



**HAL**  
open science

# A Multi-Agent System for Energy Management in a Dynamic and Open Environment: Architecture and Optimisation

Oudom Kem, Feirouz Ksontini

► **To cite this version:**

Oudom Kem, Feirouz Ksontini. A Multi-Agent System for Energy Management in a Dynamic and Open Environment: Architecture and Optimisation. IEEE/WIC/ACM International Conference on Web Intelligence (WI), Oct 2019, Thessaloniki, Greece. pp.348-352, 10.1145/3350546.3352545 . hal-04573500

**HAL Id: hal-04573500**

**<https://hal.science/hal-04573500>**

Submitted on 15 May 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Multi-Agent System for Energy Management in a Dynamic and Open Environment: Architecture and Optimisation

Oudom Kem

CEA, LIST, 91191 Gif-sur-Yvette cedex, France  
oudom.kem@cea.fr

Feirouz Ksontini

CEA, LIST, 91191 Gif-sur-Yvette cedex, France  
feirouz.ksontini@cea.fr

## ABSTRACT

The increasing presence of smart and dynamic environments with IoT devices creates new challenges in energy optimisation such as handling environments' dynamics and privacy concerns. In this paper, we aim at optimising energy consumption in such environments, exploiting building flexibility to reduce energy bills, while respecting user preferences as well as device constraints and addressing the complexity of the environment. We propose a multi-agent optimisation system based on Alternating Direction Method of Multipliers to solve the optimisation problem. Various evaluations of the proposal show significant energy cost savings, while addressing dynamics and preserving privacy.

## KEYWORDS

Energy optimisation, MAS, smart buildings, IoT, dynamics, privacy

## 1 INTRODUCTION

These days, the shifting towards smart buildings along with Internet of Things (IoT), while offering numerous advantages, presents also challenges. Smart buildings are not limited to residential buildings, but cover a wide range of building types from connected industrial buildings to smart transit stations. Such buildings are dynamic and open in the sense that devices may enter or exit the environments at any time. They may contain multiple users, each owning a set of devices and having different constraints. Devices may be activated or deactivated at by their owners, resulting in inclusion in or exclusion from the optimisation. Moreover, privacy, one of the main concerns in IoT [13], needs to be addressed, especially in the presence of multiple users. In this paper, we address energy optimisation in such complex and multi-user environments, handling the dynamics, openness, and privacy concern.

Many algorithms and techniques have been proposed for building energy optimisation. Some are based on artificial neural network [3, 9], multi-agent system [7], distributed model predictive [11], ant colony optimisation [5], and reinforcement learning [10], to name a few. The existing works address specific aspects of the optimisation. Most of the works focus on thermal comfort, user preferences, and energy saving. To the best of our knowledge, hardly any consider the dynamics and privacy issue in the optimisation.

An environment may contain a significant number of devices, each with its own dynamic constraints, objectives, and preferences provided by its user. Performing an optimisation for such an environment over a time horizon entails dealing with a large number of variables, making it computationally impractical to solve in a centralised manner [8]. Solving complex problems using a distributed technique has become prevalent in the literature [2, 6, 8, 12]. The algorithm in [8] significantly parallelises the problem to be executed by solver agents. In addition, it uses the approach commonly

known as Lagrangian relaxation, which preserves the privacy of the cost function and local constraints of each agent. Advances in decomposition methods such as alternating direction method of multipliers (ADMM) [1, 4] have been applied to solve optimisation of energy flow due to their robustness and privacy-preserving features. In this paper, we propose a multi-agent energy management system (MEM) based on ADMM to solve energy optimisation in a distributed fashion. Agents representing the devices perform local optimisation with local data, avoiding any privacy issue. An inherently distributed, dynamic, and open system itself, MEM proves efficient in handling dynamics and distribution of its components.

The rest of the paper is organised as follows. First, we present the context of the work. Second, a concise description of the distributed optimisation is provided. Third, we present our main contribution, MEM. Fourth, different evaluations of the proposal are presented. Finally, we discuss and conclude the paper.

## 2 CONTEXT

This work addresses energy management in environments equipped with a wide range of energy-consuming/producing devices (e.g., household appliances, photovoltaic, and local generators). We consider both fixed devices and controllable devices (i.e. controllable consumption/production) providing some sort flexibility.

Let  $U$  be a set of users in an environment  $E$ ,  $D$  a set of energy-consuming/producing devices,  $C$  a set of device constraints, and  $UPref$  user preferences on how each device should operate. Then, an environment at a given time  $t$  is defined as follows:

$$E_t = (U_t, D_t, C_t, UPref_t, S_t) \quad (1)$$

where  $E_t$  represents the environment  $E$  at time  $t$ ,  $U_t$  a set of users in  $E_t$ ,  $D_t$  a set of devices in  $E_t$  and considered in the optimisation,  $C_t$  a set of device constraints,  $UPref_t$  a set of user preferences, and  $S_t$  a set of external energy sources. We separate static devices  $SD_t \subset D_t$  whose state is constant from dynamic devices  $DD_t \subset D_t$  whose state may evolve over time. A device  $d \in D_t$  has a set of constraints  $C_d \subset C_t$ . A user  $u \in U_t$  owns a set of devices  $D_u \subset D_t$ , and has a set of preferences on their devices  $UPref_u \subset UPref_t$ . For each device they own  $d_u \in D_u$ , the user may configure  $d_u$  based on their preferences  $UPref_u^d \subset UPref_u$ .

## 3 DISTRIBUTED OPTIMISATION

The devices, whose energy consumption or production profile is to be optimised, are connected at the building level, forming a network of devices. Optimising energy flow in such a network is to minimise the network objective function subject to the constraints of each device in the network. We model such a network as an energy coordination network [8] composed of a set of terminals  $T$ , a set of devices  $D$ , and a set of nets  $N$ . A terminal models a transfer point

through which the energy flows between a device and a net. A net represents an exchange zone that constrains the energy schedules of its associated devices. Each device and each net is associated with a set of terminals.

Each terminal  $t \in T$  has a schedule  $p_t = ((p_t(1), \dots, p_t(H)) \in \mathbb{R}^H$  over a time horizon  $H \in \mathbb{N}^+$  (e.g., 24 hours). Then,  $p_t(\tau)$  where  $\tau \in [1, H]$  is the amount of energy consumed ( $p_t(\tau) > 0$ ) or generated ( $p_t(\tau) < 0$ ) by device  $d \in D$  in time period  $\tau$  through terminal  $t$ , where  $t$  is associated with  $d$ . For each device  $d \in D$ , we use ' $d$ ' to refer to both the devices and the set of terminals associated with the device. Each device  $d \in D$  has a set of energy schedules denoted by  $p_d = \{p_t | t \in d\}$ , possesses a set of  $|d|$  terminals and has an objective function  $f_d : \mathbb{R}^{|d| \times H} \rightarrow \mathbb{R}$ . Then,  $f_d(p_d)$  is the cost of operating device  $d$  according to the schedule  $p_d$ . Every device has a set of constraints  $C_d$  which  $p_d$  must satisfy. Similarly, each net  $n \in N$  has a set of energy schedules  $p_n = \{p_t | t \in n\}$ , a set of  $|n|$  terminals, an objective function  $f_n : \mathbb{R}^{|n| \times H} \rightarrow \mathbb{R}$ , and a set of constraints  $C_n$  to satisfy. Provided an energy coordination network, we define the optimisation problem as follows:

$$\begin{aligned} \min_{p \in \mathbb{R}^{H \times |T|}} & \sum_{d \in D} f_d(p_d) + \sum_{n \in N} f_n(p_n) \\ \text{subject to} & p_d \in C_d, \forall d \in D \\ & p_n \in C_n, \forall n \in N \end{aligned} \quad (2)$$

To solve the optimisation problem specified in Equation 2, we implemented a solution based on ADMM. ADMM iteratively solves the problem until the convergence is reached. In each iteration, ADMM performs the following steps:

Step 1: Device-minimisation executed in parallel by each device

$$\forall d \in D, \hat{p}_d^{k+1} = \underset{p_d \in C_d}{\operatorname{argmin}} (f_d(p_d) + \frac{\rho}{2} \|p_d - \hat{p}_d^k + v_d^k\|_2^2) \quad (3)$$

Step 2: Net-minimisation executed in parallel by each net

$$\forall n \in N, \hat{p}_n^{k+1} = \underset{p_n \in C_n}{\operatorname{argmin}} (f_n(p_n) + \frac{\rho}{2} \|p_n - \hat{p}_n^k + v_n^k\|_2^2) \quad (4)$$

Step 3: Price update executed in parallel by each net

$$\forall n \in N, v_n^{k+1} = v_n^k + (p_n^{k+1} - \hat{p}_n^{k+1}) \quad (5)$$

First, each device computes in parallel its best response to the price and energy requested by nets. At any iteration  $k + 1$ ,  $p_d^{k+1}$  represents the device's response to the request  $\langle \hat{p}_d^k + v_d^k \rangle$ . Second, upon receiving the offers from all the devices connected to it, each net checks if the convergence has been reached. If there is no convergence, nets compute a new request for the devices considering the devices' previous offers and send the new request to the devices. Third, nets update the scaled dual variables (refer to [8] for further details on ADMM). The cost functions and constraints of the devices differ according to their types, and are presented in Section 4.3.

## 4 MULTI-AGENT ENERGY MANAGEMENT

MEM is composed of agents assuming different roles required to carry out the optimisation, modelled as follows:

$$MEM = (NA, DA) \quad (6)$$

where  $NA$  is the model of net agents assuming the role of the nets in the energy coordination network and  $DA$  models the static device

agents representing the static devices as well as the dynamic device agents representing the dynamic devices. The role of an agent determines the knowledge it possesses, the actions it can perform, and the behaviour it comports.

### 4.1 Net agents

The role of a net agent is to ensure that, in each time period, there is a balance between the energy flowing into and out of its terminals. Net agents are modelled as follows:

$$NA = (Opt_{net}, C_{net}, Comm_{net}) \quad (7)$$

where  $Opt_{net}$  contains the optimisation steps (equation 4 and equation 5) executed locally by a net agent,  $C_{net}$  represents the energy balance constraint upheld by a net, and  $Comm_{net}$  is the communication ability of the agent, enabling it to send and receive messages with other agents (e.g., functions required for sending, receiving, and processing messages). Each net agent iteratively computes the optimisation steps  $Opt_{net}$  until a convergence is reached, while ascertaining that  $C_{net}$  is respected.

### 4.2 Device agents

The role of a device agent is to perform the local optimisation, while respecting the constraints of its user(s) and the device. The optimisation result is communicated to its associated net agent(s). Formally, the model of a device agent is as follows:

$$DA = (Opt_{dev}, C_{dev}, Pref, I_{dev}, Comm_{dev}, Sen, State) \quad (8)$$

where  $Opt_{dev}$  is the device-minimisation step (equation 3) executed by the device agent (see 4.3),  $C_{dev}$  represents device constraints,  $Pref$  consists of user preferences regarding the manner in which the device may operate (e.g., the interval in which the device is allowed to be switched on or off),  $I_{dev}$  refers to additional information required by the device agent to perform the optimisation (e.g., energy prices),  $Comm_{dev}$  is the communication capacity of the agent required for communicating with its associated net agent(s),  $Sen$  is the agent's ability (e.g., functions to retrieve data directly from sensors or from data stores) to acquire the current state of the device, and  $State$  models the information concerning the device's state as part of the agent's knowledge.

Each dynamic device agent needs to consider the current state of the device in its optimisation. For instance, the agent is provided access to the sensor installed on the device for which the agent is responsible. In this way, the agent can retrieve data from the sensor to monitor the state of the device. For static device agents,  $Sen$  and  $State$  are irrelevant. Each device agent executes its device-minimisation step iteratively considering the updates from its associated net(s).

### 4.3 Device optimisation model

**4.3.1 Fixed load (FL).** FL [8] models a device whose power consumption profile must be satisfied. Fixed loads have a zero cost function. Their constraint  $c_{FL} \in C_{dev}$  ensures that the required consumption  $p_{req}$  is satisfied in each time period, formally:  $p_{FL}(\tau) = p_{req}(\tau)$ ,  $\tau = 1, \dots, H$  where  $p_{FL}$  is the actual consumption of FL.

**4.3.2 Shiftable load (SL).** SL [8] models a device that must consume a certain amount of energy (i.e., the volume) within a given

time interval. The interval allowed to activate the device is a user preference  $pref_{SL} \in Pref$  and imparted to the device agent as part of its knowledge. SL has a zero cost function. It encodes a hard constraint  $c_{SL} \in C_{dev}$  mandating that the required amount of consumption  $V$  is satisfied between an earliest time period  $A$  and latest time period  $D$ , formally:  $\sum_{\tau=A}^D p_{SL}(\tau) = V$  where  $p_{SL}$  is the actual consumption of SL.

**4.3.3 Sheddable load (ShL).** ShL [8] models a device whose consumption can be shed at a cost when there is a need to reduce demands (e.g., lighting system - on/off or selecting a level of brightness). The cost function of sheddable loads considers inconvenience  $p_{inc} \in Pref$  resulting from the shedding, formally:  $f_{ShL}(\tau) = p_{inc}(\tau) * (p_{req}(\tau) - p_{ShL}(\tau))$  where  $p_{req}$  is the required consumption and  $p_{ShL}$  the actual consumption after shedding. ShL's constraint  $c_{ShL} \in C_{dev}$  ensures that the consumption after shedding does not surpass the required consumption, formally:  $0 \leq p_{ShL}(\tau) \leq p_{req}(\tau)$ ,  $\tau = 1, \dots, H$ .

**4.3.4 Curtailable generator (CG).** CG models an energy source whose production can be curtailed (e.g., photovoltaic). The cost function of curtailable generators incorporates a curtailment cost  $p_{curt} \in Pref$ , formally:  $f_{CG}(\tau) = p_{curt}(\tau) * (p_{gen}(\tau) - p_{CG}(\tau))$  where  $p_{gen}$  is the expected production and  $p_{CG}$  the production after curtailment.

**4.3.5 External tie.** ET [8] models a connection to an external source of energy. Transactions with ET consist in pulling energy from the source or injecting energy to it. Its cost function factors in the prices of importing  $p^{imp} \in I_{dev}$  and exporting  $p^{exp} \in I_{dev}$  energy:

$$\begin{aligned} p_{ET}(\tau) < 0, f_{ET}(\tau) &= P^{exp} * p_{ET}(\tau) \\ p_{ET}(\tau) = 0, f_{ET}(\tau) &= 0 \\ p_{ET}(\tau) > 0, f_{ET}(\tau) &= P^{imp} * p_{ET}(\tau) \end{aligned} \quad (9)$$

where  $p_{ET}$  is the amount of pulled (positive value) or injected (negative value). Its constraint  $c_{ET} \in C_{dev}$  restricts importing and exporting energy by some specified limit  $P^{max}$ , formally:  $|p_{ET}(\tau)| \leq P^{max}$ ,  $\tau = 1, \dots, H$ .

## 4.4 Constraints, privacy, and dynamics

**4.4.1 Constraints.** Device constraints  $C_{dev}$  and the device user's preferences  $Pref$  are imparted to a device agent as a part of its knowledge when the agent is instantiated. While conducting its local optimisation, the agent ensures that the constraints and preferences are not violated.

**4.4.2 Privacy.** User preferences and device constraints are known only to the corresponding device agent. They are not shared with other agents. Moreover, employing ADMM enables the coordination exchanges in which the preferences, the constraints, and the cost structures of each device are reflected in its private cost function. More precisely, as described in Section 3, the coordinated messages  $p_d^{k+1}$  sent to the associated net agent are simply the representation of the device agent's response to the net agent's request  $\langle \hat{p}_d^k + v_d^k \rangle$ .

**4.4.3 Dynamics.** We address two types of dynamics: devices' evolving state and activation/deactivation of devices. First, to handle the

evolving state of the devices, device agents are equipped with the ability to acquire the information concerning the current state of the devices. This allows device agents to perform the optimisation with accurate information about the devices. Second, a device that is already registered in the system may be activated or deactivated at any time by its user, which results in including the device in or excluding it from the optimisation, respectively. To address this situation, the device agent responsible, upon having detected the activation or deactivation, informs its associated net agent of the event. Accordingly, the net agent includes the device in or excludes it from the optimisation.

The result of the optimisation is the consumption and/or production profile of each device for the next time horizon discretised into a number of time periods (e.g., 24 hours divided into 96 time periods of 15 minutes). During the horizon in which the optimised profiles are applied to the devices, changes impacting the environment and thus rendering the current profiles non-optimal may occur. The agents, having detected such changes, trigger a new optimisation process with the update-to-date state of the environment.

## 5 EVALUATIONS

The evaluations are conducted using a case study of a prosumer building with a connection to an energy supplier (i.e., external tie) and equipped with a photovoltaic (PV) for local uses. The time horizon for the experiments is 24 hours divided into 96 time periods (TP) of 15-minute interval. We use the consumption data from UK Elexon's non-domestic unrestricted customers<sup>1</sup>. In each scenario, the fixed consumption represents a certain percentage of the overall consumption, and a certain percentage is considered flexible. PV production is based on data from one of our projects. We use two-banded Time-Of-Use tariffs from EDF (i.e., an energy supplier in France) consisting of a peak price (between 7:00 and 23:00): €0.158/kWh and an off-peak price (between 23:00 and 7:00): €0.11/kWh. In the experiments, flexibility is concretely modelled by means of shiftable devices, which consume the amount of flexibility provided in each scenario.

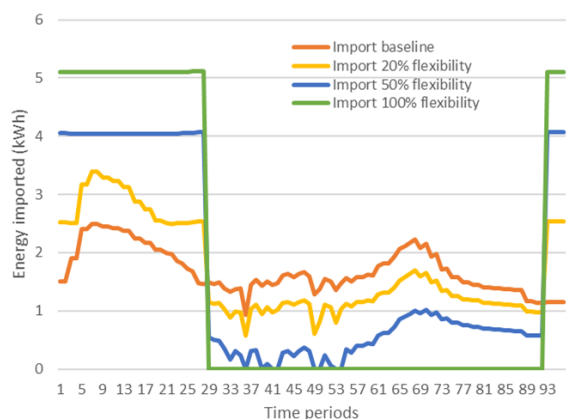
User preferences and device constraints are encoded in the corresponding device agents. It is important to note that each device agent is instantiated with the knowledge of its user preferences and device constraints, unknown to other agents, in order to preserve privacy. To address the dynamics and openness, the optimisation is executed iteratively every TP (i.e., 15 minutes). The pertinent changes to the devices or addition/removal of devices happening during the previous TP are detected by each device agent and taken into account in the next iteration.

### 5.1 Result analysis

In the first evaluation, we experiment with four different scenarios, with varying percentages of flexibility proportional to the total consumption of the building. The first scenario has no flexibility at all, second 20% flexibility, third 50%, and fourth 100%. Figure 1 depicts the energy imported for each of the scenarios. The imported energy is the energy pulled from the supplier via the external tie. Since the building also produces energy, the imported energy is the amount required after self-consumption using the energy from PV.

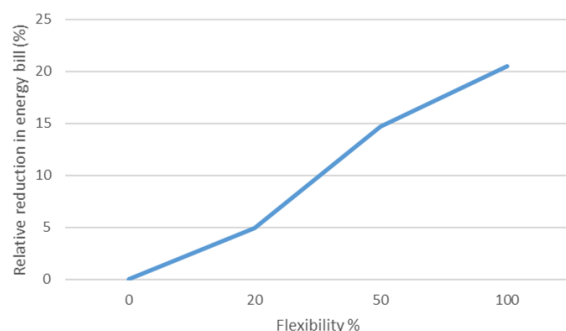
<sup>1</sup><https://www.elexon.co.uk/knowledgebase/profile-classes/>

The first scenario (shown as Import baseline) supposes that all the consumption is fixed. Therefore, the system is not able to optimise the consumption based on the energy tariffs. By introducing flexibility, the system can optimise the flexible consumption to benefit the off-peak price, the result of which can be seen in Figure 1. The imported energy during peak period between TP 29 (i.e., 7:00) and TP 91 (i.e., 23:00) decreases as the flexibility increases, with no import when the flexibility reaches 100%. The reason is that the system schedules the flexible consumption in the off-peak periods.



**Figure 1: Comparison of energy imported for different amounts of flexibility**

Figure 2 illustrates the reduction of energy bills as the amount of flexibility increases. In the best-case scenario, in which all the consumption is flexible and can be shifted, the reduction attains over 20%. In a more realistic scenario where approximately 20%<sup>2</sup> of the consumption is shiftable, the energy bill decreases roughly 5%.



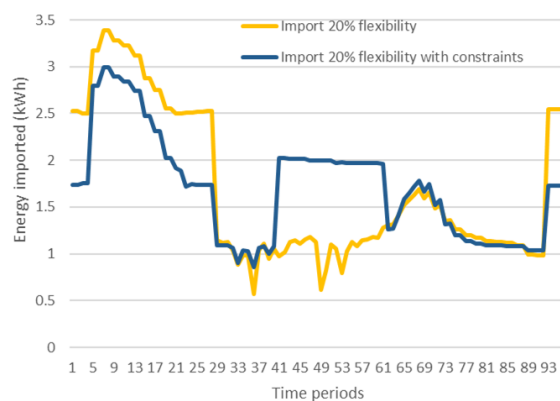
**Figure 2: Reduction of energy bills**

It is noteworthy that, in the aforementioned results, all the flexible consumption can be scheduled freely by the system without any constraints. In reality, it is common for users to specify the interval in which the shiftable consumption is allowed to be scheduled. To evaluate the system's capacity to respect such preferences from users, in the second evaluation, we simulate user preferences on shiftable devices. For this evaluation, a set of shiftable devices are

<sup>2</sup>Different studies have shown that around 10-20% of demand can be time-shifted [14]

only allowed to be shifted within an interval between TP 41 and 61 (i.e., 10:00-15:00).

Figure 3 demonstrates the comparison of imported energy between the scenario of 20% flexibility without any constraint and that with the previously mentioned constraint. Without constraints, all the shiftable devices are scheduled to operate in off-peak periods, while, with the constraint, they are set to execute as specified in the constraint regardless of the energy price.



**Figure 3: Satisfying user preferences on shiftable loads**

## 6 DISCUSSION AND CONCLUSION

Currently, each agent detects the changes and acts accordingly. These changes are taken into account in the next optimisation. A remaining question is when to carry out the next optimisation to consider the changes. A promising solution may be to determine which types of changes should trigger immediate re-execution of the optimisation and which types can wait for the next periodic iteration (e.g., every 15 minutes). Further investigation is necessary to determine the compatible mechanism for different types of environment dynamics.

It has been shown that the system is able to reduce energy bills up to over 20% in the best-case scenario, while respecting user preferences and device constraints. The distributed architecture of the system and the inherent characteristics of ADMM enable each agent's knowledge to be private, thus preserving privacy. Detecting and handling such dynamics and openness of the environment are carried out by the agents in cooperation. The system is designed to support dynamic changes of their components and to allow dynamic additions of new components.

## REFERENCES

- [1] Jonathan Eckstein. 1994. Parallel alternating direction multiplier decomposition of convex programs. *Journal of Optimization Theory and Applications* 80, 1 (1994), 39–62.
- [2] David A Fisher and Howard F Lipson. 1999. Emergent algorithms—a new method for enhancing survivability in unbounded systems. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. 1999. HICSS-32. Abstracts and CD-ROM of Full Papers*. IEEE, 10–pp.
- [3] Danilo Fuselli, Francesco De Angelis, Matteo Boaro, Stefano Squartini, Qinglai Wei, Derong Liu, and Francesco Piazza. 2013. Action dependent heuristic dynamic programming for home energy resource scheduling. *International Journal of Electrical Power & Energy Systems* 48 (2013), 148–160.

## Multi-Agent System for Energy Management

- [4] Daniel Gabay and Bertrand Mercier. 1975. *A dual algorithm for the solution of non linear variational problems via finite element approximation*. Institut de recherche d'informatique et d'automatique.
- [5] Azzaya Galbazar, Safdar Ali, and D Kim. 2016. Optimization approach for energy saving and comfortable space using aco in building. *International Journal of Smart Home* 10, 4 (2016), 47–56.
- [6] Balho H Kim and Ross Baldick. 2000. A comparison of distributed optimal power flow algorithms. *IEEE Transactions on Power Systems* 15, 2 (2000), 599–604.
- [7] Laura Klein, Jun-young Kwak, Geoffrey Kavulya, Farrokh Jazizadeh, Burcin Becerik-Gerber, Pradeep Varakantham, and Milind Tambe. 2012. Coordinating occupant behavior for building energy and comfort management using multi-agent systems. *Automation in construction* 22 (2012), 525–536.
- [8] Matt Kraning, Eric Chu, Javad Lavaei, Stephen Boyd, et al. 2014. Dynamic network energy management via proximal message passing. *Foundations and Trends® in Optimization* 1, 2 (2014), 73–126.
- [9] Jin Woo Moon and Jong-Jin Kim. 2010. ANN-based thermal control models for residential buildings. *Building and Environment* 45, 7 (2010), 1612–1625.
- [10] June Young Park, Thomas Dougherty, Hagen Fritz, and Zoltan Nagy. 2019. LightLearn: An adaptive and occupant centered controller for lighting based on reinforcement learning. *Building and Environment* 147 (2019), 397–414.
- [11] Helton F Scherer, Manuel Pasamontes, Jose Luis Guzmán, JD Álvarez, E Camponogara, and JE Normey-Rico. 2014. Efficient building energy management using distributed model predictive control. *Journal of Process Control* 24, 6 (2014), 740–749.
- [12] Paul Scott and Sylvie Thiébaux. 2015. Distributed multi-period optimal power flow for demand response in microgrids. In *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*. ACM, 17–26.
- [13] Masood Shah and Abdul Salam Shah. 2016. Appraisal of the Most Prominent Attacks due to vulnerabilities in cloud computing. *Int. J. Grid Distrib. Comput* 9, 7 (2016), 13–22.
- [14] Smart Grid Program team. 2015. *The journey to Green Energy or a Quest for Flexibility*. Technical Report. Eandis. [https://www.edsoforsmartgrids.eu/wp-content/uploads/Vision-Paper\\_Journey-to-Green-Energy\\_Eandis.pdf](https://www.edsoforsmartgrids.eu/wp-content/uploads/Vision-Paper_Journey-to-Green-Energy_Eandis.pdf)