* [24].

Moreover, we further investigate the convergence time of each learning scheme to the Nash equilibrium and its dependence on the number of consumers taking part in the DSM game.

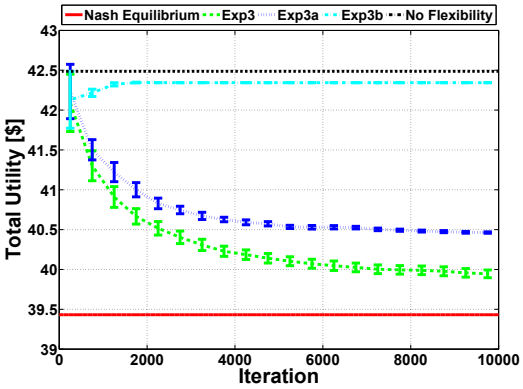
Note that, for each case defined in our tests, we generate 5 different instances. In Subsection V-B, we only report the average results obtained for each test scenario.

B. Performance Evaluation

Figures 2, 3, and 4 show the numerical results obtained in our tests as a function of the number of iterations of the learning algorithms, with a population of 50 consumers. Specifically, in these figures, we divide the time horizon into periods of 500 iterations, for each of which we report the mean value and the 95% confidence interval of the observed results. Moreover, in order to investigate the convergence of the proposed MAB methods, we also represent the Nash equilibrium of the DSM game and the performance of the appliance scheduling game without scheduling flexibility, i.e., when the usage of electric devices is fixed and cannot be modified by the DSM system.



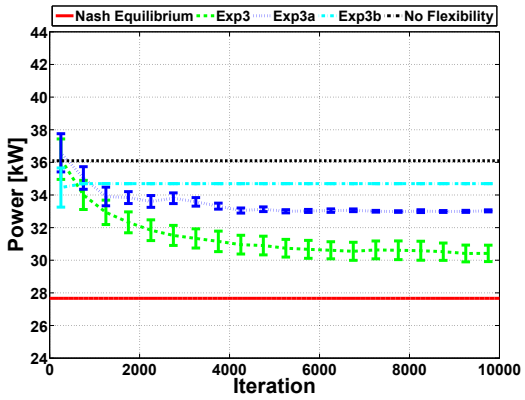
(a) Low Flexibility



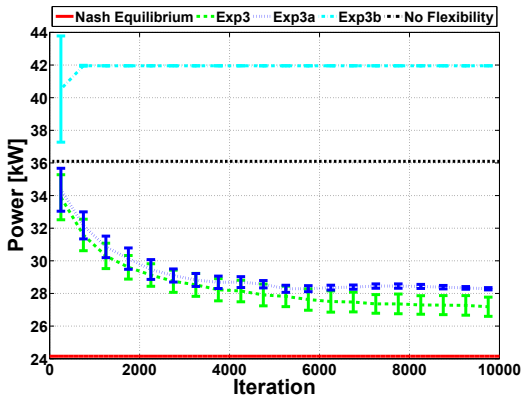
(b) High Flexibility

Fig. 2. Aggregated utility with 50 users.

Numerical results show that the basic MAB Exp3 algorithm has the best performance in the long-run since it yields to outcomes which are closer to the Nash Equilibrium than those obtained with the other methods. However, this result comes at the cost of a worse stability and rate of convergence. Indeed, in the Exp3 case, players do not reach a stable equilibrium state, as it can be noted by analyzing the confidential intervals. This highlights the inability of this algorithm to anticipate the players "behavior" in the next iteration, which causes an extended exploration phase in which devices keep switching between different schedules, in an almost cyclic manner. As for the low convergence speed, it is explained by the fact that, at each iteration, the Exp3 algorithm updates its schedules distribution probabilities $p^k(\cdot)$ based only on the actual bill paid by playing the chosen schedule, without estimating the potential bills associated with the other schedules. As a consequence, a prolonged exploration stage is required to evaluate all the scheduling combinations.



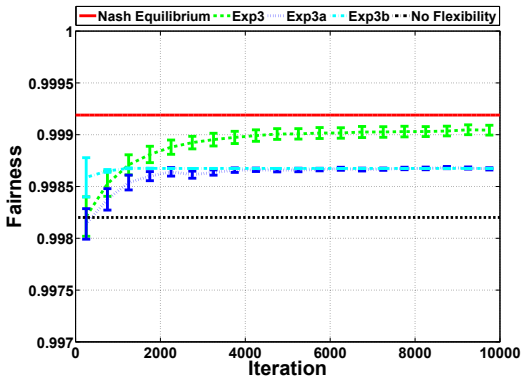
(a) Low Flexibility



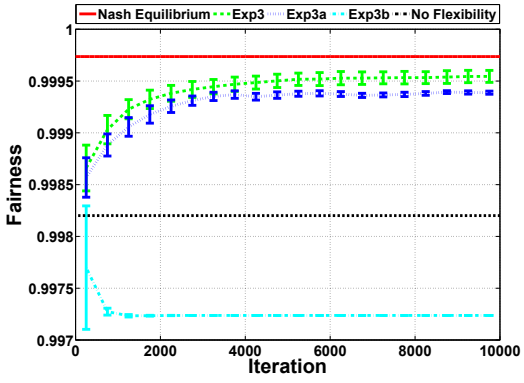
(b) High Flexibility

Fig. 3. Aggregated peak power demand with 50 users.

As illustrated in Figures 2- 4, the stability and convergence rate of the learning process can be improved with the proposed variants of the Exp3 algorithm. Specifically, in the Exp3a case, only a few devices are actually allowed to change their strategies with respect to the previous iteration, thus reducing the game dynamics and improving the stability of the game outcome. Moreover, in the case of the Exp3b learning method, the exploration phase of the learning process is shortened



(a) Low Flexibility



(b) High Flexibility

Fig. 4. Fairness of the DSM solution with 50 users.

by updating the schedules distribution probabilities based on the bills associated with all the potential schedules, thus determining a lower convergence time.

It can be further observed that the system flexibility in scheduling the electric devices notably influences the convergence speed of the MAB algorithms. More specifically, the learning algorithms have higher convergence rates with a short flexibility level. In fact, in this case, the solution space of the problem is smaller since each player has a lower number of strategies to try, therefore reducing the exploration phase of the learning process. However, as expected, shorter convergence times come at the cost of higher bills and peaks of energy demand, as well as lower fairness. Nevertheless, even with a low flexibility level, the MAB algorithms allow players to achieve better results than those observed with fixed schedules (i.e., without the DSM system), except for the Exp3b algorithm which has a too short exploration phase.

In addition to the schedule flexibility, the convergence speed of the proposed MAB learning algorithms also depends on the number of shiftable devices which take part in the loads scheduling game. Indeed, as illustrated in Table I for the Exp3 method, the greater the number of appliances (i.e., users), the faster the convergence. These results affirm the applicability of such learning algorithms to real use-case scenarios, where thousands of users would participate in the DSM game with the expectation of reducing their bills even in the short-term.

TABLE I
CONVERGENCE TIME (MEASURED AS NUMBER OF ITERATIONS) WITH EXP3 ALGORITHM AND 1% AND 5% NASH EQUILIBRIUM GAPS

		NE Gap 5 %		**NE Gap 1 %**	
	Flexibility	Low	High	Low	High
Users	10	51	622	3491	-
	20	3	588	3362	-
	30	1	447	3217	-
	40	1	450	2673	9403
	50	1	447	2648	8897

VI. CONCLUSIONS

Demand-side management systems are currently considered a very effective solution to control users' power loads in the smart grid. In this paper, we proposed several solutions specially tailored for a DSM framework used to schedule the electric devices of residential consumers on a daily basis in a distributed fashion, with the goal of minimizing their bills. In our vision, each appliance decides autonomously (and independently of other appliances) its schedule for the next iteration based on multi-armed bandit algorithms, finding an optimal balance between the exploration and exploitation efforts. In particular, at each iteration, each player chooses an action (i.e., a feasible appliance schedule) which minimizes the expected regret with respect to the cost-optimal strategy.

We measured the performance of our proposed methods and scheduling algorithms using realistic instances of the DSM load scheduling game, showing their effectiveness in converging to very efficient operating points characterized by low tariffs. Moreover, our proposed methods do not require any communication among players, therefore addressing both security and privacy concerns that affect other solutions proposed in the literature. For this reason, our proposed schemes represent a promising and very efficient solution to implement DSM systems in next generation smart grid infrastructures. In such context, this work proposed to model the problem of appliances scheduling as a multi-armed bandit problem.

VII. ACKNOWLEDGMENTS

This work has been partially funded by the Italian MIUR project SHELL "Ecosistemi domestici condivisi e interoperabili per ambienti di vita sostenibili, confortevoli e sicuri" and Regione Lombardia project SCUOLA "Smart Campus as Urban Open LAbs".

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] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Foundations and Trends in Machine Learning*, vol. 5, no. 1, pp. 1–122, 2012.

[3] V. Berinde, "Iterative approximation of xed points," *Springer, 2nd edition*, 2007.

[4] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Trans. on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, 2011.

[5] G. Strbac, "Demand side management: Benefits and challenges," *Energy Policy*, vol. 36, no. 12, pp. 4419–4426, 2008.

[6] M. A. A. Pedrasa, T. D. Spooner, and I. F. MacGill, "Coordinated scheduling of residential distributed energy resources to optimize smart home energy services," *IEEE Trans. on Smart Grid*, vol. 1, no. 2, pp. 134–143, 2010.

[7] A. Barbato, A. Capone, G. Carello, M. Delfanti, D. Falabretti, and M. Merlo, "A framework for home energy management and its experimental validation," *Energy Efficiency*, pp. 1–40, 2014.

[8] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," *Smart Grid Communications, 2010 First IEEE International Conference on*, pp. 238 –243, oct. 2010.

[9] W. Saad, Z. Han, H. V. Poor, and T. Basar, "Game-theoretic methods for the smart grid: an overview of microgrid systems, demand-side management, and smart grid communications," *Signal Processing Magazine, IEEE*, vol. 29, no. 5, pp. 86–105, 2012.

[10] A.-H. Mohsenian-Rad, V. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Trans. on Smart Grid*, vol. 1, no. 3, pp. 320–331, 2010.

[11] D. Fudenberg, *The theory of learning in games*. MIT press, 1998, vol. 2.

[12] M. H. Bowling, "Convergence and no-regret in multiagent learning," *Neural Information Processing Systems (NIPS)*, 2004.

[13] R. S. Sutton and A. G. Barto, *Introduction to reinforcement learning*. MIT Press, 1998.

[14] P. Stone and M. Veloso, "Multiagent systems: A survey from a machine learning perspective," *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, 2000.

[15] L. Rose, S. Lasaulce, S. M. Perlaza, and M. Debbah, "Learning equilibria with partial information in decentralized wireless networks," *IEEE Communications Magazine*, vol. 49, no. 8, pp. 136–142, 2011.

[16] Q. Yu, J. Chen, Y. Sun, Y. Fan, and X. Shen, "Regret matching based channel assignment for wireless sensor networks," *IEEE ICC*, pp. 1–5, 2010.

[17] C. Ibars, M. Navarro, and L. Giupponi, "Distributed demand management in smart grid with a congestion game," 2010, pp. 495–500.

[18] W. Wang, Y. Xu, and M. Khanna, "A survey on the communication architectures in smart grid," *Computer Networks*, vol. 55, no. 15, pp. 3604–3629, 2011.

[19] K. Wang and L. Chen, "On optimality of myopic policy for restless multi-armed bandit problem: an axiomatic approach," *IEEE Transactions on Signal Processing*, vol. 60, no. 1, pp. 300–309, 2012.

[20] B. Shengrong, F. Yu, P. Liu, and P. Zhang, "Distributed scheduling in smart grid communications with dynamic power demands and intermittent renewable energy resources," *IEEE ICC*, pp. 1–5, 2011.

[21] A. Barbato, A. Capone, L. Chen, F. Martignon, and S. Paris, "A distributed demand-side management framework for the smart grid," 2014, preprint, http://arxiv.org/abs/1405.1964.

[22] ECORET Project, Official web site (ITA), http://www.rse-web.it/progetti.page?RSE_originalURI=/progetti/progetto/documento/178/312827&objId=178&typeDesc=Rapporto&RSE_manipulatePath=yes&docIdType=1&country=ita, apr 2014.

[23] MICENE Project,Official web site (ITA), http://www.eerg.it/index.php?p=Progetti_-_MICENE, apr 2014.

[24] R. Jain, *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley - Interscience, 1991.