

A Local Decisional Algorithm Using Agent-Based Management in Constrained Energy Environment

C. Adam, G. Henri, T. Levent, J-B Mauro, and A-L Mayet

Abstract—Energy Efficiency Management is the heart of a worldwide problem. The capability of a multi-agent system as a technology to manage the micro-grid operation has already been proved. This paper deals with the implementation of a decisional pattern applied to a multi-agent system which provides intelligence to a distributed local energy network considered at local consumer level. Development of multi-agent application involves agent specifications, analysis, design, and realization. Furthermore, it can be implemented by following several decisional patterns. The purpose of present article is to suggest a new approach for a decisional pattern involving a multi-agent system to control a distributed local energy network in a decentralized competitive system. The proposed solution is the result of a dichotomous approach based on environment observation. It uses an iterative process to solve automatic learning problems and converges monotonically very fast to system attracting operation point.

Keywords—Energy Efficiency Management, Distributed Smart-Grid, Multi-Agent System, Decisional Decentralized Competitive System.

I. INTRODUCTION

BUILDING up efficient energy network decisional pattern using agent-based management needs to bring together two very basic notions (energy scheduling and multi-agent system) and to adapt artificial intelligence tools with them.

Concerning energy scheduling, the main target is to reduce energy consumption at peak-times in the local network. For this purpose, several methods already exist. The first one is a price incentive method [1] which consists in trying to reduce the peak demand at its lower level by an adapted price policy. However, this method has its own limits. The second way to reach the target is also a cost incentive method the distinctiveness of which allows technological inputs. This method uses client-imposed constraints and splits the objects in three different categories: malleable, moldable and rigid [2]. The main idea in this approach consists in introducing a provisional dimension in energy consumption by adapting it both to price and to consumption profiles of the equipment to be used. A third method consists on reducing the global peak demand while optimizing consumption cost for one household inside a given area. To achieve this goal, a distributed algorithm derived from a Game-Theoretic analysis needs to be used [3].

C. Adam, G. Henri, T. Levent, J-B. Mauro, and A-L. Mayet are with ECE Paris School of Engineering, Paris, 75015 France (e-mail: adam@ece.fr, ghenri@ece.fr, levent@ece.fr, mauro@ece.fr, mayet@ece.fr).

Multi-agent systems offer solutions to manage energy consumption because they make it possible to account for the fact that agents may correspond to different consumer behaviors.

However, an overly homogeneous optimized consumption due to the use of autonomous software agents in smart houses can generate problems in the network [4]. To avoid this situation, a decentralized system can also be part of the solution to reduce peak demand and carbon emissions. Otherwise, use of multi-agent systems in smart grid technologies, offers local solutions to general loads shedding [5], while providing a communication framework between each member of the system. A multi-agent system can also be used at a district scale [6] to optimize the energy management of specific entities such as buildings or factories.

After evoking both energy scheduling and multi-agent systems concepts, it is advisable to identify the nature of inputs artificial intelligence can bring to get an optimized energy management.

Various publications already dealt with artificial intelligence in Energy sector [7]. In the following, artificial intelligence is associated with energy scheduling and multi-agent system to propose a new way to reach easily a high level of energy efficiency management through a decisional pattern that will combine demand side management with production and grid constraints.

II. CONTEXT AND CONSTRAINT

The model consists in contracting with each consumer a maximal load level he can use at any time and that he will not be allowed to outreach. In other terms, production capacity will be the main constraint. By this way, the demand will be adapted to the production exclusively by optimizing the consumption with no interference with production side.

To reduce the risks related to either the use of a centralized model or the negotiation phase between agents, and to get an energy efficient management, the decisional pattern will be placed in a decentralized competitive context. Furthermore, to be more effective, all objects in the system will be split into three different categories, malleable, modulable and rigid defined as follows:

- **Modulable:** objects for which starting time and attributed consumption level are decided.
- **Malleable/adjustable:** objects for which starting time and attributed consumption level are decided, with this

level being flexible during the use of the object.

- **Rigid:** objects for which only starting time is decided as their use need continuous energy supply.

to which there corresponds three agent types each one with its own decisional pattern. Moreover, agents will be provided with adaptability. They must be autonomous and make their own decision without any extra intervention. Thus, agents should follow the way they work to become intelligent. Decisional pattern methods and algorithms will be drawn from elements provided in the introduction.

Various solutions already dealt with adaptability. Here, methodology developed in [8] is interesting for solving automatic learning problems. It associates agent states to action using rewards matrix. Thus, by adding a learning factor γ an agent will be able to benefit from its previous errors by assimilating them thanks to its learning environment. Moreover dichotomy method [9], [10] is appropriate to automatic learning issue. It is an iterative process where, at every step, research space is cut into two not necessarily equal parts before restriction of initial space to one of these two parts. In present case, time and power can be used as research spaces to construct decisional pattern.

III. TOOL IMPLEMENTATION FOR TESTING

To provide an efficient interface for testing each part of decisional pattern, a simulator has been built up within Jade framework [11], [12] allowing easy implementation and development for multi-agent solution using Java and in particular, for messages exchange designed to be compatible with IP-based network (IP = Internet Protocol) based on the IEEE standard on Foundation for Intelligent Physical Agent (FIPA). Furthermore, it will facilitate the seamless transition from grid connected to an island mode when upstream outages will be detected.

The simulator will allow check all possible combinations of scenarios and strategies by returning graph showing energy consumption vs. time. In this way, the best interface will be provided for tests to implement the new smart-grid decisional pattern using decentralized agent-based management.

IV. BASIC DECISIONAL PATTERN

As indicated above, to be more effective objects are split into three different categories: malleable, moduable and rigid. For this precise reason, one decisional pattern per category has to be provided based on the following agent states.

An agent will have only three possible states:

- Off
- Asking energy
- Run

An asking agent will have only two possible positions:

- Start
- Doesn't start

At the beginning the agent corresponding to rigid objet will define the time as a space to build the iterative process. It will have only two possible positions:

- Start

- Doesn't start

So the agent will ask its environment to provide some energy to start. Once launched, the agent will record the period in which it has been able to start.

To build a more efficient system, each agent is given a 24x2 matrix, which will be its memory. The first row incorporates time periods of the day where agent could start. The second row includes a weight which represents the number of launch per period. By this way any object will remember the most likable period to start (where the weight is at his highest level).

Thus, at the end of one cycle, each agent endowed with artificial intelligence will have by this token mechanically integrated the precise period during which it will be able to start the next cycle.

Concerning moduable objects, it is noticed that they can start even if energy maximal power is not reach. So to formalize the decisional pattern of moduable category, a constraint is brought into previous rigid algorithm. Moduable agent will be able to start at its minimal Power instead of its maximal Power. Here again, to provide an efficient system artificial intelligence will be used. With same matrix system the weight will be built by adding a new variable $x = (1*(P_used/P_max))$.

For malleable objects, decisional pattern will only permit the agent to give and to receive energy constantly as a function of disposed energy. They will play the role of compensator to flat energy consumption curve.

Finally, one must also integrate unexpected starting events into future tests to keep some realistic feature (for instance, when inviting people at home, this will emulate the use of unusual agents during cycle of permanent objects). The program file is given in Appendix.

V. SCENARIO TESTING

It is first necessary to define the elements which should be observed. In present case these will be:

- A curve showing the consumption vs. time
- A bar graph showing scheduling of the agents
- A collected history of each day if agent has launched or not
- A matrix for each agent to count the number of launches by time period.

Once the indicators are defined, the communication between each agent has to be checked. To test viability and efficiency of the system, the time period will be adapted depending on the elements to observe.

The first test will be short operating only with a rigid and a malleable agent. After stabilization, a moduable agent will be added. This operation will give, if required, the possibility to modify the algorithm before the second phase.

The second phase will consist in testing decisional pattern in a real environment. For this purpose, three typical scenarios will be considered:

- The first one will refer to a student flat consumption model,

- The second one will refer to a familial house consumption model,
 The third one will refer to a familial green house consumption model.

VI. TEST RESULTS

First a consumption curve for a malleable object is obtained.

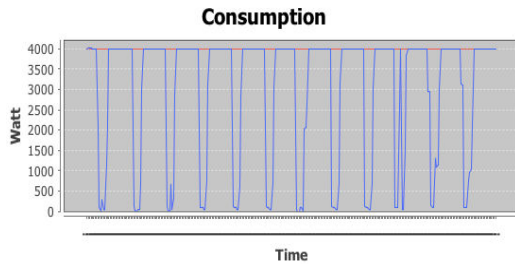


Fig. 1 Consumption of malleable object vs time

It is observed that the curve strongly varies with time, indicating whether the object is giving or receiving energy. From Fig. 1 its energy consumption varies between 4000W and 0W.

At 4000W the malleable object is at full power and has received all necessary energy for its proper functioning. At 0W it does not work and therefore gives energy. Next, the algorithm is tested for a typical set of rigid objects, here a laptop, a dishwasher, a washing machine and a dryer. Each object has its own energy requirement for its own start. In order not to interfere with user comfort, simulation tests are performed over the interval [10,17], starting from first day equi-probable initial distribution $D(k) = -10 + 10[\Gamma(k-10) - \Gamma(17-k)]$ where $k \in [0,23]$ represents the day hour. Daily consumption histogram is obtained as shown on Figs. 2-4.

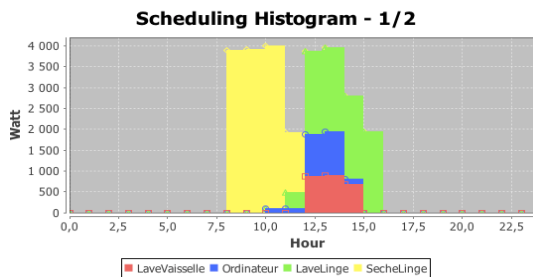


Fig. 2 Daily consumption histogram 1st day

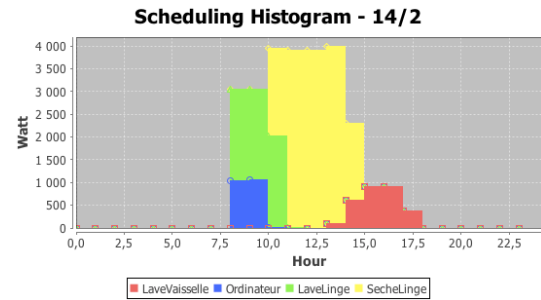


Fig. 3 Daily consumption histogram 14th day

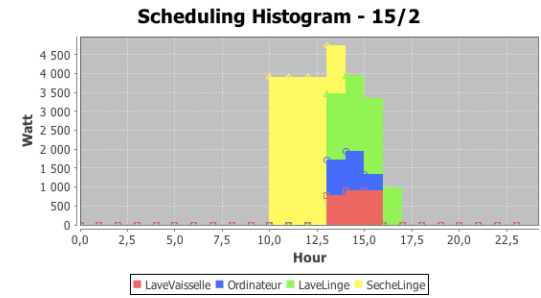


Fig. 4 Daily consumption histogram 15th day

Fig. 2 displays first day scheduling of the cycle. Fig. 3 displays scheduling a few days later and Fig. 4 at cycle end. It is seen that different results are shown. Indeed, in Fig. 2, where dryer was starting first, three agents (a laptop, a dishwasher and a washing machine) cannot start in interval [10,12] but only in interval [12,14], when the dryer has completed its own cycle, due to its too high consumption. On Fig. 3 the laptop and the washing machine adapt their consumption to the environment, and start their own cycle in interval [8,11]. It is only when their cycles are completed, that the dryer can start its cycle. Finally, when dryer consumption allows it, the dishwasher starts its cycle.

On Fig. 4, the laptop, the dishwasher and the washing machine try to adapt all together their consumptions to their environment. Fig. 4 shows that they become able to schedule themselves in a shorter time period starting in the interval [13,17] i.e. after dryer cycle is completed.

Figs. 2-4 show that with its actual rules proposed algorithm works, in that the three rigid agents are able to reassess their working organization. They avoid the mismatch on day 1 (Fig. 2) and later schedule themselves much better over time to comply with energy constraints (Figs. 3 and 4).

After simulation is completed, the following values regarding matrix records are obtained for the different objects.

Object	0	12
LaveVaisselle	-10	1
Ordinateur	-10	68
LaveLinge	-10	7
SecheLinge	-10	0
	-10	0
	-10	0
	-10	-10
	-10	-10
	1	-10
	-10	-10
	0	-10
	2	-10

Fig. 5 Matrix Records - Dishwasher

LaveVaisselle	0	-10	12	1
Ordinateur	1	-10	13	62
LaveLinge	2	-10	14	10
SecheLinge	3	-10	15	1
	4	-10	16	0
	5	-10	17	0
	6	-10	18	-10
	7	-10	19	-10
	8	6	20	-10
	9	-10	21	-10
	10	0	22	-10
	11	0	23	-10

Fig. 6 Matrix Records – Computer

LaveVaisselle	0	-10	12	0
Ordinateur	1	-10	13	74
LaveLinge	2	-10	14	5
SecheLinge	3	-10	15	0
	4	-10	16	0
	5	-10	17	0
	6	-10	18	-10
	7	-10	19	-10
	8	-1	20	-10
	9	-10	21	-10
	10	0	22	-10
	11	1	23	-10

Fig. 7 Matrix Records - Washing Machine

LaveVaisselle	0	-10	12	0
Ordinateur	1	-10	13	0
LaveLinge	2	-10	14	0
SecheLinge	3	-10	15	0
	4	-10	16	0
	5	-10	17	0
	6	-10	18	-10
	7	-10	19	-10
	8	2	20	-10
	9	-10	21	-10
	10	68	22	-10
	11	9	23	-10

Fig. 8 Matrix Records - Dryer

The right column $[0,23]$ indicates the time in the day. Left values in the interval $[10, infinity]$ correspond to the weight. Negative values correspond to the period during which the agents cannot ask for energy. Positive values correspond to the number of times during which agents have adapted to the situation.

Matrix on Fig. 5 gives information on dishwasher behavior during a cycle. It can be noticed that this agent had five departing time periods (corresponding to positive numbers on second row): first one, in interval $[8,9]$ with 1 succeeding launch; second one, in interval $[11, 12]$ with 2 succeeding launches; third one, in interval $[12,13]$ with 1 succeeding launch; fourth one, in interval $[13,14]$ with 68 succeeding launches and last one, in interval $[14,15]$ with 7 succeeding launches. Here interpretation of matrix results indicates that time interval $[13,14]$ is best for dishwasher launching. Similarly, from second matrix on Fig. 6, best time interval for laptop start is also $[13,14]$ with 62 succeeding days. In the same way, from Fig. 7 this interval is best time period for washing machine launch with 74 performed launches. Finally, last matrix on Fig. 8 indicates that best launch time period for dryer is in interval $[10,11]$ with 68 succeeding launches. Overall, agents not requiring too much energy can work together with larger agent. Interestingly, results seem conclusive in all respects when present analysis concerning a microscopic scale is only using a microscopic solution, showing that there is no needs to call for a global heading control. For completion in present simulation one should also

introduce unexpected events to give the system the ability to schedule the agents by making them adaptive.

VII. CONCLUSION

Analysis of efficient energy network decisional pattern using agent-based management has been set up to optimize energy consumption at local level where objects are split into three different groups with different consumer behaviors. To this aim decentralized multi-agent system has been preferred to give agents adaptive dimension for matching objects consumption requirements. Based on the idea that one class of objects can play the role of a fictive energy source, a simulator has been developed with corresponding rules and different scenarios have been tested. They mainly show that system is largely able to locally adapt energy demand by self-organization without needing a global level control. At more general system level, present results indicate that utilized simple local rules are leading to asymptotically stable and converging system dynamics toward its optimum operating point. This is suggesting the importance of adjustment between physical laws system elements are actually obeying and their distribution, from which a robustness ball counting percentage of hits toward optimal operating point defined by chosen rules vs. available energy could be defined.

From this initial work, larger scale simulation can be considered to check up to what level proposed (microscopic scale) solution is still valid at macroscopic scale. On the other hand the flexibility of simulator algorithmic structure allows at the same time to test different and more sophisticated object behaviors. In this way, a new interesting element can be brought to the very open question of controlling complex systems by intelligence delegation as opposed to centralized classical control.

APPENDIX

Algorithm of dichotomous programming for permanent rigid and/or modulable object:

```
//Variable and initialization
start = False //Boolean, Agent started
end_cycle = False //Boolean, End Cycle
t_start = 0 min //hour of the start
t_scenario = 0 min //hour of the scenario
P_used = 0 min //amount of energy used by the agent
P_ch //available energy depending on agent nature
Agent
Coeff //adjustable parameter regulating flux distribution
according to agent nature
P_min //minimal amount of energy needed for the agent to run
P_max //amount of energy needed for the agent to run
If Agent ==Modulable
    P_ch = P_min
    Coeff = 1*(P_used/P_max)

Else if Agent ==Rigid
    P_ch = P_max
    Coeff = 1
```

Matrix(24)(2)

//method when energy demand

While (end_cycle = False)

t_scenario=getTime()

//We synchronize with the scenario

For i from 0 to 24:

t_debut=Matrix[getLigne(Matrix.getMax(Matrix[i][2]))][1]

// it is verified that no energy is received

if (message != null)

P_used = P_used + agent.getMessage()

End if

If ((t_scenario >= t_start) && (t_scenario < (t_start + 60min)) && (P_used >= P_ch))

//So the agent starts (we have to see if one variable may suffice)

start = True

end_cycle = True

Matrix[t_start][2] = Matrix[t_start][2] + Coeff

Else If ((t_scenario >= t_start) && (t_scenario < t_start + 60min) && (P_used < P_ch))

//Send a request for asking energy
agent.send(P_max-P_used)

Else //object doesn't start

fori from 0 to 24:

t_debut =

Matrix[Matrix.getLine(Matrix.getMax(Matrix[i>t_debut][2]))][1]

End if

End While

ACKNOWLEDGMENT

The authors are very much indebted to ECE for having provided the environment where the project has been developed, to Pr C. Duhart for his help during the completion of the work and to Pr. M. Cotsaftis for preparation of the manuscript.

REFERENCES

- [1] A Mohsenian-Rad, V. Wong, J Jatskevich, R. Schober, A Leon-Garcia, "Autonomous Demand-Side Management Based on Game Theoretic Energy Consumption Scheduling for the Future Smart Grid", in *IEEE Trans. on Smart Grid*, vol. 1(3), December 2010.
- [2] T. Zimmerman, S Smith, A Unahalekhaka, "Client Side Power Scheduling", *2nd Intern. Workshop ATEs 2011*, in *10th Intern. Conf. AAMAS*, Taipei, RoC, May 26 2011.
- [3] A. Mohsenian-Rad, V Wong, J Jatskevich, R. Schober, A Leon-Garcia, "Optimal and Autonomous Incentive-based Energy Consumption Scheduling Algorithm for Smart Grid", *Proc. IEEE ISGT Conf.*, Gaithersburg, MD, Jan. 19-21 2010.
- [4] S. Ramchurn, P. Vytelingum, A. Rogers, N. Jennings, "Agent-Based Control for Decentralised Demand Side Management in the Smart Grid", *2nd Intern. Workshop ATEs 2011*, in *10th Intern. Conf. AAMAS*, Taipei, RoC, May 26 2011.

- [5] M. Pipattanasomporn, H. Feroze, S. Rahman, "Multi-Agent Systems in a Distributed Smart Grid Design and Implementation", *Proc. PSCE'09. IEEE/PES*, Seattle, WA, March 15-18 2009.
- [6] XingyuCai, Chun Zhang, Hao Yu, RadhikaBhar, HoayBengGooi, "Uncertainty aware minority game based energy management system for smart buildings", *Proc. IEEE ISGT Conf.*, Tianjin, China, May 21-24 2012.
- [7] P. Palensky, D. Dietrich, "Demand Side Management: Demand Response intelligent Energy Systems and Smart Loads", *IEEE Trans. on Indus. Informatics*, vol.7(3), pp.381-388, August 2011.
- [8] C. J.C.H. Watkins, P. Dayan "Q-Learning", *Machine Learning*, vol.8, pp.279-292, *Kluwer Acad. Publ.*, Boston, 1992.
- [9] B. Zeddini, "Modèles d'Auto-Organisation Multi-Agents pour le Transport à la Demande", *PhDThesis*, Le Havre University, 2009.
- [10] S. Pinson, M. F. Shakun "An extended multi-agent negotiation protocol. Autonomous Agents and Multi-Agent Systems", *Autonomous Agents and Multi-Agent Systems*, vol.8(1), pp.545, 2004.
- [11] "Jade documentation" by Jade.tilab.com
- [12] F.L. Bellifemine, G. Caire, D. Greenwood "Developing Multi-agents with Jade", *Wiley Series in Agent Technology*, England, 2007.