Optimal Scheduling of On/Off Cycles: A Decentralized IoT-Microgrid Approach

Fernando Lezama[1(✉)], Jorge Palominos[1], Ansel Y. Rodríguez-González[1], Alessandro Farinelli[2], and Enrique Munoz de Cote[1]

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica, 72840 Puebla, Mexico
{f.lezama,jpalominos,ansel,jemc}@inaoep.mx
[2] University of Verona, 37134 Verona, Italy
alessandro.farinelli@univr.it

Abstract. The current energy scenario requires actions towards the reduction of energy consumptions and the use of renewable resources. To this end, the energy grid is evolving towards a distributed architecture called Smart Grid (SG). Moreover, new communication paradigms, such as the Internet of Things (IoT), are being applied to the SG providing advanced communication capabilities for management and control. In this context, a microgrid is a self-sustained network that can operate connected to the SG (or in isolation). In such networks, the long-term scheduling of on/off cycles of devices is a problem that has been commonly addressed by centralized approaches. In this paper, we propose a novel IoT-microgrid architecture to model the long-term optimization scheduling problem as a distributed constraint optimization problem (DCOP). We compare different multi-agent DCOP algorithms using different window sizes showing that the proposed architecture can find optimal and near-optimal solutions for a specific case study.

Keywords: Multi-agent · Smart Grid · IoT · Microgrid · Optimization

1 Introduction

The current world scenario including global warming, increase in carbon emissions, and the growing world population and power demand has led to governments, energy utilities, and research centers to take concrete actions towards the reduction of energy consumptions and the use of renewable resources [1].

Due to this, the electric grid has evolved over the last decades to a highly automated energy network, widely known as Smart Grid (SG). The SG is an advanced power network that incorporates two-way communication for efficient control, reliability and safety [2]. Moreover, the SG abandons the centralized nature of the traditional electric grid towards a decentralized architecture in which the electricity is produced in a distributed way, and customers can be producers and consumers (i.e., prosumers) at the same time [3].

In this scenario the concept of microgrid provides a complementary solution to achieve more efficient energy management in small areas [4]. A microgrid is a

E. Sucar et al. (Eds.): AFI 2016, LNICST 179, pp. 79–90, 2017.
DOI: 10.1007/978-3-319-49622-1_10

small self-sustained power network, with local distribution and local generators that uses renewable energy (such as solar, wind, biomass, among others). Moreover, a *smart* microgrid can be seen as an independent home management system that uses a combination of the electric home network and the Internet to manage home appliances and local generators in an intelligent and efficient way. Both energy consumption and generation should be considered to save energy cost at the user end [5]. In addition, a key element for a microgrid is peer-to-peer communication and plug-and-play functionalities, allowing distributed control and scalability without significant modifications of the grid.

Different technologies used in other industrial applications, such as sensor or wireless networks, can be adopted for the communication of the devices within the SG. Nevertheless, to reduce the number of communication protocols and to handle a significant amount of data, the Internet of things (IoT) arises as one of the most recent enablers of the SG, and it is expected to continue playing a crucial role in the evolution of the SG [2,6].

Figure 1 shows a SG and a microgrid as a home management system. One of the main problems to tackle in a microgrid is the control and management of resources through the scheduling of on/off cycles of devices. A trivial solution can be to turn on the micro generators all the time and storage the exceeded energy in batteries. The main problem with this trivial solution is that life cycle of generators and batteries is drastically reduced, and there is a cost associated with excessive energy generation. For this reason, a more efficient solution to minimize the cost involves the scheduling of on/off cycles of the devices (also known as the dispatch problem) in the microgrid.

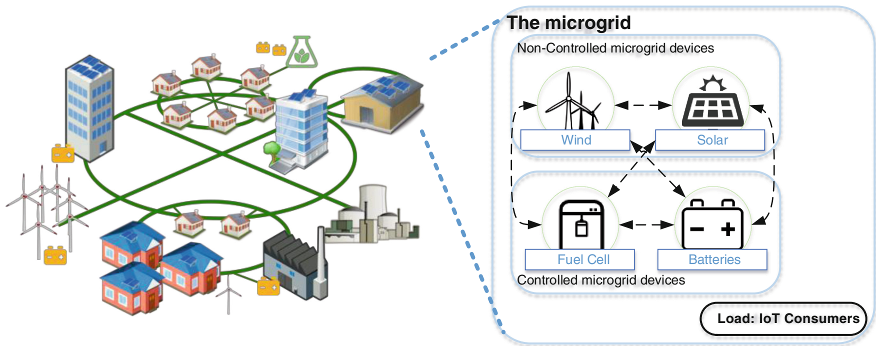

Fig. 1. Smartgrid and microgrid.

The optimal scheduling of on/off cycles of devices in the microgrids can be done in two ways, centralized and decentralized [7]. The centralized approach has the advantage of treating the system as a whole, hence allowing global optimization. However, a centralized approach lacks flexibility since adding new devices to the system implies the recalculation of the entire scheduling, and robustness because losing the central unit shut down all the system [8–10]. On the other hand, the decentralized approach is more flexible, allowing the addition of new devices and performing the optimization in a distributed way according to the tendency of intelligent distribution networks (i.e., the SG).

So far, the decentralized approach has been applied to the optimization scheduling for a particular time without considering long-term optimization [11–13]. However, recently several efforts have been devoted to extend the Distributed Constraint Optimization Problem (DCOP) model so to take system dynamics into consideration. The most prominent approach in this perspective is the Dynamic DCOP model (D-DCOP), where the system evolution is modeled as a sequence of canonical DCOPs computing a new solution each time the system changes (trying to re-use as much as possible the previous solution) [14].

While D-DCOP techniques introduce dynamism to DCOPs, they generally do not capture the sequential nature of the problem: they simply react to changes but do not plan for the best sequence of actions.

Moreover, in many real applications, we also need to take into account the uncertainty related to system dynamic. Recently Markov models have been used to capture the coupling aspect of D-DCOPs in which the DCOP in the next time step is a function of the value assignments in the current time step [15]. A major challenge with this interesting approach is that the problems become rapidly intractable when the size of the problem (e.g., the number of variables for the underlying DCOP) grows.

In this paper, we propose to solve optimal scheduling of on/off cycles for a home microgrid as a multi-agent decentralized approach explicitly considering long-term optimization into account. To this end, each controllable device is modeled as an independent agent with the ability of peer-to-peer communication with other devices (i.e., other agents) in the microgrid. Specifically, we model the problem as a DCOP and we use different off-the-shelf approaches to solve such problem[1].

In more detail, we consider Synchronous Branch and Bound (SynchBB), Distributed Pseudotree-Optimization Procedure (DPOP), Memory-Bounded DPOP (MB-DPOP) and Asynchronous Forward Bounding (AFB) [16], applied to the long-term optimization task.

We compare the performance of such algorithms against the optimal solution returned by a centralized approach. Moreover, we compare the run time, the number and the size of messages for the DCOP algorithms. To solve the long-term optimization problem while maintaining the model tractable, we split the problem into time windows. The results show that, even when the multi-agent distributed approach provides optimal and near-optimal solutions for small window sizes, it pays a large computational cost associated with the interaction of agents for large window sizes.

[1] We implemented the DCOP algorithms in FRODO2 and JaCoP. Both available in http://frodo2.sourceforge.net and http://www.jacop.eu respectively.

2 Problem Formulation

We model the microgrid as an IoT-microgrid architecture shown in Fig. 2. This IoT-microgrid architecture has two main blocks. The first block is composed of generators (G) and storage devices (S) (bottom right side of Fig. 2). The generators and storage devices can be grouped as controlled (e.g., fuel cell generators and batteries) and non-controlled (e.g., generators dependent on weather conditions) devices. The second block correspond to the Load formed by IoT consumers (C) (bottom left side of Fig. 2). Such IoT consumers can be different smart appliances, such as smart metering infrastructure, sensors, and other smart devices used for home automation.

The two blocks (i.e., generators and storage devices, and IoT consumers) are linked together and have a connection to external entities, such as the cloud and the SG respectively (upper side of Fig. 2). From the cloud, weather predictions and other information, such as temperature or electric load forecasting, can be retrieved from devices for management and control. Consumers and non-controlled devices provide information as input for the optimization task while the controllable devices can perform actions to modify the conditions of the IoT-microgrid.

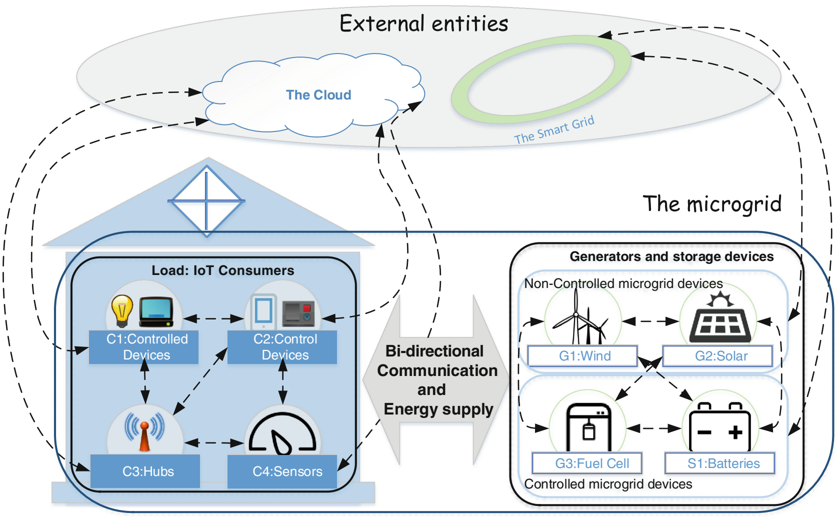

Fig. 2. IoT-microgrid architecture.

In the following, $P_{type,i}(t)$ refers to the production or quantity of energy provided by generators $i = \{1, 2..., N_{type}\}$ of some type of energy $type = \{1, 2..., N\}$ (e.g., wind or solar predictions provided by the cloud) at time t. $P_{BC,j}(t)$ and $P_{BD,j}(t)$ correspond to the energy charge/discharge status respectively of battery $j = \{1, 2, ..., M\}$ at time t. $P_{UE(t)}$ and $P_{EE(t)}$ is the amount of undelivered and exceeded energy at time t. Finally, C is the cost associated with the production or use of each energy.

Our objective is to minimize the cost of generated (E_{gen}) and storage energy ($E_{storage}$), while at the same time keep the balance between production and consume ($E_{disbalance}$) into the microgrid. The objective function is defined as:

$$Minimize \quad f = \sum_{t=1}^{T}(E_{gen}(t) + E_{storage}(t) + E_{disbalance}(t)) \tag{1}$$

where:

$$E_{gen}(t) = \sum_{type=1}^{N} \sum_{i=1}^{N_{type}} P_{type,i}(t) * C_{type,i}(t) \tag{2}$$

$$E_{storage}(t) = \sum_{j=1}^{M}(-P_{BC,j}(t) * C_{BC,j} + P_{BD,j}(t) * C_{BD,j}) \tag{3}$$

$$E_{disbalance}(t) = P_{UE}(t) * C_{UE} - P_{EE}(t) * C_{EE} \tag{4}$$

Subject to the following constraints:

– Kirchhoff law or power balance:

$$\sum_{type=1}^{N} \sum_{i=1}^{N_{type}} P_{type,i}(t) + \sum_{j=1}^{M} P_{BD,j}(t) + P_{UE} = Load(t) + \sum_{j=1}^{M} P_{BC,j}(t) + P_{EE}(t); \forall t \tag{5}$$

where $Load(t)$ is the energy required for all the consumers at time t. Prediction of the $Load$ can be provided by the cloud in the IoT-microgrid architecture.

– Energy type production limits at time t

$$P_{type,i}(t) \leq P_{lim_{type,i}}; \quad \forall type, i, t \tag{6}$$

– Storage, charge and discharge battery limits at each time t

$$P_{Storage,j}(t) \leq P_{lim_{Storage,j}}; \quad \forall j, t \tag{7}$$

$$P_{BD,j}(t) \leq P_{lim_{BD,j}} * \mathrm{X}(t); \quad \forall j, t, \mathrm{X} \in 0, 1 \tag{8}$$

$$P_{BC,j}(t) \leq P_{lim_{BC,j}} * \ \mathrm{Y}(t); \quad \forall j, t, \mathrm{Y} \in 0, 1 \tag{9}$$

where $P_{Storage,j}$ is the maximum power capacity of the jth battery, and X and Y are boolean variables used to avoid that the jth battery charge and discharge at the same time (i.e., $\mathrm{X}(t) + \mathrm{Y}(t) \leq 1; \forall t$).

– Charge and discharge limits at time t considering period $t-1$

$$P_{BD,j}(t) - P_{Storage,j}(t-1) \leq 0; \forall j, t \tag{10}$$

$$P_{BC,j}(t) + P_{Storage,j}(t-1) \leq P_{lim_{Storage,j}}; \forall j, t \tag{11}$$

– State balance of the battery

$$P_{Storage,j}(t) = P_{Storage,j}(t-1) - P_{BD,j}(t) + P_{BC,j}(t); \forall j, t \tag{12}$$

This formulation can be optimally solve using Mixed-integer linear programming in a centralized fashion [7]. Different from [7], in the next section we present a long-term decentralized approach that allows solving the problem using agents and DCOP algorithms.

3 Long-Term Multi-agent Optimization

As stated in Sect. 2, in this paper we consider a microgrid with consumers, generators and storage devices. For simplicity, in this section we consider a microgrid with only one solar generator, one wind generator, one fuel cell generator, and one storage device (i.e., one battery). We also assume perfect predictions of solar and wind generation. Moreover, the consumers are grouped all together as a joint *Load*, also known in advanced.

With these considerations, in Fig. 3 we present a distributed multi-agent model for the long-term optimization scheduling of on/off cycles. In this model, agents are the elements (e.g., the fuel cell generator (FC), battery charge (BC) and battery discharge (BD)) that can perform an on/off action to optimize the total cost of energy production in the long-term. These agents receive information of the *Load* and weather conditions (i.e., consumers and renewable energy generation) as input, and perform optimization in a distributed way.

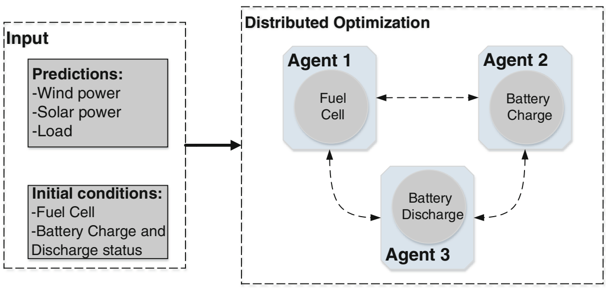

Fig. 3. Multi-agent distributed model.

The decentralized multi-agent architecture enables to treat the problem as a DCOP, allowing the use of distributed multi-agent algorithms such as AFB, DPOP, MB-DPOP, and SynchBB.

The multi-agent algorithms to solve DCOP (e.g., AFB, DPOP, MB-DPOP and SynchBB) distribute the processing among agents. However, optimally solving a DCOP is known to be an NP-complete problem, hence solving the long-term optimization problem directly will be impractical even for a short optimization horizon.

For this reason, we propose to split the problem in time windows. In this way, for a period T, we can solve the problem by dividing such period T in $N_{windows} = \mathrm{T}/n$, where n is the size of the window. This means that for a size

$n = 1$ we will solve T windows, for $n = 2$ we will solve T/2 windows and so on. Figure 4 shows the scheme of optimization in time windows. The input for the fist window corresponds to initial conditions of the microgrid (e.g., the battery could start with an initial charge of 100 W, and the fuel-cell generator in off state ready for being activated).

Then, optimization for that window is done by using any of the algorithms to solve the DCOP (e.g., AFB, DPOP, MB-DPOP and SynchSBB). The result obtained in that window is used as input for the next window. The process is repeated sequentially until a solution for the long-term period T is obtained.

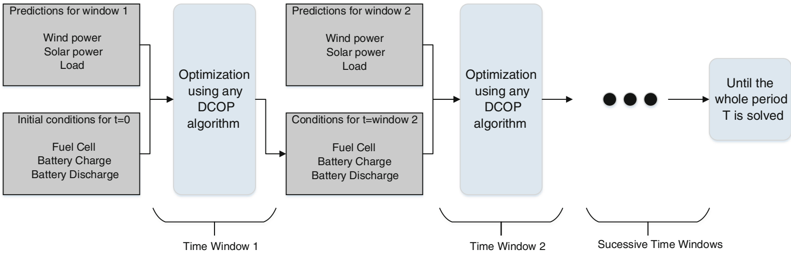

Fig. 4. Optimization using time windows.

One disadvantage of this approach is that the optimal solution cannot be guaranteed and depends directly on the size of the window chosen. Small time windows (e.g., size 1) are solved fast since the optimization is done without knowledge of the future, keeping the information and variable relations low. However, the quality of the solution may not be good enough because of the lack of global vision for the conditions of the successive periods (e.g., wasting all the resources in the current time without considering the demand for the successive time).

It is expected that the quality of the solution improves by increasing the size of the windows. However, as shown in Fig. 5, the number of variables grows significantly with the size of the windows, hence resulting into an exponential growth in the search space of the number of possible configuration to explore. With this in mind, a trade-off between running time (i.e., small sizes) and quality of the solution (i.e., large sizes) has to be considered when solving the long-term optimization problem.

Heuristics and pre-processing strategies can be applied to reduce the configuration search space significantly reducing the run time of the approach in the average case. Nevertheless, in this paper the main objective is to propose a decentralized architecture for the long-term optimization hence the use of such heuristics is out of the scope and will be considered as future work.

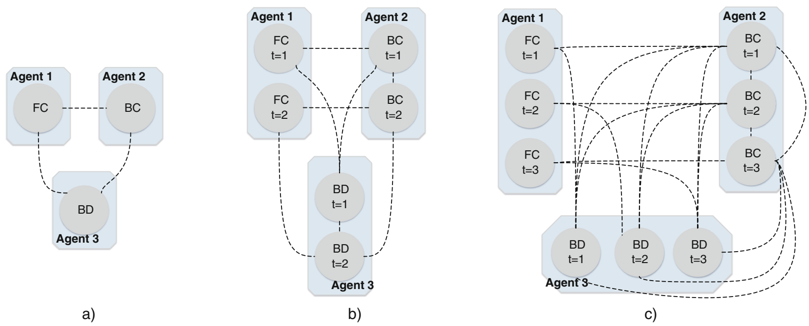

Fig. 5. Model for different window size. Agents and variable relations with a window size of: (a) 1, (b) 2 and (c) 3.

4 Results and Discussion

The results section is divided into three parts. First, in Sect. 4.1 we present the particular scenario considered in this paper. In Sect. 4.2, we present optimal results using the centralized approach from [7]. As previously stated, the centralized approach is less flexible to scalability. However, the optimal solution using such centralized approach is taken as a benchmark. Finally, in Sect. 4.3 we present the results obtained with our decentralized long-term optimization approach.

4.1 Case Study

The reported results consider the Budapest Tech case study presented in [7]. This scenario considers a microgrid with one wind generator (W), one solar generator (S), one fuel cell generator (FC), and one battery. Energy costs are considered constants for simplicity. Such costs and power limits are presented in Table 1.

Table 1. Costs of energy generation and production limits taken from [7].

Costs	Production limits
$C_W = 0.4$ W/h	$P_{lim_W} = 400$ W
$C_S = 0.4$ W/h	$P_{lim_S} = 150$ W
$C_{FC} = 0.9$ W/h	$P_{lim_{FC}} = 80$ W
$C_{BC} = 0.4$ W/h	$P_{lim_{Storage}} = 200$ W
$C_{BD} = 0.6$ W/h	$P_{lim_{BD}} = [0 - 50]$ W
$C_{UE} = 1.5$ W/h	$P_{lim_{BC}} = [0 - 200]$ W
$C_{EE} = 0$ W/h	

Also, the optimization of on/off cycles was done for a period of 24 h (i.e., T = 24) in intervals of 1 h. The first 4 columns of Table 2 present the forecast of wind energy (P_W), solar power (P_S), and the joint *Load* at each time t used as an input of the optimization task. Also, we considered that the battery starts fully charged with 100 W.

4.2 Centralized Optimal Solution

We solved the scheduling problem from a centralized perspective as an mixed-integer linear programming (MILP) using Java and ILOG CPLEX. Table 2 shows two different configurations of the controllable devices for the analyzed scenario. Each column has the configuration of fuel cell power (P_{FC}), battery charge (P_{BC}), battery discharge (P_{BD}), undelivered energy (P_{UE}), exceeded energy (P_{EE}), and the cost associated with such configuration at each time t. The total cost of the solution for the long-term optimization in a period T = 24 h is also reported.

We noticed that there were different configurations with the same global optimal value. The problem of multiple configurations that satisfied all the restrictions with minimum cost is known as the degenerated problem [17]. Different configurations were found varying the order in which the restrictions were considered, or the version of CPLEX solver.

The execution times were in the order of 70 mS with a standard deviation of 10 mS after 1000 experiments. We used a PC with Processor Intel(R) Core(TM) i7-4770 @ 3.40 GHz and 16 GB of RAM.

4.3 Decentralized Long-Term Scheduling Results

In this section, we present the results of the proposed decentralized approach for the scenario presented in Sect. 4.1. The optimization procedure was done for different window sizes as explained in Sect. 3. The experiments were run 100 times each. We present the mean value and standard deviation (Std) of those 100 experiments for different DCOP algorithms (i.e., AFB, DPOP, MB-DPOP, and SynchBB).

In general, in this model for an arbitrary window size (WS) and considering 3 agents (i.e., FC, BC, and BD), the number of variables is $3 * WS$, the number of constrains is $3 * WS + battery_{status} * WS$, and the domain size is $\mathrm{FC}_{domain}^{WS} * \mathrm{BC}_{domain}^{WS} * \mathrm{BD}_{domain}^{WS}$. It can be notice that the domain of each variable grows exponentially along the window size, making the problem not tractable for large window sizes.

Table 3 shows the results of various window sizes reporting the percent error (i.e., the percent error between the long-term decentralized approach and the optimal solution found with the centralized approach of Sect. 4.2), running time, the total number of messages and the total size of messages. It can be seen that DPOP, MB-DPOP, and SynchBB have the same performance. This is because these three algorithms are exact. These three approaches provide an error of 3.64 % for a window size of 1. The error decreases when the window size increases to 2. On the other hand, AFB algorithm found the optimal value for window sizes 3, but the time required to reach the solution was approximately nine

Table 2. Results obtained using Java-ILOG CPLEX. Two optimal configurations.

t	P_W	P_S	$Load$	P_{FC}	P_{BC}	P_{BD}	P_{UE}	P_{EE}	Cost	P_{FC}	P_{BC}	P_{BD}	P_{UE}	P_{EE}	Cost
1	130	0	160	0	0	30	0	0	70	0	0	30	0	0	70
2	150	0	140	0	10	0	0	0	56	0	0	0	0	10	60
3	140	0	150	0	0	10	0	0	62	0	0	10	0	0	62
4	160	0	120	0	40	0	0	0	48	0	0	0	0	40	64
5	100	0	110	0	0	10	0	0	46	0	0	10	0	0	46
6	120	5	100	0	25	0	0	0	40	0	0	0	0	25	50
7	150	10	170	0	0	10	0	0	70	0	0	10	0	0	70
8	180	30	180	0	30	0	0	0	72	0	0	0	0	30	84
9	170	60	200	0	30	0	0	0	80	0	0	0	0	30	92
10	160	100	220	0	0	0	0	40	104	0	0	0	0	40	104
11	120	130	230	0	20	0	0	0	92	0	0	0	0	20	100
12	130	140	240	0	5	0	0	25	106	0	0	0	0	30	108
13	150	150	240	0	0	0	0	60	120	0	50	0	0	10	100
14	176	140	230	0	0	0	0	86	126.4	0	50	0	0	36	106.4
15	185	130	220	0	0	0	0	95	126	0	50	0	0	45	106
16	120	100	210	0	0	0	0	10	88	0	10	0	0	0	84
17	130	60	210	20	0	0	0	0	94	0	0	20	0	0	88
18	140	30	220	0	0	50	0	0	98	0	0	50	0	0	98
19	170	10	230	0	0	50	0	0	102	0	0	50	0	0	102
20	190	5	240	25	0	20	0	0	112.5	45	0	0	0	0	118.5
21	120	0	250	80	0	50	0	0	150	80	0	50	0	0	150
22	170	0	200	30	0	0	0	0	95	30	0	0	0	0	95
23	130	0	190	60	0	0	0	0	106	60	0	0	0	0	106
24	150	0	180	0	0	30	0	0	78	0	0	30	0	0	78
				Total 2141.9						Total 2141.9					

Table 3. Comparison of DCOP algorithms for long-term optimization. Running time are shown in seconds. Messages size are shown in MegaBytes.

Window	AFB	Std	DPOP	Std	MB-DPOP	Std	SynchBB	Std
Percent error								
Size 1	6.34 %	0.0	3.64 %	0.0	3.64 %	0.0	3.64 %	0.0
Size 2	8.30 %	8.3	0.42 %	0.0	0.42 %	0.0	0.42 %	0.0
Size 3	0 %	—	—	—	—	—	—	—
Time								
Size 1	8.1	0.1413	17.3	0.2218	17.2	0.1941	12.3	0.2815
Size 2	2101.3	13.3349	176.6	0.5569	296.7	56.9301	2741.2	137.5269
Size 3	757754.7	—	—	—	—	—	—	—
Number of messages								
Size 1	52785	5.5959	480	0.0	528	0.0	107076	0.0
Size 2	314767	394.3478	628	0.0	688	0.0	5492875	0.0
Size 3	185255337	—	—	—	—	—	—	—
Total size of messages								
Size 1	2.6	$3.87 * 10^{-4}$	0.8	$7.58 * 10^{-6}$	0.8	$6.75 * 10^{-6}$	2.6	$8.56 * 10^{-6}$
Size 2	28	0.0396	2.8	$8.35 * 10^{-6}$	2.9	$8.98 * 10^{-6}$	154.7	0.0
Size 3	2351.7	—	—	—	—	—	—	—

days (e.g., 757754 s). The other three algorithms did not finish the optimization procedure after those nine days. Such behavior can be explained because a small window size (i.e., 1 and 2) does not provide enough information to the agents, limiting the capacity to find the global optimal solution in the long-term. With a window size 3, the agents have sufficient information to find the optimal solution in the long-term but the optimization procedure takes more time since there are more variables to handle by the agents and the search space is too large. The exponential increase in complexity can be also appreciated in the number and size of messages exchanged by the agents in the optimization process.

The results also open research directions for the use of multi-agent systems in the long-term optimization. For instance, it is clear the necessity of faster procedures of search, allowing agents to handle large window sizes and improve the quality of the solutions.

5 Conclusions and Future Work

In this paper, using the concepts of IoT and microgrids, we proposed a distributed architecture of agents for a decentralized management of smart devices. By doing that, the problem of scheduling on/off cycles can be treated as a DCOP, and different multi-agent algorithms can be used to find the solution. Also, knowing that multi-agent algorithms typically do not integrate temporality in their frames due to the large size of variables and search space, we proposed the use of optimization windows to solve the long-term optimization problem sequentially.

Results show that the distributed architecture and time window can be used to find solutions comparable to the optimal for the long-term optimization. Small size windows boost the optimization speed by losing information, limiting the capability of DCOP algorithms. Large window sizes provide more information leading even to the optimal solutions (i.e., with AFB and window size 3), but the search space increases as well, making the optimization task complex. A trade-off must be taken into account between efficiency (i.e., speed) and efficacy (i.e., better solution) when the optimization is performed.

As further work, this paper opens new research directions in different areas. Regarding the DCOP representation, one direction could be the formulation of distributed problems considering time in a compact and natural way. The creation of new restrictions would be needed to include time specifying where and how the variables are connected through time. Another direction, regarding the variable domain of a multi-agent system, will require the develop of new techniques that help to reduce the domain size in challenging scenarios. A way to attack such problem could be through heuristics that help algorithms to find efficiently optimal or near-optimal solutions. Another option would be the implementation of pre-processing techniques to reduce the domain of variables before the application of the optimization algorithms.

References

1. Gungor, V., Sahin, D., Kocak, T., Ergut, S., Buccella, C., Cecati, C., Hancke, G.: A survey on smart grid potential applications and communication requirements. IEEE Trans. Ind. Inf. **9**(1), 28–42 (2013)
2. Yan, Y., Qian, Y., Sharif, H., Tipper, D.: A survey on smart grid communication infrastructures: motivations, requirements and challenges. IEEE Commun. Surv. Tutorials **15**(1), 5–20 (2013)
3. Karnouskos, S.: The cooperative internet of things enabled smart grid. In: IEEE International Symposium on Consumer Electronics (2010)
4. Vega, A., Santamaria, F., Rivas, E.: Modeling for home electric energy management: a review. Renew. Sustain. Energy Rev. **52**, 948–959 (2015)
5. Han, J., sic Choi, C., Park, W.K., Lee, I., Kim, S.H.: Smart home energy management system including renewable energy based on ZigBee and PLC. IEEE Trans. Consum. Electron. **60**(2), 198–202 (2014)
6. Karnouskos, S.: Smart houses in the smart grid and the search for value-added services in the cloud of things era. In: IEEE International Conference on Industrial Technology, pp. 2016–2021, February 2013
7. Morais, H., Kádár, P., Faria, P., Vale, Z.A., Khodr, H.M.: Optimal Scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming. Renew. Energy **35**, 151–156 (2010)
8. Bazmohammadi, N., Karimpour, A., Bazmohammadi, S.: Optimal operation management of a microgrid based on MOPSO and differential evolution algorithms. In: IEEE Iranian Conference on Smart Grids, pp. 1–6 (2012)
9. Chaouachi, A., Kamel, R.M., Andoulsi, R., Nagasaka, K.: Multiobjective intelligent energy management for a microgrid. IEEE Trans. Ind. Electron. **60**(4), 1688–1699 (2013)
10. Su, W., Wang, J., Roh, J.: Stochastic energy scheduling in microgrids with intermittent renewable energy resources. IEEE Trans. Smart Grid **5**(4), 1876–1883 (2014)
11. Hiremath, R., Shikha, S., Ravindranath, N.: Decentralized energy planning; modeling and application - a review. Renew. Sustain. Energy Rev. **11**(5), 729–752 (2007)
12. Logenthiran, T., Srinivasan, D., Khambadkone, A.M.: Multi-agent system for energy resource scheduling of integrated microgrids in a distributed system. Electr. Power Syst. Res. **81**(1), 138–148 (2011)
13. Miller, S., Ramchurn, S.D., Rogers, A.: Optimal decentralised dispatch of embedded generation in the smart grid. In: Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, pp. 281–288 (2012)
14. Petcu, A., Faltings, B.: Superstabilizing, fault-containing distributed combinatorial optimization. In: Proceedings of the National Conference on Artificial Intelligence, pp. 449–454 (2005)
15. Nguyen, D.T., Yeoh, W., Lau, H.C., Zilberstein, S., Zhang, C.: Decentralized multi-agent reinforcement learning in average-reward dynamic DCOPs. In: Proceedings of the International Conference on Autonomous Agents and Multi-agent Systems, pp. 1341–1342 (2014)
16. Petcu, A.: A Class of Algorithms for Distributed Constraint Optimization, vol. 194. IOS Press, Amsterdam (2009)
17. Chvatal, V.: Linear Programming: Series of Books in the Mathematical Sciences. W.H. Freeman, New York (1983)