

A Robust Deterministic Energy Smart-Grid Decisional Algorithm for Agent-Based Management

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

Abstract—This paper is concerning the application of a deterministic decisional pattern to a multi-agent system which would provide intelligence to a distributed energy smart grid at local consumer level. Development of multi-agent application involves agent specifications, analysis, design and realization. It can be implemented by following several decisional patterns. The purpose of present article is to suggest a new approach to control the smart grid system in a decentralized competitive approach. The proposed algorithmic solution results from a deterministic dichotomous approach based on environment observation. It uses an iterative process to solve automatic learning problems. Through memory of collected past tries, the algorithm monotonically converges to very steep system operation point in attraction basin resulting from weak system nonlinearity. In this sense, system is given by (local) constitutive elementary rules the intelligence of its global existence so that it can self-organize toward optimal operating sequence.

Keywords—Decentralized Competitive System, Distributed Smart Grid, Multi-Agent System.

I. INTRODUCTION

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

The main objective of energy scheduling is to reduce energy consumption at peak-times. For this purpose, several methods already exist. First one is a price incentive method [1] consisting in trying to reduce the peak demand at lowest level by adapted price policy. A 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 into three different categories: malleable, moldable and rigid [2]. The main idea here consists in introducing a provisional dimension in energy consumption by adapting it both to price and to consumption profiles of used equipment. A third method consists in reducing 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].

To manage energy consumption multi-agent systems offer solutions making it possible to account for the agents to correspond to different consumer behaviors.

C. Adam, G. Henri, T. Levent, J.-B. Mauro, A.-L. Mayet are Undergraduate Students, ECE-Paris Graduate School of Engineering, Paris, France (G. Henri is Corresponding Author e-mail: gonzague.henri@alumni-ece.org).

However, overly homogeneous and optimized consumption resulting from use of autonomous software agents in smart houses can generate problems in the grid [4]. To avoid this situation, a decentralized system can also be part of the solution to reduce peak demand. 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 system member. A multi-agent system can also be used at larger scale [6] to optimize the energy management of specific entities such as buildings or factories.

To proceed, artificial intelligence [7] is here associated with energy scheduling and multi-agent system to introduce a new way for reaching “easily” a high level of energy efficiency management through a decisional pattern that will combine demand side management with production and grid constraints. To qualify the word “easily”, and to test the idea in compatibility with academic environment, simple deterministic and synchronous analysis will be in a first step conducted here on a model system representative of a local consumption unit. It will be verified that results exhibit fast convergence with extremely steep optimal peaks, indicating strong system robustness within its attraction basin. Extension to stochastic and asynchronous inputs will be discussed elsewhere.

II. ENVIRONMENT AND CONSTRAINT

The model consists in contracting with each consumer a maximal load level available at anytime and he will not be allowed to outreach. In other words, production capacity will be the main constraint. In this way, the demand will be adapted to 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 have already been dealing 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 errors by assimilating them while learning its working 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 checking all possible combinations of scenarios and strategies by returning graphs 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. DECISIONAL PATTERN STRUCTURE

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

A)-At the beginning the agent corresponding to rigid objet will define the time interval on which is built 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 at which it has been able to start.

To build an efficient system, each agent is given a 24x2 matrix, which will be its "memory". The first row incorporates day time periods where the agent can start. The second row includes a weight which represents the number of launch per period. In this way any object can 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 mechanically integrated token the precise period during which it will be the most able to start the next cycle.

B)-To obtain the decisional pattern of moduable agent category, a specific condition is added in previous rigid algorithm. Moduable agent will be able to start at minimal power P_{min} instead of maximal power P_{max} . This expresses the property that moduable object should be able to start even if maximal power is not available. 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 $x = (1 * (P_{used} / P_{max}))$.

C)-For malleable objects, decisional pattern will only permit the agent to give and to receive energy constantly as a function of disposable 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. SCENARIOS TESTING

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

- The consumption vs. time
- A bar graph showing scheduling of the agents
- A collected history of each day agent launching
- A matrix counting for each agent 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 consumption scenarios have been considered:

- The student flat consumption model,
- The family house consumption model,
- The green house consumption model.

VI. CASE ANALYSIS

Here in first case scenario, the algorithm is tested for a

typical set of rigid objects, a laptop, a dishwasher, a washing machine and a dryer, with their own power requirement for their own start. In order not to interfere with user comfort, simulation tests have been performed over daily interval $[10,17]$ during a tri-month cycle, starting from first day initial equi-probable distribution:

$$D(k) = -10 + 10[I(k-10) - I(17-k)] \quad (1)$$

where $k \in [0,23]$ represents the day hour. Daily consumption histogram is obtained as shown on Figs. 1-3.

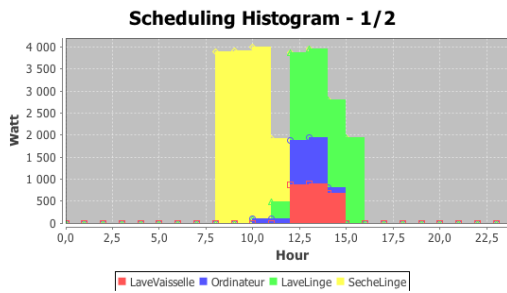


Fig. 1 Daily consumption histogram 1st day

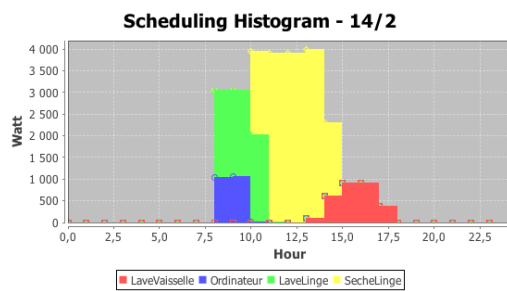


Fig. 2 Daily consumption histogram 14th day

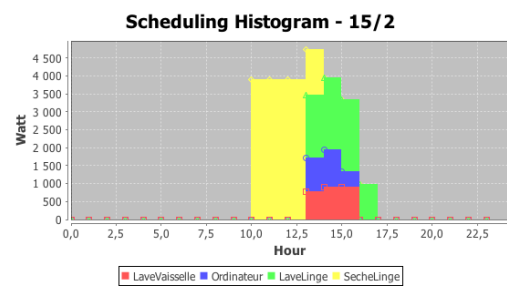


Fig. 3 Daily consumption histogram 15th day

Fig. 1 displays first day scheduling of the cycle. Fig. 2 displays scheduling a few days later and Fig. 3 one day later. On Fig. 1, 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. 2 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 finished, that

the dryer can start its cycle. Finally, when dryer consumption allows it, the dishwasher starts its cycle.

On Fig. 3, the laptop, the dishwasher and the washing machine try to adapt all together their consumptions to their environment. Fig. 3 shows that they schedule themselves in a shorter time period starting in the interval $[13, 17]$ i.e. after dryer cycle is completed.

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

VII. CONVERGENCE AND ROBUSTNESS

After simulation is completed, collected matrix records are obtained for the different objects in each hour interval over a full working day as displayed on Fig. 4. Values on vertical axis are ranging in interval $[-10, \infty]$. 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 and been able to start their duty cycle. For instance, $[dw]$ curve indicates that dishwasher got finally five departing times, but clearly time interval $[13, 14]$ is best for its start with 68 successful launchings. In the same way, it is observed on the other curves that there exists for all other system components a very privileged time interval where they concentrate by far their departure within the setting of actual energy flow rules.

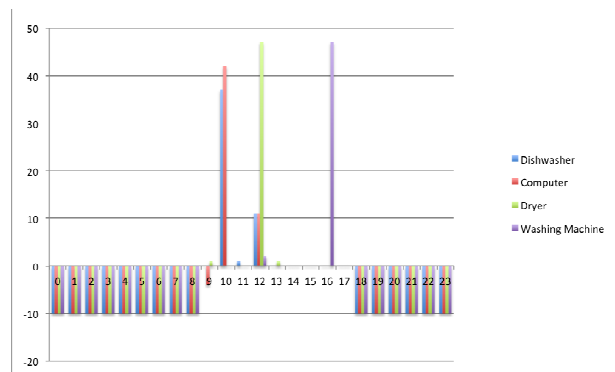


Fig. 4 Stationary Starting Time Distribution for System Components

So the natural working organization emerging from proposed simple algorithm is converging toward unique, strongly localized and sharp solution with extremely steep values, see Fig. 5. Main reason for this great coherence is the very easy albeit rich enough algorithm structure which covers the complete space of possible system states within attraction basin defined by its actual setting and initial conditions. Also, fast system evolution toward “best” operating sequence is made monotonic (non oscillatory) by the weight of previous memorized starts, and is thus more robust to initial system parameter modification. This is observed on Figs. 2 and 3, where change of starting order due to random perturbation

added on 14th day is immediately corrected the day after and returns to previous one.

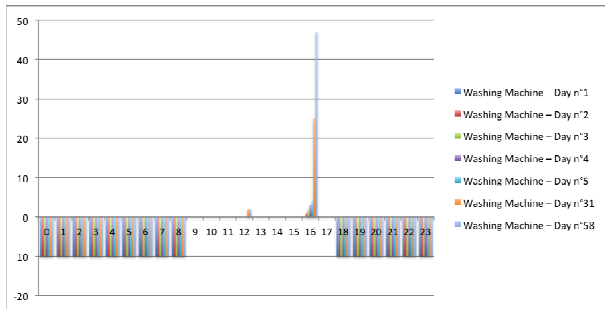


Fig. 5 Convergence of Start Time Distribution vs. Number of Tries for Dishwasher

It should further be related to “soft” nature of system nonlinearity, which only indirectly affects in each time interval the behavior of each element through consumption of all the other ones in the system. An interesting consequence of present launch distribution is that, being extremely well defined, present analysis could be more easily fitted in a hierarchical structure (each “unit” at each level just communicates a global demand to other units), giving the system more robustness to exterior conditions up to a “safety” level which in discussed example is around 4kW.

As expectable, agents not requiring too much energy can work together with larger agent. Interestingly, results concerning a microscopic scale problem are dealt with in present analysis when 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.

VIII. CONCLUSION

Analysis of efficient smart-grid 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 indicating 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 checkup 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

Dichotomous Programming Algorithm 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, in which the project has been developed, to Pr C. Duhart for his help during the completion of the work and to Pr. M. Cotsaft is 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" *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. 1921 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 1518 2009.
- [6] Xingyu Cai, Chun Zhang, Hao Yu, Radhika Bhar, Hoay Beng Gooi, "Uncertainty aware minority game based energy management system for smart buildings", *Proc. IEEE ISGT Conf., Tianjin, China*, May 2124 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.381388, August 2011.
- [8] C. J.C.H. Watkins, P. Dayan "Q-Learning", *Machine Learning, vol.8, pp.279292, Kluwer Acad. Publ., Boston, 1992*.
- [9] B. Zedini, "Modèles d'Auto-Organisation Multi-Agents pour le Transport à la Demande", 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*.