Multistage Energy Management of Coordinated Smart Buildings: A Multiagent Markov Decision Process Approach

Georgios Tsaousoglou, Nikolaos Efthymiopoulos, Prodromos Makris, Emmanouel Varvarigos

***Abstract*—Smart buildings provide an important opportunity for large-scale development of demand response, due to their existing flexibility that can be harvested through internet-of-things technologies with minimal cost of new equipment. However, after taking an energy management action, the resulting energy consumption of a building depends on several uncertain factors. Thus, the consumption of the smart building is not directly controllable and, contrary to the typical approach taken in the literature, it cannot be modeled as a decision variable in practice. In this paper, we consider the problem of coordinating the stochastic load control actions of multiple smart buildings under such endogenous uncertainties. We model the problem as a Multi-agent Markov Decision Process and, after reformulations, we bring it to a solvable decomposed form. Our simulations compare the proposed approach with a myopic approach that does not consider future uncertainties, and also quantify the trade-off between cost-effectiveness and computational time in terms of the look-ahead horizon length.**

***Index Terms*—smart buildings, stochastic control, demand response, multiagent systems, Markov Decision Process**

I. Introduction

A. End-use flexibility

In traditional energy systems, all actions concerning energy balancing and cost optimization take place exclusively on the supply side while the demand side passively consumes energy, being unaware of the system's state and real-time operational costs. The increasing penetration of Renewable Energy Sources (RES), presents a new reality where energy supply becomes contingent to weather conditions. This transition brings challenges that relate to lower levels of supply side dispatchability and higher levels of uncertainty. An envisioned remedy is to draw on the flexibility capabilities of the demand side, by leveraging modern information and communication technologies.

Integrating and managing the distributed flexible resources is a major topic of research. Especially the introduction of new electricity markets, that can incorporate the characteristics of demand-side flexibility into their market design, forms a predominant challenge. A recent review of energy flexibility markets, [1], provides a relevant introduction to such topics, while [2] provides a discussion and review focused on market models and mechanisms for local electricity markets.

The authors are with the National Technical University of Athens (NTUA), Greece, and with the Institute of Communications and Computer Systems (ICCS), Greece.

This work was partly funded from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 863876 in the context of the FLEXGRID project.

An early modelling framework can be found in [3], where each flexible resource is characterized by a utility function and its local constraints, while the system's cost depends on the aggregated consumption of all flexible resources. The authors present a standard Lagrangian decomposition towards coordinating the dispatch decisions of the energy consumers via the iterative exchange of price signals. Later works in this direction, elaborate more on the user models [4], [5], on the strategic aspects of user participation [6], [7], and on the inclusion of distribution network constraints into the market mechanism [8], [9], while [10] proposes a game-theoretic mechanism that accounts for truthful participation in a constraints-informed low voltage flexibility market.

While the studies mentioned above use deterministic models, a second branch of the literature takes various types of uncertainty into account. The relevant studies can be categorized into three approaches: scenario-based stochastic programming, learn-to-optimize, and model-free methods.

In the first approach, the decisions are stochastically optimized using scenarios of uncertainty realization and scenario-dependent decision variables. The scenario tree grows exponentially in the number of look-ahead stages. However, when the uncertainty is *exogenous* (i.e., the uncertain parameters evolve independently of our control decisions), this issue can be managed by applying scenario-reduction techniques. For example, electricity prices or RES generation are factors of uncertainty that do not depend on the energy management decisions. Thus, the resulting uncertainty can be managed by considering a set of representative scenarios for the decision horizon's prices or RES generation. In this path, [11] presents a hybrid robust-stochastic approach for the scheduling of an electric vehicles' fleet, under uncertain electricity prices and real-time charging needs. The authors in [12], stochastically optimize the dispatch of different energy devices (electrical and gas-fired) under uncertain electricity and gas prices. In [13], the authors consider a community of prosumers and a set of predefined scenarios for the community's RES output. A stochastic day-ahead market is cleared towards scheduling the day-ahead commitments of each prosumer. In [14], the authors present the generic modelling framework for two-stage stochastic market-clearing of economic dispatch problems, using realization scenarios for uncertain parameters. As elaborated in [15], the number of scenarios needed to adequately capture the characteristics of a stochastic process is usually too large, rendering the associated stochastic programming problem computationally intractable. Conclusively, the optimality

of such methods depends on the number of scenarios sampled and hence is in trade-off with the method's computational time.

In the learn-to-optimize approach, a realization scenario for the (exogenously) uncertain parameters is again sampled, but utilized to solve a deterministic dispatch problem offline. By solving multiple such problem instances, one can create a dataset where an instantiation of the uncertain parameters is mapped to a particular optimal dispatch solution. A machine learning algorithm can be trained on that dataset, offline, so as to learn to predict a good dispatch, online, once provided with an up-to-now uncertainty realization. In this direction, [16] presents a stochastic gradient boosting trees approach that learns to solve the optimal power flow problem in a low-voltage distribution grid. The authors in [17] take this approach a step further by using the partial derivatives of the dispatch optimization problem to train a sensitivity-informed neural network. Finally, in [18] a neural network is trained on the optimal dual variables of the economic dispatch problem instead of the primal, which is shown to enhance performance while guaranteeing constraint satisfaction.

Model-free methods take a black-box approach where the decision-maker tries different actions with the purpose of gradually learning a good policy. In [19], a demand response aggregator faces a cost function for electricity, and a disutility cost from its users. The aggregator decides the retail price using Q-learning. Q-learning is also used in [20], towards controlling the power consumption of residential appliances. The authors configure the algorithm with a forecasting module that predicts the realization of uncertain parameters. In [21], the authors take on the problem of deciding the dispatch of electricity appliances, in a setting where the electricity price depends on the aggregated consumption. An actor-critic architecture is proposed, with the critic neural network placed at a coordinator, whereas an actor neural network is placed with each user. Thus, the users decide on local actions but system-wide knowledge is integrated through the critic. An advantage of model-free methods is that they can handle *endogenous* uncertainty (uncertain factors that depend also on the control decisions), and in fact they do not even require statistical knowledge over the uncertainty. The downside of such methods is that an online learning process can be prohibitively costly, while (near-)optimality and constraint satisfaction are not guaranteed.

B. The case of smart buildings

Of all distributed resources capable of providing flexibility, smart buildings constitute a particularly interesting case, as buildings account for about 40% of global energy use, and can readily offer control capabilities by using the already available flexibility of existing assets. Moreover, this flexibility can be drawn at a very small cost and with minimal new equipment by using Internet-of-Things (IoT) technologies. In particular, conventional electricity-consuming assets can be configured with sensors, actuators and controllers, such that data-collection and energy management decisions become possible. An overview of IoT solutions for smart energy management in buildings can be found in [22]. In the face of this opportunity, a number of studies have presented IoT-based solutions for building energy management systems (EMS).

Demonstrations of real-life smart buildings, with IoT-based EMS, have been presented in [23] and [24]. In [25], the authors propose a thermostat control for air-conditioning units in a smart building. These works use heuristic algorithms to make energy management decisions, e.g. based on labeled device priorities. In [26], the energy consumption of the appliances of a smart home is predicted using a neural network. By using the neural network as a prediction module, the authors scheduled the home's appliances using a genetic algorithm.

The authors in [27] explain the difficulty of formulating an accurate physics-based thermal model for a building. Such difficulties have motivated multiple related studies that use reinforcement learning techniques for building EMS. A review paper focusing especially on such studies can be found in [28]. Towards addressing the problem of thermal model development, the authors in [29] propose an IoT-informed learning module that automatically constructs a building thermal model, describing how indoor temperature changes under different environmental conditions and thermostat control actions. Similarly, [30] presents a method on deriving a stochastic model for the building's consumption evolution, based on measurement data. In particular, the authors discretized the building's state variables (namely the building's energy consumption) into different levels, and experimentally estimated the probability of transitioning to each consumption level at the next time instance, given that the building is at a certain consumption level at the present time. It was shown that 10-20 consumption levels were adequate to represent a fairly accurate building model.

In contrast to physics-driven modeling methods, a learning-based IoT solution does not need a domain expert to manually design a thermal model for each particular building, but can operate in a plug-and-play fashion, dramatically enhancing the widespread deployment of building EMS in practice. Combined with the minimal cost needed for new equipment, IoT-based smart buildings form an opportunity towards the large-scale adoption of end-use flexibility technologies.

C. Research gap and contribution

By contrasting the literature on coordination frameworks for multiple flexible resources, discussed in subsection I-A, against the literature on real-life building EMS, discussed in subsection I-B, one can notice a bold difference: The former uses simplified models, such as first order temperature dynamics, and sophisticated algorithmic techniques for energy management decisions. On the other hand, the latter focuses more on implementation aspects for IoT-based energy management (usually for a single building), while the EMS decisions are based on heuristic/meta-heuristic algorithms or on model-free methods, suggesting that physics-based building models are not practical or suitable for real-life building EMS. Reasons include:

- Estimating the actual power consumption resulting from an energy management action is difficult in practice, as a building's energy consumption is influenced by various internal and external factors (e.g. room occupancy,

available daylight, user behavior) that create considerable uncertainties [27].
- Thermostat control is preferable to power control for thermal devices, as the latter can interfere with user comfort and/or cause equipment damage [25].
- Constructing physics-based transition models for each building appliance is impractical, as they require domain-knowledge and building-specific or appliance-specific technical studies [29].

The emerging research question is: how can we coordinate a set of multiple smart buildings to jointly provide coordinated energy management services under the real conditions of endogenous multistage uncertainty? In this paper, we adopt statistical models for buildings (e.g. as proposed in [29], [30]), and propose a coordination mechanism that is able to accommodate the inherent uncertainty of IoT-informed building EMS. By adopting such models, we depart from the existing literature of economic dispatch studies (subsection I-A) that model the consumption of a flexible resource (building) as a control variable. In contrast, we adopt the more realistic approach, where a building's EMS can only configure a set of actions (e.g. altering thermostat setpoints, dimming lights, etc) that affect the building's consumption stochastically, but do not determine it decisively. Moreover, the EMS ensures the building's smooth operation, e.g. avoids dysfunctional situations that cause user discomfort. More specifically, when asked to save energy the EMS chooses which loads can be curtailed without compromising the building's smooth operation. Conclusively, a building's energy consumption is only partially controllable in practice.

Putting a smart building into economic operation, by affecting thermostats, dimming lights and curtailing non-essential loads, stochastically affects the building's energy consumption, which in turn affects the decision on the next actions, and so on. This endogenous uncertainty does not allow for scenario reduction methods to be applied, since the uncertainty realization at each stage is entangled with the decision variables of that stage. For this reason, a realization scenario cannot be generated independently from the decision variables, which impedes the applicability of learn-to-optimize methods as well. The predominant formal framework for multistage decisions under uncertainty is that of Markov Decision Processes (MDP) – a cornerstone of Artificial Intelligence - while in the presence of multiple interacting agents (i.e., buildings), the relevant framework is called a Multiagent MDP (MMDP).

In what follows, we model the economic dispatch problem of a set of smart buildings as an MMDP. By leveraging the available statistical information, we present an optimal solution method, while handling the curse of dimensionality that typically characterizes MMDPs without resorting to model-free methods. The preliminaries of MDPs are presented in the next section (Section II). Accordingly, Section III presents the modeling of a single smart building as an MDP. In Section IV, we formulate the problem and present the proposed approach towards reaching the optimal solution. Section V presents the simulation setting and results, whereas Section VI concludes this paper.

II. Preliminaries and Notation

An MDP comprises a set of *states*, a set of *actions*, a *transition function*, a *cost* function, and a set of decision *stages*. We use calligraphic letters, and brackets to denote sets. Namely, $\mathcal{S} = \{s_1, s_2, ...\}$ denotes the set of states, $\mathcal{A} = \{a_1, a_2, ...\}$ the set of actions, and $\mathcal{T} = \{1, 2, ..., T\}$ the set of decision stages within a finite horizon. Parenthesis brackets are used to define a tuple of variables. Namely, $s \triangleq (x, y, z)$ means that a state $s \in \mathcal{S}$ is defined by the state variables x, y, z, and $a \triangleq (u, v, w)$ means that an action $a \in \mathcal{A}$ is defined by the action variables u, v, w. Also, $f(s, a) \equiv f(x, y, z, u, v, w)$ means that f is a function of the state and action variables of s and a.

Suppose that at some stage $t \in \mathcal{T}$, the control agent finds itself in some state $s \in \mathcal{S}$. By taking an action $a \in \mathcal{A}$, the agent suffers a cost, defined by the cost function $c(t, s, a)$, while in the next stage $t + 1$ the system transitions to a next state $\hat{s}$ with probability $p(\hat{s}|t, s, a)$. The set of all such probabilities (for all possible stage-state-action-state tuples) constitutes the MDP's transition function. A *policy* is a mapping from states to actions, i.e., a particular way of deciding an action once presented with any system state. The solution concept of an MDP is to identify an optimal policy, i.e., a policy that minimizes the expected cost over the whole horizon of decision stages. Equivalently, this model can also be seen as a finite horizon discrete-time dynamical system, where the policy is the control law [31].

An optimal policy can be defined in terms of the so-called *value function* $v(s, t)$ of each state and stage, as prescribed by the Bellman equation:

$$v(s,t) = \min_{a \in \mathcal{A}} \left\{ c(t,s,a) + \sum_{\hat{s} \in \mathcal{S}} p(\hat{s}|t,s,a) \cdot v(\hat{s}, t+1) \right\}. \quad (1)$$

Intuitively, a state's value is the optimal (over actions) expected cost when starting at that state, consisting of the here-and-now cost $c(t, s, a)$ of taking action a, plus the future cost of transitioning, with probability $p(\hat{s}|t, s, a)$, to a state $\hat{s}$ that has a value $v(\hat{s}, t + 1)$.

In the case where different action variables are controlled by different agents, we have an MMDP. An MMDP is not trivially decomposable into a set of MDPs, since the optimal agents' actions are generally coupled through the joint cost and joint transition functions. Let $\mathcal{N}$ denote the set of agents. We use $\boldsymbol{s} = (s_n)_{n \in \mathcal{N}}$, or $\hat{\boldsymbol{s}}$, to denote a joint state and $\boldsymbol{a} = (a_n)_{n \in \mathcal{N}}$ to denote a joint action. The respective set $\boldsymbol{\mathcal{S}}$ of joint states, is defined by the Cartesian product of local state sets $\mathcal{S}_n$, as in $\boldsymbol{\mathcal{S}} \triangleq \times_{n \in \mathcal{N}} \mathcal{S}_n$, and similarly for the set of joint actions. The cardinality of $\boldsymbol{\mathcal{S}}$ is $|\boldsymbol{\mathcal{S}}| = |\mathcal{S}_n|^{|\mathcal{N}|}$. This reveals the typical challenge in MMDPs, which is the curse of dimensionality, i.e., the fact that the number of joint states and joint actions grows exponentially in the number of agents, which renders the tracking of all MMDP states and actions impractical.

III. System Model

Let us consider a set $\mathcal{N} = \{1, 2, ..., N\}$ of buildings and a set $\mathcal{T} = \{1, 2, ..., T\}$ of timeslots within a finite horizon (e.g. a day). A smart building $n \in \mathcal{N}$ features an EMS that, through

certain measures, can configure a mode for the building's operation in the upcoming timeslot. More specifically, the EMS can bring the building into one of four modes, namely `Normal` (undisturbed) operation, `Economic` (energy saving) operation, `Emergency` operation (only critical loads are served) and, finally, `Charging` operation where the building consumes more than its normal demand, e.g., by charging batteries or preheating rooms in order to take advantage of a low-price time. These operational modes define the building's *Action* space

$$\mathcal{A}_n = \{\texttt{Normal, Economic, Emergency, Charging}\}, \tag{2}$$

where an action variable $a_{n,t} \in \mathcal{A}_n$ denotes the operational control action chosen by the building's EMS at timeslot t. Note that regular buildings $\nu \in \mathcal{N}_{\text{reg}} \subset \mathcal{N}$, that do not feature control capabilities, can be modeled as a special case of this representation by fixing $a_{\nu,t} = \texttt{Normal}$, for all $t \in \mathcal{T}$.

At each timeslot t, the building's operational mode $\mu_{n,t}$ is directly determined by the action chosen in the previous timeslot and therefore takes its values from the set $\mathcal{M}_n \equiv \mathcal{A}_n$. The building's energy consumption is discretized into L levels $\{\mathrm{e}_n^1, \mathrm{e}_n^2, ..., \mathrm{e}_n^L\}$, such that the actual building's consumption is rounded to a level $l_{n,t} \in \mathcal{L}_n = \{1, 2, ..., L\}$ corresponding to a consumption of e_n^l kWh in t. Thus, the building's *State* space is defined by

$$\mathcal{S}_n = \mathcal{M}_n \times \mathcal{L}_n, \tag{3}$$

where a state tuple $s_{n,t} \triangleq (\mu_{n,t}, l_{n,t})$ defines the building's operation mode and energy demand level at a timeslot t.

After an action is taken, the system transitions to the next timeslot, and the building reaches a new state $\hat{s}_{n,t+1} \triangleq (\mu_{n,t+1}, l_{n,t+1})$. The new operational mode is directly defined by the action $a_{n,t}$ chosen in the previous timeslot, i.e.

$$\mu_{n,t+1} = a_{n,t}. \tag{4}$$

The building's new consumption level $l_{n,t+1}$ stochastically depends on the building's previous state $s_{n,t}$ and on the action taken, based on the following descriptions. First, let us define a continuous variable $\widetilde{e}_{n,t+1}(s_{n,t}, a_{n,t})$ to denote the building's *expected* energy consumption at $t+1$ as a function of $s_{n,t}, a_{n,t}$. For $a_{n,t} = \texttt{Normal}$, the building's expected consumption $\widetilde{e}_{n,t+1}(s_{n,t}, \texttt{Normal})$ is derived through a stochastic process that will be defined shortly. If action `Economic` is chosen at t instead, then the building will operate in economic mode in $t+1$, resulting in an energy curtailment of $\beta_{n,t+1}(\texttt{Economic}) \cdot \widetilde{e}_{n,t+1}(s_{n,t}, \texttt{Normal})$ with respect to the building's would-be expected consumption under normal operation in $t+1$, as in

$$\widetilde{e}_{n,t+1}(s_{n,t}, \texttt{Economic}) = (1 - \beta_{n,t+1}(\texttt{Economic})) \cdot \widetilde{e}_{n,t+1}(s_{n,t}, \texttt{Normal}), \tag{5}$$

where $\beta_{n,t+1}(\texttt{Economic}) \in (0, 1)$, such that the curtailed demand is a percentage of the building's would-be normal consumption $\widetilde{e}_{n,t+1}(s_{n,t}, \texttt{Normal})$. Similarly, for $a_{n,t} = \texttt{Emergency}$, we have a curtailment factor of $\beta_{n,t+1}(\texttt{Economic}) < \beta_{n,t+1}(\texttt{Emergency}) < 1$, while, for normal operation, we obviously set $\beta_{n,t+1}(\texttt{Normal}) = 0$. Finally, for $\mu_{n,t} = \texttt{Charging}$, we have $\beta_{n,t+1}(\texttt{Charging}) < 0$. In general, the expected consumption in a timeslot $t+1$ is a function of the building's operation mode at $t+1$ (defined by the action taken in t), and the building's consumption at $t+1$ under the `Normal` operation mode, as in

$$\widetilde{e}_{n,t+1}(s_{n,t}, a_{n,t}) = (1 - \beta_{n,t+1}(a_{n,t})) \cdot \widetilde{e}_{n,t+1}(s_{n,t}, \texttt{Normal}). \tag{6}$$

As for the building's expected consumption under normal operation $\widetilde{e}_{n,t+1}(s_{n,t}, \texttt{Normal})$, this is defined as

$$\widetilde{e}_{n,t+1}(s_{n,t}, \texttt{Normal}) = \zeta_{n,t} \cdot \widetilde{e}_{n,t} + \gamma_{n,t} + \delta_{n,t}(a_{n,t-1}) \cdot \beta_{n,t}(a_{n,t-1}) \cdot \widetilde{e}_{n,t}. \tag{7}$$

The first term of (7), models the building's loads consuming at state $s_{n,t}$, (i.e. timeslot t), that continue to consume at $\hat{s}_{n,t+1}$ (i.e., $t+1$). Clearly, we always have $\zeta_{n,t} \leq 1$. The second term, $\gamma_{n,t}$, denotes the new demand arrivals at the end of t. The last term models the backlog demand that was not served in t and requests to be served in $t+1$. The backlog demand for $t+1$ is a percentage $\delta_{n,t} \in [0, 1]$ of the demand curtailed at t, which depends on the action taken at $t-1$, as defined previously. Note that, in (7), we can replace $a_{n,t-1}$ with $\mu_{n,t}$, based on (4), so that the Markov property is preserved.

Given the expected consumption defined by Eq. (6), we can now present how the transition function is built. By examining all possible consumption levels in $\mathcal{L}_n$, we calculate a distance metric

$$d(l, \widetilde{e}_{n,t+1}) = |\mathrm{e}_n^l - \widetilde{e}_{n,t+1}| \tag{8}$$

for each $l \in \mathcal{L}_n$. We then set the probability $p(\hat{s}_{n,t+1}|t, s_{n,t}, a_{n,t})$ of state $\hat{s}_{n,t+1} = (\mu_{n,t+1}, l_{n,t+1})$ as

$$p(\hat{s}_{n,t+1}|t, s_{n,t}, a_{n,t}) = \begin{cases} 0, & \text{for } \mu_{n,t+1} \neq a_{n,t} \\ \frac{1/d(l_{n,t+1}, \widetilde{e}_{n,t+1})}{\sum_{l \in \mathcal{L}_n} \left(1/d(l, \widetilde{e}_{n,t+1})\right)}, & \text{for } \mu_{n,t+1} = a_{n,t}. \end{cases} \tag{9}$$

Intuitively, Eq. (9) assigns higher probability to consumption levels close to the expected consumption, and lower probability to distant levels, while the building's operational mode transitions deterministically based on Eq. (4), i.e. for modes different than the selected action, the probability is zero. Based on the above, the *Transition function*: ${\mathcal{S}_n}^2 \times \mathcal{A}_n \times \mathcal{T} \rightarrow [0, 1]$, from state-timeslot-action-state tuples to probabilities, is fully defined.

A graphical illustration of the building's transition is provided in Fig. 1. In the figure's example, the building finds itself in a certain consumption level (in yellow) in timeslot 1, and expects a peak demand in timeslot 2 and a valley in timeslot 3 under normal operation (grey). By putting the building into economic operation in timeslot 2, the expected consumption is flattened since there is an energy curtailment in timeslot 2, and a respective rebound of the backlog demand in timeslot 3. The uncertainty is depicted in the figure by the horizontal arrows (in blue for the chosen action and in grey for no action), where

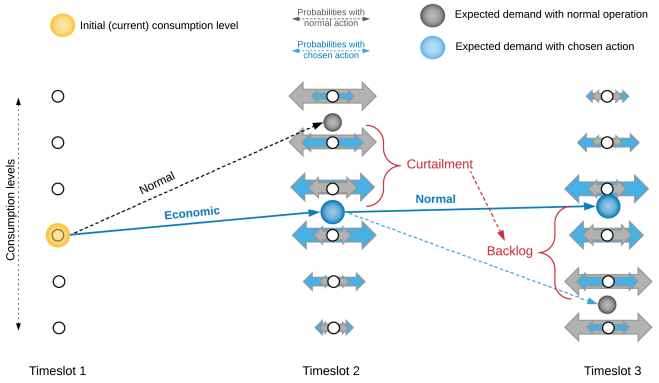


Fig. 1: Transition of a building's expected energy consumption in three timeslots. Horizontal arrows illustrate the probability magnitudes of each consumption level under normal operation (grey) and under `Economic` mode for Timeslot 2 (blue).

the size of each arrow expresses the probability of the building actually landing in the respective energy consumption level.

A *Cost function* $c_n : S_n \times \mathcal{A}_n \times \mathcal{T} \to \mathbb{R}$ is implemented in the EMS agent, mapping a timeslot-state-action triple $(t, s_{n,t}, a_{n,t})$ to a monetary cost. This cost relates to expenses or user dissatisfaction that stem from deploying the control action $a_{n,t}$ from state $s_{n,t}$ at timeslot t. The methods to be presented can accommodate a large variety of relevant cost functions. However, for exemplary purposes, we present one plausible choice:

$$c_n(t, s_{n,t}, a_{n,t}) = \omega_{n,t}(\mu_{n,t}) + \eta_n(\mu_{n,t}, a_{n,t}) + \theta_n \cdot C\left(\sum_{n\in\mathcal{N}} \mathrm{e}_n^l\right). \quad (10)$$

The first term expresses the disutility of the building for operating at mode $\mu_{n,t}$ at t. Naturally, it is $\omega_{n,t}(\texttt{Normal}) = \omega_{n,t}(\texttt{Charging}) = 0$, and $\omega_{n,t}(\texttt{Emergency}) > \omega_{n,t}(\texttt{Economic}) > 0$, for each n and t. The second term poses a penalty η for bringing the building into a restricted mode for two timeslots in a row. Namely, it is $\eta_n(\mu_{n,t}, a_{n,t}) = 0$ if $\mu_{n,t}$ or $a_{n,t}$ is `Normal` or `Charging`, and for the rest of the cases, it is

$$\begin{aligned}\eta_n(\texttt{Emergency},\texttt{Emergency}) > \\ \eta_n(\texttt{Economic},\texttt{Emergency}) > \\ \eta_n(\texttt{Emergency},\texttt{Economic}) > \\ \eta_n(\texttt{Economic},\texttt{Economic}).\end{aligned}$$

The last term refers to the energy cost of the whole set of buildings, where θ_n is a factor denoting n's share of the total cost $C\left(\sum_{n\in\mathcal{N}} \mathrm{e}_n^l\right)$, such that

$$\sum_{n\in\mathcal{N}} \theta_n = 1. \quad (11)$$

This share of energy cost, can refer to various use cases of aggregation schemes, e.g., a community of buildings that jointly participate in the wholesale market. For the function $C(\cdot)$ we adopt the quadratic cost function typically used in the related literature:

$$C\left(\sum_{n\in\mathcal{N}} \mathrm{e}_n^l\right) = \mathrm{w}_1 \cdot \sum_{n\in\mathcal{N}} \mathrm{e}_n^l + \mathrm{w}_2 \cdot \left(\sum_{n\in\mathcal{N}} \mathrm{e}_n^l\right)^2. \quad (12)$$

Such quadratic cost functions are widely used to model the fuel cost of the system's marginal generator or to increasingly penalize deviations from a given dispatch profile.[1]

IV. Problem Formulation and Solution Method

Our objective is to calculate a joint policy $\boldsymbol{\pi}$ that minimizes the total expected cost, as in

$$\min_{\boldsymbol{\pi}} \left\{ \mathbb{E}_{\psi\sim\boldsymbol{\pi}} \left[\sum_{n\in\mathcal{N}} c_n \right] \right\}, \quad (13)$$

where $\psi \sim \boldsymbol{\pi}$ is the set of joint *State-Action* trajectories, conditioned over joint policy $\boldsymbol{\pi}$. By mentally extending Fig. 1 into multiple timeslots, buildings, and action choices, the problem at hand can be visualized as a path selection in a tree graph. From a mathematical viewpoint, this formulation can also be viewed as a generalization of the (deterministic) demand *scheduling* problem frequently encountered in the smart grid literature: When there is no uncertainty, i.e. $p(\hat{s}_{n,t+1}|t, s_{n,t}, a_{n,t}) = 1$ for some state $\hat{s}_{n,t+1}$ and every (t, s, a), then we have a deterministic scheduling problem.

Recall that $\boldsymbol{\mathcal{S}} \triangleq \times_{n\in\mathcal{N}}\mathcal{S}_n$ stands for the set of joint states and $\boldsymbol{\mathcal{A}} \triangleq \times_{n\in\mathcal{N}}\mathcal{A}_n$ the set of joint actions. The Bellman equation of the MMDP defines the optimal solution by instantiating a *value function* $v_{\boldsymbol{s},t}$ that represents the expected future cost when starting at a particular joint state $\boldsymbol{s} \triangleq (s_{n,t})_{n\in\mathcal{N}} \in \boldsymbol{\mathcal{S}}$ and timeslot t. This value function adheres to a recursive definition, where the value $v_{\boldsymbol{s},t}$ of a state at t, depends on the here-and-now cost $\sum_{n\in N} c_n(t, s_{n,t}, a_{n,t})$ of the optimal actions, and on the expected future cost $\sum_{\hat{\boldsymbol{s}}\in\mathcal{S}} p(\hat{\boldsymbol{s}}|t, \boldsymbol{s}, \boldsymbol{a}) \cdot v_{\hat{\boldsymbol{s}},t+1}$, which consists of the value $v_{\hat{\boldsymbol{s}},t+1}$ of the next state $\hat{\boldsymbol{s}}$ times the conditional probability of reaching $\hat{\boldsymbol{s}}$ by taking action $\boldsymbol{a}$ from state $\boldsymbol{s}$ at t. This definition allows us to formulate problem (13), as the calculation of the state values through a linear program:

$$\max_{v_{\boldsymbol{s},t}} \left\{ \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{S}} v_{\boldsymbol{s},t} \right\} \quad (14)$$

$$\text{s.t. } v_{\boldsymbol{s},t} \le \sum_{n\in N} c_n(t, \boldsymbol{s}, \boldsymbol{a}) + \sum_{\hat{\boldsymbol{s}}\in\mathcal{S}} p(\hat{\boldsymbol{s}}|t, \boldsymbol{s}, \boldsymbol{a}) \cdot v_{\hat{\boldsymbol{s}},t+1}, \quad \forall \boldsymbol{s} \in \boldsymbol{\mathcal{S}},\ t \in \mathcal{T}/\{T\},\ \boldsymbol{a} \in \boldsymbol{\mathcal{A}}, \quad (15)$$

while, for $t = T$, we set $v_{\boldsymbol{s},t} = \sum_{n\in N} c_n$. Observe that constraint (15) enforces the value function definition.

The challenge with problem (14)-(15) is that the number of variables $v_{\boldsymbol{s},t}$ grows exponentially with the number of buildings, since we need one such variable for every possible

[1] In the latter case the cost function takes the form $\mathrm{w}_2 \cdot \left(\sum_{n\in\mathcal{N}} \mathrm{e}_n^l - \mathrm{D}\right)^2$ where D is the target consumption. For our purposes, the inclusion of a constant term D would not affect the proposed approach. Thus the method to be presented is applicable to both the use cases mentioned.

joint state. Towards handling this intractability, we proceed with building a decomposable reformulation of problem (14)-(15). First, we consider the dual problem, which reads as

$$\min_{x_{t,\boldsymbol{s},\boldsymbol{a}}} \left\{ \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \left(x_{t,\boldsymbol{s},\boldsymbol{a}} \cdot \sum_{n\in\mathcal{N}} c_n(t,\boldsymbol{s},\boldsymbol{a}) \right) \right\} \tag{16}$$

$$\text{s.t.} \sum_{\hat{a}\in\mathcal{A}} x_{t+1,\hat{\boldsymbol{s}},\hat{\boldsymbol{a}}} = \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} p(\hat{\boldsymbol{s}}|t,\boldsymbol{s},\boldsymbol{a}) \cdot x_{t,\boldsymbol{s},\boldsymbol{a}}, \quad \forall t \in \mathcal{T}/\{T\}, \hat{\boldsymbol{s}} \in \mathcal{S}. \tag{17}$$

In the dual program, variables $x_{t,\boldsymbol{s},\boldsymbol{a}}$ represent the probability that the system finds itself in joint state $\boldsymbol{s}$, timeslot t, and takes joint action $\boldsymbol{a}$. By using Eq. (11), we can write the term $\sum_{n\in\mathcal{N}} c_n(t,\boldsymbol{s},\boldsymbol{a})$ as

$$\sum_{n\in\mathcal{N}} c_n(t,\boldsymbol{s},\boldsymbol{a}) = \sum_{n\in\mathcal{N}} c_n^*(t,\boldsymbol{s},\boldsymbol{a}) + C\left(\sum_{n\in\mathcal{N}} \mathrm{e}_n^l\right), \tag{18}$$

where

$$c_n^*(t,\boldsymbol{s},\boldsymbol{a}) = \omega_{n,t}(\mu_{n,t}) + \eta_n(\mu_{n,t}, a_{n,t}). \tag{19}$$

The term $\sum_{n\in\mathcal{N}} c_n^*(t,\boldsymbol{s},\boldsymbol{a})$ is directly decomposable to local (per building) states and actions. Towards handling the term $C\left(\sum_{n\in\mathcal{N}} \mathrm{e}_n^l\right)$, we introduce the auxiliary continuous variables $\widetilde{E}_t$ (one per timeslot), to represent the expected aggregate consumption at timeslot t. Under these considerations, let us define local variables $x_{n,t,s_{n,t},a_{n,t}}$ which, with slight abuse of notation, we will write as $x_{n,t,s,a}$. Using these local variables, we can reformulate problem (16) as

$$\min_{x_{n,t,s,a}} \left\{ \sum_{n\in\mathcal{N}} \sum_{t\in\mathcal{T}} \sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} \left(x_{n,t,s,a} \cdot c_n^*(t,\boldsymbol{s},\boldsymbol{a}) \right) + \sum_{t\in\mathcal{T}} C(\widetilde{E}_t) \right\} \tag{20}$$

subject to

$$\sum_{\hat{a}\in\mathcal{A}_n} x_{n,t+1,\hat{s},\hat{a}} = \sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} p(\hat{s}_{n,t+1}|t,s_{n,t},a_{n,t}) \cdot x_{n,t,s,a}, \quad \forall n \in \mathcal{N}, t \in \mathcal{T}/\{T\}, \hat{s} \in \mathcal{S}_n \tag{21}$$

$$\sum_{n\in\mathcal{N}} \sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} x_{n,t,s,a} \cdot \mathrm{e}_n^l \le \widetilde{E}_t, \qquad \forall t \in \mathcal{T}/\{1\}. \tag{22}$$

where $p(\hat{s}_{n,t+1}|t,s_{n,t},a_{n,t})$ is given by Eq. (9). We also define each building's initial state, $s_{n,1}^*$ by setting

$$\sum_{a\in\mathcal{A}_n} x_{n,1,s_{n,1}^*,a} = 1 \tag{23}$$

$$\sum_{s\neq s_{n,1}^*} \sum_{a\in\mathcal{A}_n} x_{n,1,s,a} = 0. \tag{24}$$

Observe that, in problem (20)-(24), the decision variables of different buildings are coupled only via the constraints (22). This allows us to decompose the problem into a set of local subproblems. Specifically, we bring the term

$$\sum_{n\in\mathcal{N}} \sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} x_{n,t,s,a} \cdot \mathrm{e}_n^l - \widetilde{E}_t$$

into the objective function along with a Lagrange multiplier, and use the Alternate Direction Method of Multipliers (ADMM) to iteratively update the decision variables (in parallel) and the multipliers. Let k denote the iteration number and $\lambda_t^{(k)}$ the respective multiplier for t. The algorithm's updates read as

For each building n:

$$x_{n,t,s,a}^{(k+1)} = \text{argmin}_{x_{n,t,s,a}} \left\{ \sum_{t\in\mathcal{T}} \left[\sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} \left(x_{n,t,s,a} \cdot c_n^*(t,\boldsymbol{s},\boldsymbol{a}) \right) + \lambda_t^{(k)} \cdot \left(\sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} x_{n,t,s,a} \cdot \mathrm{e}_n^l + \sum_{i\in\mathcal{N}/\{n\}} \sum_{s\in\mathcal{S}_i} \sum_{a\in\mathcal{A}_i} x_{i,t,s,a}^{(k)} \cdot \mathrm{e}_i^l - \widetilde{E}_t^{(k)} \right) + \frac{\rho}{2} \cdot \left(\sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} x_{n,t,s,a} \cdot \mathrm{e}_n^l + \sum_{i\in\mathcal{N}/\{n\}} \sum_{s\in\mathcal{S}_i} \sum_{a\in\mathcal{A}_i} x_{i,t,s,a}^{(k)} \cdot \mathrm{e}_i^l - \widetilde{E}_t^{(k)} \right)^2 \right] \right\}$$

$$\text{subject to} \quad (21), (23), (24). \tag{25}$$

Expected Aggregated Demand $\widetilde{E}_t$:

$$\widetilde{E}_t^{(k+1)} = \text{argmin}_{\widetilde{E}_t} \left\{ \sum_{t\in\mathcal{T}} C(\widetilde{E}_t) + \lambda_t^{(k)} \cdot \left(\sum_{n\in\mathcal{N}} \sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} x_{n,t,s,a}^{(k)} \cdot \mathrm{e}_n^l - \widetilde{E}_t \right) + \frac{\rho}{2} \cdot \left(\sum_{n\in\mathcal{N}} \sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} x_{n,t,s,a}^{(k)} \cdot \mathrm{e}_n^l - \widetilde{E}_t \right)^2 \right\}. \tag{26}$$

Lagrange multiplier λ_t:

$$\lambda_t^{(k+1)} = \lambda_t^{(k)} + \rho \cdot \left(\sum_{n\in\mathcal{N}} \sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} x_{n,t,s,a}^{(k+1)} \cdot \mathrm{e}_n^l - \widetilde{E}_t^{(k+1)} \right). \tag{27}$$

The variable updates presented above are iteratively executed until the convergence criterion is met, i.e.,

$$\left| \sum_{n\in\mathcal{N}} \sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} x_{n,t,s,a}^{(k)} \cdot \mathrm{e}_n^l - \widetilde{E}_t^{(k)} \right| \le \varepsilon, \ \forall t \in \mathcal{T}. \tag{28}$$

Notice that the local problem (25) of a building n is optimized based only on the building's local variables $x_{n,t,s,a}$. The summation $\sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} x_{n,t,s,a}$ within the last (quadratic) term of (25) creates bilinear terms that can, however, be sidestepped by introducing auxiliary variables $y_{n,t}$, as in

$$x_{n,t,s,a}^{(k+1)} = \text{argmin}_{x_{n,t,s,a}, y_{n,t}} \left\{ \sum_{t\in\mathcal{T}} \left[\sum_{s\in\mathcal{S}_n} \sum_{a\in\mathcal{A}_n} \left(x_{n,t,s,a} \cdot c_n^*(t,\boldsymbol{s},\boldsymbol{a}) \right) \right. \right.$$

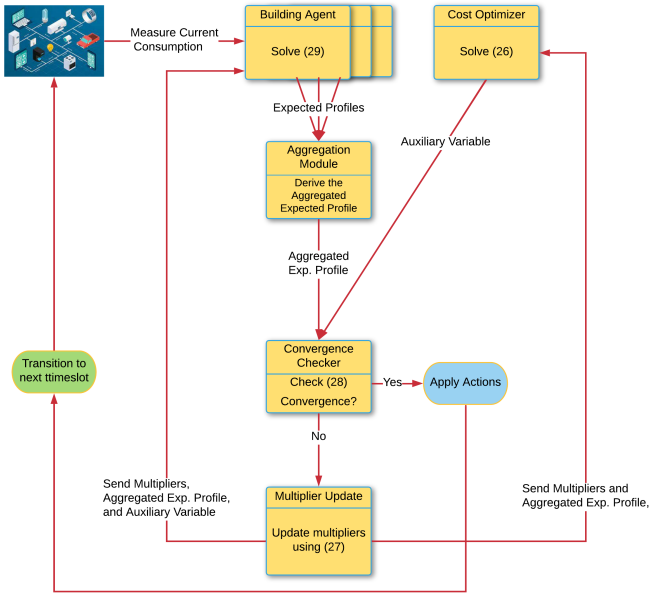


Fig. 2: Flowchart and information exchange of the proposed algorithm.

$$+\lambda_t^{(k)}\cdot\left(y_{n,t}+\sum_{i\in\mathcal{N}/\{n\}}\sum_{s\in\mathcal{S}_i}\sum_{a\in\mathcal{A}_i}x_{i,t,s,a}^{(k)}\cdot \mathrm{e}_i^l-\widetilde{E}_t^{(k)}\right)$$

$$+\frac{\rho}{2}\cdot\left(y_{n,t}+\sum_{i\in\mathcal{N}/\{n\}}\sum_{s\in\mathcal{S}_i}\sum_{a\in\mathcal{A}_i}x_{i,t,s,a}^{(k)}\cdot \mathrm{e}_i^l-\widetilde{E}_t^{(k)}\right)^2\Bigg]\Bigg\}$$

subject to

$$y_{n,t}=\sum_{s\in\mathcal{S}_n}\sum_{a\in\mathcal{A}_n}x_{n,t,s,a}\cdot \mathrm{e}_n^l,\qquad \forall t\in\mathcal{T}.$$
$$(21),(23),(24). \tag{29}$$

Therefore, the final form of the problem is a decomposed convex program, which the ADMM algorithm can tackle efficiently. The algorithm is implemented in a rolling horizon fashion for a simulation horizon $\mathcal{T}$ of 24 timeslots, where in each timeslot $t\in\mathcal{T}$ the system looks $\mathfrak{T}$ timeslots ahead, i.e. at each stage the system is optimized for the period $[t,\min\{T,t+\mathfrak{T}\}]$. After the optimal decisions are reached, the system transitions to the next timeslot $t+1$, where each building's initial consumption and mode are determined by the decisions made in t, and the system is optimized for the period $[t+1,\min\{T,t+1+\mathfrak{T}\}]$, and so on.

The procedure is demonstrated graphically in Fig. 2 with red arrows depicting the information flows. The aggregation module sums the expected consumptions $\sum_{s\in\mathcal{S}_n}\sum_{a\in\mathcal{A}_n}x_{n,t,s,a}^{(k)}\cdot \mathrm{e}_n^l$ of buildings for each timeslot to derive the aggregated expected profile $\left(\sum_{n\in\mathcal{N}}\sum_{s\in\mathcal{S}_n}\sum_{a\in\mathcal{A}_n}x_{n,t,s,a}^{(k)}\cdot \mathrm{e}_n^l\right)_{t\in\mathfrak{T}}$. The convergence checker compares this profile with the auxiliary variable $\widetilde{E}_t^{(k)}$ (calculated by the cost optimizer module as in (26)) and checks the criterion (28). If the criterion is met, the buildings are signaled to apply the decided actions. Otherwise, the multipliers are updated, using (27) and the updated values of $\lambda_t,\widetilde{E}_t^{(k)}$ and $\sum_{n\in\mathcal{N}}\sum_{s\in\mathcal{S}_n}\sum_{a\in\mathcal{A}_n}x_{n,t,s,a}^{(k)}\cdot \mathrm{e}_n^l$, for each timeslot, are communicated to the buildings to repeat the procedure.

TABLE I: Distributions of setting's parameter values

Parameter	Average Value	Standard deviation
$\beta_{n,t}(\texttt{Normal})$	0	0
$\beta_{n,t}(\texttt{Economic})$	0.15	0.02
$\beta_{n,t}(\texttt{Emergency})$	0.25	0.02
$\beta_{n,t}(\texttt{Charging})$	-0.15	0.02
$\delta_{n,t}(\texttt{Normal})$	0	0
$\delta_{n,t}(\texttt{Economic})$	1	0
$\delta_{n,t}(\texttt{Emergency})$	1	0
$\delta_{n,t}(\texttt{Charging})$	0.5	0.1
$\omega_{n,t}(\texttt{Normal})$	0	0
$\omega_{n,t}(\texttt{Economic})$	10	3
$\omega_{n,t}(\texttt{Emergency})$	30	10
$\omega_{n,t}(\texttt{Charging})$	0	0
$\eta_{n,t}(\texttt{Economic},\texttt{Economic})$	10	3
$\eta_{n,t}(\texttt{Emergency},\texttt{Economic})$	20	5
$\eta_{n,t}(\texttt{Economic},\texttt{Emergency})$	20	5
$\eta_{n,t}(\texttt{Emergency},\texttt{Emergency})$	50	10

Observe that a building is able to derive the term $\sum_{i\in\mathcal{N}/\{n\}}\sum_{s\in\mathcal{S}_i}\sum_{a\in\mathcal{A}_i}x_{i,t,s,a}^{(k)}\cdot \mathrm{e}_i^l$ by simply subtracting its own profile from the expected aggregate consumption $\sum_{n\in\mathcal{N}}\sum_{s\in\mathcal{S}_n}\sum_{a\in\mathcal{A}_n}x_{n,t,s,a}^{(k)}\cdot \mathrm{e}_n^l$ received. Note that the buildings do not share their local information with other buildings or with any other party, since only the aggregated term is needed for all the other modules (i.e. the convergence check, the multipliers update and the cost optimizer). This property allows the proposed algorithm to fully maintain the buildings' privacy, since it is well-established that the aggregation module can be implemented in a fully distributed fashion e.g. using Distributed Hash Tables (see [6] for a detailed elaboration).

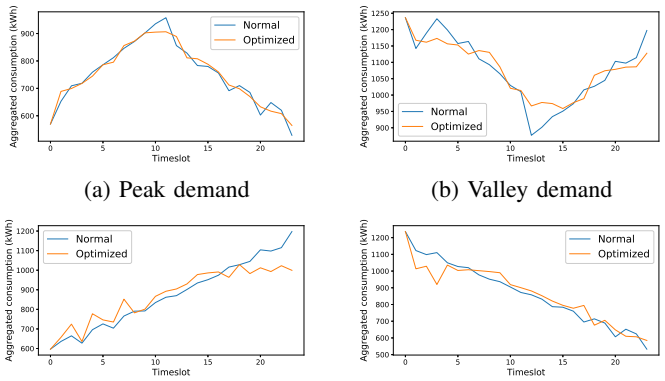
In the next section we present an evaluation setup and discuss the algorithm's performance with respect to various metrics and design choices.

V. Simulation Setting and Results

We considered a setting with $|\mathcal{N}|=50$ buildings. The consumption of each building is discretized into $|\mathcal{L}_n|=10$ levels with equal distance amongst them. Assuming buildings of an average living area of 500 sq. meters, a base consumption e_n^0 for each building is chosen randomly from $[5,15]$ kWh, based on the average consumption of European buildings [32]. The consumption of level l is set to $\mathrm{e}_n^l=\mathrm{e}_n^0+l\cdot\mathrm{e}_n^0\cdot 0.2$. The values of parameters $\beta_{n,t},\delta_{n,t},\omega_{n,t},\eta_{n,t}$ are generated by random normal distributions (the same distributions for all buildings and timeslots), depending on the chosen action, as presented in Table I[2]. The cost function parameters were set as $\mathrm{w}_1=0.02,\mathrm{w}_2=0.004$. All experiments were run in a i5-7300U CPU, 2.60GHz laptop computer with 8GB of RAM, using the CPLEX solver and the Pyomo environment.

In the first simulation, the algorithm was run in rolling horizon with a look-ahead period of $\mathfrak{T}=10$ timeslots and for different demand patterns. The different demand patterns are generated by changing the values of parameters $\zeta_{n,t}$ and $\gamma_{n,t}$ (i.e. the percentage of consumption that continues from t to $t+1$ and the amount of new demand). In Fig. 3, we present the system's aggregated energy consumption, under

[2]In this paper we have assumed that statistical information for each building's consumption behavior is available. In a real system, the prerequisite step would be to estimate the MDP parameters using historical data for each building. A method on how to derive the values for the MDP parameters, based on measurement data, is presented in [30].



(a) Peak demand (b) Valley demand

(c) Increasing demand (d) Decreasing demand

Fig. 3: Aggregated energy consumption throughout the horizon for different demand patterns.

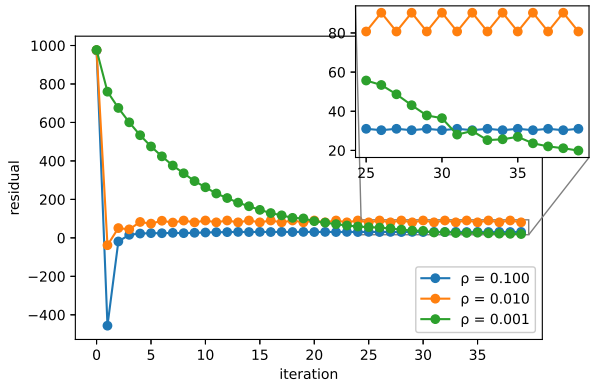


Fig. 4: Convergence behavior under different values of ρ.

the always normal operation and under the optimal decisions, throughout the horizon T. Four cases were considered for the normal demand: peak demand in the middle of the horizon (upper left), demand valley in the middle of the horizon (upper right), continually increasing demand (lower left), continually decreasing demand (lower right). We can observe the algorithm's tendency to flatten the aggregated consumption. For the rest of the experiments the peak demand case was used.

The respective consumption of a single building demonstrates similar patterns to the ones of Fig. 3, although the consumption of buildings with higher values of ω_n tends to be closer to their normal consumption, while the consumption of buildings with higher values of η_n exhibits more abrupt oscillations, since the algorithm avoids to put the building in a restricted mode for two timeslots in a row, resulting in an immediate rebound effect.

The number of iterations needed for the ADMM algorithm to converge, depends on the value of parameter ρ. In Fig. 4, we present the value of the residual

$$\max_{t\in[0,\mathcal{T}]}\left\{\left|\sum_{n\in\mathcal{N}}\sum_{s\in\mathcal{S}_n}\sum_{a\in\mathcal{A}_n} x^{(k)}_{n,t,s,a}\cdot \mathrm{e}^l_n - \widetilde{E}^{(k)}_t\right|\right\}$$

as a function of the iteration k for different cases of ρ. The case of $\rho = 0.001$ is always convergent when used with a tolerance of $\varepsilon = 1$.

By leveraging the optimality guarantees of the ADMM, the proposed method is able to achieve the optimal solution. Nevertheless, two factors can cause inefficiencies in practice: inaccurate assessment of the system's transition function, and a limited look-ahead horizon. In the next experiment we evaluate the method's performance under inaccurate transition functions and for different lengths of the look-ahead horizon.

Towards testing the sensitivity/robustness of the method to forecast inaccuracies, we use an error factor, such that the parameters $\widetilde{\beta}_{n,t}, \widetilde{\gamma}_{n,t}, \widetilde{\delta}_{n,t}, \widetilde{\zeta}_{n,t}$ that are used to build the estimated transition function, differ from the actual ones within the error factor. For example, if the error is 10%, then the algorithm is executed with a value $\widetilde{\beta}_{n,t}$ that is chosen randomly between $0.9\beta_{n,t}$ and $1.1\beta_{n,t}$. The same applies to all four parameters. While the decisions are taken using the resulting (altered) transition function, the actual transitions happen based on the true transition function. We tested the system for errors of 10%, 20% and 30%. With respect to the chosen number of look-ahead timeslots $\mathcal{T}$, it is subject to a trade-off between the method's optimality and the method's computational time. For shorter look-ahead horizons the decisions are more myopic (i.e. suboptimal).

The results for different error factors and various lengths for the look-ahead horizon are presented in Fig. 5. The case of zero error and a full look-ahead $\mathcal{T} = |\mathcal{T}|$ represents the optimal solution. Note that for $\mathcal{T} = 1$, the proposed algorithm reduces to the myopic approach that is often undertaken in the literature (e.g. [3]). Therefore, the figure demonstrates a comparison between the proposed method and the typical myopic approach, showcasing the cost savings gained by the proposed method's consideration of future uncertainty. As observed by the figure, a 10% error has a very small effect on the method's optimality. The reason is that a small error is usually not enough to alter the decided actions. Nevertheless, higher levels of inaccuracy can negatively impact the method's optimality. Interestingly, for an error factor of 30% a longer look-ahead horizon does not help anymore, and it may even have a negative impact. The reason is that, with high levels of inaccuracy, a longer look-ahead horizon propagates the accumulated error of more timeslots, causing higher amounts of inaccuracy. For more accurate forecasts, though, the longer the look-ahead horizon, the better the method's performance.

On the other hand, increasing the look-ahead period increases the complexity of local problems (29), resulting in higher computational times per algorithm iteration. In Fig. 6, we present the average computational time needed for a decision in a given timeslot t, as a function of the number of buildings, and for various cases of the look-ahead horizon length. As observed, the algorithm scales well with the number of buildings, which is expected since the local problems (29) are solved in parallel. With a longer look-ahead horizon, the computational time naturally increases, although it remains within acceptable levels for power systems applications.

VI. Conclusions

In this paper, we modeled the energy consumption behavior of a smart building as a Markov Decision Process.

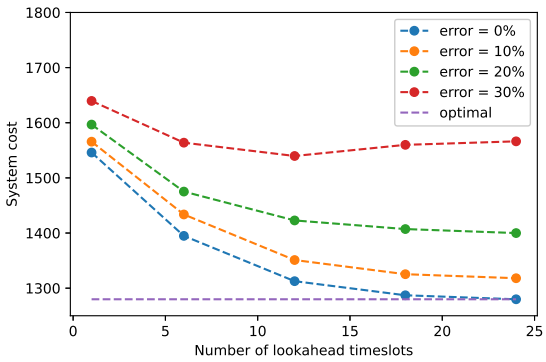


Fig. 5: System cost as a function of the look-ahead horizon length for different cases of forecast accuracy.

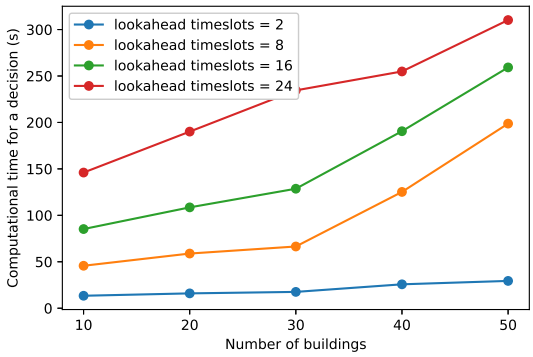


Fig. 6: Scalability for different horizon lengths.

The considered problem was the coordination of multiple smart buildings' energy management under uncertainty and partial consumption controllability. An iterative algorithm was proposed for making globally optimal local decisions under uncertainty. We compared the proposed approach with the myopic approach that is often adopted in the literature and in practice. Our results demonstrate the cost-effectiveness of taking future uncertainties into account and quantify the trade-off between cost-effectiveness and computational time, as a function of the look-ahead horizon length.

Among the strengths of the proposed algorithm is the ability to scale well to large numbers of buildings and the compatibility with the characteristics of smart buildings configured with internet-of-things solutions for energy management. On the other hand, the method necessitates the availability of statistical information over the buildings' transition dynamics. Such information, however, is easy to derive if there is an extended network of sensors deployed. A threat that one need to be aware of, is the sensitivity of the method's performance with respect to inaccuracies in the modeling of the building's transition dynamics. Overall, we believe that the low cost of new equipment for smart buildings, combined with our promising results towards practical and cost-effective building coordination, could provide an important opportunity towards large-scale development of demand-side flexibility.

Future work can integrate the proposed method with appliance-level models of building energy consumption, while an important milestone is to test the proposed technique in hardware-in-the-loop simulations, using actual IoT-configured smart buildings.

REFERENCES

[1] X. Jin, Q. Wu, and H. Jia, "Local flexibility markets: Literature review on concepts, models and clearing methods," Applied Energy, vol. 261, p. 114387, 2020.

[2] G. Tsaousoglou, J. S. Giraldo, and N. G. Paterakis, "Market mechanisms for local electricity markets: A review of models, solution concepts and algorithmic techniques," Renewable and Sustainable Energy Reviews, vol. 156, p. 111890, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032121011576

[3] P. Samadi, A.-H. Mohsenian-Rad, R. Schober, V. W. Wong, and J. Jatskevich, "Optimal real-time pricing algorithm based on utility maximization for smart grid," in 2010 First IEEE International Conference on Smart Grid Communications. IEEE, 2010, pp. 415–420.

[4] S. Moon and J.-W. Lee, "Multi-residential demand response scheduling with multi-class appliances in smart grid," IEEE Transactions on Smart Grid, vol. 9, no. 4, pp. 2518–2528, 2018.

[5] N. G. Paterakis, O. Erdinc, A. G. Bakirtzis, and J. P. Catalao, "Optimal household appliances scheduling under day-ahead pricing and load-shaping demand response strategies," IEEE Transactions on Industrial Informatics, vol. 11, no. 6, pp. 1509–1519, 2015.

[6] G. Tsaousoglou, K. Steriotis, N. Efthymiopoulos, P. Makris, and E. Varvarigos, "Truthful, practical and privacy-aware demand response in the smart grid via a distributed and optimal mechanism," IEEE Transactions on Smart Grid, vol. 11, no. 4, pp. 3119–3130, 2020.

[7] G. Tsaousoglou, P. Pinson, and N. G. Paterakis, "Transactive energy for flexible prosumers using algorithmic game theory," IEEE Transactions on Sustainable Energy, 2021.

[8] W. Liu, Q. Wu, F. Wen, and J. Østergaard, "Day-ahead congestion management in distribution systems through household demand response and distribution congestion prices," IEEE Transactions on Smart Grid, vol. 5, no. 6, pp. 2739–2747, 2014.

[9] G. Tsaousoglou, P. Soumplis, N. Efthymiopoulos, K. Steriotis, A. Kretsis, P. Makris, P. Kokkinos, and E. Varvarigos, "Demand response as a service: Clearing multiple distribution-level markets," IEEE Transactions on Cloud Computing, 2021.

[10] G. Tsaousoglou, J. S. Giraldo, P. Pinson, and N. G. Paterakis, "Mechanism design for fair and efficient dso flexibility markets," IEEE Transactions on Smart Grid, vol. 12, no. 3, pp. 2249–2260, 2021.

[11] S. Minniti, A. Haque, N. Paterakis, and P. Nguyen, "A hybrid robust-stochastic approach for the day-ahead scheduling of an ev aggregator," in 2019 IEEE Milan PowerTech. IEEE, 2019, pp. 1–6.

[12] N. Neyestani, M. Yazdani-Damavandi, M. Shafie-Khah, G. Chicco, and J. P. Catalão, "Stochastic modeling of multienergy carriers dependencies in smart local networks with distributed energy resources," IEEE Transactions on Smart Grid, vol. 6, no. 4, pp. 1748–1762, 2015.

[13] J. L. Crespo-Vazquez, T. AlSkaif, Á. M. González-Rueda, and M. Gibescu, "A community-based energy market design using decentralized decision-making under uncertainty," IEEE Transactions on Smart Grid, vol. 12, no. 2, pp. 1782–1793, 2020.

[14] J. Kazempour, P. Pinson, and B. F. Hobbs, "A stochastic market design with revenue adequacy and cost recovery by scenario: Benefits and costs," IEEE Transactions on Power Systems, vol. 33, no. 4, pp. 3531–3545, 2018.

[15] A. J. Conejo, M. Carrión, J. M. Morales et al., Decision making under uncertainty in electricity markets. Springer, 2010, vol. 1.

[16] J. S. Giraldo, M. Salazar, P. P. Vergara, G. Tsaousoglou, J. Slootweg, and N. G. Paterakis, "Optimal operation of community energy storage using stochastic gradient boosting trees," in 2021 IEEE Madrid PowerTech, 2021, pp. 1–6.

[17] M. K. Singh, S. Gupta, V. Kekatos, G. Cavraro, and A. Bernstein, "Learning to optimize power distribution grids using sensitivity-informed deep neural networks," in 2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). IEEE, 2020, pp. 1–6.

[18] G. Tsaousoglou, K. Mitropoulou, K. Steriotis, N. G. Paterakis, P. Pinson, and E. Varvarigos, "Managing distributed flexibility under uncertainty by combining deep learning with duality," IEEE Transactions on Sustainable Energy, 2021.

[19] B.-G. Kim, Y. Zhang, M. van der Schaar, and J.-W. Lee, "Dynamic pricing and energy consumption scheduling with reinforcement learning," IEEE Transactions on Smart Grid, vol. 7, no. 5, pp. 2187–2198, 2016.

[20] X. Xu, Y. Jia, Y. Xu, Z. Xu, S. Chai, and C. S. Lai, "A multi-agent reinforcement learning-based data-driven method for home energy management," IEEE Transactions on Smart Grid, vol. 11, no. 4, pp. 3201–3211, 2020.

[21] H.-M. Chung, S. Maharjan, Y. Zhang, and F. Eliassen, "Distributed deep reinforcement learning for intelligent load scheduling in residential smart grids," IEEE Transactions on Industrial Informatics, vol. 17, no. 4, pp. 2752–2763, 2020.

[22] D. Minoli, K. Sohraby, and B. Occhiogrosso, "Iot considerations, requirements, and architectures for smart buildings—energy optimization and next-generation building management systems," IEEE Internet of Things Journal, vol. 4, no. 1, pp. 269–283, 2017.

[23] M. Collotta and G. Pau, "A novel energy management approach for smart homes using bluetooth low energy," IEEE Journal on Selected Areas in Communications, vol. 33, no. 12, pp. 2988–2996, 2015.

[24] M. Shakeri, M. Shayestegan, S. S. Reza, I. Yahya, B. Bais, M. Akhtaruzzaman, K. Sopian, and N. Amin, "Implementation of a novel home energy management system (hems) architecture with solar photovoltaic system as supplementary source," Renewable energy, vol. 125, pp. 108–120, 2018.

[25] X. Zhang, M. Pipattanasomporn, T. Chen, and S. Rahman, "An iot-based thermal model learning framework for smart buildings," IEEE Internet of Things Journal, vol. 7, no. 1, pp. 518–527, 2019.

[26] B. Yuce, Y. Rezgui, and M. Mourshed, "Ann–ga smart appliance scheduling for optimised energy management in the domestic sector," Energy and Buildings, vol. 111, pp. 311–325, 2016.

[27] D. Azuatalam, W.-L. Lee, F. de Nijs, and A. Liebman, "Reinforcement learning for whole-building hvac control and demand response," Energy and AI, vol. 2, p. 100020, 2020.

[28] K. Mason and S. Grijalva, "A review of reinforcement learning for autonomous building energy management," Computers & Electrical Engineering, vol. 78, pp. 300–312, 2019.

[29] A. Saha, M. Kuzlu, and M. Pipattanasomporn, "Demonstration of a home energy management system with smart thermostat control," in 2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT). IEEE, 2013, pp. 1–6.

[30] R. Pop, A. Hassan, K. Bruninx, M. Chertkov, and Y. Dvorkin, "A markov process approach to ensemble control of smart buildings," in 2019 IEEE Milan PowerTech. IEEE, 2019, pp. 1–6.

[31] D. P. Bertsekas et al., Dynamic programming and optimal control: Vol. 1. Athena scientific Belmont, 2000.

[32] [Online]. Available: https://ec.europa.eu/energy/content/energy-consumption-m%C2%B2_en

Georgios Tsaousoglou Dr. Georgios Tsaousoglou received his PhD from National Technical University of Athens (NTUA) in 2019. He then joined the Greek Transmission System Operator as an electricity markets expert. Soon after, he became a postdoctoral researcher and Marie Curie Fellow in Eindhoven University of Technology. As of January 2022, he is a senior researcher at National Technical University of Athens. His research interests include decision making under uncertainty, multiagent systems, and algorithmic game theory, applied to the areas of electricity markets, demand response and power systems.

Nikolaos Efthymiopoulos Dr. Nikolaos Efthymiopoulos is currently a senior researcher at National Technical University of Athens, Greece. Since 2010 he holds a PhD degree in Computer Science. His research activities span around: smart grids, energy markets, computer networks, social networks, peer- to-peer, optimization, theory of dynamical systems, game theory/auctions.

Prodromos Makris Dr. Prodromos Makris is currently a senior researcher at National Technical University of Athens (NTUA). He holds a PhD (2013) from University of the Aegean, Greece. From 2017, he has served as technical coordinator for H2020- SOCIALENERGY and H2020-FLEXGRID projects. From 2020, he also serves as Adjunct Lecturer in University of the Aegean.

Emmanouel Varvarigos Prof. Emmanouel Varvarigos received his Ph.D. degree in Electrical Engineering and Computer Science from the MIT, Cambridge, MA in 1992. In 2015, he joined as a Full Professor the School of Electrical and Computer Engineering of the National Technical University of Athens (NTUA/ICCS). He is the coordinator of VIMSEN and SOCIALENERGY, which are HORIZON2020 projects relevant with the exploitation of energy data analytics and optimization towards the development of ICT platforms that develop DSM (DR) services and flexibility markets towards energy efficiency.