Factored MDPs for Optimal Prosumer Decision-Making

Angelos Angelidakis
Electronic and Computer Engineering
Technical University of Crete
Greece, GR73100
aggelos@intelligence.tuc.gr

Georgios Chalkiadakis
Electronic and Computer Engineering
Technical University of Crete
Greece, GR73100
gehalk@intelligence.tuc.gr

ABSTRACT

Tackling the decision-making problem faced by a prosumer (i.e., a producer that is simultaneously a consumer) when selling and buying energy in the emerging smart electricity grid, is of utmost importance for the economic profitability of such a business entity. In this paper, we model, for the first time, this problem as a factored Markov Decision Process. By so doing, we are able to represent the problem compactly, and provide an exact optimal solution via dynamic programming—notwithstanding its large size. Our model successfully captures the main aspects of the business decisions of a prosumer corresponding to a community microgrid of *any* size. Moreover, it includes appropriate sub-models for prosumer production and consumption prediction. Experimental simulations verify the effectiveness of our approach; and show that our exact value iteration solution matches that of a state-of-the-art method for stochastic planning in very large environments, while outperforming it in terms of computation time.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

General Terms

Algorithms, Experimentation

Keywords

energy; smart grid; factored MDPs; decision-making

1. INTRODUCTION

In recent years, the term *prosumer* has been coined in order to describe an entity that both produces and consumes energy, implying that prosumers possess the ability to play a key role to the stabilization of the electricity network [4, 20]. As such, and assuming prosumers are able to adjust their behaviour according to dynamic indicators, their smooth integration into the shaping *Smart Grid* is of critical importance [22]. Viewed as a business entity, a prosumer could correspond to a single residence, a specific industry, or to whole neighborhoods of houses that are served by a dedicated microgrid—which may or may not be connected to the rest of the electricity Grid. Our focus of attention in this paper will be optimizing the business decisions of a micro grid-corresponding prosumer, producing electricity from (mainly) renewable energy resources, and which has the option of buying and selling energy from utility companies residing in the larger electricity Grid. Paradigms of such community-oriented and renewable energy-relying microgrids are expected to be commonplace in the near future [4]. Naturally, the viability (economic and otherwise) of such an entity is tightly connected to the quality of its business decisions: i.e., whether to buy, sell, or store energy, within some decisions horizon, in order to possibly make a profit while ensuring the smooth operation of its energy-consuming units; and ensuring this viability is key to the smooth integration of prosumers into the Smart Grid.

Notwithstanding its importance, essentially no work to date has, to the best of our knowledge, attacked this specific problem heads on. By contrast, in this paper, we model, for the first time, the decision problem faced by a microgrid-prosumer planning its energy production, storage and usage strategy for the day ahead as a *factored Markov Decision Process* [7]. Our formulation enables us to provide *an exact optimal solution* (bar certain discretization-related modeling decisions) for the problem faced by a prosumer corresponding to a microgrid of essentially *any* size. In addition, we equip our consumer with specific consumption and production-predicting submodels, which provide it with the necessary input signals on which to base its decisions. As part of our work, we show that *Gaussian processes* and *Bayesian linear regression* techniques can be successfully used for consumption prediction.

Given our model, the solution to the prosumer decision problem can then be computed using standard dynamic programming techniques. In this work, we employed value iteration to this purpose. The effectiveness and efficiency of our approach is verified by comparisons to the performance of *SPUDD*, a state-of-the-art method for stochastic planning in large environments. Our value iteration method, operating over a problem horizon corresponding to 24 hours, is shown to produce policies that coincide with those produced by SPUDD [12]. However, as we explain in Section 6 of the paper, SPUDD has to operate over a state space that is artificially larger, while, at the same time, it does not possess enough structure. This creates a need to build huge input files for SPUDD to operate on, resulting to a huge pre-processing time for the algorithm. As a result, while our method can scale to larger state spaces for our problem, SPUDD cannot produce a solution in such cases within the required 24-hour timeframe.

The rest of this paper is organized as follows. Section 2 provides a brief background on factored MDPs and reviews related work; Section 3 then describes our model, while the value iteration algorithm is described in Section 4; Section 5 presents our methods for predicting the future production and consumption of the prosumer; Section 6 presents our simulation experiments; and, finally, Section 7 concludes this paper and outlines future work.

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

2. BACKGROUND AND RELATED WORK

Factored Markov Decision Processes (FMDPs) [7] provide a *compact* alternative to standard MDP representation. Specifically, they decompose states into sets of *state variables* in order to represent the transition and model compactly—since transitions and rewards may rely on specific model aspects, corresponding to subsets of variables only. Thus, the set of states in a factored MDP representation correspond to multivariate random variables, $\boldsymbol{s} = \langle s_i \rangle$, with the s_i variables taking on values in their corresponding $DOM(s_i)$ domains. Intuitively, state variables correspond to a selection of *features* which are sufficient to describe the system state. In FMDPs, actions are also quite often described as random variables, while reward functions used are assumed to be factored into specific (usually additive) components. Furthermore, FMDP models allow for *external signals*, described by *signal variables*, affecting state variables; while *temporal Bayesian networks (TBNs)* and *influence diagrams* can be employed to represent the effects of actions on state transitions and rewards. A multitude of techniques that exploit the resulting representational structure can then be used to solve large problems, at least approximately (e.g., linear value functions, approximate linear programming, stochastic algebraic decision diagrams, and so on) [11, 7].

Stochastic Planning Using Decision Diagrams (SPUDD) [12], in particular, is a well-known algorithm for finding (near-)optimal policies in very large problems represented as factored MDPs. It is essentially a value iteration algorithm that uses algebraic decision diagrams (ADDs) [5] to represent value functions and policies, assuming an ADD input representation of the FMDP. Specifically, SPUDD operates over an input script describing the factored states and actions, and the FMDP transition model and reward function.

Now there have been a few recent papers dealing with optimal decision-making when buying and selling energy in the Smart Grid. However, most of them do not focus on prosumers. For instance, TacTex [24] was the champion agent for the 2013 Power Trading Agent Competition (PowerTAC). In PowerTAC, several self-interested, autonomous agents corresponding to *brokers* compete with each other with the goal of maximizing profits through energy trading. TacTex does not model the decision making problem of a microgrid prosumer, as we do, but that of a broker simultaneously participating in tariff and wholesale markets. As such, its utility measure is the cash amount existing in a bank, while the energy amount to buy is not considered part of the decision making problem: it is simply set to the difference between predicted demand and the energy that is already procured for the targeted time period. Moreover, there are only 26 states in the MDP solved by TacTex, and a state transition leads to one of only two potential states (by contrast, we tackle MDPs with state-action spaces encompassing hundreds of thousands of elements).

Similarly to TacTex, the work of Peters *et al.* [19] also deals with optimising the long-term behaviour of broker agents during retail electricity trading. They employ the classic SARSA reinforcement learning algorithm [23] for selecting actions in a tariff market. However, it is less flexible than TacTex's tariff market strategy, which is not constrained to a finite set of actions.

We are only aware of two papers that focus on prosumer decision-making. First, Nikovski and Zhang [17] propose a method for finding the optimal conditional operational schedule for a set of power generators, assuming stochastic electricity demand and stochastic generator output. However, in contrast to our work here, they do not tackle the problem of selling or storing the generated power. Second, Kanchev *et al.* [13] propose an energy management system which could be employed by a prosumer managing photovoltaic generators, storage units, and a gas microturbine. However, they assume a deterministic system, not accounting for uncertainty and errors that may occur during the prosumer's operation time.

3. OUR MODEL

The prosumer we consider in this work corresponds to a *microgrid* distributing power to a community. As such, it produces energy by means of *renewable energy sources*, and is responsible for the well-being of *residential consumers*. Moreover, the prosumer has access to *storage devices (batteries)*, which it can use to store energy for future use. Our prosumer is connected to the wider Grid, and it has to take decisions regarding the amounts of energy to purchase or sell to the Grid at pre-specified intervals during the next day. We assume that the Grid is represented by some utility company that can specify *tariffs* determining the sell and buy prices of electricity, to which the prosumer can subscribe (at any one of the aforementioned time intervals). The tariffs available to prosumers for the day-ahead are announced by the utility company at the beginning of each day. Then, the problem facing the prosumer is taking the right decisions as to which tariff to subscribe to and what amounts of energy to buy, sell, or store at any given interval of the day-ahead—so as to meet demand at a minimum cost and make a profit by selling the electricity to the utilities.

We acknowledge that this model formulation, presented in detail below, seemingly disregards the complexity of modern and anticipated electricity markets. Indeed, prosumers could be faced with complex decisions during their simultaneous participation in markets of various types (e.g., *spot*, *forward*, *balancing*, or even *futures*). Despite this fact, we believe that solving the simpler problem of viewing the prosumer as an entity interacting with the wider electricity Grid via pre-specified tariffs determining energy prices (which, however, can be "variable" or to an extent "real-time" themselves), is key to determining behaviour in more complex business environments. In addition, ours is a model that corresponds to conceivable situations in the immediate near future, where (largely) energy self-sufficient communities will be operating their private microgrids, and only occasionally use energy from the wider Grid—essentially as a fallback strategy.

In the rest of this section, we first describe our factored states and actions, and present certain physical constraints they have to adhere to; and we then present the transition model, and our choices for representing the reward function so that it realistically captures the gains and costs from selling and purchasing energy. Importantly, our reward function takes into account periodic operation costs of the prosumer related with subscription to a tariff, as well as its costs because of accumulating battery life losses due to discharging. Moreover, there is nothing in the formulation below that precludes the applicability of our model and proposed solution to microgrid prosumers of a particular size or type.

3.1 Factored Representation

We now describe our problem's factored representation in detail. To begin, the factored states can be described as a multivariate random variable $\boldsymbol{s} = \langle s_i \rangle$, where each variable s_i can take a value in its domain DOM(s_i). There are three factored state variables, listed in Table 1. The first one, *tms*, takes as values the specific time steps at which the prosumer is able to act. Its domain is originally set to [1 ... 24] (one time step per hour in the day). However, as we later explain, we can drop this state variable altogether from the representation, and incorporate it in the problem horizon over which our value iteration method operates; moreover, we also conduct experiments that require the prosumer to act on a half-hourly basis. The second one, *bat*, corresponds to the amount of energy available in the batteries, and its domain is [0 ... $Battery_{max}$], with

$Battery_{max}$ corresponding to the maximum capacity of the storage device(s). Note that *bat* is a naturally continuous state variable, but it was discretized in order to enable its processing by existing FMDP solvers (such as SPUDD). Finally, *tf* corresponds to the tariff the prosumer has assigned to at the moment, and its domain is the enumerated tariffs that the utility offers during the day. That is, DOM(*tf*)=$\{tf_1, \cdots, tf_i, \cdots, tf_K\}$, with K being the number of tariffs available on a specific day. Each tf_i tariff is characterized by a buying and a selling price, denoted $buying_i$ and $selling_i$ respectively, and communicated to the prosumer via external signals.[1]

Then, actions can be described as a multivariate random variable $\boldsymbol{a} = \langle a_i \rangle$ where each variable a_i affects the transition from some factored state to another, and takes a value in its domain DOM(a_i). The discretization for each DOM(a_i) is performed dynamically: it is based on the discretization of the DOM(s_i) domains, in a way that from any given state, actions can lead to any other.

There are three factored actions. First, action *buy*, which describes the amount of energy bought from the electric utility. Positive values for *buy* denote the actual buying of energy from the utility, while negative values mean the prosumer sells energy to the utility. With $Load_{max}$ being the maximum total expected residential consumption load, and the nominal power generating capacity of the renewable energy sources denoted by RES_{nom}, the domain for *buy* is set to [-$RES_{nom} \ldots Load_{max}$]. Second, factored action *chg*, which signifies the attempt to store an amount of energy to the batteries. Its value range is [-$Battery_{max} \ldots Battery_{max}$]. Positive values represent charging the battery, and negative values represent discharging the battery. Finally, the third action, sel_{tf}, corresponds to a selection of tariff by the prosumer. Its domain is [0 ... K]. The value 0 signifies a choice to remain attached to its current tariff, while values 1 to K signify a choice to move to some other of the K tariffs available.[2]

Now, there are three types of external signals the prosumer receives. These are listed in Table 3, and can be described as multivariate random variable $\boldsymbol{sg} = \langle signal_i \rangle$ where each variable $signal_i$ can take a value in its domain DOM($signal_i$). The prosumer employs these signals to help her determine her actions. The first two, *prod* and *cons*, specify the current estimates about the production and consumption levels of the prosumer (at a specific time step). Their domains are defined given the RES_{nom} and $Load_{max}$ values introduced above. Thus, DOM(*prod*)= [0 ... RES_{nom}] , and DOM(*cons*)=[0 ... $Load_{max}$]. The third signal type, $price_{tf}$, specifies, once a day, the buy and sell prices ($buying_i$ and $selling_i$) for each one of the K tariffs, and for each t time step of the day ahead.

Notice that all factored variables in our formulation are independent of the size of the prosumer microgrid—i.e., they are not affected by the number of generators or homes populating it. Moreover, despite the complexity of the problem, the temporal dependencies among the state variables in our model are in fact quite simple, as seen in the 2-stages temporal Bayesian network (2-TBN) of Fig. 1. It can be seen there that a variable value at $t+1$ depends only on the variable's value at t (with value changes triggered, for *bat* and *tf*, by actions *chg* and sel_{tf} respectively).

Features	Denoted	Description
time-step	*tms*	current time step within the operational day
battery	*bat*	amount of stored energy (at a specific time step)
tariff	*tf*	tariff currently in effect (at a specific time step)

Table 1: Factored states

Actions	Denoted	Description
buy	*buy*	buy from the utility
charge	*chg*	charge battery
select tariff	sel_{tf}	select a tariff to subscribe to

Table 2: Factored actions

Signals	Denoted	Description
production	*prod*	predicted levels of energy production
consumption	*cons*	predicted levels of energy consumption
$\{buying_i, selling_i\}$	$price_{tf}$	buying and selling price for tariff i

Table 3: Signal types

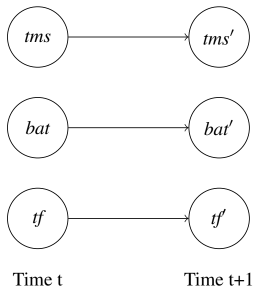


Figure 1: Temporal dependencies among state variables

In what follows, we use the notation x_t to denote the value of a state, action, or signal variable at time t.

3.2 Physical Constraints

There are certain constraints that our state and action variables must adhere to. First, in a setting involving energy exchanges, the *balance energy constraint* [14, 2] must be respected at all times. This means that, at any time step t, power produced (including that bought) should match power consumed (including that stored):

$$prod_t - cons_t - chg_t + buy_t = 0 \tag{1}$$

The second constraint refers to the storage unit(s) of the prosumer. A storage unit cannot be charged over its capacity:

$$chg_t \leq Battery_{max} - bat_t \tag{2}$$

Similarly, the energy quantity discharged from a unit cannot exceed that currently stored in the unit:

$$-chg_t \leq bat_t \tag{3}$$

Finally, for safety reasons, the battery storage level must be always kept between 20% and 100% [10]:

$$0.2 \leq bat_t / Battery_{max} \leq 1 \tag{4}$$

[1] Notice that tariffs can be key to group together a range of consumer preferences, that would have had to be represented by distinct state or action variables otherwise. For instance, one would have wished to represent preferences to consume when buying prices are low, e.g. at night, and sell when selling prices are high—and distinct sell and buy variables would have been required to allow this. Tariffs could potentially incorporate more information, such as special discounts, and so on. Thus, the use of tariffs can be key at reducing the state-action space in such problems.

[2] The additional 'stay-with-current-tariff' action is required as subscribing and resubcribing would entail a subscription cost (thus the action protects the prosumer from that cost).

3.3 Transition Function

State transitions in our model will be in general stochastic, since faults may occur while taking actions like charging or discharging the storage devices and buying or selling energy to the utility. The variable *tms* is an exception to this rule—since one specific time step is always followed by the next one. That is, $Pr(tms_{t+1} = t+1 | tms_t = t) = 1$. For the rest of the variables, we define certain *bounded regions* (with distinct boundaries for each variable), which include a subset of discrete factored states lying close to the factored state intended to transition to by performing a factored action taken at time t. The boundaries can be set to any values required.

Thus, (factored) actions are assumed to have the intended result with some probability p (arbitrarily set to 0.9 in our experiments); while, with probability $1-p$, they transition to some (factored) state within the bounded region (chosen uniformly at random). For instance, assuming that N *bat* states lie within a prespecified bound$_{bat}$ bounded region, the action of charging the battery with an energy amount c at time t (action $chg_t = c$) is successful with probability p:

$$Pr(bat_{t+1} = bat_t + c \mid chg_t = c, bat_t) = p$$

whereas with probability $1-p$ it fails, leading to any potential factored battery state within the bound$_{bat}$ region:

$$Pr(bat_{t+1} = bat \in \text{bound}_{bat} \mid chg_t = c, bat_t) = (1-p)/N$$

Since distinct factored actions can be simultaneously utilized—i.e., the prosumer can select a new tariff, buy energy, and charge the battery at the same time step t— the overall transition probability is given by Eq. 5 as follows.

$$Pr(tms_{t+1}, bat_{t+1}, tf_{t+1} | tms_t, bat_t, tf_t, chg_t, sel_{tf,t}) = \\ Pr(bat_{t+1} | bat_t, chg_t) \cdot Pr(tf_{t+1} | tf_t, sel_{tf,t}) \quad (5)$$

given that the battery level at any time step depends on the previous battery level state and on whether a *chg* action was used, while the tariff in place is affected by a tariff selection action. Notice also that, in our model, buying or selling energy does not have a direct effect on a state variable, thus no state transitions need to be defined for action *buy*. It is thus implicitly assumed that *buy* (a positive or negative energy amount) always succeeds. This assumption is quite realistic, and it is motivated from the need to respect the constraint in Eq. 1 above: choosing how much energy to buy/sell depends on the production and consumption estimates, and on the results of charging the battery. In practice, the latter is an action whose outcome is indeed more uncertain than that of buying/selling energy.

3.4 Factored Reward Representation

The next step is to determine the reward function for our factored MDP. The reward function is associated with *(a)* either the gain from selling power to the utility or the cost of buying power in a certain price; *(b)* the running costs for being subscribed to a tariff; and *(c)* the operation costs of using the storage devices. As such, we choose to represent the reward function as a cost function with three main components. Specifically, the function describing the immediate cost for a transition from state $\boldsymbol{s}_t$ to $\boldsymbol{s}'_{t+1}$ by executing some $\boldsymbol{a}_t$ at time-step t, is defined as follows:

$$Cost(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}'_{t+1}) = Cenergy + C_{period} + C_{bl} \quad (6)$$

We now explain its components in turn. The first component, *Cenergy*, captures the cost per Wh for buying electricity (or the profits from selling it to the utility), given the buy/sell rates prescribed by the tariff in effect:

$$Cenergy(tf_{t+1}, buy_t) = \begin{cases} buy_t \cdot \text{buying}_{tf_{t+1}} & \text{if } buy_t \geq 0 \\ buy_t \cdot \text{selling}_{tf_{t+1}} & \text{if } buy_t < 0 \end{cases} \quad (7)$$

The second component captures the periodic costs $C_{periodic}$ inflicted on the prosumer for being subscribed into a tariff. Naturally, one would expect that "better" tariffs for a prosumer—that is, tariffs specifying high selling prices and low buy prices—will actually incur higher periodic costs (flat rates). Due to this, in our model we make the assumption that periodic costs drop exponentially with decreasing tariff quality (i.e., as the difference between buying price and selling price increases):

$$C_{period}(tf_{t+1}, price_{tf}^{t+1}) = C_1 \ \exp\{-C_2 \cdot (\text{buying}_{tf}^{t+1} - \text{selling}_{tf}^{t+1})\} \quad (8)$$

with C_1 = 0.013, C_2 = -2.7. The function is plotted in Fig. 2.

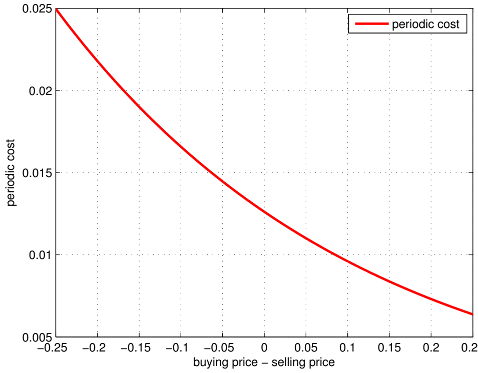


Figure 2: Periodic costs as a function of tariff quality.

The third component of the cost function, C_{bl}, captures the costs associated with *battery life losses*. That is, the costs inflicted from charging (or discharging) the storage devices (batteries) with a charge amount of chg_t, at a given time-step t when the stored energy amount is at bat_t. To estimate this component, we assume the use of deep-cycle batteries, which are lead-acid batteries designed to be regularly deeply discharged (using most of their capacity) [25].

The C_{bl} cost of an attempted *chg* action can then be viewed as a fraction of the $C_{init-bat}$ *initial investment cost* for the batteries:

$$C_{bl} = L_{loss} \cdot C_{init-bat} \quad (9)$$

The "life loss" L_{loss} factor in the above equation is affected by the *effective throughput* A_c of the battery over a certain charge period (measured in Ah) [25]:

$$L_{loss} = \frac{A_c}{A_{total}}$$

Here, A_{total} is the total cumulative throughput (in Ah) during the battery's lifetime. A battery size of $Q\ Ah$ will deliver an effective $A_{total} = 390 \cdot QAh$ over its lifetime [25].

Now, A_c above related to the operating *state of charge* (SOC) and the *actual throughput* A'_c. The latter can be calculated, given the voltage of the battery, as:

$$A'_c = \frac{chg_t}{V_{battery}}$$

To calculate A_c, we first have to define the *state of charge* (SOC) of the battery, as the fraction of its total $Battery_{max}$ capacity covered by its currently stored energy amount, bat_t:

$$SOC = \frac{bat_t}{Battery_{max}}$$

and its value has to be kept always between 0.2 and 1, for safety reasons [25]. A_c is then expressed as

$$A_c = \lambda_{soc} A'_c$$

where λ_{soc} is an *effective weighting factor* given the battery's state of charge. When SOC is between 0.2 and 1, λ_{soc} is approximately linear with SOC [25], which can be expressed as

$$\lambda_{soc} = k \cdot SOC + d$$

In our work, the values of k and d in the previous equation were set to -0.7594 and 1.43 respectively, as a result of applying linear fitting over certain empirically set (SOC, λ_{SOC}) data points reported in [25]. The resulting fitted line is depicted at Fig. 3.

With λ_{soc} at hand, we can then fully determine the C_{bl} component, and use it to determine the life loss cost incurred on batteries during their charge (or discharge) by the application of a chg_t action at time-step t.

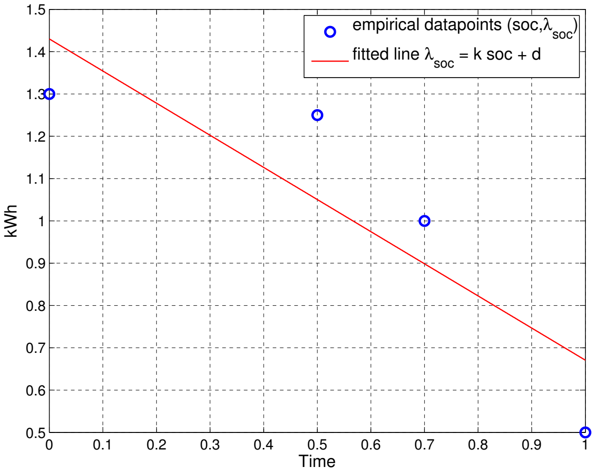


Figure 3: Estimating the λ_{soc} weighting factor.

4. SOLVING THE FMDP

With the above FMDP at hand, the optimal policy can be derived by solving the corresponding Bellman equations. Dynamic programming (DP) methods can be used to obtain the optimal solution [23]. In our work here we used *value iteration (VI)* as the DP method of choice. Interestingly, our experiments confirm that our formulation permits VI to provide us with the solution within a reasonable time, when run on everyday desktops or laptops. This is despite the large state-action space (in the order of hundreds of thousands)–while, at the same time, SPUDD sometimes fails to compute a solution within a reasonable time, when taking its pre-processing requirements into account. We will discuss our experimental results in length in Section 6. For now, we simply outline the instantiation of the VI algorithm in this domain.

Our problem is naturally a finite-horizon problem, thus we employed a finite-horizon VI method. By setting the horizon T to be equal to the number of time steps at which the prosumer is required to act, we can incorporate the *tms* factored state into the problem's horizon, thus effectively reducing the size of the state space.

```
for all instantiations of $\boldsymbol{s}$ do
    set $V_{T+1}(\boldsymbol{s}) = 0$
end
for all time-steps t in descending order
    (i.e., with $1, \cdots, T$ stages-to-go) do
    for all instantiations of $\boldsymbol{s}_t$ do
        $V_t(\boldsymbol{s}_t) \leftarrow \max_{\boldsymbol{a}_t} \sum_{\boldsymbol{s}'_{t+1}} Pr(\boldsymbol{s}'_{t+1} \,|\boldsymbol{a}_t, \boldsymbol{s}_t) \cdot$
            $\big(R(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}'_{t+1}) + V_{t+1}(\boldsymbol{s}'_{t+1})\big)$
    end
end
for all instantiations of $\boldsymbol{s}$ and all time-steps $t$ do
    $\pi(\boldsymbol{s}, t) =$
    $\arg\max_{\boldsymbol{a}} \sum_{\boldsymbol{s}'} Pr(\boldsymbol{s}' \,|\boldsymbol{a}, \boldsymbol{s}) \left(R(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') + V_{t+1}(\boldsymbol{s}')\right)$
end
```

Algorithm 1: Value iteration for solving the FMDP

Then, with $\boldsymbol{s}'_t$ denoting the potential successor states of $\boldsymbol{s}_t$; with $Pr(\boldsymbol{s}'_{t+1} \,|\boldsymbol{a}_t, \boldsymbol{s}_t)$ denoting the probability of state transitions from $\boldsymbol{s}_t$ to possible successor states $\boldsymbol{s}'_{t+1}$, given that action $\boldsymbol{a}_t$ was taken; and $R(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}'_{t+1}) = -Cost(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}'_{t+1})$ denoting the corresponding immediate reward (the negative immediate cost), the VI algorithm iteratively estimates the value function for the factored states, and outputs an optimal policy π, as shown in Alg. 1.

5. PROSUMER PRODUCTION AND CONSUMPTION MODELS

Naturally, the estimated production from the renewable energy sources distributed on the microgrid, and the predicted load consumption of the connected consumers, affect the policy of the prosumer. The prosumer is notified about the expected production and consumption values via the *prod* and *cons* signals. Thus, it is necessary to predict values for those signals that are as accurate as possible, to assist the decision-making process of the prosumer.

5.1 Production Prediction

To obtain the production estimates of the photovoltaic systems (PVS) and wind turbine generators (WTG) of our microgrid, we employ *RENES* [18], a web-based PVS and WTG production prediction tool. RENES generates PVS and WTG production estimates given time, geographical coordinates and online weather forecasts, and it comes with specific performance guarantees. For PVS production predictions, RENES utilizes non-linear approximation components for turning cloud-coverage into radiation forecasts, which are then used for production prediction. It has an interactive web-based interface, along with an API providing XML responses to prediction requests. New production estimates are provided every half an hour. RENES predictions are provided free-of-charge. The tool and API can be accessed at www.intelligence.tuc.gr/renes/ .

5.2 Consumption Prediction

Here we show how to employ two regression methods to predict the load consumption of the prosumer: Gaussian Process (GP) and Bayesian Linear Regression. To begin, the input data of our model, and the output data whose values we are trying to predict, correspond to the factored state *tms* and the signal variable *cons* :

$$\boldsymbol{x} = (\mathit{tms}_1, \dots, \mathit{tms}_n) \tag{10}$$

$$\boldsymbol{y} = (\mathit{cons}_1, \dots, \mathit{cons}_n) \tag{11}$$

that is, they are sequences of consumption data, containing information about time-steps and the respective load consumption.

Our goal in regression, is to make predictions of the target variables for new inputs. Given a set of output data

$$\boldsymbol{y} = (y_1, \dots, y_n)^T$$

corresponding to input values $(x_1, \dots, x_n)$, where n is the length of the time sequence we use, we predict the target variable y_{n+1} for a new input vector with an additional x_{n+1} value.

Bayesian Linear Regression.

The first method we use for prediction is *Bayesian linear regression.* To begin, we define a model parameter $\boldsymbol{w}$

$$\boldsymbol{w} = [\boldsymbol{x}\ \boldsymbol{y}]$$

with x, y as in Eqs. 10 and 11 above. For a set of training samples, $\mathcal{D} = \{(x_j, y_j), j = 1, ..., n\}$ (x_j inputs and y_j outputs) we need to predict the posterior distribution of w given the target values y.

Now, the conjugate prior of $\boldsymbol{w}$ is a *Gaussian distribution*:

$$p(\boldsymbol{w}) = \mathcal{N}(\boldsymbol{w}|\mu_0, \sigma_0^2)$$

where μ_0 is the mean and σ_0^2 the variance noise; while the *likelihood function* $p(y|w)$ is given also by a *Gaussian distribution* of the form

$$p(\boldsymbol{y}|\boldsymbol{w}) = \mathcal{N}(\boldsymbol{y}|\ \Phi\boldsymbol{w}, \beta^{-1}I)$$

where β is noise single precision parameter, and Φ is a polynomial basis function.

With conjugate prior and likelihood function at hand, the *posterior distribution* is computed using Bayes theorem for Gaussians [6]. In order to find the posterior distribution, we just require the mean and the variance:

$$p(\boldsymbol{w}|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{w}|\mu_n, S_n),\ where$$

$$\mu_n = S_n(S_0^{-1}\mu_0 + \beta\Phi^T\boldsymbol{y})$$

$$S_n^{-1} = S_0^{-1} + \beta\Phi^T\Phi$$

In this work, we adopt a zero-mean isotropic Gaussian, governed by a single precision parameter α, so that:

$$p(\boldsymbol{w}|\alpha) = \mathcal{N}(w|0, \alpha^{-1}I)$$

Then, the corresponding posterior distribution $p(\boldsymbol{w}|\boldsymbol{y})$ has:

$$\mu_n = \beta S_n\Phi^T\boldsymbol{y}$$

$$S_n^{-1} = \alpha I + \beta\Phi^T\Phi$$

Evidence approximation [6] is utilised to calculate the optimal values of the hyper-parameters α and β.

Gaussian Process Regression.

The second regression method that we use is Gaussian Process (GP) with two form of kernels, a gaussian and a polynomial one. The use of a GP with a Gaussian kernel appears to be the better choice for our setting, as we demonstrate in Sec. 5.3 below. We note that Gaussian Processes have also been recently applied for consumption reduction prediction in electricity demand management settings [3, 16, 21].

Gaussian processes can be used for regression and classification without a parametric model assumption. For a set $\mathcal{D} = \{(x_j, y_j), j = 1, ..., n\}$ of training samples, with x_j inputs and y_j noisy outputs, we need to predict the distribution of the noisy output at some test locations. We assume the model:

$$y_j = f(x_j) + \epsilon_j,\ \text{where } \epsilon_j \sim \mathcal{N}(0, \sigma_{noise}^2)$$

with σ_{noise}^2 the variance noise.

GP regression is a Bayesian approach that assumes a priori that function values follow: $p(\mathbf{f}|x_1, x_2, ..., x_n) = \mathcal{N}(\mathbf{0}, K)$ where $\mathbf{f} = [f_1, f_2, ..., f_n]^T$ is the vector of latent function values, $f_j = f(x_j)$ and K is the covariance matrix that is computed by a "kernel" covariance function $k(\cdot, \cdot)$: $K_{jk} = k(x_j, x_k)$.

The kernel functions used in this work are given by a polynomial and Gaussian form [6] respectively:

$$k(x_j, x_k) = \theta_0 + \theta_1(x_j^T x_k) + \theta_2(x_j^T x_k)^2$$

$$k(x_j, x_k) = \theta_0\ \exp\big(-\frac{(x_j - x_k)^T(x_j - x_k)}{2(\theta_1)^2}\big)$$

where the θ_* are the model's hyper parameters. Their optimal values can be found by maximizing the log likelihood [6], for instance using *backtracking line search* [8], as we do in this work.

Finally, in order to proceed to the inference, we must combine the joint GP prior obtained by the test values with the likelihood $p(\mathbf{y}|\mathbf{f})$, via Bayes rule. The joint GP prior and the independent likelihood are both Gaussian with mean and variance at a test point x_* as follows:

$$GP_\mu(x_*, \mathcal{D}) = K_{*,f}(K_{f,f} + \sigma_{noise}^2 I)^{-1}\mathbf{y} \tag{12a}$$

$$GP_\sigma(x_*, \mathcal{D}) = K_{*,*} - K_{*,f}(K_{f,f} + \sigma_{noise}^2 I)^{-1}K_{f,*} \tag{12b}$$

5.3 Comparing Regression Methods

To choose a Φ polynomial basis function to use for Bayesian linear regression (BLR), we performed cross-validation with random subsampling repeated 10 times for different polynomial functions [15]. For this, we split our consumption dataset (described in Section 6 below) to an 80% part for training, and a 20% one for testing. The results, in terms of *mean square error (MSE)*, are shown of Table 4. The degree of the polynomial with the minimum average MSE is D=5: thus, this was the polynomial of choice for *BLR*. We then compared the performance of *BLR* against that of a Gaussian process (GP) that employed either the polynomial (GP-poly) or the Gaussian (GP-G) kernel mentioned above. The prediction performance of the methods is depicted in Fig. 4; and Table 5 contains the methods' MSE. Results show that *GP-G* does much better than *GP-poly* and *BLR* in terms of MSE, achieving a quite low MSE value. Moreover, the prediction mean of the *GP-G* method apparently follows more closely the actual consumption pattern, as emerging from the actual (x, y) data points.

6. EXPERIMENTS AND RESULTS

We evaluate our model by examining a residential prosumer at New Hampshire, New England, northeastern United States. The data used in our prediction of residential load consumption for the area, comes from the Public Service Company of New Hampshire, and is freely available in their website (http://www.psnh.com/). Our simulated prosumer serves 30 households and includes 20 photovoltaic modules with nominal power 60kW, 2 windturbines with nominal power 1000kW and 24 deep cycle 12Volts batteries 212AH C20 /

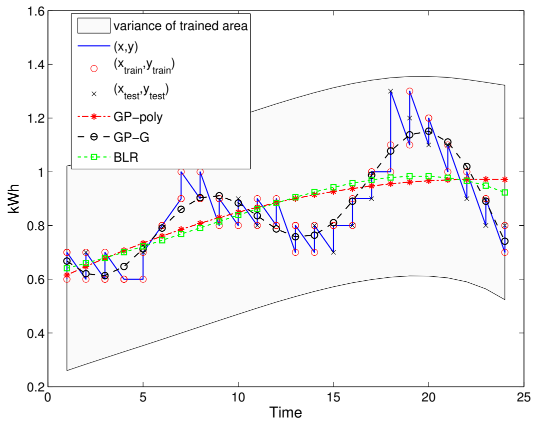


Figure 4: Prediction performance of *GP-poly*, *GP-G*, and *BLR*. The (x, y) input-target pairs are actual consumption data points. The *GP-G* mean matches the (typical) daily electricity demand curve of our dataset, with two consumption peaks.

Degree of Polynomial	MSE
1	0.022372
2	0.021312
3	0.020175
4	0.017679
5	0.016861
6	0.017329
7	0.017355
8	0.017167
9	0.017399
10	0.017611

Table 4: MSE of Bayesian linear regression Φ functions

FMD200 – VRLA/AGM, with cost of each battery 269,00 €. Estimated battery lifetime is 10-12 years. As mentioned earlier, we employ RENES (www.intelligence.tuc.gr/renes/) to obtain predictions regarding the power production of the prosumer's renewable energy generators; the services provided by RENES are also free of charge. Our simulations were conducted with data regarding a specific day-ahead (24 / 10 / 2014), at which date the predicted electricity consumption and electricity production profile of the particular residential prosumer was as presented in Fig. 5. All experiments were conducted on a 2.10 GHz x 4 Intel Core i3-2310M processor, with 8GB of memory.

We now discuss our choices for the factored MDP representation for our simulated prosumer, and proceed to compare the performance of our value iteration solution method with that of SPUDD.

We initially adopted the following discretisation for our state and action variables (signals are not discretised, but simply communicate the production and consumption predictions, and tariff characteristics to the prosumer). The discretisation step size is shown inside the range of the factored state *bat* (corresponding to the prosumer's batteries' array), and the action *chg* below:

$$bat = [0kWh \ : \ 1kWh \ : \ 60kWh]$$

$$chg = [-60kWh \ : \ 1kWh \ : \ 60kWh]$$

GP with polynomial kernel (GP-poly)	0.0173
GP with Gaussian kernel (GP-G)	0.006943
Bayesian linear Regression (BLR)	0.0169

Table 5: MSE of GP & Bayesian Linear Regression

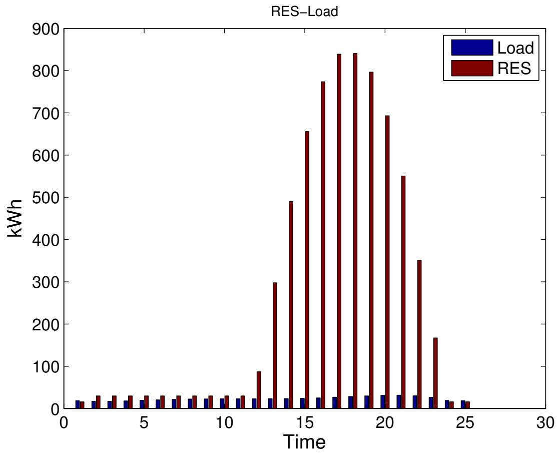


Figure 5: Predicted production of renewable energy sources (RES) and predicted load consumption of prosumer (Load)

We also defined nine tariffs, which are as follows:

$$\mathit{tf}_1 = \{0.1€, 0.1€\} \ \ \mathit{tf}_4 = \{0.2€, 0.1€\} \ \ \mathit{tf}_7 = \{0.3€, 0.1€\}$$

$$\mathit{tf}_2 = \{0.1€, 0.2€\} \ \ \mathit{tf}_5 = \{0.2€, 0.2€\} \ \ \mathit{tf}_8 = \{0.3€, 0.2€\}$$

$$\mathit{tf}_3 = \{0.1€, 0.3€\} \ \ \mathit{tf}_6 = \{0.2€, 0.3€\} \ \ \mathit{tf}_9 = \{0.3€, 0.3€\}$$

which thus give rise to 10 possible sel_{tf} tariff selection actions (9 corresponding to choosing one of the tariffs+1 for choosing to stay with their current one).

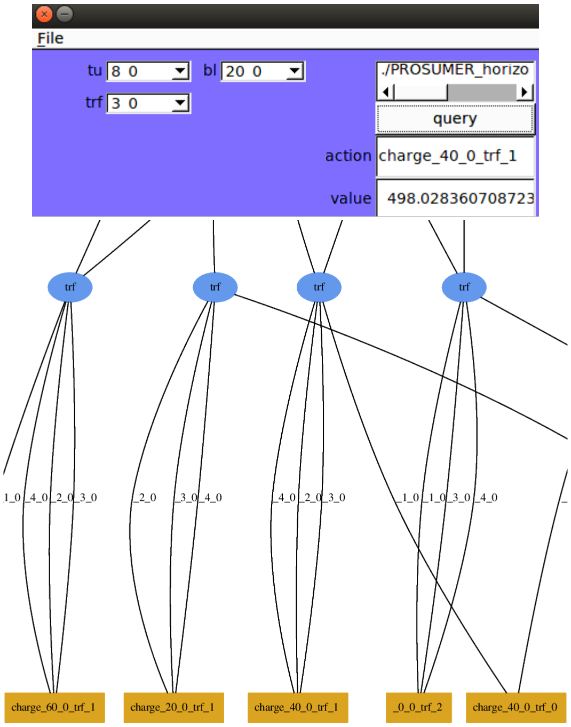
The transition boundaries for our state variables were initially set to boundary$_{bat}$=1kWh and boundary$_{tf}$=0.1€. Given those boundaries, the maximum number of transitions leading from one state to another are $\sim$ 15.

Now, the discretisation above resulted to a state-action space size of $|S \times A| = 664290$. Notice, however, that in order for SPUDD to be able to model our problem, there was a need to add an additional state variable, *tms* , due to the fact that SPUDD does not allow us to incorporate the time-step into the problem horizon, but requires a complete representation for all states. Simply put, formulating the problem requires the states to be "stamped" by the time-step, so as to keep them distinct from each other, for the SPUDD solver to be able to operate upon the representation. Thus, in the case of SPUDD, state-action spaces shown in Table 6 are expanded by a factor of $|DOM(tms)|$ (i.e., 24 or 48, for our experiments). We also note that a policy extracted by SPUDD can be presented through policy diagrams and the *pquery* SPUDD GUI tool. Figure 6 provides an insight on how such diagrams look like, for a toy example (a smaller instance of our problem).

Horizon	$\|S \times A\|$	bounded region size	value iteration (hours)	SPUDD (hours)	
				Script	Runtime
24	664290	15	1.76	13.4992	0.184
		90	15.84	46.9188	1.19
	2624490	15	8.7603	36.98	0.73975
48	664290	15	3.5	16.8221	0.4271

Table 6: Running time of value iteration and SPUDD for four different scenarios. "Script" refers to the pre-processing time required for the SPUDD input files to be generated, while "Runtime" denotes the subsequent SPUDD execution time.

We compared SPUDD to value iteration for this initial discretisation, and observed that the optimal policy computed through value iteration and SPUDD for the day-ahead coincide with each other. Nevertheless, value iteration produced the optimal policy in approximately 15% of the required time for SPUDD to extract the same policy. The exact running times are presented in Table 6.



tu 8 0
bl 20 0
trf 3 0
./PROSUMER_horizo
action charge_40_0_trf_1
value 498.028360708723

Figure 6: Part of the SPUDD's optimal policy (below), for a toy example with $|S| \cdot |A| = 63000$, and 15 factored states at most within any bounded region. Running time to create the script was: 29.7 sec and to execute it: 1.46 sec. Variables tu, bl and trf presented in pquery correspond to the factored states *tms*, *bat* and *tf* respectively. Variables are presented in blue bubbles, with factored actions in yellow squares, e.g. $charge\_40\_0\_trf\_1$ represents the actions *chg*= 40kWh and *sel*$_{tf}$ = 1. The pquery GUI tool is shown above.

Following that, we increased the size of the transition boundaries so as to contain 90 state variables instead of 15. The boundaries used for the transitions from one state to another in this case are: $boundary_{bat}$=10kWh and $boundary_{tf}$=0.2€. SPUDD could not produce a solution within the time that our "planning-for-the-day-ahead" problem must be solved (maximum 24 hours). By contrast, the running time for the simple value iteration method was approximately 15.8 hours, as shown in Table 6.

We then increased the size of state and action spaces to $|S \times A| = 2624490$ (by reducing the discretisation step sizes for our factored variables), but kept the bounded regions for state transitions quite small ($boundary_{bat}$=1kWh and $boundary_{tf}$=0.1€). SPUDD, once more, was not able to produce a solution within 24 hours, and could not generate a final policy with the available memory, in contrast to our value iteration method (Table 6).

Finally, we also experimented with a scenario involving 48 (half-hour) time steps at which the prosumer is required to act (as is usually the case in electricity markets). In this case, we had $|S \times A| = 664290$, $boundary_{bat}$=1kWh , and $boundary_{tf}$=0.1€. Once again, value iteration provided us with the same (optimal) policy as SPUDD, but in approximately 25% of the time (Table 6).

The experiments above demonstrate the limitations of SPUDD when used for problems that do not possess enough structure to allow for a compact enough representation of the required transitions in its input files. Both SPUDD and value iteration provide us with the same optimal policies in all experiments–that is, policies which intuitively maximize profits from selling/buying decisions while ensuring that consumer needs are satisfied. Nevertheless, value iteration required a fraction of SPUDD's total required time to produce the solution. We note that this is despite the fact that we took special care to make our factored representation as compact as possible for SPUDD to operate upon.

7. CONCLUSIONS

This paper employs, for the first time, factored MDPs to model the decision problem faced by a prosumer planning its energy flow management for the day-ahead. Our model incorporates the key factors responsible for the effective operation of a microgrid prosumer, regardless of its size; and allows us to obtain the exact optimal solution to the problem. We used a simple value iteration algorithm to compute the solution to this sequential decision making problem, and demonstrated our method's effectiveness and efficiency by comparing it to the performance of *SPUDD*. By so doing, we exposed the limitations of this particular FMDP solver. While our model enables the simple VI method to compute the optimal solution within a reasonable time, the problem does not have enough structure to allow the creation of a compact input file for SPUDD to operate on, resulting to poor performance. In addition, this work provides specific predictive tools for obtaining prosumer consumption and production estimates, and exhibits how *Gaussian processes* and *Bayesian linear regression* techniques can be used for consumption prediction in this setting (with Gaussian processes emerging as the most successful).

Our model and solution technique allow the determination of optimal policies regarding the main prosumer activities. However, additional state and action variables can be added to the model, to allow for additional operations to take place (e.g., choosing to alter the projected production and consumption levels for increased economic benefits). In future work, we intend to enrich our model in that direction, and also study the performance of more-powerful-than-VI factored MDP solution techniques, such as approximate linear programming and approximate policy iteration [11]. Applying these techniques in this domain is not a straightforward task, since it requires the careful determination of appropriate basis value functions. Finally, we intend to use these ideas in order to smoothly incorporate prosumers within *cooperatives* that are fast emerging in the Smart Grid [9, 1].

8. ACKNOWLEDGEMENTS

We are grateful to Dr. Jesse Hoey for help with using SPUDD and fruitful discussions.

REFERENCES

[1] Federation of groups and cooperatives of citizens for renewable energy in Europe. http://www.rescoop.eu.

[2] T. Ackermann (ed.). *Wind power in power systems*. J. Wiley & Sons, 2005.

[3] C. Akasiadis and G. Chalkiadakis. Stochastic filtering methods for predicting agent performance in the Smart Grid. In *Proc. of ECAI/PAIS-2014*, pages 1205–1206, 2014.

[4] P. Asmus. Microgrids, virtual power plants and our distributed energy future. *The Electricity Journal*, pages 72–82, 2010.

[5] R. Bahar, E. Frohm, C. Gaona, G. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Algebric decision diagrams and their applications. *Formal methods in system design*, pages 171–206, 1997.

[6] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[7] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of Artificial Intelligence Research (JAIR)*, pages 1–94, 1999.

[8] S. Boyd and L. Vandenberghe. *Convex Optimization*. 2004.

[9] G. Chalkiadakis, V. Robu, R. Kota, A. Rogers, and N. Jennings. Cooperatives of distributed energy resources for efficient virtual power plants. In *Proc. of AAMAS-2011 - Volume 2*, pages 787–794, 2011.

[10] J. Chiasson and B. Vairamohan. Estimating the state of charge of a battery. *IEEE Transactions on Control Systems Technology*, pages 465–470, 2005.

[11] C. Guestrin, D. Koller, R. Parr, and S. Venkataraman. Efficient solution algorithms for factored MDPs. *Journal of Artificial Intelligence Research (JAIR)*, pages 399–468, 2003.

[12] J. Hoey, R. St-Aubin, A. J. Hu, and C. Boutilier. SPUDD: Stochastic planning using decision diagrams. Proc. of UAI-1999, pages 279–288, 1999.

[13] H. Kanchev, D. Lu, F. Colas, V. Lazarov, and B. Francois. Energy management and operational planning of a microgrid with a PV-based active generator for Smart Grid applications. *Industrial Electronics, IEEE Transactions on*, pages 4583–4592, 2011.

[14] D. Kirschen and G. Strbac. *Fundamentals of Power System Economics*. J. Wiley & Sons, 2005.

[15] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proc. of IJCAI-95*, pages 1137–1143, 1995.

[16] J. Kolter and J. Ferreira. A large-scale study on predicting and contextualizing building energy usage. In *AAAI*, pages 1349–1356, 2011.

[17] D. Nikovski and W. Zhang. Factored markov decision process models for stochastic unit commitment. In *IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply (CITRES)*, pages 28–35, 2010.

[18] A. Panagopoulos, G. Chalkiadakis, and E. Koutroulis. Predicting the power output of distributed renewable energy resources within a broad geographical region. In *Proc. of ECAI /PAIS 2012*, pages 981–986, 2012.

[19] M. Peters, W. Ketter, M. Saar-Tsechansky, and J. Collins. A reinforcement learning approach to autonomous decision-making in smart electricity markets. *Machine learning*, 92(1), pages 5–39, 2013.

[20] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. Jennings. Putting the 'smarts' into the Smart Grid: A grand challenge for Artificial Intelligence. *Commun. ACM*, 55(4), pages 86-97, 2012.

[21] A. Rogers, S. Maleki, S. Ghosh, and N. Jennings. Adaptive home heating control through gaussian process prediction and mathematical programming. In *Proc. of ATES 2011*, pages 71–78, 2011.

[22] A. Rogers, S. Ramchurn, and N. Jennings. Delivering the Smart Grid: Challenges for autonomous agents and multi-agent systems research. In *Proc. of AAAI-2012*, pages 2166–2172, 2012.

[23] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction, 1998.

[24] D. Urieli and P. Stone. Tactex'13: a champion adaptive power trading agent. In *Proc. of AAMAS-2014*, pages 1447–1448, 2014.

[25] B. Zhao, X. Zhang, J. Chen, C. Wang, and L. Guo. Operation optimization of standalone microgrids considering lifetime characteristics of battery energy storage system. *Sustainable Energy, IEEE Transactions on*, pages 934–943, 2013.