

Research Paper

Finding the optimal asynchronous decentralized power flow and its impact on cybersecurity in electrical and telecommunication networks

Taqi Mohammed Khattab Al-Rubaye
Department of Medical Laboratory Technology,
College of Medical Technologies,
The Islamic University Najaf , Iraq
kaboos287@gmail.com

Abstract

The power flow calculation is historically carried out by the network operator after the resolution of the electricity market to check that all physical constraints are respected. If this is not the case, the latter encourages market agents to move their operating points until the solution respects the network constraints. This method therefore requires several round trips between the market operator and the network operator to find a feasible operating point for a given future moment. The calculation of the Optimal Power Flow (OPF) makes it possible in a single step to find the optimal solution both from the economic point of view of the market agents, but also from the point of view of the electricity network by ensuring the feasibility of the solution. It introduces the complex voltage variables as well as complex powers in the lines, and ensures that these variables remain within imposed intervals. The OPF has so far been carried out in a centralized manner, i.e. by a single agent. Since the separation of the electricity market and network managers, it is no longer possible to consider that a single entity has all the information necessary to solve the OPF problem. Indeed, the data of the cost functions as well as the power limits of producers and consumers must be communicated if we want to solve the OPF in a centralized manner, which undermines the confidentiality of each agent's data. In addition, with the advent of distributed renewable energies and diverse energy communities, the number of actors that can interact through the electricity network is likely to increase significantly, which makes the centralized OPF calculation more complicated. The decentralization of the OPF makes it possible to overcome these problems by making it possible to scale up, to take into account a large-scale electricity network, as well as the confidentiality of each agent's data. On the other hand, it requires many iterations and communication between agents at each iteration. This makes the decentralized resolution very sensitive to communication hazards.

An asynchronous version is then proposed for each decentralized OPF algorithm, which makes the resolution more robust to communication hazards. Thus, the asynchronous versions of the presented algorithms allow a real implementation less sensitive to the different communication hazards. After presenting the Optimal Power Flow problem in section 1, the centralized

resolution will be explained in section 2 by highlighting its shortcomings for a large-scale implementation. The principle of decomposition of the problem for its decentralized resolution will be presented in section 3. We will then present two decentralized algorithms, one decomposed by nodes in section 4 and the other decomposed by boundaries in section 5. For each of these decentralized algorithms, the asynchronous versions will be introduced and compared to the synchronous versions to show the interest of asynchronous implementations in various conditions of the communication network and various cases with regard to network partitioning.

Novelty : It addresses the communication vulnerability of decentralized OPF by proposing and analyzing asynchronous versions of these algorithms. This means that agents don't have to wait for each other to complete their calculations and communicate. They can operate more independently, making the system more resilient. The article compares the performance of synchronous and asynchronous versions, showing the advantages of the asynchronous approach in realistic communication scenarios. It also describes two different decentralized OPF algorithms, one decomposed by nodes and the other decomposed by boundaries, then applies the asynchronous methods to each. In essence, the article moves beyond the limitations of synchronous decentralized OPF by providing a more practical and reliable solution for real-world implementation, where communication issues are inevitable.

Keywords: Optimal, Power Flow, communication, OPF algorithm.

Introduction

The massive increase in the share of intermittent and difficult to predict renewable energies in the electricity mix leads to major challenges to overcome for the electricity grid. One of the avenues studied to meet these challenges is the more "intelligent" management of the electricity grid, in particular through the Smart Grids paradigm. This paradigm juxtaposes the electricity grid with a communication network and advanced management algorithms, and thus makes it possible to overcome the historical limits linked to the replacement of large non-renewable power plants by a large number of small renewable and distributed power plants[1]. Smart Grids introduce the concept of decentralized management, which makes it possible to coordinate a very large number of agents of the electricity grid, which is difficult to achieve in a centralized version because of the bottleneck of communications to the central agent. In addition, decentralized algorithms offer advantages compared to centralized alternatives, such as data confidentiality and heterogeneous preferences, in particular through the fact that decentralized agents have the power to choose the agents with whom they exchange[2]. Decentralized algorithms fragment a global problem into several local sub-problems solved by dispersed machines. These machines coordinate iteratively by communicating their progress to neighboring machines at each iteration. Despite their advantages, decentralized algorithms rely heavily on the communication network. Indeed, the slightest slowdown in communication impacts the overall progress of the algorithm because each agent must wait for all the messages from its neighbors at each iteration. The definitive loss of a message can even lead to the complete stop of the resolution[3].

In the work described in this article, we seek to quantify the effects of communication network hazards on the resolution of decentralized power grid management algorithms. A thorough analysis of the state of the art shows that these effects are little studied and analyzed, particularly in the case of peer-to-peer configurations that take into account physical constraints. The work presented here attempts to partially fill this gap[4]. We focus in particular on three types of algorithms: a peer-to-peer market algorithm where each market agent communicates exclusively with the agents with which it negotiates, an Optimal Power Flow algorithm that seeks to minimize the costs of agents while taking into account the physical constraints of the power grid, and finally an endogenous peer-to-peer market algorithm that adds an agent to the peer-to-peer market that takes into account the constraints of the power grid. We also introduce the principle of asynchronous algorithm, which makes it possible to limit the waiting times at each iteration, and we study the effects of communication network hazards on the resolution of asynchronous versions of these algorithms.

1- Formalization of the problem

The Optimal Power Flow problem consists in minimizing the costs of all consumers of the network, as in the electricity market problem, while determining the values of the physical variables of the electrical network and ensuring the feasibility of the solution[5].

Thus, the variables considered in the resolution of the OPF are the following:

— the complex voltages at each node n of the network, V_n

— the complex powers of consumers i , $S_i = P_i + jQ_i$

The formulation of the OPF is described in (1).

$$\min \sum_{i \in C} f_i(P_i) \quad (1-1)$$

$$\text{If } P_i + jQ_i \quad (1-1b)$$

$$S_{nm} \in C \quad n \in N, \quad m \in N_m \quad (1-1c)$$

$$\sum_{i \in C_n} S_i = \sum_{m \in N_n} S_{nm} \quad (1-1d)$$

$$S_{nm} = y_{nm}^* V_n (V_n - V_m)^* \quad (1-1e)$$

$$\underline{S_{nm}} \leq |S_{nm}| \leq \overline{S_{nm}} \quad (1-1f)$$

$$\underline{\theta_{mn}} \leq -(V_n V_m^*) \leq \overline{\theta_{mn}} \quad (1-1g)$$

$$(-V_n) = 0 \quad (1-1h)$$

Note There are various formulations depending on the level of network modeling. Here, we consider a line model with a simple admittance, but we could also consider a pi model as in matpower[6], or a transformer followed by an asymmetric pi model as in PowerModels[7]. Equations (1.1b)-(1.1c) represent respectively the constraints on the voltage amplitude at each node and on the complex apparent power of each consumer. Equation (1.1d) describes the constraint of flux conservation at each node, with S_{nm} representing the complex power flowing from node n to node m , the definition of which is given in (1.1e), and the amplitude of which is limited in (1.1f). The difference in voltage angle between two nodes connected by a line is

limited in (1.1g). Finally, the angle of the reference node is fixed in (1.1h). Note that in this formulation, we have chosen to represent producers and consumers in the same form of "consumactors", and to distinguish them from the nodes to which they are connected.

Classically, we find for example equation (4.1d) in the form of

$$S_n^g - S_n^d = \sum_{m \in N_n} S_{nm}$$

S_n^g and S_n^d represent respectively the complex power of the generator at node n and that of the consumer at node n. This formulation does not allow to have several consumers or producers on the same node, and does not allow to have a consumer that can produce during one moment of the day, then consume during another.

It should also be noted that in the literature, consumption is traditionally considered as non-flexible, hence the formulation given in the previous paragraph. In our formulation, non-flexible consumption is represented by consumer actors whose lower and upper bounds are equal $\underline{S}_i = \overline{S}_i$. This can also be transposed to the case of non-flexible producers, such as renewable producers[7].

However, in this case, we can always imagine a degree of load shedding flexibility, where we would then have $\underline{S}_i = 0$ and $\overline{S}_i = S_{i,max}$.

2 Centralized Resolution

2.1 Synchronous Centralized Resolution

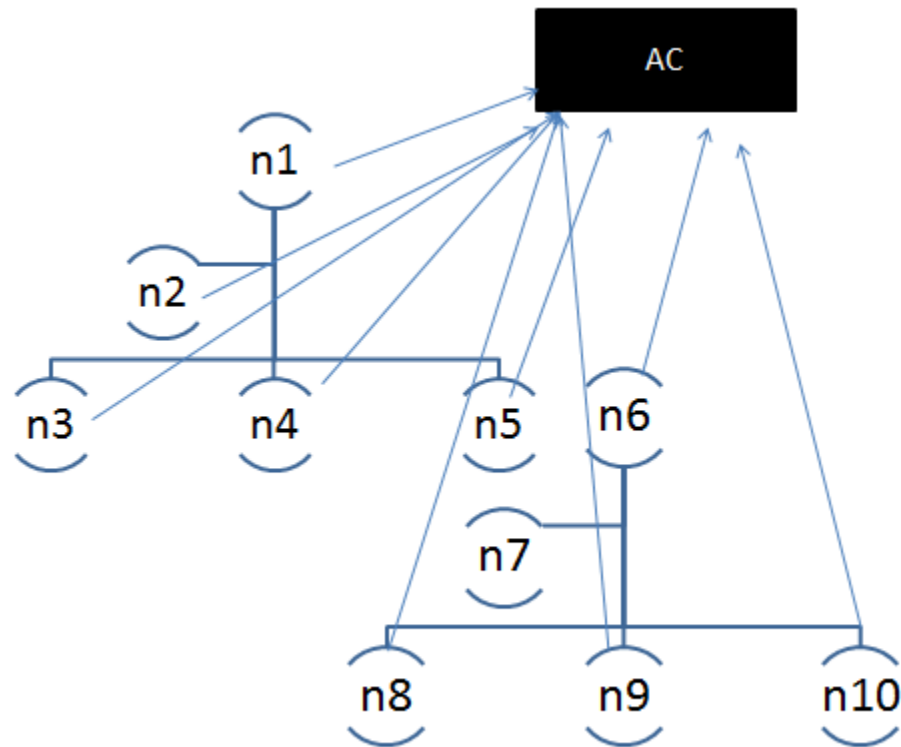


Figure 1 – A network diagram example where the OPF problem is solved by a central agent."

AC. The yellow arrows represent the communication links between the nodes of the network and the central agent.

The centralized resolution of the OPF is done by a central agent, on a single computing machine having all the data necessary to solve the problem, in particular the information relating to:

- the set of nodes N of the electrical network, the lines connecting these nodes and the associated admittance matrix Y ;
- the set of consumers C_n associated with each node $n \in N$, as well as their cost functions and their power limits;
- the limit values of the powers in the lines, of the angle differences of the voltage on either side of each line and of the absolute value of the voltage of each node[8].

The central agent "AC" as well as its communication links with the nodes of a test case are represented in Figure 1. A "synchronous" resolution in the centralized scheme would correspond to the fact that before each resolution of the OPF, all consumers communicate with the central agent their data, i.e. their cost function and their production and consumption limits for a future delivery time. The duration of the centralized resolution

is then composed first of a duration of information collection, secondly of the duration of calculation of the OPF, and finally, of the duration of returning the solution to all agents. In the same way as for an electricity market, the resolution of an OPF is done for a given future delivery time (time T) and makes it possible to determine the values of the powers to be injected and consumed by the flexible agents of the electricity network. It is then necessary that the calculation of the OPF and the communication step from the central agent to the consumers of the market ends before the time T for which the calculation was first done[11].

However, the same difficulties as for the centralized market arise when considering the centralized resolution of the OPF. First, the fact that the central agent collects all the market data causes a communication bottleneck. Also, the duration of the communication steps increases with the percentage of message losses on the communication network, and which is all the higher as the number of agents considered. Finally, the calculation time of the OPF problem increases with the size of the network considered.

Thus, the presence of communication hazards could strongly disrupt the smooth running of centralized calculations. Indeed, if at the time of launching the calculations for a given delivery time, the central agent has not received all the necessary data, it would then be necessary to proceed with a portion of obsolete data which would imply that the solution found is suboptimal or not feasible because it cannot respect the physical constraints of the network[10].

2.2 Asynchronous centralized resolution

In this part, we consider the case of a resolution of the centralized Optimal Power Flow problem where the central agent has not received all the updated data from the consumers.

We apply the calculations on a test case with 118 agents completed by temporal consumption data with a sampling period equal to the minute, from the UMass 1 dataset [99]. We perform the OPF calculation on the 1440 points of a day chosen arbitrarily from the dataset and in application it may even fail to respect the physical constraints of the network given that the input data of the problem are different from the data actually applied by all the agents.

Figure 2 shows the percentage of occurrences (on the 1440 points of the day considered) of the relative gap between the optimal power produced obtained by synchronous centralized resolution and the power produced deduced from the result of the asynchronous centralized OPF, as a function of the percentage of data actually up to date. The quartiles and medians are specified in Table 1

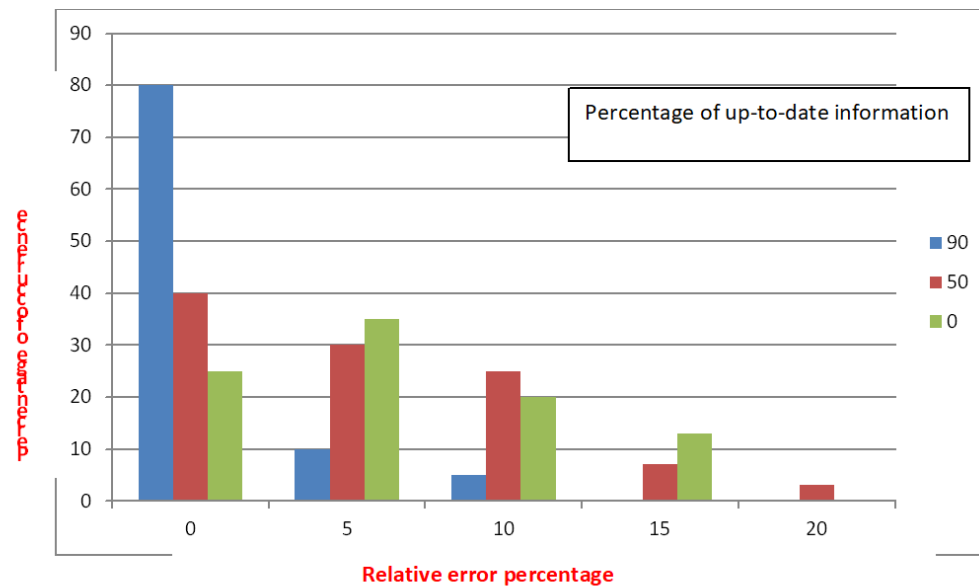


Figure .2 – Histogram of Relative Error in Centralized OPF Results at Varying Levels of Up-to-Date Information

Percentage of data updated	90	50	0
Quartile 25%	0.04%	1.03%	2.39%
Median	0.24%	3.10%	6.17%
Quartile 75%	1%	7.43%	11.32%

Table 1 – Quartiles of Asynchronous Centralized Resolution Relative Errors Based on Data Update Percentage

We observe that the gap between the optimal solution and the real solution increases statistically as the rate of obsolete data increases.

— For 90% of data actually updated, the relative gap does not exceed 1% in three quarters of the cases.

— For 0% of data updated, that is to say that all the data are inherited from the previous time step, then the relative gap is much more dispersed, being able to exceed 11% in a quarter of the cases. This gap is limited by the temporal correlation of the data: the power consumed by an agent at time T has a high chance of being equal to or very close to the consumption at the previous time when the sampling period is small (of the order of a few seconds or even a minute). The gap then increases with the sampling period[13].

If we want to overcome message losses from one time step to another by replacing the missing data with those previously received, then we are exposed to suboptimal solutions that do not necessarily respect the physical constraints of the network. Finally, and even more so for the OPF problem which presents many more variables and constraints than the market problem, the computation time increases significantly when considering large-scale networks, as illustrated in Figure 3 which presents the computation time of the OPF problem as a function of the number of variables considered .

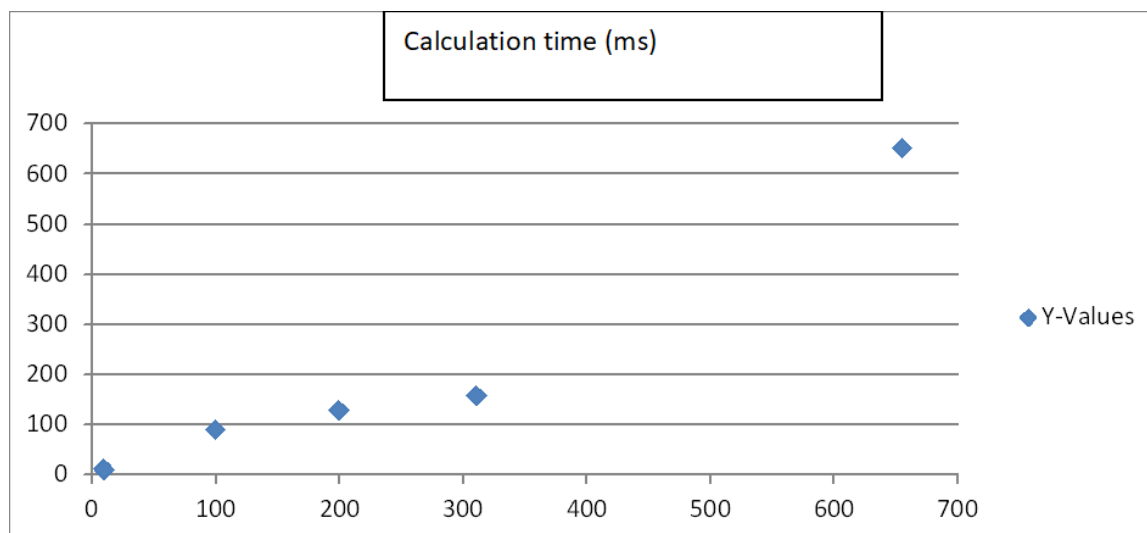


Figure .3 – OPF Computation Time vs. Number of Variables for Different Test Cases

The x and y scales are logarithmic. The number of variables takes into account complex voltages, complex powers in the lines as well as complex powers injected by consumers. shows that the problems of the centralized resolution of the electricity market and those of the centralized OPF are similar. This observation therefore calls for mobilizing the asynchronous decentralization solutions proposed , this time in the context of solving the Optimal Power Flow problem.

3 Decomposition of the problem for the decentralization of its resolution

In order to solve the problems raised in the previous section, we propose to look at decentralized OPF algorithms, that is to say where the calculation of the OPF is no longer solved by a single central agent but by a multitude of agents who each have a role equivalent to the other agents. The electrical network is decomposed into several regions, each region includes a certain number of nodes, most often connected by power lines[14]. We consider that a local machine in each

region has the necessary information to solve a local OPF: the values of the admittances of the lines of the region, as well as the cost functions of the producers and consumers internal to the region. Each region solves a local OPF and communicates with the regions that are neighbors, a neighboring region being defined as having at least one node connected by a physical line to the region considered. We can find the diagram of a partitioning of an electrical network into several regions in Figure 4a. The communication matrix of each region is determined by the presence or absence of physical links between two regions. The decentralized resolution process is iterative, which pushes neighboring regions to communicate with each other at each iteration of the algorithm.

Note that the major difference with the peer-to-peer market problem presented in the previous Chapter, in addition to the higher number of variables and the non-convex constraints, lies in the form of the communication matrix: exchanges are made from one to another, that is, from neighboring regions to neighboring regions until the entire electrical network is covered. On the other hand, in the peer-to-peer market, exchanges are made between consumers regardless of their position on the physical electrical network. It then takes many more iterations to solve the decentralized OPF than to solve the peer-to-peer market because of the propagation of the solution over the entire network[12].

The function to be minimized (4.1a) is a sum of cost functions local to each consumer, therefore local to each node, which makes it easily decomposable. On the other hand, the constraints (4.1d) to (4.1g) are not easily decomposable between the different local problems. It is necessary to create local copies of the complex voltage variables of all the bordering nodes of a region to be able to completely decompose the problem. Several decomposition methods are possible: consensus and innovation (C+I), ADMM, Lagrange multiplier method, etc. We will study in this article two decentralized OPF algorithms based on the ADMM method, the first coming from the article by Erseghe [54], the second being a redundant decomposition by boundaries which allows an easier asynchronization of the algorithm[13].

4 Decentralized resolution - node decomposition

The OPF algorithm from [54] which presents a node decomposition, that is to say that for each node bordering a region, a copy of its complex voltage is created and shared between the different neighboring regions of this node. In his article, Erseghe illustrates the convergence of his algorithm on three test cases, including the IEEE test case with 118 nodes that we will use in the following sections.

4.1 Synchronous formulation

Each region k is represented by a set R_k of nodes of the electrical network. An example of an electrical network as well as its partitioning into three regions is presented in Figure 4a. For a node n , we denote by N_n the set of nodes connected to node n by a power line. Among the nodes of a region, some are internal to the region, that is, they are only connected to other nodes of the same region, n_9 and n_{10} are internal to region 3 for example. Other nodes are border nodes, that is, they have at least one physical connection with a node of another region, for example n_7 and

n_8 are border nodes of region 3. Nodes n_2 and n_6 are also border nodes of region 3 even if they are not part of it. We designate O_k as the set of border nodes of a region k [14].

The decomposition by regions of the algorithm is possible by creating duplicate variables of the border nodes. Indeed, because of constraints (4.1d) to (4.1g), local computations at region 3 require knowledge of the voltage of nodes $n_2 \in R_1$ and $n_6 \in R_2$, and conversely regions 1 and 2 need to know the voltages of nodes $n_8, n_7 \in R_3$ respectively. Figure .4b shows the links between boundary nodes and regions:

if a node n has a link with a region k , then there is a copy of the voltage of node n local to k : v_n^k

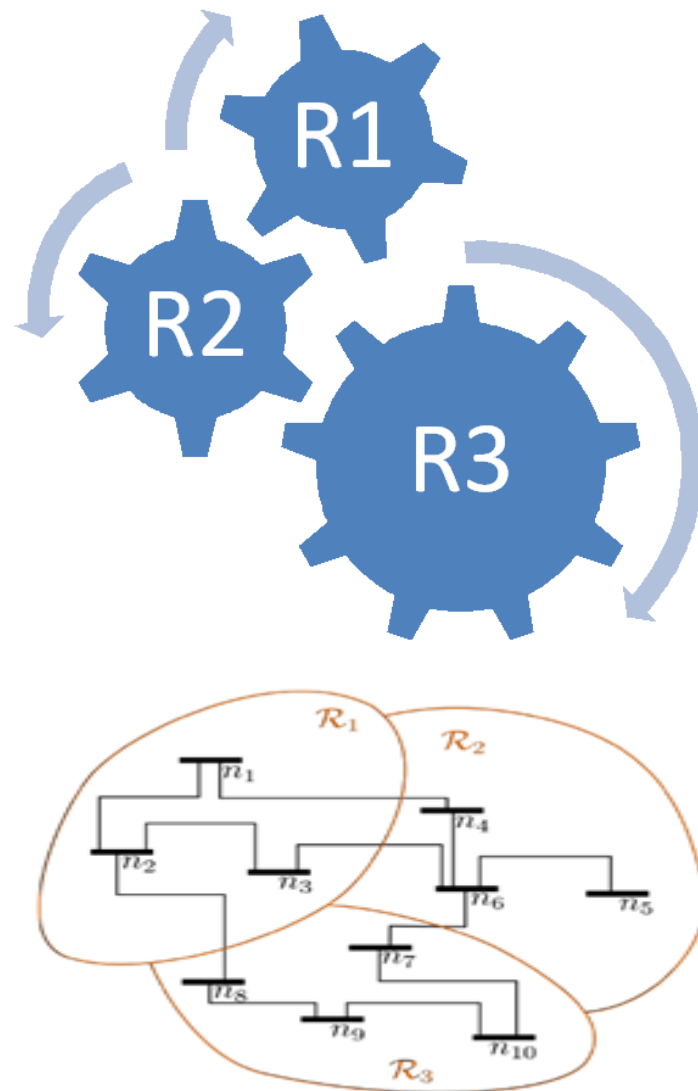


Figure .4 – Power Grid Partitioning: Schematic of a Three-Region Example.

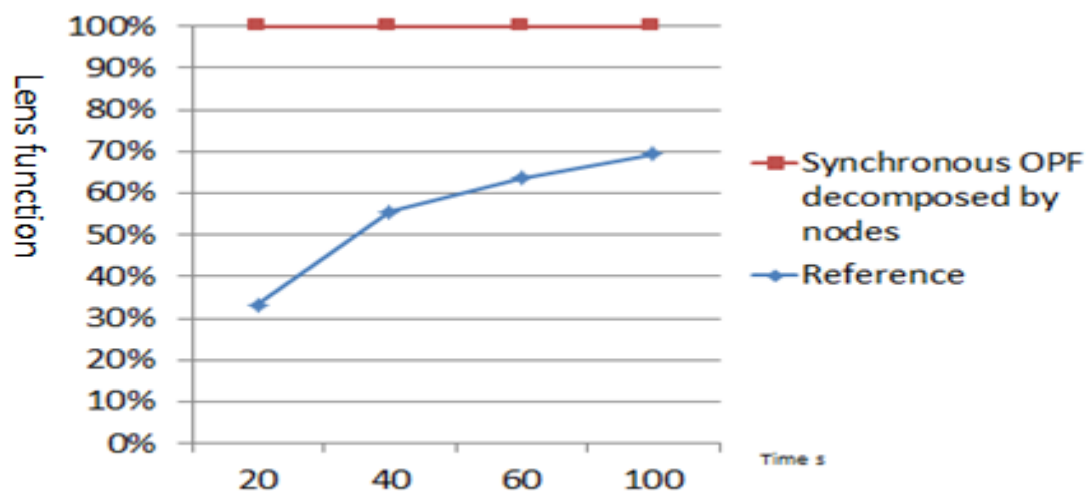
Figure 4a shows the nodes of the power grid and their physical connections via power lines. Figure 4b shows the nodes whose complex voltages are duplicated between the different regions. Any link between a region and a node means that there is an internal variable in the region representing the state of that node.

$$\text{Arg min} \sum_{k=1}^R \sum_{n \in R_k} \sum_{i \in C_n} f_i(P_i) \quad (2)$$

The formulation of the problem including copies of the boundary voltages is given in (2). The physical constraints are found in equations (2.). The equality constraint of the different copies of the composite voltages is given in (2). A consensus is then achieved between the copied voltages using the ADMM method, by setting m_{hn} as the dual variable of each constraint of (2), for any n boundary node and for any region h neighboring node n . The details of the calculations are given in [54]. We then obtain an iterative algorithm where each region proceeds according to Algorithm 1.1.

The copies of the boundary voltages v_{kn} local to region k are updated according to (4.3). This amounts to performing a local calculation of Optimal Power Flow on a part of the electrical network, in particular on the nodes of region k and the neighboring nodes, with a modified cost function (4.3a) presenting a term coming from the local cost functions of each agent and another term coming from the ADMM consensus on the border voltages[15]. Once these copies are updated for all the border nodes $n \in O_k$ of the local region k , they are communicated to all the neighboring regions which also have a copy of the same nodes. Conversely, the local region then waits to receive all the updates of the copies of the neighboring regions. Once all these copies are received, the calculation can continue: we update intermediate values with (4.4) (value of "mixing"), (4.5) (value of "memory") and (4.6).

Application on a test case The presented algorithm is tested dataset with 118 nodes, 53 generators and 186 lines, partitioned into 10 different regions A.II. We consider constant communication delays equal to 50ms, and computation times from the model presented in section 2



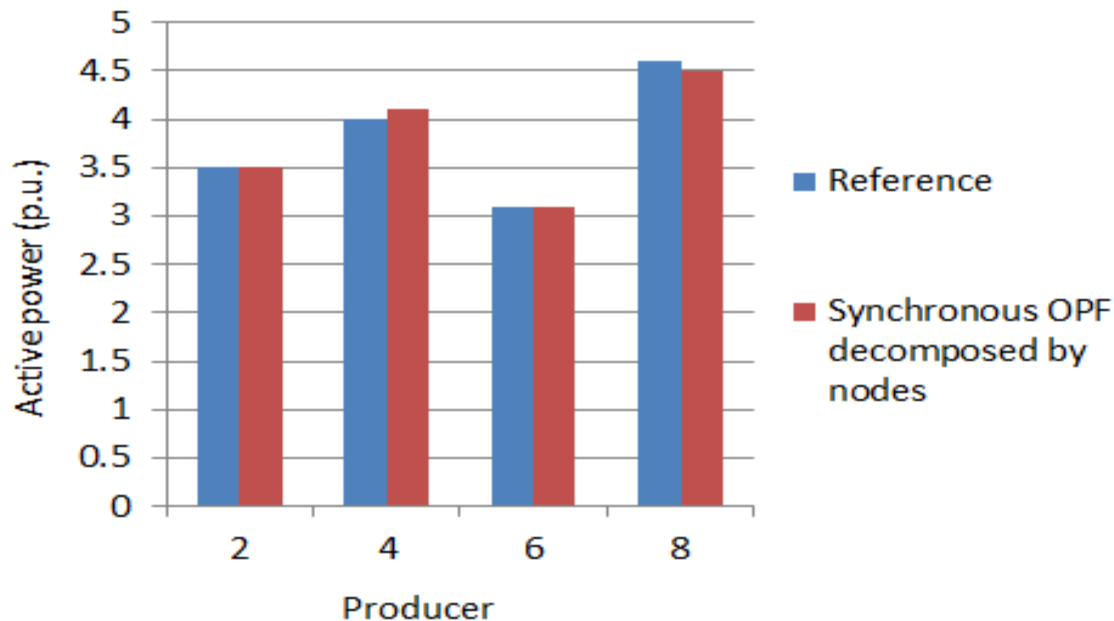


Figure 5 – Decentralized OPF Algorithm Results for 118-Node, 10-Region Test Case: Comparison with Synchronous Centralized Solution

In Figure 5a, the value of the global objective function is displayed over the simulation time. In Figure 5b, the active powers of 9 producers after 60 seconds of calculations are compared to the optimal centralized solution[16].

The results of the synchronous version of the algorithm are given in Figure .5. It shows the value of the global objective function over time as well as the final values of the active powers compared to the optimal solution. It can be observed that the algorithm converges to the optimal solution after about 20 seconds (according to the simulation clock), counting the communication delays and the calculation times. This result will be compared with the asynchronous version of the algorithm.

4.2 Asynchronous formulation

As explained waiting for messages from all neighboring regions at each iteration significantly slows down the progress of the algorithm, especially if the communication delays are varied. We then seek to trigger the calculations at each iteration after receiving only a fraction δ of the expected messages. Messages that have not arrived in time to be processed at the current iteration

will be processed in the following iterations. We have seen that each border node n of a region k could be copied as many times

as the number of regions close to this node. However, the dual variable m_k of the consensus on the voltage v_n is unique regardless of the number of copies. Its update requires a weighted average (4.4) of the difference between the local copy v_k and the neighboring copies v_h . The asynchronization of the algorithm can then be done in two different ways.

— Either we wait for a given number of messages, regardless of the origin and the nodes concerned. In this case, the updates of the voltage v_{kn} and of the associated dual variable m_n are

carried out if at least one message concerning node n is received among the messages taken into account at a given iteration.

— Or we introduce a way of counting the information received that takes into account all the copies of the same node. Let us take an example where three regions k, h, l have copies of the same node n . For region k to be able to carry out the updates of the local variable v_{kn} and of the associated dual variable m_{kn} , then it must have received the latest information v_{hn} and v_{ln} from regions h and l . We then no longer count the number of messages received for the triggering of an iteration, but the number of "activated" nodes, that is to say the number of nodes from which we have received all the copies of the neighboring regions

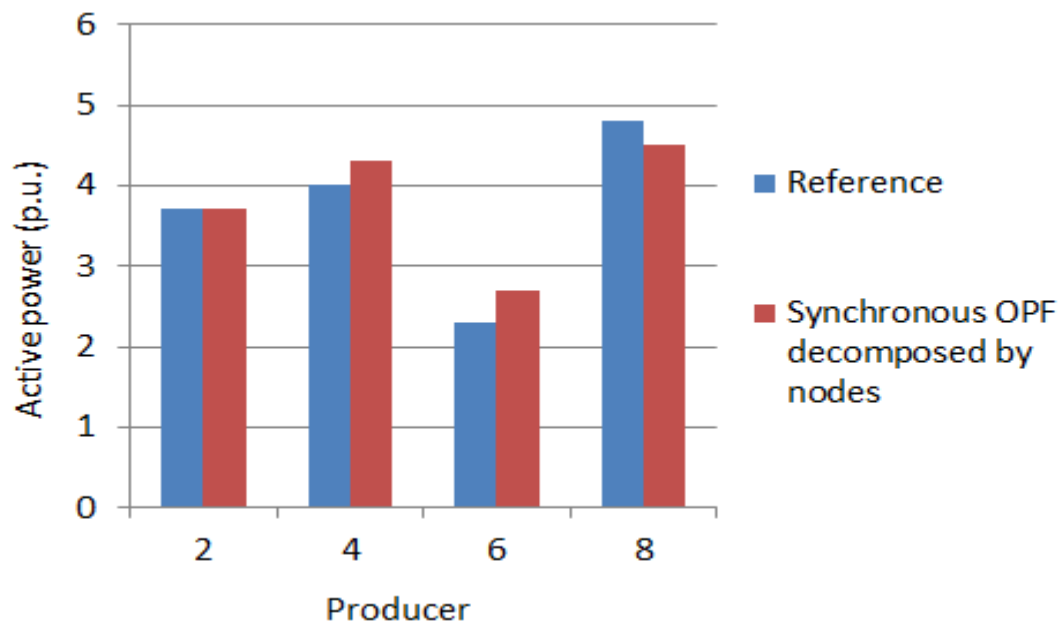
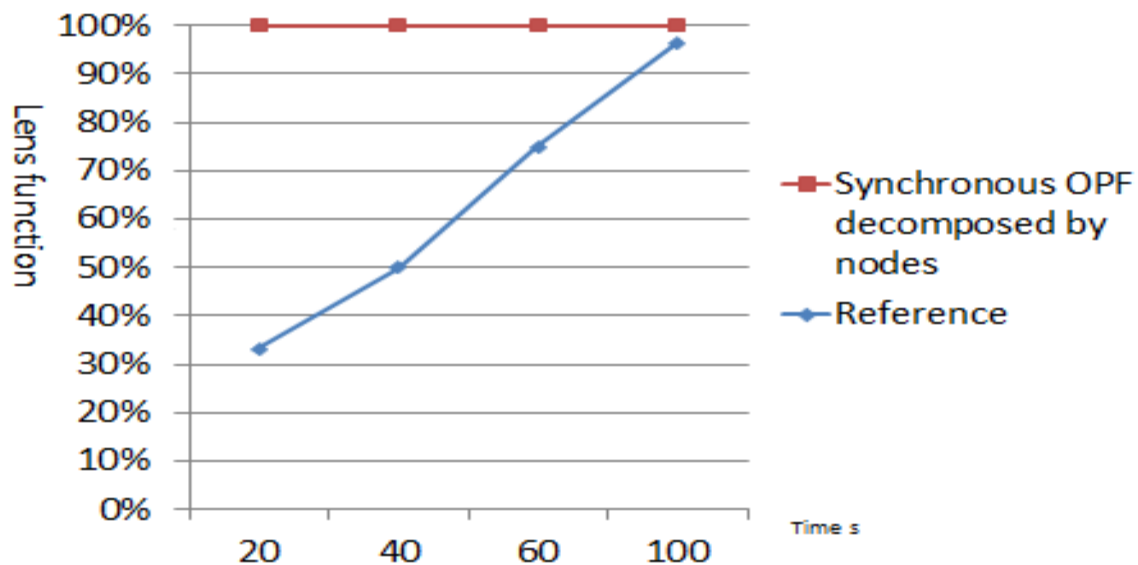


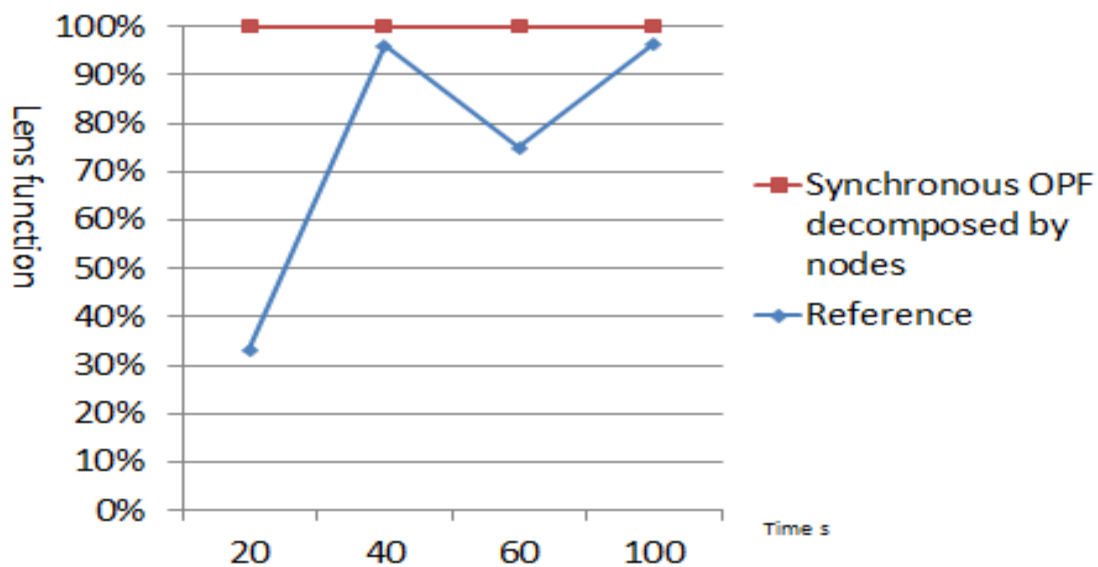
Figure 6 – Partially Asynchronous Decentralized OPF Results for 118-Node Test Case: Comparison with Synchronous Centralized Optimal Solution

The agents are triggered after receiving 10% of the expected messages. In Figure 6a, the value of the global objective function is displayed over time. In Figure 6b, the active powers of 9 producers after 60 seconds of computation are compared with the optimal solution.

By applying the first asynchronous method to our OPF problem, we realize in Figure 6 that, although the algorithm converges, it does not converge to the optimal solution. Indeed, the fact that the intermediate value in (4.4) associated with node n depends on the values v_h of all neighboring regions requires that all these values are effectively updated during the same iteration.

The second method allows to take into account all these values at the same time in (4.4), which amounts to saying that considering a node $n \in O_k$, the operation remains synchronous. On the other hand, considering all the nodes in O_k , it is not necessary to have received all the information from all these nodes to be able to continue the calculations. This is an asynchronous operation but synchronous by node. It is then necessary to modify the trigger of the asynchronous implementation[17].

Figure 7 shows the resolution of the asynchronous algorithm as described in Algorithm 2. We observe that the final solution is indeed the optimal solution sought. From now on, we only refer to the "node-synchronous" asynchronous algorithm when we talk about asynchronous algorithms.



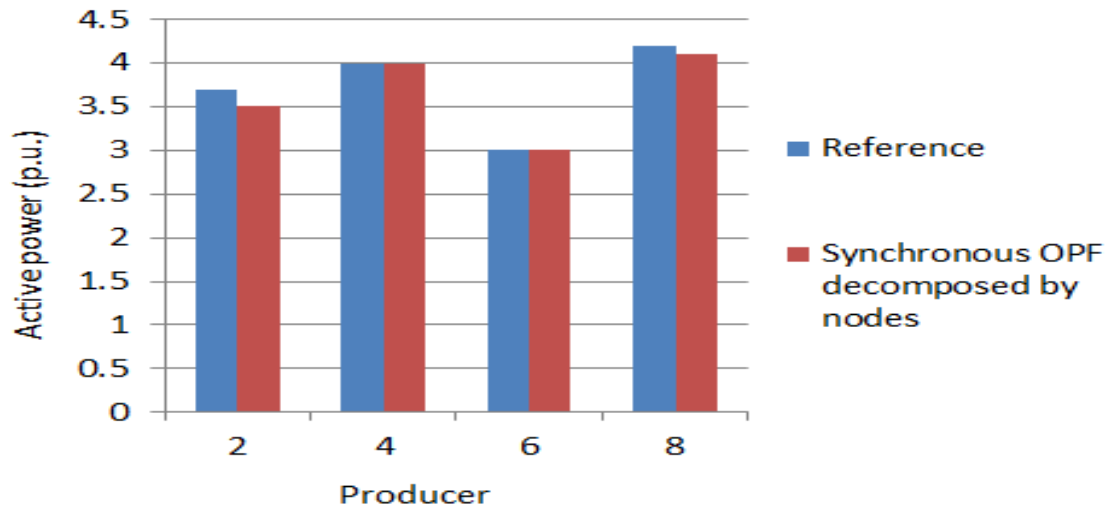


Figure 7 Parameters Used in OPF Algorithm Simulations Decomposed by Nodes

In Figure 7a, the value of the global objective function is displayed over time. In Figure 7b, the active powers of 9 producers after 60 seconds of calculations are compared with the optimal solution. Convergence criterion The definition of primal or dual residues in asynchronous in the case of this particular decomposition is complex since the different regions advance at different rates and are therefore not at the same iteration as a function of time. We then propose in (4.13) a convergence criterion different from the previous chapter, based on the minimization of the error between the known optimal solution s^* and the solution of the decomposed algorithm at a given time s . The algorithm is considered to have converged when the error falls below a threshold ε . Parameters of the studies carried out The following sections describe the different studies carried out on the asynchronous algorithm described above. These studies are applied to a test case with 118 nodes, 53 generators and 186 lines. The local optimization problems of the different regions are solved via the Julia PowerModels package [106] which is based on the Ipopt solver [108] using an interior point method. The different parameters of the communication delay model as well as the computation time model are likely to change during the studies. Their values are compiled in Table 2.

Section	Case	Δ_{comm}	Δ_{calc}
4.4.3	118 nodes 10 regions	$\alpha = 0$ $\beta = 50 \text{ ms}$ Deterministic $P_{pertes} = 0\%$	Affine model 2.2

4.44	118 nodes 10 regions	$\alpha = 0$ $\beta = 50$ ms Deterministic with stochastic losses	Affine model 2.2
4.4.5	118 nodes 10 regions	$\alpha = 0$ Deterministic with stochastic losses Ppertes = 5%	Affine model 2.2
4.4.6	118 knots	$\alpha = 0$ $\beta = 50$ ms Deterministic with stochastic losses Ppertes 5% =	Affine model 2.2

Table 2 synchronous OPF Algorithm Performance: Convergence, Calculation, and Attentive Times vs. Asynchronism Parameter (118-Node Test Case)"

Note that the chosen communication delays are all equal regardless of the regions that communicate with each other, given that the parameter α is always zero. This is a deliberate choice that will allow us to easily compare the communication delays Δ_{comm} to the average computation time later. Given that we consider a small number of large regions, we can assume that the computing machines associated with each region are located in subnetworks that are far from each other, but nonetheless equidistant in terms of communication time. We start by studying the influence of the asynchronism parameter in section 3.

The influence of communication hazards is studied in sections 4 and 5, in particular the influence of message losses and the influence of communication delays compared to computation times. We end with a study of the influence of the size of the regions in Section 6.

4.3 Influence of the asynchronism parameter

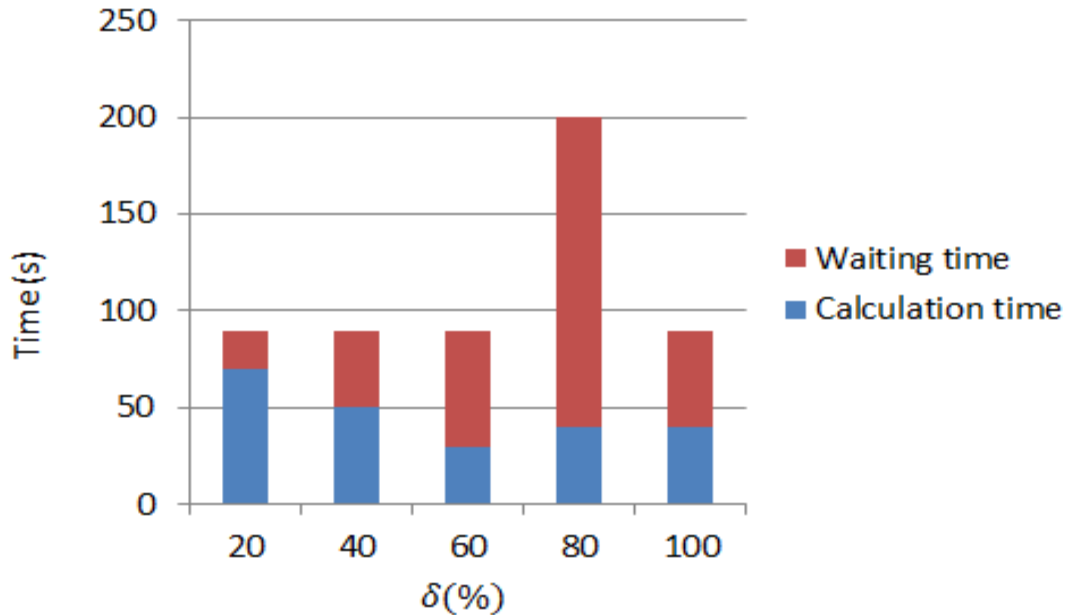


Figure 8 - Convergence Time vs. Message Loss Rate for Asynchronous OPF Algorithm (118-Node Test Case): Impact of $\delta = 10\%$ and $\delta = 100\%$, with Standard Deviation Error Bars".

The influence of the synchronization parameter δ , set in (12), at the convergence temps shown in Figure 8 for a previous start that does not contain any messages. When the calorie temps remain in different regions, as long as the temps enter the iteration. When the algorithm is $\delta = 100\%$, the algorithm is synchronized, which means that all new ones in this region are active before the pouvoir passes through the iteration that suits them. Ainsi, on observant that the algorithm is more rapidly converging in this case[18]. The temps of the calculus tend to increase and the temps of the attention are lowered to the extent that it is fixed. Seulement, the gain réalisé on the temps d'attente n'est pas to compenser the augmentation of the calculus temps to ensure that the algorithm deviates from more and more asynchronously. On observation autour de $\delta = 80\%$ uncomportement that contrasts this tendance. This is at an auto-blocage phone that monitors the attendant areas and receives a message from the home to respond to the calculations in their respects. Then some of the files that appear after sending messages, stay blocked and don't finish with the global advance of the algorithm. We have been able to use a mechanism that allows calculus agents to deblock this situation after a certain delay (i.e. 5 seconds). To observe also a très forte augmentation of d'attente temps donné que ce auto-blocage phénomène se product assez fréquemment pour cette valeur de δ . Augmentation of calculus temps, the number of daily iterations, ensures that the increased asynchronism increases due to the need to combine the information provided by the augmentation system. Number of iterations to bring the whole

family closer together. When you see in the simulation conditions, with this case testing in 10 regions and in the case où all communication connections are strong, the versions with synchronization are more closely aligned with the synchronized version. This result is different from pair to pair as soon as our previous ones are calculating the calculus temps in the past 4 days, and the number of iterations augmentation for the effect of the augmenter The global calculus temps are synchronized.

4.4 Influence of message loss rate

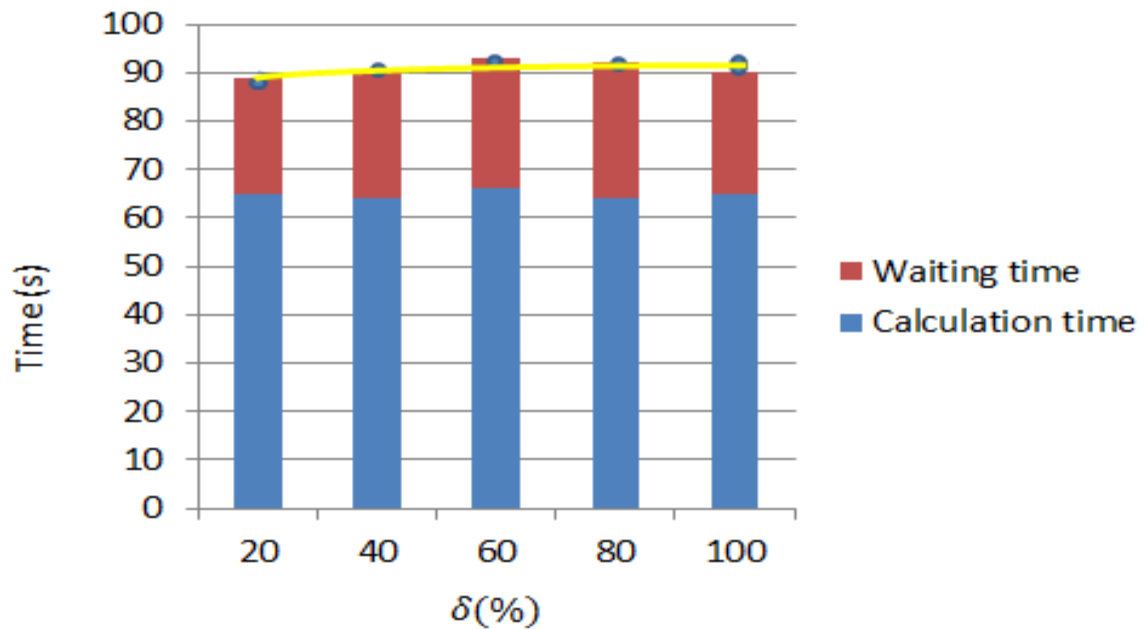


Figure 9 – Activated Node Percentage in Region R4 vs. Communication Delay (Δ_{comm}) for Asynchronism Parameters $\delta = 10\%$, 50% , and 100% (118-Node Test Case)

We now observe the influence of the message loss rate on the convergence time of the algorithm. We recall that the message loss rate is defined in section When a message is lost in the communication network, the agent that sent the message sends it again after a delay equal to a round-trip time between the sender and the recipient. Thus, the total communication delay for a message that has been lost once is 3 times the communication delay without losses. Figure 9 compares the convergence time of the synchronous case at $\delta = 100\%$ to the asynchronous case at $\delta = 10\%$ for different values of the loss rate pperstes. When pperstes is close to zero, few messages are lost, so over all exchanges, the median value of the communication delays remains identical to that where there are no losses. Thus, the communication delays become more varied but retain the same median value. The convergence time of the algorithm increases as the loss rate increases, which is expected. On the other hand, this increase is more pronounced in the synchronous case than in the asynchronous case, which makes the asynchronous algorithm faster from pperstes = 2%. Indeed, the increase in convergence time is on average 11% in asynchronous between pperstes = 0% and pperstes = 10% compared to 70% for the synchronous case. Messages that have been lost at least once must still be waited for by the agents in the synchronous version, which systematically increases the overall waiting time of the algorithm. On the other hand,

these messages are not waited for in the asynchronous version since the agents trigger their iterations before their reception, which limits the increase in the overall waiting time[20]. The error bars in Figure 9 represent the standard deviation of the computation time as well as the waiting time. In the synchronous case, whatever the message loss rate, the average overall computation time is constant. This is explained by the need for a fixed number of iterations to reach the same level of convergence.

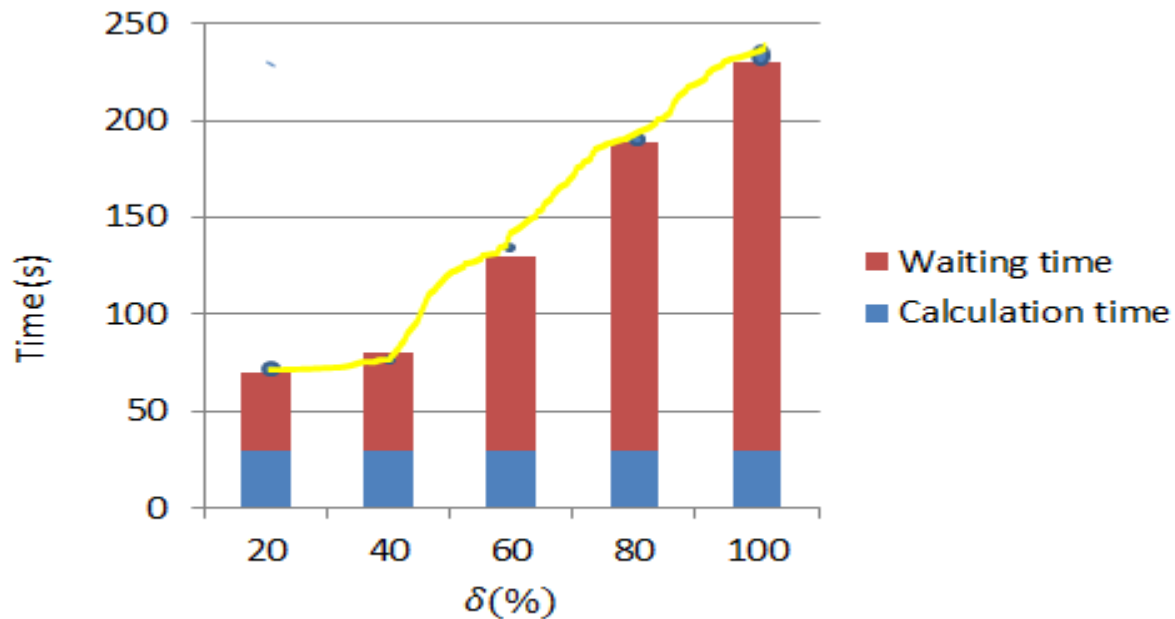


Figure 10 – Convergence time, consisting of the average computation time and the average waiting time, as a function of the communication delay for three values of the asynchronism parameter $\delta = 10\%$, 50% or $\delta = 100\%$, applied to the test case with 118 nodes.

We now observe the influence of the communication delays Δ_{comm} on the convergence time of the OPF algorithm, at a fixed message loss rate $p_{pertes} = 5\%$ chosen arbitrarily. In this study, the communication delay, identical for all communications, Δ_{comm} varies from 25 to 200 ms. Figure 10 presents the average computation times, the average waiting times as well as the average convergence times as a function of the communication delay Δ_{comm} and for three values of the asynchronism parameter $\delta \in \{10, 50, 100\}\%$. We first notice that the convergence time increases as the communication delays increase, which is natural. In the same way as in the previous section, the increase in the convergence time is less in asynchronous compared to the synchronous version of the algorithm. Indeed, by going from $\Delta_{comm} = 25\text{ms}$ to $\Delta_{comm} = 200\text{ms}$, the convergence time increases by 370% in synchronous compared to 168% in asynchronous at $\delta = 10\%$. We can conclude that the asynchronous version is less sensitive to the increase in communication delays than the synchronous version. The average calculation times depend on the number of local iterations, so it is normal to note that they are constant whatever the communication delays in synchronous version $\delta = 100\%$ at equal convergence level. On the

other hand, in asynchronous, we note that the calculation times vary slightly for low communication delays. Indeed, for low communication delays at $\delta = 10\%$, the computation times are slightly smaller than for higher communication delays, which reflects a smaller number of iterations.

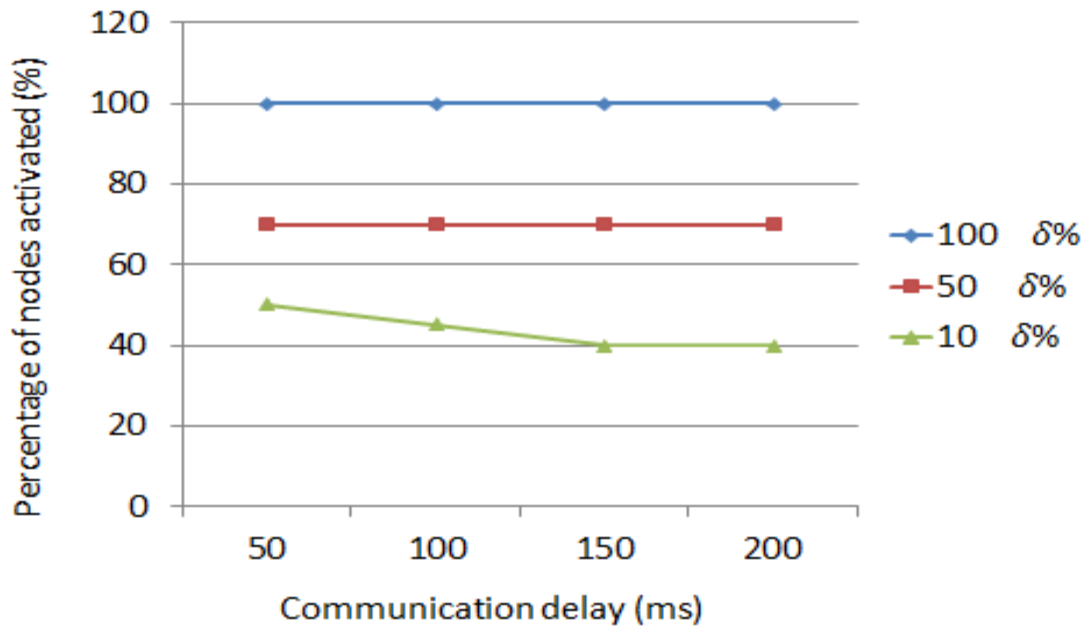


Figure 11 – Percentage of activated nodes of region R4 as a function of the communication delay Δ_{comm} for three values of the asynchronism parameter $\delta = 10\%$, 50% or $\delta = 100\%$, applied to the test case with 118 nodes.

Figure 11 gathers the information necessary to explain this phenomenon. It presents the effective percentage of activated nodes with respect to the communication delay Δ_{comm} . As explained in the note of section .2, the real number of activated nodes is different from the number given on the right of inequality (4.12), which only represents the criterion for triggering the computations. Reducing the communication delays to fixed computation times allows more messages to reach the destination regions before they have finished their previous computations. The node is immediately activated at the end of the local computation because all the expected messages arrived while the agent was computing. Thus, the percentage of effective activated nodes is all the greater as the communication delays are short, as illustrated in Figure 11 for the asynchronous cases. At $\delta = 10\%$, we clearly observe that the percentage of activated nodes decreases with Δ_{comm} until stabilizing after $\Delta_{comm} \geq 150$ ms. This variation reflects the variation in computation time for $\delta = 10\%$ in Figure 10: the more nodes are activated at each iteration, the fewer iterations are needed to reach the same level of convergence.

4.6 Influence of the size of the regions

The resolution of the OPF using the decentralized algorithm decomposed by nodes requires very fine tuning of the parameters related to each region w_k , otherwise one will be faced with a very slow resolution of the problem or even not being able to converge towards the optimal solution.

This makes it difficult to apply the algorithm on a large number of partitionings. We were still able to test it on 4 different partitionings (in 10, 8, 5 and 3 regions), based on the 118-node case, described in Table II.2 in the appendix. These partitionings were built by merging the regions of the 10-region case used so far in previous simulations, which makes it easier to find the wk parameters of the algorithm, which are shared in Table II.2 in Appendix A.II. There are not enough points in this study to draw statistically accurate results, but it still allows us to have a preliminary idea of the influence of asynchronism on different partitionings of the same test case

Asynchronism parameter δ

corresponding to acceleration (%)

Number of regions	Synchronous convergence time (s)	Maximum asynchronous acceleration (%)	Asynchronism parameter δ corresponding to acceleration (%)
3	35	8.27	10
5	47	12.3	30
8	138	12.4	20
10	103	18.8	60

Table .3 – Average convergence time for the synchronous decentralized algorithm, with a message loss rate of $p_{pertes} = 5\%$, and the maximum speedup observed by applying an asynchronous implementation, the speedup being defined as the difference between the asynchronous and synchronous convergence time, divided by the synchronous convergence time. The results in terms of synchronous and asynchronous convergence times are compiled in Table 3. It is observed that the convergence time for the synchronous algorithm increases as the number of regions increases. This is due to the fact that there are more variables copied between regions for which a certain consensus would have to be reached. Regarding the possible time savings brought by the asynchronous implementation of the algorithm (see the "Maximum asynchronous speedup" row of Table 3), it is a priori higher for a higher number of regions. Indeed, since there are more messages to exchange between regions, a synchronism has a greater capacity to reduce waiting times. The maximum acceleration observed is not at the same degree of a synchronism depending on the partitioning. The algorithm would have to be applied to a higher number of partitioning to be able to draw conclusive remarks.

4.7 Conclusions on the asynchronous OPF algorithm decomposed by nodes

"Decentralized OPF utilizes regional computational agents for iterative problem-solving via inter-agent communication. The node-decomposed algorithm [14] requires region-specific parameterization, impacting solution sensitivity and cross-case comparisons. A synchronization, though complex due to message condition requirements for synchronous equivalence, permits

convergence time analysis against a synchronism levels. Homogeneous delays reveal increased asynchronous convergence time due to iterative compensation, despite reduced waiting times. However, asynchronous convergence demonstrates greater resilience to communication delay and message loss variations compared to synchronous methods. Speedup advantages increase with region count, potentially maximizing with fully distributed (node-per-region) OPF." The decentralized Optimal Power Flow (OPF) approach distributes computational tasks among regional agents, who iteratively solve the problem through communication. Our study focuses on a node-decomposed algorithm [14], which demands careful tuning of region-specific parameters, making comparative analysis challenging. A synchronizing this algorithm adds complexity, necessitating message conditions to ensure equivalence with the centralized synchronous solution. Nonetheless, we can examine convergence time relative to a synchronism. With uniform communication delays, asynchronous operation lengthens convergence due to increased computational overhead for information compensation. Conversely, the asynchronous method demonstrates improved robustness against variable communication delays and message losses compared to synchronous execution. The efficiency gain of a synchronization increases with the number of regions, reaching its peak when each node constitutes a separate region, effectively fully distributing the OPF problem." "Implementing a decentralized OPF solution involves regional agents iteratively solving the problem through communication. The node-decomposed algorithm [14], while powerful, presents parameterization challenges that complicate test case comparisons. A synchronizing this algorithm requires careful consideration of message conditions to maintain consistency with the centralized synchronous solution. Despite these complexities, analyzing convergence time in relation to a synchronism provides valuable insights. While asynchronous operation introduces a convergence time penalty under homogeneous delays due to increased computational needs, it exhibits greater resilience to communication variability and message loss. Notably, the asynchronous approach offers significant speedup advantages as the number of regions increases, with potential for maximum efficiency in fully distributed scenarios." "The decentralized OPF algorithm divides the computational load among regional agents, who iteratively solve the power flow problem through communication. Based on a node decomposition [16], it requires region-specific tuning, making cross-case comparison difficult. Adding a synchronicity requires message condition checks to ensure the result is the same as the centralized synchronous one. However, studies on convergence time vs. a synchronicity are possible. With uniform communication delays, convergence time lengthens in asynchronous mode due to added calculations per iteration, even though waiting time decreases. However, the asynchronous approach is more robust to varying communication delays and message loss than the synchronous one. The speedup of asynchronous vs synchronous increases with region count, with maximum speedup in fully distributed systems (one region per node)."

Reference

1. Aslani, Mehrdad, et al. "A novel clustering-based method for reliability assessment of cyber-physical microgrids considering cyber interdependencies and information transmission errors". (2022). doi: 10.1016/j.apenergy.2022.119032. url:

- <https://doi.org/10.1016/j.apenergy.2022.119032>
2. Baringo, Luis and Morteza Rahimiyan. "Virtual Power Plants". In: Virtual Power Plants and Electricity Markets: Decision Making Under Uncertainty. Cham: Springer International Publishing, 2020, p. 1-7. isbn: 978-3-030-47602-1, doi: 10.1007/978-3-030-47602-1_1, url: https://doi.org/10.1007/978-3-030-47602-1_1
 3. Baroche, T., F. Moret and P. Pinson. "Prosumer markets: A unified formulation". In: 2019 IEEE Milan PowerTech, PowerTech 2019, Institute of Electrical and Electronics Engineers Inc., 2019. isbn: 9781538647226, doi: 10.1109/PTC.2019.8810474.
 4. Chenine, Moustafa, Kun Zhu and Lars Nordström. "Survey on priorities and communication requirements for PMU-based applications in the nordic region". In: 2009 IEEE Bucharest PowerTech: Innovative Ideas Toward the Electrical Grid of the Future (2009). doi: 10.1109/PTC.2009.5281956.
 5. Das, Laya, et al. "Measuring smart grid resilience: Methods, challenges and opportunities". In: Renewable and Sustainable Energy Reviews 130 (2020), p. 109918. issn: 1364-0321, doi: 10.1016/J.RSER.2020.109918.
 6. Dorsch, Nils, Stefan Böcker and Christian Wietfeld. "ICT Requirements and Recent Developments". In: Pathways to a Smarter Power System, Academic Press, 2019, chap. 12, p. 343-369. isbn: 9780081025925, doi: 10.1016/B978-0-08-102592-5.00012-0.
 7. Eid, Cherrelle, et al. "Aggregation of demand side flexibility in a smart grid: A review for European market design". In: International Conference on the European Energy Market, EEM 2015-August (2015). issn: 21654093, doi: 10.1109/EEM.2015.7216712.
 8. Gavriluta, Catalin, et al. "Cyber-physical framework for emulating distributed control systems in smart grids". In: International Journal of Electrical Power and Energy Systems 114 (2020), p. 105375. issn: 0142-0615, doi: 10.1016/J.IJEPES.2019.06.033.
 9. Gazafroudi, Amin Shokri, et al. "Evolving New Market Structures". In: Pathways to a Smarter Power System, Academic Press, 2019, chap. 6, p. 183-203. isbn: 9780081025925, doi: 10.1016/B978-0-08-102592-5.00006-5.
 10. Ghasemkhani, Amir, et al. "Optimal Design of a Wide Area Measurement System for Improvement of Power Network Monitoring Using a Dynamic Multiobjective Shortest Path Algorithm". In: IEEE Systems Journal 11.4 (2015), p. 2303-2314. issn: 1932-8184, doi: 10.1109/JSYST.2015.2469742.
 11. Hadjsaid, Nouredine, et al. "Modeling cyber and physical interdependencies - Application in ICT and power grids". In: 2009 IEEE/PES Power Systems Conference and Exposition, PSCE 2009 (2009). doi: 10.1109/PSCE.2009.4840183.
 12. Kirschen, Daniel S. and Goran Strbac. Fundamentals of Power System Economics, 2nd Ed. Wiley, 2018.
 13. Lardier, William, Quentin Varo and Jun Yan. "Quantum-sim: An open-source co-simulation platform for quantum key distribution-based smart grid communications". In: 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2019 (2019). doi: 10.1109/SMARTGRIDCOMM.2019.8909806.

14. Le Goff Latimier, Roman. “Management and dimensioning of a fleet of electric vehicles associated with a photovoltaic power plant: stochastic and distributed co-optimization”. In: IEEE 2011 EnergyTech, ENERGYTECH 2011 (2011). doi: 10.1109/ENERGYTECH.2011.5948542.
15. Liberatore, Vincenzo and Ahmad Al-Hammouri. “Smart grid communication and co-simulation”. In: IEEE 2011 EnergyTech, ENERGYTECH 2011 (2011). doi: 10.1109/ENERGYTECH.2011.5948542.
16. Morstyn, Thomas, et al. “Using peer-to-peer energy-trading platforms to incentivize prosumers to form federated power plants”. In: Nature Energy 2018 3:2 3.2 (2018), p. 94-101. issn: 2058-7546, doi: 10.1038/s41560-017-0075-y, url: https://www.nature.com/articles/s41560-017-0075-y.
17. RTE, Energy Futures 2050 Main Results, tech. report, 2021.
18. Van Der Velde, Dennis, Omer Sen and Immanuel Hacker. “Towards a scalable and flexible smart grid co-simulation environment to investigate communication infrastructures for resilient distribution grid operation”. In: SEST 2021 - 4th International Conference on Smart Energy Systems and Technologies (2021). doi: 10.1109/SEST50973.2021.9543387.
19. Wang, Yu, et al. “A Distributed Control Scheme of Microgrids in Energy Internet Paradigm and Its Multisite Implementation”. In: IEEE Transactions on Industrial Informatics 17.2 (2021), p. 1141-1153. issn: 19410050, doi: 10.1109/TII.2020.2976830.
20. Wang, Yu, et al. “Cyber-Physical Design and Implementation of Distributed Event-Triggered Secondary Control in Islanded Microgrids”. In: IEEE Transactions on Industry Applications 55.6 (2019), p. 5631-5642. issn: 19399367, doi: 10.1109/TIA.2019.2936179