

Algorithmes d'optimisation distribuée pour la gestion de smart grids

Simulation et implémentation

Baptiste HAMARD et Alexandre FAYE-BEDRIN
Ecole Normale Supérieure de Rennes

INTRODUCTION

La gestion de réseaux électriques est une des composantes fondamentales du génie électrique. Elle permet d'assurer la stabilité d'un réseau sur lequel sont connectés des producteurs et des consommateurs. La multiplication d'acteurs non conventionnels sur ces réseaux – énergies renouvelables, charges flexibles, véhicules électriques, et caetera – ainsi que l'essor des technologies de communication ont donné lieu au concept de smart grid reposant sur l'interaction entre un réseau de puissance et un de communication.

Afin d'assurer l'équilibre entre les puissances qui sont injectées sur le réseau et celles qui sont consommées, des problèmes de grande dimension doivent être résolus de manière très rapide. Dans le cadre de grands marchés de l'énergie, plusieurs milliers d'acteurs doivent convenir d'un prix, qui ne sera valide que pour une période de 10 minutes. Pour résoudre ces problèmes efficacement, le recours à des algorithmes distribués est donc primordial afin d'assurer la réactivité du système, ainsi que la confidentialité des données de chaque acteur.

Dans un premier temps, le problème lié à l'équilibre des puissances dans un réseaux sera explicité. Puis, dans un second temps, deux manières envisageables de résoudre ce problème : une version centralisée et un algorithme Peer to Peer. Enfin, un cas d'étude sera étudié pour analyser l'impact des différents paramètres.

I. EXPRESSION DU PROBLÈME

Le problème précédent peut se traduire par plusieurs situations réelles. L'objectif général est d'atteindre l'équilibre des puissances produites et consommées. Cette flexibilité est d'autant plus importante aujourd'hui avec l'essor des énergies renouvelables variables qui multiplient les sources d'incertitudes.

(les éoliennes dépendent du vent, les panneaux photovoltaïques du soleil etc.). Pour cela on doit établir des règles qui déterminent et répartissent la puissance demandée aux agents producteurs en fonction des agents consommateurs. Cependant, si le problème peut être de taille importante, cela doit aussi pouvoir s'adapter à une communauté et une consommation locale de la puissance.

Ce problème posé en partie précédente peut être envisagé de la manière suivante : on dispose de N acteurs a_i , chacun consommant une puissance P_i sur le réseau, et possédant une

fonction f_i qui associe un coût à P_i . Formellement, le problème à résoudre se ramène alors à la minimisation de $\sum f_i(P_i)$ tout en respectant la contrainte $\sum P_i = 0$.

$$\begin{aligned} \min_P \sum_{i=1}^N f_i(P_i) \\ \text{tel que } \sum_{i=1}^N P_i = 0 \\ \forall i, P_i \in [P_i^b; P_i^a] \end{aligned} \quad (1)$$

Les contraintes sont intrinsèquement liées à la puissance produite ou fournie par les agents. Résoudre ce problème sous contraintes est assez complexe, en effet sa dimension peut être très importante. De plus on souhaite que les données des utilisateurs restent confidentielles. On va résoudre ce problème de deux manières différentes :

- Dans un premier temps, on va le résoudre avec un agent central qui s'occupera de calculer la fonction de coût. Chaque autre agent aura des comptes à lui rendre sur la consommation et la production qu'il voit.
- dans un second temps nous allons le résoudre de manière peer to peer, chaque agent communiquera avec les autres et calculera son propre coût pour la puissance fournie en fonction de celle produite.

section Résolution centralisée

A. Alternating Direction Method of Multipliers (ADMM)

L'Alternating Direction Method of Multipliers (ADMM) est un algorithme d'optimisation distribuée qui permet de résoudre les problèmes du type :

$$\begin{aligned} \text{mimimum de } f(x) + g(z) \\ \text{sujet à la contrainte } Ax + Bz = c \end{aligned}$$

avec $x \in \mathbb{R}^n$ et $z \in \mathbb{R}^m$, où $A \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$ et $c \in \mathbb{R}^p$. On suppose que $f(x)$ et $g(z)$ sont des fonctions convexes. La valeur optimale du problème sera notée :

$$p^* = \inf \{ f(x) + g(z) \mid Ax + Bz = c \} \quad (2)$$

On note le lagrangien augmenté :

$$L_p(x, z, y) = f(x) + g(z) + y^T (Ax + Bz - c) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2 \quad (3)$$

L'ADMM consiste à effectuer les itérations suivantes :

$$x^{k+1} := \underset{x}{\operatorname{argmin}} L_\rho(x, z^k, y^k) \quad (4)$$

$$z^{k+1} := \underset{z}{\operatorname{argmin}} L_\rho(x^{k+1}, z, y^k) \quad (5)$$

$$y^{k+1} := y^k + \rho(x^{k+1} + z^{k+1} - c) \quad (6)$$

avec $\rho > 0$. La méthode consiste à minimiser L_ρ par rapport à x (équation 4), puis L_ρ par rapport à z (équation 5) et à mettre à jour y à l'aide d'une descente de gradient. On pose alors k la variable d'itération.

Dans le cas de notre problème, qui est de la forme :

$$\text{minimiser } \sum_{i=1}^N f(x_i) \text{ sujet à la contrainte } \sum_{i=1}^N x_i = 0$$

avec $x_i \in \mathbb{R}^n$, $i = 1, \dots, N$ où f_i représente le coût de la fonction pour le sous système i . C'est un problème de partage où l'objectif partagé g est la fonction indicateur de l'ensemble $\{0\}$. Les composantes du vecteur x_i représentent des quantités échangées par les N différents sous-systèmes. Le problème peut être résolu via l'ADMM en simplifiant les équation et en le traitant comme le problème comme un problème convexe sous contrainte avec :

$$\mathcal{aC} = \{x \in \mathbb{R}^{nN} \mid x_1 + \dots + x_N = 0\}. \quad (7)$$

$$x_i^{k+1} := \underset{x_i}{\operatorname{argmin}} (f_i(x_i) + \frac{\rho}{2} \|x_i - x_i^k + \bar{x}^k + u^k\|_2^2) \quad (8)$$

$$u^{k+1} := u^k + \bar{x}^{k+1} \quad (9)$$

avec $u = \frac{y}{\rho}$ et \bar{x} la moyenne.

Grâce à cet algorithme le problème sous contraintes devient résoluble de manière itérative et fait converger la variable duale réduite u (qui correspond au prix).

L'étape décrite par l'équation (8) est distribuable : la mise à jour de chacune des composantes du vecteur x est indépendante, en ce sens que les calculs ne font pas intervenir des composantes du vecteur différentes. L'algorithme est donc parallélisable et chaque agent peut le faire de son côté sans connaître les informations des autres agents.

La distribution du calcul revient donc à l'implémentation du calcul de x_i^{k+1} sur de manière indépendante, à la mise en commun des résultats, au calcul de u_{k+1} et \bar{x}_i^{k+1} puis à la redistribution de ces deux variables.

Les fonctions de coût peuvent ainsi être connues uniquement des nœuds concernés, qui effectuent les calculs de manière confidentielle et ne communiquent que les résultats.

Cependant ce programme est limité car tous les agents communiquent leur coût avec un agent centralisé or cela peut poser quelques problèmes liés à la confidentialité des données traitées ou à la sauvegarde de ces données (en cas d'accident sur le point névralgique).

II. RÉOLUTION PEER TO PEER (P2P)

Afin de surmonter les problématiques de confidentialité et de passage à l'échelle dans le cas d'un très grand nombre d'acteurs, une résolution P2P est mise en place. Chaque agent établit sa propre variable duale u et communique avec les autres agents. Au final, le coût doit converger vers une valeur similaire à la partie précédente sans qu'un agent central n'ait à traiter toutes les données.

A. Modification du problème initial et algorithme de résolution basé sur l'ADMM

Dans ce cas le problème est un peu différent.

En effet, il n'y a plus d'agent central et chaque agent calcule sa propre fonction de coût.

On regarde la consommation d'un agent i avec un autre agent j . On note $P_{i,j}$ la puissance échangée entre un agent i et un agent j . Le but est alors d'avoir $P_{i,j} = -P_{j,i}$. Si $P = P_{i,j}$ alors on retrouve $P_{i,j} = -P_{j,i}$. Soit le problème suivant :

$$\begin{cases} \min_x & \sum_i f_i(\sum_j x_j) \\ \text{Ct} & P_{i,j} = P_{j,i} \end{cases} \quad (10)$$

Pour obtenir cela on peut poser la contrainte

$$P_{i,j} = Q_{i,j}$$

pour :

$$Q_{i,j} = \frac{P - P^T}{2}$$

On peut alors écrire le lagrangien augmenté du problème pour chaque agent :

$$L_\rho(P_{i,j}, \lambda_{i,j}) = \sum_i (f_i(P_{i,j}) + \sum_j \lambda_{i,j} (P_{i,j} + Q_{i,j})) + \frac{\rho}{2} \|P_{i,j} + Q_{i,j}\|^2 \quad (11)$$

De manière similaire à la partie antérieure, on peut simplifier la résolution de ce problème en deux étapes :

$$\underset{P_{i,j}}{\operatorname{argmin}} (f(\sum_j P_{i,j}) + \frac{\rho}{2} \sum_j (P_{i,j} - Q_{i,j} \frac{\lambda_{i,j}}{\rho})^2) \quad (12)$$

$$\lambda^{k+1} := \lambda^k + \rho(P_{i,j} - Q_{i,j}) \quad (13)$$

Chaque agent va minimiser de son côté la fonction 12, encore une fois en suivant la méthode de l'ADMM. Chaque agent aura donc 3 opérations fondamentales, on peut considérer le problème suivant :

$$\begin{cases} \min_x & \sum_i f_i(x_i) + g(\sum_i z_i) \\ & x_i - z_i = 0, i = 1, \dots, N \end{cases} \quad (14)$$

$$x_i^{k+1} := \underset{x_i}{\operatorname{argmin}} (f_i(x_i) + \frac{\rho}{2} \|x_i - z_i^k + u_i^k\|_2^2) \quad (15)$$

$$z^{k+1} := \underset{z}{\operatorname{argmin}} (g(\sum_i z_i) + \frac{\rho}{2} \|z_i^k - x_i^{k+1}\|_2^2) \quad (16)$$

$$u_i^{k+1} := u_i^k - x_i^{k+1} - z_i^{k+1} \quad (17)$$

B. Application au problème peer to peer

Cette fois-ci, tous les calculs s'effectuent de manière indépendante sur les différents nœuds. Il y a juste une étape d'échange d'informations : entre l'étape décrite en (12) et (13), les différents nœuds s'échangent leurs $P_{i,j}$ de manière à avoir accès aux composantes de P^T qui les intéressent pour calculer les $Q_{i,j}$.

Tel que l'algorithme est défini, on a, par rapport à la méthode de résolution centralisée, *a priori* une confidentialité de la fonction de coût, mais une charge de calcul augmentée pour chacun des nœuds du réseau.

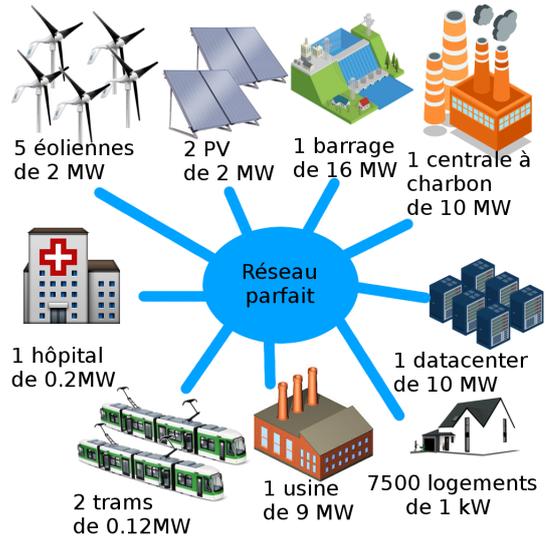


FIGURE 1: Schéma du réseau

III. IMPLÉMENTATION ET ANALYSE DES RÉSULTATS

A. Construction d'un cas d'étude

À un instant donné, on dispose d'un réseau électrique supposé parfait (une plaque de cuivre idéale). On a plusieurs agents connectés à ce réseau, qu'on définit tels que suit :

Producteur	Nombre	Flexibilité	Capacité
Centrale à charbon	1	Faible	10 MW
Barrage réversible	1	Très forte	16 MW
Éolienne	5	Moyenne	2 MW
Panneau photovoltaïque	2	Moyenne	2 MW

Consommateur	Nombre	Flexibilité	Consommation
DataCenter	1	Faible	10 MW
Logement	7500	Moyenne	1 kW
Usine	1	Faible	9 MW
Tramway	2	Faible	120 kW
Hôpital	1	Très faible	200 kW

Chacun de ces agents peut être associé à une fonction de coût, qui à une puissance consommée ou produite associe un coût relatif, c'est-à-dire qui indique les préférences de puissance de l'agent.

Dans notre cas, on décide d'utiliser des fonctions polynomiales de la forme $f(P) = a(P - P_0)^2 + b$, avec :

- P la puissance électrique injectée sur le réseau par l'agent ($P < 0$ représentant alors une consommation)
- P_0 la puissance idéalement fournie par l'agent au réseau
- $a > 0$ un coefficient rendant compte de la pénibilité de s'éloigner de la puissance idéale (pour cause de baisse de rendement par exemple)
- b le coût associé à la puissance idéale de l'agent

Arbitrairement, les unités utilisées par la suite sont les MW pour les valeurs de puissance, et les \mathcal{M} (Mécatreuros) pour les valeurs de coût.

Agent	P_0	a	b
Centrale à charbon	8.5	10	1000
Éolienne	1.9	1	10
Barrage	3	0.1	100
Panneau photovoltaïque	1.9	0.9	11
DataCenter	-10	5	20
Logement \times 7500	-7.5	1	10
Usine	-9	5	8
Tramway	-0.12	6	9
Hôpital	-0.2	200	30

Les agents possèdent des limites intrinsèques de production d'énergie, que les fonctions de coût ne peuvent pas forcément exprimer correctement. On définit donc les limites de production de puissance P_{max} suivantes :

Agent	$P_{max}(MW)$
Centrale à charbon	10
Éolienne	2
Barrage	16
Panneau photovoltaïque	2
Consommateurs	0

En revanche, on considère ici que chaque agent peut consommer sans limite, et que seules les fonctions de coût éviteront à la puissance passant par le réseau de diverger vers des valeurs invraisemblables. On va utiliser ce cas d'étude dans deux scénarii distinct pour tester la flexibilité et la robustesse du réseau :

- 1) La puissance produite est conséquente et une partie de la puissance fournie provient d'énergie renouvelable
- 2) La puissance produite est diminuée car il n'y a plus de vent (diminution de l'énergie renouvelable)

On cherche à appliquer ces algorithmes aux deux différentes méthodes de résolution de manière distribuée pour vérifier la convergence des valeurs.

B. Scénario 1 : Puissance Maximale

1) Gestion par ADMM

: L'algorithme est implémenté en langage C, avec la biblio-

thèque OpenMPI pour prendre en charge la distribution du calcul.

Le programme est lancé sur une machine seule, puis sur une grappe d'ordinateurs pour montrer l'effectivité de l'indépendance des nœuds.

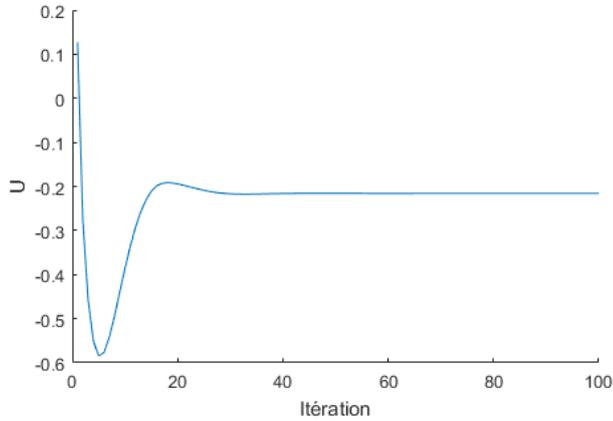


FIGURE 2: Convergence de la variable duale selon la résolution centralisée

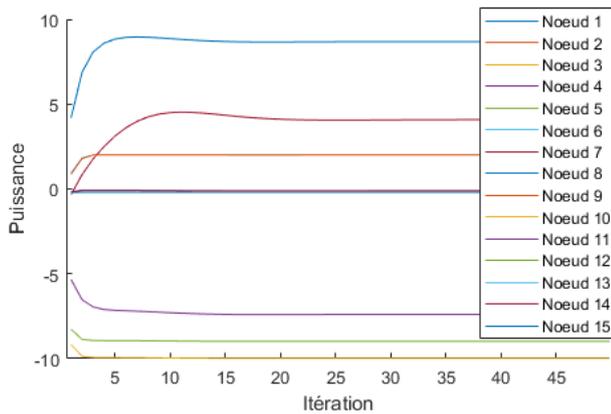


FIGURE 3: Convergence des puissances des agents selon la résolution centralisée

On observe la convergence de la variable duale u à partir de la figure 2 et la convergence des puissances fournies et produite par agent à partir de la figure 3. La variable duale semble tendre vers $-0.22 \text{ M} / \text{Watt}$. Cette valeur sert de comparaison lorsque nous allons traiter la résolution peer to peer. Le nombre d'itération pour la convergence de cette variable duale ou des puissances dépend du coefficient du lagrangien augmenté.

Pour cela nous allons étudier l'impact de ce facteur de pénalité (coefficient de violation de contrainte), repéré ρ . Ce coefficient correspond à la violation de la contrainte définie dans le problème mathématique ($Ax + By - C = 0$). On peut observer l'impact de ce facteur sur le nombre d'itérations nécessaire à la convergence de la variable duale (Figure 4).

On remarque sur la figure 4 que moins l'on viole la contrainte (plus ρ est faible) moins la fonction met de temps pour converger. Il semble d'ailleurs que les prix tendent vers 0 lorsque ρ augmente. Cependant l'impact de cette violation

de contrainte se remarque sur la convergence des puissances. Nous n'avons pas implémenté de conditions d'arrêt à part le nombre d'itérations

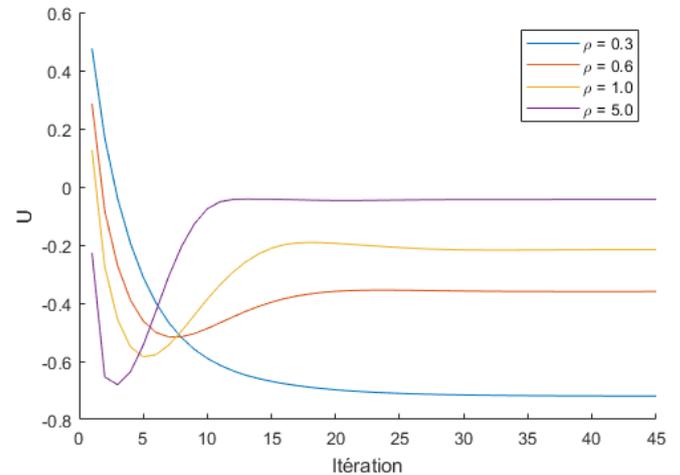


FIGURE 4: Impact de ρ sur la convergence de la variable duale et sa valeur

Plus on viole la contrainte du problème plus il faut un grand nombre d'itération à l'algorithme pour converger. Cependant contrairement au cas de la variable duale, les puissances semblent converger après un certain nombre d'itérations vers une valeur similaire sauf pour des valeurs très grande de ρ (> 20) où les puissances ne convergent plus à cause de l'instabilité numérique, l'algorithme en lui-même devant converger pour toute valeur strictement positive de ρ .

Agent	Puissance fournie	P_0 (MW)
Centrale à charbon	8.68MW	8.50
Éolienne	2.00MW	1.90
Barrage	4.08MW	3.00
Panneau photovoltaïque	2.00MW	1.90
DataCenter	-9.98MW	10.00
Logement $\times 7500$	-7.39MW	-7.50
Usine	-8.98MW	-9.00
Tramway 1	-0.10MW	-0.12
Tramway 2	-0.11MW	-0.12
Hôpital	-0.20MW	-0.20

La puissance totale fournie au réseau est bien nulle, et les valeurs de puissance sont restées dans les limites imposées.

2) Peer to peer

: On applique l'algorithme peer to peer vu précédemment.

Agent	Puissance fournie	P_0 (MW)
Centrale à charbon	8.68MW	8.50
Éolienne	2.00MW	1.90
Barrage	4.08MW	3.00
Panneau photovoltaïque	2.00MW	1.90
Data Center	-9.98MW	10.00
Logement \times 7500	-7.39MW	-7.50
Usine	-8.98MW	-9.00
Tramway 1	-0.10MW	-0.12
Tramway 2	-0.11MW	-0.12
Hôpital	-0.20MW	-0.20

Analysons dans un premier temps la convergence de la variable duales selon les nœuds (Figure 5). Pour les première itération (dépend de la valeur de ρ), on observe sur chaque nœuds une valeur différentes pour la variable duale. Assez rapidement les valeurs convergent vers la même valeur que l'ADMM ($-0.22 \text{ \$/Watt}$).

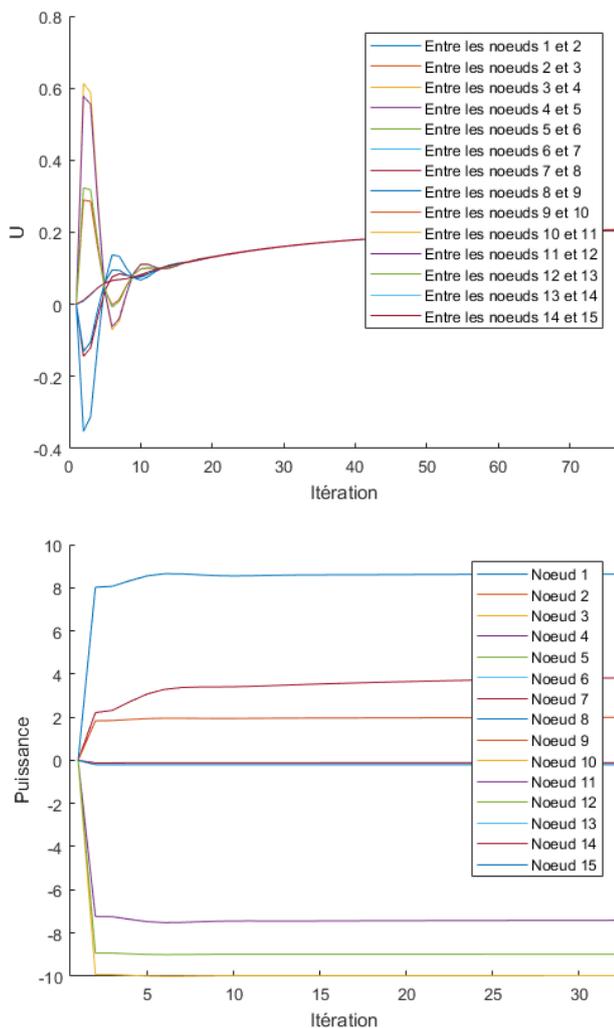


FIGURE 5: Convergence des variables duales et des puissances par nœud selon la résolution peer to peer

On remarque que la résolution peer to peer converge vers des valeurs similaires, aussi bien pour les variables duales que pour les puissances. Cependant on augmente le nombre

d'itérations (à ρ fixé) pour atteindre la valeur de convergence avec l'algorithme peer to peer.

C. Scénario 2 : Diminution des énergies renouvelables

On reprend les valeurs définies pour le scénario, à ceci près que les éoliennes ont une puissance maximum nulle.

1) *ADMM*: Les opérations sont les mêmes que pour l'autre scénario. On obtient les valeurs suivantes :

Variable duale : $-1.72 \text{ \$/Watt}$

Agent	Puissance fournie	P_0 (MW)
Centrale à charbon	9.93MW	8.50
Éolienne	0.00MW	1.90
Barrage	11.59MW	3.00
Panneau photovoltaïque	2.00MW	1.90
DataCenter	-9.82MW	10.00
Logement \times 7500	-6.64MW	-7.50
Usine	-8.82MW	-9.00
Tramway 1	0.00MW	-0.12
Tramway 2	-0.03MW	-0.12
Hôpital	-0.20MW	-0.20

On note que la puissance fournie au réseau est toujours bien nulle au total.

Cette fois-ci, on peut voir des problèmes d'alimentation en énergie : un tramway est arrêté, l'autre fonctionne au ralenti. Plus généralement, la valeur du prix est bien plus importante : cela reflète un rapport offre/demande plus faible. Cela s'explique par un sous-dimensionnement du réseau, déjà évoqué pour le premier scénario, ainsi que par une mauvaise définition des fonctions de coût : si un tramway est un équipement qu'on ne peut pas arrêter tous les quarts d'heure, il faut lui affecter une flexibilité bien plus faible.

2) *Peer to peer*: De même que précédemment, on retrouve exactement les mêmes valeurs de puissance que pour la résolution centralisée :

Variable duale : $1.72 \text{ \$/Watt}$

Agent	Puissance fournie	P_0 (MW)
Centrale à charbon	9.93MW	8.50
Éolienne	0.00MW	1.90
Barrage	11.59MW	3.00
Panneau photovoltaïque	2.00MW	1.90
DataCenter	-9.82MW	10.00
Logement \times 7500	-6.64MW	-7.50
Usine	-8.82MW	-9.00
Tramway 1	0.00MW	-0.12
Tramway 2	-0.03MW	-0.12
Hôpital	-0.20MW	-0.20

Cela conforte les conclusions déjà atteintes : l'algorithme permet de retrouver les mêmes résultats sans ordinateur central, et le réseau n'est pas bien dimensionné.

On note que la valeur la variable duale (correspondant à un prix) est pour la résolution en peer to peer celle en résolution centralisée, au signe près. Cela vient d'un changement de définition, qui n'affecte en rien les valeurs des puissances échangées ni la convergence finale des algorithmes.

La complexité dans le calcul est liée à l'opérateur $x_i^{km} = \text{argmin}()$. Dans la résolution du problème centralisée, on a une complexité $\mathcal{O}(KN)$, avec K le nombre d'itération et N le nombre d'agent. Dans le cas du Peer to Peer, chaque agent communique avec tous les autres agents, et des itération locale (L) sont faite par chaque agent. Le problème semble alors être en $\mathcal{O}(KLN^2)$. Cependant on fixe un nombre maximal d'agent (M) avec lequel un agent communique fixé (qui n'augmentera pas si la communauté s'agrandit). Dès lors le problème est en $\mathcal{O}(KLM)$ ce qui est plus intéressant à résoudre.

CONCLUSION

L'objectif général était d'atteindre l'équilibre des puissances produites et consommées en ayant un réseau fiable et flexible.

La résolution Peer to Peer et l'algorithme centralisé font converger les valeurs de prix et de puissance vers des valeurs semblables. Ces méthodes ont permis la gestion de réseaux électriques complexes tout en assurant leur stabilité.

Cependant, il existe des problématiques liées à l'échange d'informations entre des acteurs du réseau. De plus puisque les tests ont été réalisés à petite échelle, les temps de calculs nécessaires pour gérer un réseau de taille très importante restent inconnus. Enfin il y a les problématiques liées au monde réel, *id est* le réseau n'est pas parfait (nous avons supposé une plaque de cuivre parfaite) et les fonctions de coûts ne suivent pas un polynôme connu.

Ces résultats pourraient être améliorés en effectuant des tests grandeurs nature, ou tout du moins représentatifs des caractéristiques d'un réseau réel pour observer les impacts liés à ce dernier sur les méthodes développées pour gérer ce réseau.

BIBLIOGRAPHIE

- Boyd S., Parikh N., Chu E., Peleato B., & Eckstein, J. (2011). *Distributed optimization and statistical learning via the alternating direction method of multipliers*. *Foundations and Trends® in Machine Learning*, 3(1), 1-122.
- Cramton, P. (2003, January). Electricity market design : the good, the bad, and the ugly. In *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on* (pp. 8-pp). IEEE.
- Barker, B. (2015, January). Message passing interface (mpi). In *Workshop : High Performance Computing on Stampede* (Vol. 262).
- Le Goff Latimier, Roman (2016) *Gestion et dimensionnement d'une flotte de véhicules électriques associée à une centrale photovoltaïque : co-optimisation stochastique et distribuée*. Thèse de doctorat. Paris Saclay.