Negotiation Algorithms for Peer-to-Peer Electricity Markets: Computational Properties

F. Moret† *fmoret@elektro.dtu.dk*
T. Baroche‡ *tbaroche@ens-rennes.fr*
E. Sorin† *egsorin@elektro.dtu.dk*
P. Pinson† *ppin@elektro.dtu.dk*

***Abstract*—Building on the concepts of transactive energy and consumer-centric electricity markets, the interest in community-based and peer-to-peer structures to energy transactions and management has substantially increased over the last few years. However, several computational challenges are to be tackled in order for these approaches to be deployed in real-world applications. Our aim here is to identify and analyze these challenges, by comparing distributed community-based market approaches to decentralized and distributed versions of peer-to-peer electricity markets. Our computational analysis concentrates on three key aspects, i.e., convergence, scaling properties, and resilience to delays in an asynchronous communication framework. Our findings justify the further proposal of hybrid approaches and of sparsification of negotiation processes.**

***Index Terms*—Peer-to-peer markets, Distributed optimization, Economic dispatch, Community-based markets**

I. Introduction

Increasing deployment of distributed energy resources, prospects for increasing demand response and distributed storage (residential, electricity vehicles, etc.), as well as the rapid progress of ICT-based sensing and control, enable a profound rethinking of electricity markets in a more consumer-centric fashion. Parag and Sovacool in [1] recently classified future consumers market frameworks in micro-grid configurations (both connected to the grid or managed in island mode), community-based structures and peer-to-peer (P2P) trading mechanisms.

Already two decades ago, [2] discussed the idea of peer-to-peer transactions, referred to as coordinated multi-bilateral trades system. There, each agent is to simultaneously negotiate with any other peer based on his own cost function. Actually, depending on the overall objectives of that coordinated multi-bilateral trading, alternative organizations may be seen as most relevant, e.g., involving interconnected microgrids or virtual energy communities. Here we place emphasis on two alternative paradigms represented by a community-based market and by a full peer-to-peer framework. In a Community-based Economic Dispatch (CED), all agents communicate with a supervisory node that coordinates the process to optimality in a distributed manner [3]. In case of a Multi-Bilateral Economic Dispatch (MBED), fully decentralized P2P trades among all participants are obtained without needing third-party supervision [4]. Following [5], a third market structure, called Power Consensus Multi-Bilateral Economic Dispatch (PCMBED), considers a full peer-to-peer negotiation process handled in a distributed fashion by means of a virtual supervisory node.

Consumer-centric markets have substantial advantages, among others product differentiation, consumer involvement and (potentially) low transaction costs. However, if interaction and negotiation mechanisms are not adequately designed, market outcomes may be clearly suboptimal if compared to centralized market structures. Existing works in distributed optimization and coordination of actors on power networks, e.g. [6]–[8], support the proposal of decentralized and distributed algorithms to clear P2P and community-based markets. However, for applications of P2P markets to fully reach their potential, we argue that computation and communication complexity issues must be resolved as they represent one of the main threats to robust system operation. In practice, this may originate from a large number of agents involved in transactions, delays in the iterative exchange of information or simply the number of iterations needed by these iterative algorithms to converge to acceptable solutions. Consequently, we conduct here an extensive computational analysis of some already proposed distributed and decentralized consumer-centric market structures using multiple-core simulations. Eventually, this analysis allows us to draw conclusions on applicability of these approaches to real-world deployment, as well as providing directions for future research.

The paper is structured as following. For self-consistency, Section II briefly introduces our alternative consumer-centric markets, their formulation and solution approaches. After a description in Section III on how test cases are generated, a convergence analysis is carried out in Section IV to assess the trade-off between convergence speed and accuracy. Scaling properties are investigated in Section V as a function of the number of agents involved, considering both computational and communication burden. Resilience to delays in information exchange is analyzed in Section VI. Finally, Section VII gathers conclusions and perspectives regarding future works.

II. Market organizations

To make our consumer-centric market mechanisms comparable, for a given set of agents, all three market structures are based on total cost minimization, where each agent is either a consumer or a producer. All proposed structures aim at solving the economic dispatch problem of a local community that is assumed to be autonomous (no interaction with the system operator or grid services provided). In the case of a

† Centre for Electric Power and Energy, Technical University of Denmark, Kgs. Lyngby, Denmark
‡ SATIE Laboratory, Ecole Normale Superieure de Rennes, France

community-based market, we consider a single price system where agents do not express individual preferences. In contrast in a peer-to-peer setup, the power balance on each trade yields differentiated electricity prices.

A. Community-based market

Since focusing on the computational properties only, we adopt a simplified version of the distributed CED, proposed in [3]. Each agent j of the community Ω is to find its power set-point p_j (positive when generating) aiming to minimize its cost function $f_j(p_j)$. The problem can be formulated as

$$\min_{\Gamma} \quad \sum_{j\in\Omega} f_j(p_j) \tag{1a}$$

$$\text{s.t.} \quad \sum_{j\in\Omega} p_j = 0 \tag{1b}$$

$$\underline{p_j} \leq p_j \leq \overline{p_j} \qquad j \in \Omega \tag{1c}$$

where $\underline{p_j}$ and $\overline{p_j}$ represent the lower and upper power bounds of agent j, respectively, and the community power balance is granted through constraint (1b). Since the objective function and power boundaries are separable among agents, the market is cleared by means of a distributed optimization algorithm, i.e. the *Alternating Direction Method of Multipliers* (ADMM) [9]. A supervisory virtual agent, the so-called community manager, coordinates the negotiation process as in an optimal exchange problem. The solving iterative procedure is summarized by

$$p_j^{k+1} = \underset{p_j}{\operatorname{argmin}}(f_j(p_j) + y^k\Delta^k + \frac{\rho}{2}||\Delta^k - p_j^k + p_j||_2^2) \tag{2a}$$

$$\Delta^{k+1} = \sum_{j=1}^{n} p_j^{k+1} \tag{2b}$$

$$y^{k+1} = y^k + \rho\Delta^{k+1} \tag{2c}$$

where Δ^k represents the power balance residual constraint and y^k the electricity price at iteration k, both being computed by the central agent.

B. Decentralized peer-to-peer based market

The formulation of the decentralized P2P market extends (1), as the power set-points p_j of each agent j are defined as the sum of the power p_{jm} bilaterally traded with a set of partner agents $m \in \omega_j$. As in [4], product differentiation is modeled as an additional trading cost on each bilateral trade $c_{jm}p_{jm}$. The MBED problem can be formulated as

$$\min_{\Gamma} \quad \sum_{j\in\Omega} f_j(p_j) + \sum_{m\in\omega_j} c_{jm}p_{jm} \tag{3a}$$

$$\text{s.t.} \quad p_{jm} + p_{mj} = 0 \qquad j \in \Omega,\ m \in \omega_j \tag{3b}$$

$$p_j = \sum_{m\in\omega_j} p_{jm} \qquad j \in \Omega \tag{3c}$$

$$\underline{p_j} \leq p_j \leq \overline{p_j} \qquad j \in \Omega \tag{3d}$$

$$p_{jm} \geq 0 \qquad j \in \Omega_p,\ m \in \omega_j \tag{3e}$$

$$p_{jm} \leq 0 \qquad j \in \Omega_c,\ m \in \omega_j \tag{3f}$$

The sign constraints on power trades (3e)-(3f) force producers Ω_p and consumers Ω_c respectively to only sell and buy energy. The trading reciprocity constraint (3b) imposes physical feasibility of the trades and allows for product differentiation, reflected by the price of each and every trade. Direct methods such as the ADMM are not easily applied with models such as the MBED, as they do not allow for an effective arbitrage of the different trades inherent to a P2P structure. A *Relaxed Consensus+Innovation* (RCI) method is thus used to solve the optimization problem under the assumption that the cost functions (f_j) have a bijective gradient. Even if C+I methods are slower to converge than direct methods, they present lighter computation and a higher algorithmic flexibility. The iterative process, for a producer, reads

$$y_{jm}^{k+1} = y_{jm}^k - \beta^k(y_{jm}^k - y_{mj}^k) - \alpha^k(p_{jm}^k + p_{mj}^k) \tag{4a}$$

$$\overline{\mu_j}^{k+1} = \left|\overline{\mu_j}^k + \eta^k(p_j^k - \overline{p_j})\right|^+ \tag{4b}$$

$$\underline{\mu_j}^{k+1} = \left|\underline{\mu_j}^k + \eta^k(\underline{p_j} - p_j^k)\right|^+ \tag{4c}$$

$$p_{jm}^{k+1} = \left|p_{jm}^k + g_{jm}^k\big(f_j'^{-1}(y_{jm}^{k+1} - c_{jm} - \overline{\mu_j}^{k+1} + \underline{\mu_j}^{k+1}) - p_j^k\big)\right|^+ \tag{4d}$$

where α^k, β^k and η^k are tuning parameters while y_{jm}^k, $\underline{\mu_j}^k$ and $\overline{\mu_j}^k$ are the dual variables of trading reciprocity constraints and power boundary constraints, respectively. The coefficient g_{jm}^k is a gradient step factor defined in [4] and the operator $|\,.\,|^+$ is the positive part operator (to be replaced by the negative part for consumers). The RCI implementation defines a fully decentralized negotiation process, where all calculations are made locally by each agent.

C. Distributed peer-to-peer based market

We formulate a distributed implementation of (3) through the PCMBED, proposed in [5]. In this formulation, agents focus on reaching consensus on their local trades p_{jm} by means of a global variable z_{jm}. The RCI method (4) is adjusted as (here for a producer)

$$z_{jm}^{k+1} = \frac{p_{jm}^k - p_{mj}^k}{2} \tag{5a}$$

$$y_{jm}^{k+1} = y_{jm}^k - \beta^k(y_{jm}^k - y_{mj}^k) - \alpha^k(p_{jm}^k - z_{jm}^{k+1}) \tag{5b}$$

$$\overline{\mu_j}^{k+1} = \left|\overline{\mu_j}^k + \eta^k(z_j^{k+1} - \overline{p_j})\right|^+ \tag{5c}$$

$$\underline{\mu_j}^{k+1} = \left|\underline{\mu_j}^k + \eta^k(\underline{p_j} - z_j^{k+1})\right|^+ \tag{5d}$$

$$p_{jm}^{k+1} = \left|z_{jm}^{k+1} + g_{jm}^k\big(f_j'^{-1}(y_{jm}^{k+1} - c_{jm} - \overline{\mu_j}^{k+1} + \underline{\mu_j}^{k+1}) - z_j^{k+1}\big)\right|^+ \tag{5e}$$

The z and y updates (5a)-(5b) are operated by the central agent while the others are computed locally by each agent. Implementing this PCMBED approach allows analyzing the benefits of a distributed implementation of a P2P market compared to a decentralized framework, as well as the impact of the intermediary z-update on the RCI process.

III. Simulations description

This section describes the simulation setups and computing framework to showcase the algorithms' properties.

A. Test cases generation

To avoid possible dependencies on contingent combination of assets, we perform Monte Carlo simulations using a sample of ten randomly generated setups. The samples are drawn from uniform distribution and in such way that extreme cases (e.g. low flexibility slope and players with market power) are avoided. At first given the fixed number of agents, the number of producers and consumers are sampled randomly such that there is at least a third of each type. In addition, variations of prices and power set points range are controlled in order to have a resilient tuning. The total consumption and production are sampled randomly within a range that is proportional to the number of agents and split randomly into the individual capacity of each agent. Following a common assumption in literature, the utility curves are assumed quadratic while built according to a price range of flexibility that is sampled randomly. The product differentiation in the P2P structures is here expressed as a preference for local consumption, with trading costs that are proportional to the euclidean distance between two agents calculated from their randomly generated positions on a two dimensional map.

B. Computing infrastructure

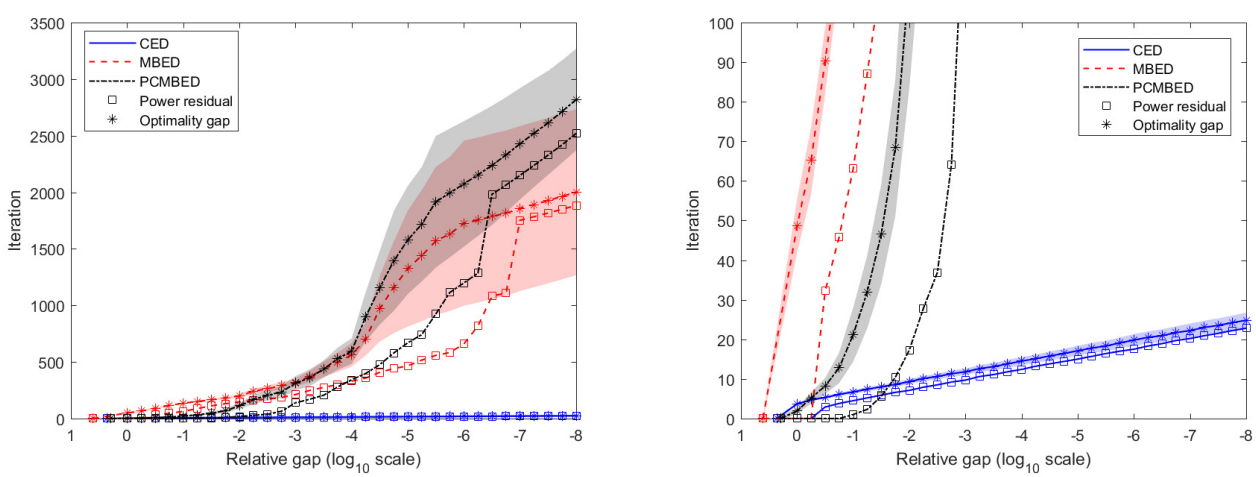
Simulations are executed on a High-Performance Computing machine located at DTU of 2.500 cores in total. For this work only 912 cores are accessible, divided in 38 nodes and connected by 10GB Ethernet cables. Each node is equipped with two Intel Xeon Processor 2650v4 (12 core, 2.20GHz) and 256 GB RAM and 480 GB-SSD disk. To study a more realistic application, we assign each agent of the community to a core of the HPC and we design a communication system through a Message Passing Interface (MPI).

Using a parallel structure for the simulations permits to better describe the agents' actual computational efforts as well as their interactions. By designing an MPI, we model communication architectures to match the three different negotiation processes [10]. For instance, in the CED and PCMBED a master core takes care of the collective computations and communications (master-to-agent and agent-to-master). On the other hand, for the MBED approach all calculations are done by the market participants and the communications are point-to-point. In section IV and V, synchronous communications are implemented through the MPI with blocking functions while in section VI, asynchronous communications are modeled through non blocking functions of the MPI.

IV. Convergence analysis

Emphasis is placed first on convergence properties. The fact that P2P markets reflect individual preferences on the contrary to community-based structure impacts negotiation mechanisms as well as computational efficiency. We benchmark the convergence of each algorithm against centralized implementation formulated as the optimization problems in (1) and (3).

We consider convergence for groups of 25 agents to verify whether and how the investigated algorithms achieve optimality. Simulations' results are only expressed in terms of iterations since synchronous communications are used for these simulations. The results discussed in this section only depend on the algorithm implemented and not on the structure of the implementation (distributed vs decentralized) or on the hardware employed. The average number of iterations required for the different algorithms to reach a given optimality gap is depicted in Figure 1 as well as the power residual used as a description of the feasibility of the solution. Additionally, the surface represents the mean absolute error at each optimality gap. Given the low number of variables of the problem, the CED approach reaches small optimality gaps considerably faster than the P2P market frameworks. The use of the z-



(a) Focus on MBED and PCMBED evolution range

(b) Focus on CED evolution range

Fig. 1: Number of iterations required to reach different levels of accuracy for the proposed algorithms

update (5a) speeds up the initial convergence and decreases the power residual for the PCMBED compared to the MBED. However, the PCMBED appears to be less efficient to reach low optimality gaps. For all algorithms, the optimality gap and of the power residual show similar patterns. This justifies for the rest of the paper, that the optimality gap can be used to describe both the convergence of the algorithm and the feasibility of the solution found.

While the CED shows on average a linear convergence rate with a Mean Relative Error (MRE) of maximum 16%, both algorithms for P2P negotiation display a change in the convergence rate when the optimality gap is below 10^{-4}. This behaviour is mainly caused by few simulations for which the algorithms are much slower to reach small optimality gaps. Indeed, for the MBED the maximum detected MRE for optimality gaps above 10^{-3} is below 25%, while it increases to around 80% for optimality gaps below 10^{-4}. The PCMBED shows a similar behaviour with two simulations that push up the average number of iterations. However, the MRE is consistently between 30% and 60% which implies a more constant dependency of the convergence speed on the setup. The convergence patterns of the P2P markets show that the tuning parameters are not able to cope with all setups efficiently, which could be solved by adaptive tuning parameters as already implemented for the CED.

V. Scaling analysis

Intuitively, prosumer-centric markets such those described will be challenged by increasingly large numbers of participants. In this section, we analyze the ability of the proposed approaches to scale up their negotiation mechanisms by investigating the time complexity of the implemented algorithms [11]. For this reason, we simulate each market framework on different community sizes – from 25 to 300 agents. By assigning each agent to a parallel thread, we limit the maximum number of agents but we introduce communication processes in the performance assessment of each proposed approach. The time complexity T^a of an algorithm $a \in \{\text{CED, MBED, PCMBED}\}$ can be split as

$$T^a(N) = t^a_{alg}(N) t^a_{str}(N). \tag{6}$$

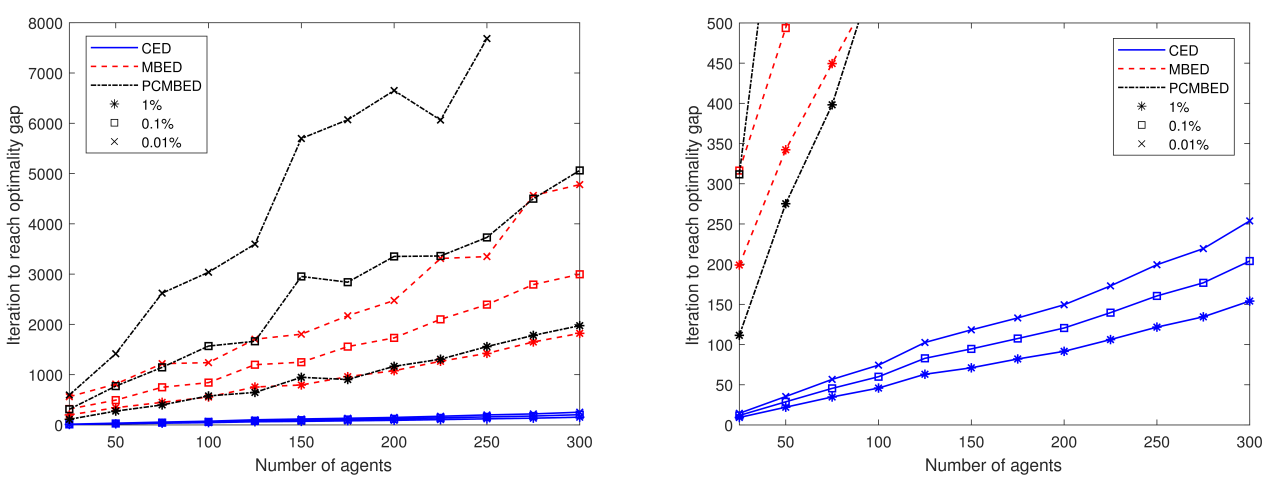
The first factor t^a_{alg}, namely the *algorithmic complexity*, expresses the dependency of the iterations number to convergence on the number of agents N. This term depends only on the algorithm implemented (in this case ADMM or RCI) but not on the structure of the implementation (distributed or decentralized). The second factor t^a_{str}, called the *structural complexity*, considers the size dependency of the computation time of each iteration. For the investigated market frameworks, we here propose empirical or theoretical expressions that we verify empirically of algorithmic and structural complexity (expressed by means of operator $\sim$) as function of the number of agents considered.

A. Algorithmic complexity

As it is difficult to implement a comparable stopping criterion for different algorithms, we investigate the number of iterations for which each algorithm is above a certain optimality gap while increasing the number of agents, as reported in Figure 2. While the CED and the MBED have a low spread between the different optimality gaps, the PCMBED is found more unstable when it comes to higher accuracy of the solution. Even if the benefit of a faster power consensus seems to fade for the PCMBED as the size of the setup increases, we can extrapolate a linear algorithmic complexity for all algorithms $t^a_{alg} \sim O(N)$ (R^2 above 0.95).

B. Structural complexity

We assess the algorithms' structural complexity through the average time to complete an iteration in a synchronous handling of the communication. As this analysis depends mostly on the structure of the implementation, the results of the PCMBED can also be transferred to a distributed implementation of the MBED (i.e. without the z-update). The



(a) Focus on MBED and PCMBED evolution range

(b) Focus on CED evolution range

Fig. 2: Evolution of the number of iterations required to reach optimality gaps of 10^{-2}, 10^{-3} and 10^{-4} over number of agents.

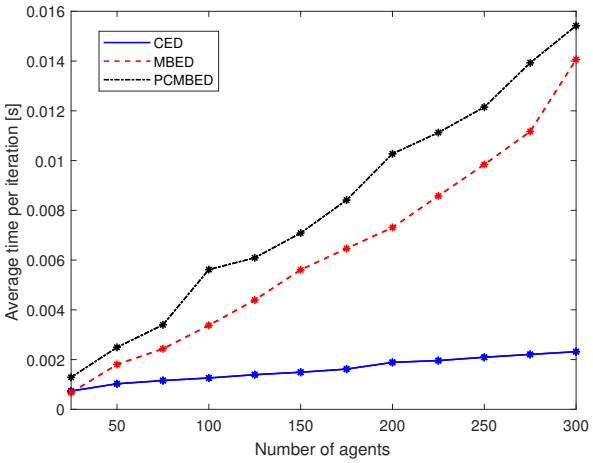


Fig. 3: Impact of scale on the average time per iteration

results, displayed in Figure 3, report a linear trend for all algorithms (R^2 between 0.97 and 0.995). However, in order to transcend from the hardware employed for these simulations and to provide a more general interpretation, we carry out a theoretical analysis of the structural complexity of each algorithm.

The structural complexity is split into computation time and communication time. Under the assumption that the time to communicate a message of size S can be expressed through a linear function $h^a_{com}(S)$, the structural complexity becomes

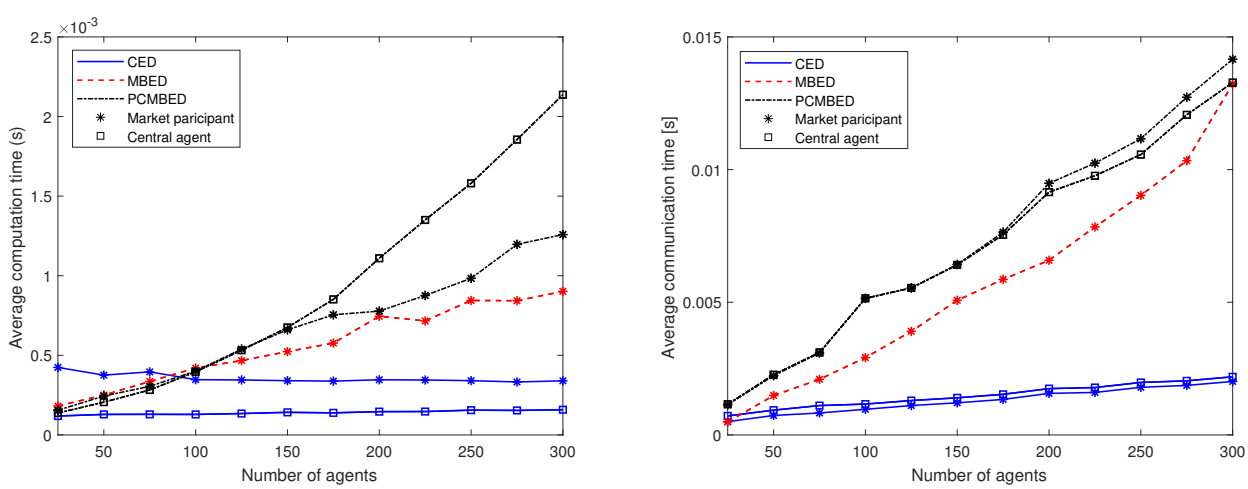
$$t^a_{str} = \delta^a_{com}(N) h^a_{com}(S^a(N)) + \delta^a_{comp}(N) \Delta t_{comp} \quad (7)$$

where δ^a_{com} and δ^a_{comp} are respectively the number of communications and computations needed for algorithm a and Δt_{comp} is the time it takes to complete one operation. In the distributed structures, different complexity can apply for the central agents and for the market participants. However, as synchronous communication are used, the maximum of the two defines the general complexity. From the structure of the algorithms we can extrapolate their complexity, as reported in Table I. It is important to notice that for P2P algorithms, all agents complexity (not the central agent in the case of the PCMBED) depend on the number of trading partners $W_n = \mid \omega_n \mid$ and not on the actual size of the setup N. Even if in our setups the trading partners are comparable to the total number of agents $W_n \sim N$, one could reduce the algorithmic complexity by limiting the number of trading partners per agent.

TABLE I: Structural complexity of the different algorithms.

Model	CED		MBED	PCMBED	
Agent type	Central	Other	Any	Central	Other
$\delta^a_{com}(N)$	O(N)	O(1)	O(W_n)	O(N)	O(1)
$S^a(N)$	O(1)	O(1)	O(1)	O(W_n)	O(W_n)
$\delta^a_{comp}(N)$	O(N)	O(1)	O(W_n)	O(NW_n)	O(W_n)
t^a_{str}	O(N)		O(W_n)	O(NW_n)	

A difference appears between the expected structural complexity of the PCMBED (quadratic) and the empirical results (linear). However, when looking separately at the results for computation and communication time, respectively the average time that each participant takes to compute the local optimization, in Figure 4a, and to transmit its messages, in Figure 4b, the expected results are verified. The computation time of the central agent in the PCMBED shows the expected quadratic increase, but as the values are smaller compared to the communication ones (specific feature of the HPC implementation) the quadratic trend is not perceived in the total complexity. Only some small differences are noted: for instance, the synchronous communications give the exact same trend for all agents (both central and non) in the CED and the PCMBED. The communication time is only slightly dependent on the size of the data transmitted ($h^a_{com}(s) \sim O(1)$) thanks



(a) (b)

Fig. 4: Evolution of average computation (a) and communication (b) time over the number of agents.

to the communication hardware used for the HPC. However, this might not be the case in practice, where different communication infrastructures can lead to more variable time to transmit messages of different sizes.

Overall these results show the importance of the characteristics of the implementation architecture used. An efficient handling of the central agent in the distributed cases reduces the structural complexity, as it does in our simulations for the CED and PCMBED. As for the communication framework, the distributed structure for instance could benefit from an efficient handling of large communications, while decentralized algorithms require sparsified and reliable communication framework to work efficiently.

VI. Resilience to asynchronous behaviour

As presented in Section V, both computation and communication complexity impact on the average time per iteration. When dealing with actual applications, the assumption of synchronous iterations implies that the time of each iteration is dictated by the slowest agent. Computation delays appear in case of non performing hardware or when the optimization sub-problems are complicated to solve, while communication delays are caused by bandwidth limits or internet traffic. The non-negligible likelihood of having significant delays justifies the analysis on how the investigated algorithms behave in case of asynchronous iterations.

For the sake of this study, we model both computation and communication delays. We account for computation heterogeneity by assigning different computation time to each agent. We fix the computation time of the central agent $\tau_C = 0.01$ seconds and sample the computation time of each other agent with a uniform distribution as $\tau_i = \tau_C + U(-\frac{\epsilon}{2}, -\frac{\epsilon}{2})$. We then vary the amplitude of $\epsilon \in [0, \tau_C]$ to investigate the resilience of the algorithms to increasing diversity. By forcing the computation time of each agent with a sleep command, we assume to model different hardware computing power and different complexity of the agents' routine. For this reason, for each simulation the sampled computation time is kept fixed, representing systematic delays in the negotiation mechanism.

Communication delays are modeled as random variables X, following an exponential distribution $\theta = \lambda e^{-\lambda x}$, as proposed in [12]. Since accounting for internet traffic and bandwidth limitations, we sample a new delay for each communication instance. By employing non blocking communication instances in the MPI, the agents can proceed with their optimization routine even if their communication is not finalized. To investigate the robustness towards different sizes of communication delays (simulating weaker and stronger networks), we vary the expected value of the exponential distribution $\mathbb{E}[X] = \frac{1}{\lambda} \in [0, \tau_C]$.

In case of distributed or decentralized systems affected by computation or communication delays, each agent can receive multiple information (e.g. of price and power set point) at each iteration. In order to manage these multiple updates, we implement three different strategies common in literature. As first attempt, we consider only the most recent information received. As we communicate not only the time stamp but also the number of iteration of the sending agent, we can identify the last updated variables. However, this strategy does not exploit all the available information. In order to take into account all updates, each agent can average all the information received at each iteration. Finally, we investigate a compromise between these two approaches by implementing an exponential weighed average over the information received. In this case, the most recent values have a bigger impact, but all the information is taken into consideration.

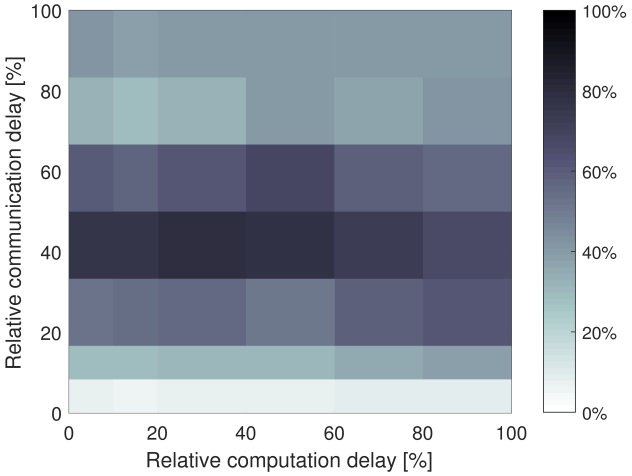
When simulating how the investigated algorithms respond to asynchronous updates, we find the P2P approaches unstable. Both communication and computation delays lead to oscillations of the negotiation process, especially when the bilateral trades have to be finely settled. Further investigation on efficient consensus algorithms in perspective with existing literature (e.g. [13]) are required to increase resilience of the RCI algorithm towards asynchronous behaviours. On the other hand, the CED is found very resilient to both computation and communication delays. The distributed structure of ADMM together with a lower number of variables allow the negotiation process to converge to optimality also when exposed to a highly asynchronous functioning.

The results given by the CED and reported in Figure 5 show that the negotiation mechanisms are generally robust towards asynchronous information updates. The time to reach 0.01% of optimality gap is at maximum doubled if compared to a synchronous system. On one hand the heterogeneity of computation time impacts the algorithmic performance linearly and with small increases, on the other hand communication delays have a more complicated influence. Depending on the strategy used to handle multiple information, the results show that in some cases higher expected values of communication delays speed up convergence.

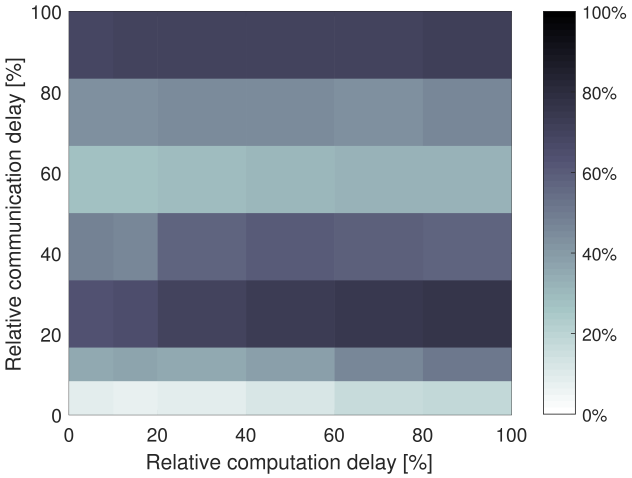
Over all simulations, the strategy that leads to the lowest relative time increase (average 41.1% and standard deviation of 19.3%) is considering only the most recent information. Since convergence is achieved without strong oscillating behaviours, as analyzed in Section IV, using the average of the information received only slows down the process (average time increase of 55.6% and standard deviation of 19.5%). Employing an exponential weighted average on the multiple updates leads to a behaviour in between the two other strategies (average time increase of 50.6% and standard deviation of 19.9%). However, in case of less smooth convergence, this strategy can allow for a good trade-off between filtering the noise of oscillating phenomena and speed of convergence. Further work on an adaptive tuning of the exponential weights employed is needed to achieve robust performances in case of more complicated negotiation mechanisms.

VII. Conclusion

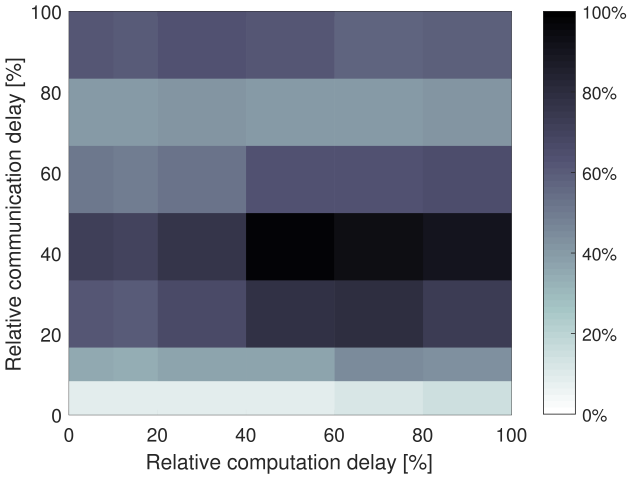
Envisaging future consumer-centric market structures based on distributed communities or peer-to-peer framework, we



(a) Last received information



(b) Mean of received information



(c) Exponential average of received information

Fig. 5: Relative time increases of the three tested strategies to reach a 0.01% optimality gap for different levels of computation and communication delays.

acknowledge that the main challenges towards real world implementations entail scalability and asynchronicity of the negotiation process. In this paper, we assessed computational properties of distributed and decentralized algorithms (ADMM and RCI) for consumer-centric market clearing. By means of parallel programming with a Message Passing Interface (MPI), we investigated their computation and communication complexity as well as their resilience towards computation and communication delays. As expected, we found the community-based distributed approach faster and more robust. However, as peer-to-peer markets are the only framework allowing for product differentiation, we identify the communication matrix sparsity, i.e. the number of possible trading partners, as a possible way to decrease the algorithm complexity and instability.

Further work needs to be carried out on efficient algorithms for consensus as well as on the design of a general market framework accounting for both community-based and peer-to-peer mechanisms. We foresee nested market models as a possibility to inherit both the efficiency and robustness of community-based market and the product differentiation of peer-to-peer mechanisms.

References

[1] Y. Parag and B. K. Sovacool, "Electricity market design for the prosumer era," *Nature energy*, vol. 1, p. 16032, Mar. 2016.

[2] F. F. Wu and P. Varaiya, "Coordinated multilateral trades for electric power networks: theory and implementation," *International Journal of Electrical Power & Energy Systems*, vol. 21, no. 2, pp. 75 – 102, 1999.

[3] F. Moret and P. Pinson, "Energy Collectives: a community and fairness based approach to future electricity markets," *IEEE Transactions on Power Systems*, 2017, working paper.

[4] E. Sorin, L. A. Bobo, and P. Pinson, "Consensus-based approach to peer-to-peer electricity markets with product differentiation," 2017, working paper.

[5] T. Baroche, "Grid Cost Allocation in a Peer-to-Peer Market," 2017, master thesis *(available upon request)*.

[6] G. Hug, S. Kar, and C. Wu, "Consensus + innovations approach for distributed multiagent coordination in a microgrid," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1893–1903, July 2015.

[7] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, and R. Baldick, "Toward distributed/decentralized dc optimal power flow implementation in future electric power systems," *IEEE Transactions on Smart Grid*, vol. PP, no. 99, pp. 1–1, 2017.

[8] A. Conejo, E. Castillo, R. Minguez, and R. Garcia Bertrand, *Decomposition techniques in mathematical programming*. Springer, 2006.

[9] S. Boyd, N. Parikh, E. Chu, and J. Eckstein, "Distributed optimization and statistical learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.

[10] J. Gerk and D. Hausheer, "Peer-to-peer market management," in *Peer-to-Peer Systems and Applications*, R. Steinmetz and K. Wehrle, Eds. Springer, 2005, ch. 29, pp. 491–507.

[11] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Prentice Hall, 1989.

[12] J. Zhang, S. Nabavi, A. Chakrabortty, and Y. Xin, "Convergence analysis of admm-based power system mode estimation under asynchronous wide-area communication delays," in *in Proc. IEEE PES General Meeting*, 2015.

[13] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, "Convergence in multiagent coordination, consensus, and flocking," in *in Proc. of the 44th IEEE Conference on Decision and Control*, 2005.