A Reputation-Based Distributed District Scheduling Algorithm for Smart Grids

D. Borra[1(✉)], M. Iori[2], C. Borean[2], and F. Fagnani[1]

[1] Dipartimento di Scienze Matematiche, Politecnico di Torino, Turin, Italy
domenica.borra@polito.it
[2] Swarm Joint Open Lab, Telecom Italia, Lazio, Italy

Abstract. In this paper we develop and test a distributed algorithm providing Energy Consumption Schedules (ECS) in smart grids for a residential district. The goal is to achieve a given aggregate load profile. The NP-hard constrained optimization problem reduces to a distributed unconstrained formulation by means of Lagrangian Relaxation technique, and a meta-heuristic algorithm based on a Quantum inspired Particle Swarm with Lévy flights. A centralized iterative reputation-reward mechanism is proposed for end-users to cooperate to avoid power peaks and reduce global overload, based on random distributions simulating human behaviors and penalties on the effective ECS differing from the suggested ECS. Numerical results show the protocols effectiveness.

Keywords: Distributed algorithms · Autonomous demand response management · Energy consumption scheduling · Smart power grids · Reputation algorithm

1 Introduction

The balance between demand and supply plays a leading role in smart grids applications and modern technologies aim to develop energy optimization algorithms able to provide efficient residential district dispatchment. A large literature has been devoted to decentralized versions of optimization algorithms applied to power systems, see, e.g., [15], due to distributed energy generation and demand, renewables such as photovoltaic resources, storage devices, with changes in real time. Multi-agent planning, as in [11], is often formulated as a combinatorial optimization problem: each agent has its own objectives, resources, constraints, and at the same time it has to share and compete for global resources and constraints. Moreover, new roles in the energy market are emerging, such as energy aggregators as intermediate between energy utilities and home users, managing uncertanties due to variable customer actions, metereology and electricity prices. Given the huge number of agents, the optimization problem is often computationally intractable in a centralized fashion, and given the time-varying cost and constraints in energy demand-response (DR) problems, a fast single-agent planning algorithm is appealing. In this paper, as in [6], customers

R. Giaffreda et al. (Eds.): IoT360 2014, Part I, LNICST 150, pp. 70–76, 2015.
DOI: 10.1007/978-3-319-19656-5_11

are incentivized to move their loads in off-peak hours despite their individual needs through marginal costs, using reputation scores as feedback. In [6] a cooperative game reduces peak-to-average ratio of the aggregate load and the Nash equilibria are reached using centralized information, whereas our approach is completely distributed. Evolutionary Game theory and Reinforcement Learning techniques have been applied to swarm intelligence problems, as in [1,5,12,14].

Our focus is on energy distribution to a residential district, according to the European Project INTrEPID [9]. In this scenery, the district global load is sensed by power meters, and using non-intrusive load-monitoring techniques (NILM, as in [8]) or smart plugs, the disaggregated data are available, turning the "blind"system to a decentralized smart grid [2]. A centralized unit senses local loads, and communicates with agents through smart-phone app or similar devices proposing day-ahead optimal Energy Consumption Schedules (ECS). Agents may accept the suggested ECS or not, according to individual needs.

Our contribution is twofold. First, we provide a mathematical formalization of the optimization problem, decoupling the global constraint through Lagrangian relaxation as in [10], see Sect. 2. Second, in Sect. 3 we design optimal ECS in a distributed fashion at two levels: at the agent level applying meta heuristic optimization techniques as QPSOL (Quantum Particle Swarm with Levy's Flights) described in [3], in order to get feasible optimal suggested ECS; at the district level a reputation-reward mechanism provides incentives for users leading to an emerging cooperative behavior. Section 4 describes the numerical results, and we draw the conclusions of our study in Sect. 5.

2 Model Description

Consider a district with N users, each i-th agent has n_i appliances that are schedulable, like washing machine (WM), dish washer (DW) and tumbler dryer (TD). Refrigerator load is also included as background profile. The state of the multi-agent system is given by $\boldsymbol{x} = (x_1, \dots, x_N)$, i.e., a vector of schedules that each user has to execute in a given time slot, and x_i is defined by the start times of all the n_i appliances of user i and their type (WM, DW, TD) with well-known load profiles. Due to energy and time constraints, the goal to find a global optimum of the constrained optimization problem, called *primal problem*:

$$\min_{\boldsymbol{x}=(x_1,\dots,x_N)} \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} g_i(x_i) = a, \;\; h_i(x_i) \le b_i, i = 1, \dots, N \qquad (1)$$

where $a, b_i \in \mathbb{R}$ and the cost function $\sum_i f_i$ is a sum of weighted norms of three factors: overload, energy cost and tardiness of the current state $\boldsymbol{x}$. The first constraint is the only coupling object: g_i denotes the peak profile of each user and the global load of the district must attain a given curve $a = a(t)$ depending on time. All the functions f_i, g_i, h_i implicitly depend on time (they span a day), discretized in minutes or hours. The inequalities involving h_i are local time and energy (usually 3 kW) constraints of each user. The Lagrange function is $\mathcal{L}(\boldsymbol{x}, \mu, \lambda_i) = \sum_{i=1}^{N} [f_i(x_i) + \lambda_i(b_i - h_i(x_i))] + \mu(g_i(x_i) - a)$ where $\lambda_i \geq 0, \mu$ are called *Lagrange multipliers*. Since λ_i can be computed locally, the Lagrange multiplier of our interest is μ, associated to the only coupling constraint. From now on, we neglect the local constraints as they can be included directly in the cost functions f_i. As detailed in [4], the corresponding relaxed *dual problem* becomes unconstrained: $\max_\mu \min_{\boldsymbol{x}=(x_1,\ldots,x_N)} \mathcal{L}(\boldsymbol{x}, \mu)$. The standard algorithm is as follows: given an initial estimate of μ, each user computes its best ECS x_i^* such that $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \mu)$. Then, $\boldsymbol{x}^*$ is sent to the central unit, and a subgradient of $\min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \mu)$ as function of μ is available. The central unit computes and sends to agents at iteration k: $\mu^{(k)} = \mu^{(k-1)} + \alpha^{(k-1)} \left(\sum_i g_i(x_i^*) - a\right)$, where $\alpha^{(k-1)}$ is the step length of the gradient descent algorithm. Since the Lagrange multiplier μ can be interpreted as the energy price, in order to decentralize the given dual problem., we split $\mu = \sum_{i=1}^{N} \mu_i$. A distributed algorithm that can be applied acts as the previous one with the only difference: agent i solves the optimization problem

$$\min_{x_i} f_i(x_i) + \mu_i \left(N g_i(x_i) - a\right), \tag{2}$$

where $N g_i(x_i) - a$ approximates the global overload $\sum_j g_j(x_j) - a$. The only computational effort of the central unit is the gradient descent step for μ. The latter optimization problem is solved by means of the population-based metaheuristic method QPSOL, see [3], that reduces a NP-hard combinatorial optimization problem to an adaptive algorithm requiring limited computational power.

The underlying idea is to split the optimization algorithm on 2 time scales: (1) the micro-scale concerns the improvement along the day of the day-ahead proposed ECS; (2) the macro-scale involves the reputation-reward mechanisms of the agents, described below, and their collective behavior.

3 Swarm Simulator Description

This simulation studies energy distribution to a city district managing its total daily power consumption without power peaks and achieving a given aggregate load curve. Users should follow utility suggestions and receive incentives according to their flexibility. Every day users compute local best ECS in a distributed way, according to their needs and utility constraints, as described in Sect. 2. In this Section we focus on the reputation mechanism defining the emerging learning process. Consider best ECS as daily input data. Agents actions define local effective ECS. Two indices evaluate end-users behaviors: (1) reputation depending on start times of effective ECS, (2) reward depending on the distance between best and effective (both local and global) load.

Reputation Definition. Each agent may accept or decline n_i suggestions, with n_i number of appliances. Denote by x_i^* the best (sub)-optimal ECS found for Eq. 2 at the end of each day, and denote by $\hat{x}_i$ the effective ECS decided by user i. Formally, the reputation of user i along the day is $r_i = 1 - \frac{|x_i^* - \hat{x}_i|}{n_i} \in [0, 1]$, where $|\cdot|$ denotes the distance between the best and effective i-th ECS in terms of start times of appliances, i.e., reputation decreases as violation rate gets high.

Reward Definition. The reward is defined in terms of credits: each agent may earn up to 24 credits each day, comparing hourly the best (b) and effective (e) two quantities: global load and local load. Formally the credits of user i at hour h is defined as $c_{ih} = 1 - \frac{|\mathrm{glob\_load_b} - \mathrm{glob\_load_e}|}{\mathrm{glob\_load_b} + \mathrm{glob\_load_e}} - \frac{|\mathrm{loc\_load_b} - \mathrm{loc\_load_e}|}{\mathrm{loc\_load_b} + \mathrm{loc\_load_e}}$. At the end of each day, credits $c_i \in [0, 1]$ are re-normalized and create rank lists.

Behavior and Learning Process Modeling. Each agent acts based on his own behavior profile, shaped according to (1) favorite start times to schedule appliances; (2) relevance given to reward and reputation by means of the weight parameter $\alpha_i \in [0, 1]$, to define reaction to feedback; (3) natural predisposition to follow advice, to set the violation probability, defined by standard deviation σ_i of a Gaussian distribution. Best ECS for utility are denoted by the start times vector x_i^* and actions are samples from Gaussian distributions $\hat{x}_i \sim \mathcal{N}(x_i^*, \sigma_i^2)$ with mean given by x_i^* and standard deviation σ_i representing flexibility. Profiles are modeled according to σ_i that is initially sampled uniformly in a given interval $[\sigma_1, \sigma_2]$. For large σ_i agents tend to selfish behaviors and do not accept suggested ECS. Another learning parameter is the weight $\alpha_i \in [0, 1]$ each agent gives to reward and reputation as feedback, i.e., after each observation period user i evaluates the linear combination of its mean reputation $\bar{r}_i$ and its mean reward $\bar{c}_i : q_i = \alpha_i \bar{r}_i + (1 - \alpha_i)\bar{c}_i$. Given the *satisfaction threshold* ϵ (in numerical experiments $\epsilon = 0.6$), if $q_i > \epsilon$, agent i is satisfied and there is a certain probability that relaxes decreasing its standard deviation σ_i, otherwise it increases according to a fixed discrete random distribution. In conclusion, behavior of agent i is defined by the Gaussian probability density function $f = f(x_i^*, \sigma_i, \alpha_i)$. At each feedback iteration the behavior parameter σ_i is updated. Houses with best and worse reputations and rewards are listed as another daily feedback, and emerging collective beahvior is described in Sect. 4.2.

4 Numerical Results

4.1 Micro-Scale Simulation

In this numerical experiments, using MATLAB software we run the simulator for small residential neighborhoods, i.e., $N = 5, N = 10$ agents and through QPSOL and Lagrangian relaxation described in Sect. 2, few iterations are sufficient to get a significant reduction of the global overload, as shown in Fig. 1. The output of such distributed algorithm are the daily suggested ECS, and the macro-scale simulator deals with the learning process acting on human decisions for ECS.

4.2 Macro-Scale Simulation

Software used for the development of macro-scale simulation is GAMA-platform [7], an agent-based, spatially explicit, modeling and simulation platform. Models

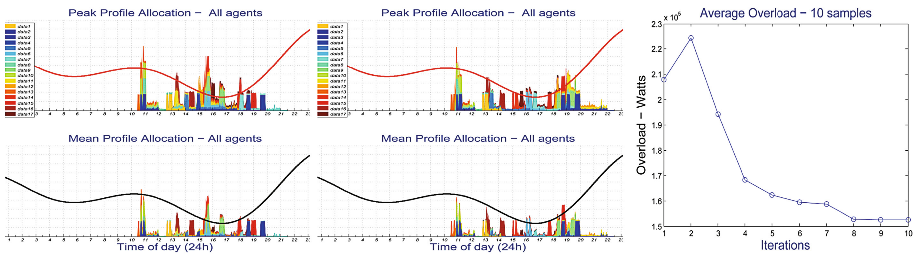

Fig. 1. In the left and center plot, the peak (upper plot) and mean (lower plot) power load (Watts) of a 5 agents neighborhood is displayed at the first and last iteration ($t = 10$) of the distributed algorithm proposed in Sect. 2. All agents are flexible during 10 am-9 pm. The right panel displays the average overload (over 10 samples), i.e., the distance between best and effective global load, as function of algorithm iterations.

are written in the GAML agent-oriented language, so that each house is considered to be an agent. We consider a district composed by $N = 100$ houses and a scheduled annual load for each resident about 1200–1400 kWh.

Appliances are distributed according to the following percentages: 99 % of houses have a WM, 70 % have a DW and 30 % have a TD. There are also some differences between user habits and families. These are modelled varying the maximum number of possible daily cycles for each appliance. In particular 40 % of residents will use every appliance no more than once a day, 50 % no more than twice and 10 % no more than three times a day. Some exceptions are considered.

The system evolution stabilizes in the presence of perturbative phenomena on the input parameters, i.e., differences between effective and best ECS. Using default value of parameters we can reach a mean percentage difference (over the best load) between the best total load and the effective total load converges to 20 % as in Fig. 2 (left).

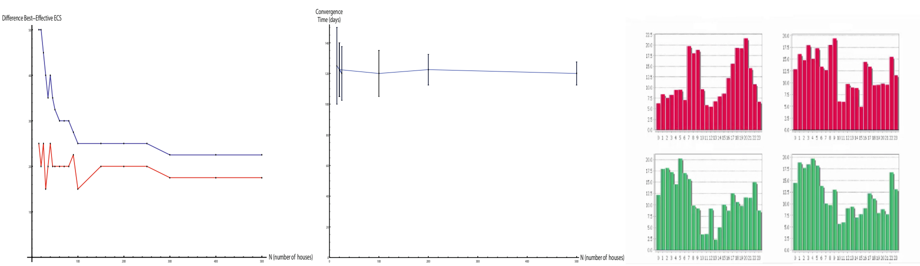

Fig. 2. The left plot shows the maximum (blue) and minimum (red) difference in percentage (converging to 20 %) between best and effective total load varying the number of houses. The central plot refers to necessary time to the district to reach a stable state (3–6 months) and a stable difference between the two loads, compared to the number of houses. Finally, the right chart is an example of total effective load (red) and total best load (green) when simulation starts (left) and at its stabilization (right) (Color figure online).

Varying the number of houses, the difference between effective and best load profile stabilizes starting from 100 houses in the district, as shown in Fig. 2 (center). Convergence time varies between 3 and 6 months. Reported values are the average over 10 simulations with the same number N of homes. Variance is greater if we consider few houses, while stabilization time increases with N.

5 Conclusions

In this paper we provide a mathematical model and a simulator of an energy distribution system applied to a residential district. Once end-users compute local optima in a distributed way, human decisions are modeled and a reputation-reward mechanism is performed on large numbers. Numerical results prove the efficiency of our algorithm: on the macroscale with few houses (150) the difference between best and effective ECS converges to 20 %, and with an average time of 3 months the district stabilizes. Future research may be devoted to apply Lagrangian Relaxation methods also to the macro time-scale, updating individual energy prices each day, as a function of the difference between best and effective ECS. Another advance is to develop asynchronous versions of the proposed algorithms adapting optimal ECS to asynchronous end-users decisions.

Acknowledgments. Authours would like to thank Ennio Grasso for the scientific hints on the mathematical modeling and numerical implementation.

References

1. Baharlouei, Z., Hashemi, M., Narimani, H., Mohsenian-Rad, H.: Achieving optimality and fairness in autonomous demand response: benchmarks and billing mechanisms. IEEE Trans. Smart Grid **4**(2), 968–975 (2013)
2. Borean, C., Ricci, A., Merlonghi, G.: Energyhome: a user-centric energy management system. Metering Int. **3**, 52 (2011)
3. Borean, C., Grasso, E.: QPSOL: quantum particle swarm optimization with levys flight. In: ICCGI 2014, Seville (2014)
4. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, UK (2004)
5. Campos-Nanez, E., Garcia, A., Li, C.: A game-theoretic approach to efficient power management in sensor networks. Operat. Res. **56**(3), 552 (2008)
6. Caron, S., Kesidis, G.: Incentive-based energy consumption scheduling algorithms for the smart grid. In: Proceedings of the 1st IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 391–396 (2010)
7. GAMA-platform. https://code.google.com/p/gama-platform/
8. Hart, G.W.: Nonintrusive appliance load monitoring. Proc. IEEE **80**(12), 1870 (1992)
9. INTrEPID, FP7-ICT project. http://www.fp7-intrepid.eu/
10. Joo, J.-Y., Ilic, M.D.: Multi-layered optimization of demand resources using lagrange dual decomposition. IEEE Trans. Smart Grid. **4**(4), 2081–2088 (2013)

11. Krause, J., et al.: A survey of swarm algorithms applied to discrete optimization problems. In: Yang, X.-S., Cui, Z., Xiao, R., Gandomi, A.H., Karamanoglu, M. (eds.) Swarm Intelligence and Bio-Inspired Computation. Theory and Applications, pp. 169–191. Elsevier Science & Technology Books, UK (2013)
12. Liyan, J., Lang, T.: Day ahead dynamic pricing for demand response in dynamic environments. In: IEEE Conference on Decision and Control (CDC), pp. 5608–5613 (2013)
13. Pinyol, I., Sabater-Mir, J.: Computational trust and reputation models for open multi-agent systems: a review. Artif. Intell. Rev. **40**(1), 1–25 (2013)
14. Tuyls, K., Nowe, A.: Evolutionary game theory and multi-agent reinforcement learning. Knowl. Eng. Rev. **20**(01), 63–90 (2005)
15. Vinyals, M., et al.: Coalitional energy purchasing in the smart grid In: Proceedings of the IEEE International on Energy Conference and Exhibition, pp. 848–853 (2012)