

Simulation sur GPU de la coordination entre marché P2P et GRD

Béatrice THOMAS^{2,*}, Roman LE GOFF LATIMIER², Abdelhafid EL OUARDI¹,
Hamid BEN AHMED², Samir BOUAZIZ¹

¹Université Paris-Saclay, ENS Paris-Saclay, CNRS, SATIE, 91190, Gif-sur-Yvette, France

²Univ Rennes, ENS Rennes, IETR, CNRS, 35170, Bruz, France

*beatrice.thomas@ens-rennes.fr

RESUME – La transition énergétique nécessite une augmentation de la production d’énergie renouvelable et de la flexibilité des consommateurs. Ces changements interviendront probablement au niveau du réseau de distribution. En outre, la flexibilité des consommateurs devrait être gérée de manière décentralisée afin de respecter la vie privée des personnes. Des solutions pour un marché décentralisé capable de gérer l’état de transport physique existent déjà dans la littérature. Néanmoins, ces implémentations ne sont pas adaptées au réseau de distribution car elles ne s’adaptent pas bien à la taille du problème lorsqu’elles sont simulées sur une seule machine. Dans cette contribution, nous proposons un marché pair-à-pair endogène sur une unité de traitement graphique (GPU) pour permettre la mise à l’échelle de ces algorithmes. L’algorithme repose sur un consensus entre un marché pair-à-pair et l’opérateur du système de distribution qui résout un optimal power flow. Comme le goulot d’étranglement de ce calcul est l’optimal power flow, nous avons utilisé l’accélération apportée par la parallélisation GPU pour pouvoir mener des études sur des cas de test significatifs afin de déterminer les interactions optimales pour accélérer les calculs. Il a été démontré que si la coordination est mal conçue, celle-ci peut empêcher le calcul de converger. Néanmoins, le temps de calcul moyen dans le cas *European Test-Feeder* (environ 900 bus) peut être divisé par deux avec une coordination différente au prix d’un plus grand nombre d’interactions. **Mots-clés** – Marché pair à pair, gestion des congestions, marché endogène, GPU.

1. INTRODUCTION

Pour atteindre la neutralité carbone en 2050, la multiplication des centrales électriques renouvelables et la généralisation de la flexibilité de la consommation sont nécessaires [1]. Chaque consommateur peut utiliser directement cette flexibilité [2], dont les véhicules électriques [3], ou la proposer via l’utilisation d’agrégateurs [4] ou de communautés. Cela conduit à un marché hétérogène avec de nombreux agents pouvant avoir des comportements différents [5]. L’électrification de la consommation entraînera une augmentation de l’utilisation du réseau qui doit rester dans les limites de ses contraintes opérationnelles tout en donnant plus de poids au gestionnaire du réseau de distribution (GRD).

La littérature propose plusieurs algorithmes pour exploiter le réseau de manière sûre et efficace, tels que *security constraints Optimal Power Flow* [6]. Pour permettre la mise à l’échelle de la gestion, la décentralisation a été proposée pour ces algorithmes [7]. Ces gestions sont adaptées à l’implémentation réelle où chaque agent calcule son problème, ce qui permet la mise à l’échelle de la gestion. Néanmoins, cette mise à l’échelle est menacée lorsque ces algorithmes sont utilisés en simulation sur un unique matériel. Or, les simulations sont nécessaires, de manière non exhaustive, pour ajuster les paramètres ou prouver la robustesse de la gestion face à des événements extrêmes. De plus, les algorithmes type OPF calculent les puissances totales optimales et ne gèrent pas directement les marchés locaux ou les échanges directs entre pairs.

Plusieurs articles ont prouvé qu’il était possible d’utiliser du matériel hautement parallèle tel que les GPU pour accélérer la simulation. En effet, certains aspects des problèmes de grille (agents, branches parallèles...) sont intrinsèquement parallèles [8]. Ainsi, le GPU peut être utilisé pour accélérer les marchés pair à pair (P2P) [9], les Power Flow (PF), ou l’Optimal Power Flow (OPF) [11]-[12]. À l’avenir, cependant, on ne sait pas exactement quelles méthodes d’interaction entre les marchés locaux et les GRD seront adoptées à grande échelle. Par conséquent, nous devons conserver une approche flexible en considérant des problèmes génériques tels que le marché P2P [13] et l’OPF. Ce problème générique, qui peut gérer des agents décentralisés tout en tenant compte des contraintes du réseau, sera appelé marché endogène [14], [15].

Les échanges et les flux physiques nécessitent des connaissances différentes de la part des divers agents. Par conséquent, nous adoptons l’approche consistant à coordonner le GRD, qui résout un OPF d’une part, et le marché P2P d’autre part. Cet article étudie donc les conditions de coordination qui permettent la convergence la plus rapide en termes de temps ou de nombre d’itérations. En effet, la coordination peut être trop lente pour converger dans les temps ou même diverger si elle est mal faite. Les résultats s’appliquent à la résolution sur une seule machine (avant déploiement) et, dans une certaine mesure, à la résolution distribuée opérationnelle.

Les principales contributions de ce travail sont les suivantes :

- la présentation d’un algorithme adapté à une implémentation GPU qui permet la résolution d’un ou plusieurs marchés pair à pair qui respectent la contrainte du réseau,
- une étude d’impact de la coordination entre le marché P2P et le calcul du GRD avec différents nombres d’itérations autorisés pour le GRD.
- un framework open source d’un marché P2P endogène sur GPU pour permettre à chacun de faire l’étude d’impact sur son cas.

Le reste du présent document est organisé comme suit. Tout d’abord, la section II présente le problème du marché endogène et l’algorithme utilisé pour le résoudre. La partie suivante sera consacrée à la validation fonctionnelle de l’implémentation GPU et au temps de calcul nécessaire pour différents cas d’étude. Enfin, l’influence de la coordination des GRD sera démontrée dans plusieurs cas d’étude dans la section IV.

2. FORMULATION DU PROBLÈME

2.1. Résolution centralisée d’un marché endogène

En considérant un réseau constitué de B bus, relié par L lignes et N agents. Les agents participent dans un marché pair à pair. On note $\mathcal{N}_i$, l’ensemble des agents positionnés sur le bus i . Les agents peuvent échanger avec leur pairs ω_n , qui peuvent être différents selon l’énergie échangée. La résolution d’un marché avec les contraintes de réseau peut être défini ainsi :

$$\begin{aligned}
\min_{T,S} \quad & \sum_{n \in \Omega} (g_n(s_n) + \sum_{m \in \omega_n} \beta_{nm} t_{nm}) \quad (1a) \\
\text{s.t.} \quad & T = -^t T \quad (\Lambda) \quad (1b) \\
& s_n = \sum_{m \in \omega_n} t_{nm} \quad (\mu) \quad n \in \Omega \quad (1c) \\
& \underline{s}_n \leq s_n \leq \bar{s}_n \quad n \in \Omega \quad (1d) \\
& \underline{t}_{nm} \leq t_{nm} \leq \bar{t}_{nm} \quad n \in \Omega \quad (1e) \\
& \text{contraintes opérationnelles} \quad i \in \mathcal{B} \quad (1f) \\
& \text{lois physiques} \quad i \in \mathcal{B} \quad (1g)
\end{aligned}$$

L'optimisation vise à trouver les transactions optimales t_{nm} qui minimisent les préférences hétérogènes $\beta_{nm} t_{nm}$ et la somme des fonction de coût $g_n = \frac{1}{2} a_n p_n^2 + b_n p_n + \frac{1}{2} a_n q_n^2 + b_n q_n$ pour tous les agents $n \in \Omega$ avec $s_n = p_n + i \cdot q_n$. La détermination de la valeur des coefficients (a_n, b_n) pour les différents agents n'entre pas dans le cadre de cet article. Il convient de noter que les échanges et la fonction coût de la puissance réactive sont utilisés à des fins mathématiques et de généralisation. Chaque agent peut choisir ses préférences hétérogènes en fonction de ses partenaires, ou le réseau peut les imposer. Les contraintes sont la balance commerciale (1b), la relation entre la puissance totale et les échanges pour chaque agent (1c), les limites de puissance (1d), et les limites d'échanges (1e). Les contraintes (1b) et (1c) seront relaxées ; par conséquent, elles sont associées à des variables duales, respectivement λ et μ . Les contraintes du réseau sont composées des lois de Kirchhoff (1g) et des contraintes opérationnelles (1f) telles que les limites de congestion ou de tension. Ces contraintes dépendent du modèle de réseau.

2.2. Organisation du marché endogène pair à pair

Le problème (1) est un problème d'optimal power flow (OPF) qui gère en plus les échanges directs d'énergie entre les agents. Les algorithmes d'OPF peuvent donc être utilisés pour résoudre ce problème d'optimisation. Néanmoins, il sera plus réaliste de considérer que le marché P2P et les contraintes du réseau seront gérés par deux entités différentes qui peuvent ne pas partager toutes les données. En outre, la complexité étant au moins quadratique avec le nombre d'agents $O(N^2)$ (en considérant un marché entièrement connecté) [17], il est préférable de résoudre le marché P2P de manière décentralisée [9]. Le système à résoudre sera considéré comme suit :

- Le marché P2P trouvera les échanges et la puissance de chaque agent,
- Le GRD résoudra un OPF pour trouver les puissances proche du marché qui respectent les contraintes du réseau,
- Ensuite, les deux parties communiquent et calculent à nouveau jusqu'à ce qu'elles parviennent à un consensus.

La contrainte de consensus ajoutée est (avec η la variable duale associée) :

$$s_n^{[GRD]} = s_n^{[P2P]} \quad (\eta) \quad n \in \Omega \quad (2)$$

Une méthode des multiplicateurs à direction alternée (ADMM), [18] est utilisée pour résoudre itérativement ce problème dit global pour chaque agent n et le GRD en parallèle. Cette organisation permet de conserver en privé les informations de tous les agents. En outre, plusieurs marchés peuvent être tenus en parallèle, et la gestion du réseau est totalement à la discrétion du GRD. Le calcul du marché devient :

$$\begin{aligned}
\min_{T,S} \quad & g_n(s_n) + \sum_{m \in \omega_n} \beta_{nm} t_{nm} \\
& + \frac{\rho}{2} (s_n - \frac{s_n^{[GRD],k} + s_n^{[P2P],k}}{2} + \eta_n^{[P2P],k})^2 \quad (3) \\
\text{s.t.} \quad & (1b) - (1e)
\end{aligned}$$

où l'optimisation du côté GRD est :

$$\begin{aligned}
\min_S \quad & \frac{\rho}{2} \sum_{n \in \omega} (s_n - \frac{s_n^{[GRD],k} + s_n^{[P2P],k}}{2} + \eta_{GRD}^{[GRD],k})^2 \quad (4) \\
\text{s.t.} \quad & (1f) - (1g)
\end{aligned}$$

La mise à jour de la variable duale est :

$$\begin{aligned}
\eta_n^{[P2P],k+1} &= \eta_n^{[P2P],k} + \frac{1}{2} (s_n^{[P2P],k} - s_n^{[GRD],k}) \\
&:= -\eta_n^{[GRD],k+1} \quad (5)
\end{aligned}$$

Les lecteurs doivent noter que $\eta_n^{[P2P]}$ et $\eta_n^{[GRD]}$ représentent la même variable duale (avec une valeur opposée). Deux notations sont utilisées ici pour conserver des expressions similaires des fonctions de coût dans (3) et (4). Pour parvenir à un consensus entre le marché et le GRD tout en tenant compte des pertes, le GRD doit intervenir sur le marché. Ce travail considère que l'agent 0 est le GRD, visant à acheter les pertes calculées par l'OPF. Pendant le calcul du GRD, cet agent n'injecte ni ne soutire de puissance du réseau, car cela entraînerait un double comptage des pertes. Nous considérons donc que $s_0^{[GRD]} = s_{loss}^{[GRD]}$. Du point de vue du marché, le GRD (ou agent 0) est un agent (un consommateur) qui ne se distingue pas des autres.

2.3. Résolution du GRD

Un avantage de cette organisation est que la convergence est indépendante de l'algorithme utilisé par la résolution GRD tant que chaque OPF converge vers le même optimum. Cependant, lorsque la précision de la résolution du GRD ou le nombre d'itérations est limité, l'algorithme utilisé pour l'OPF pourrait avoir un impact. L'algorithme et les hypothèses utilisés dans cet article seront donc présentés dans cette section pour des raisons de reproductibilité.

Nous considérons un réseau de distribution radial où la ligne i va au bus i et vient du bus A_i . Ainsi, on note v_i , s_i , l_i , S_i , respectivement, la tension au carré et la puissance injectée sur le bus i , le courant au carré et les flux de puissance sur la ligne i (ou qui arrivent au bus i).

Ainsi les contraintes du réseau (1f)-(1g) sont :

$$\sum_{n \in \mathcal{N}_i} s_n = s_i \quad i \in \mathcal{B} \quad (6a)$$

$$v_{A_i} - v_i + z_i S_i^* + S_i z_i^* - l_i |z_i|^2 = 0 \quad i \in \mathcal{B} \quad (6b)$$

$$\sum_{j \in C_i} (S_j - l_j z_j) + s_i - S_i = 0 \quad i \in \mathcal{B} \quad (6c)$$

$$|S_i|^2 = v_i l_i \quad i \in \mathcal{B} \quad (6d)$$

$$(v_0, S_0) = (v^{ref}, 0) \quad ref \quad (6e)$$

avec v^{ref} étant la tension de référence du réseau (habituellement $v^{ref} = 1pu$) et x^* étant le conjugué complexe de x . Pour résoudre le problème du GRD, l'OPF décentralisé proposé dans [19] et optimisé sur [12] sera utilisé. La contrainte (6d) est relaxée pour obtenir un problème d'optimisation convexe, et devient $|S_i|^2 \leq v_i l_i$. Cette formulation limite les données échangées car les agents ne doivent envoyer leur puissance qu'au bus où ils se trouvent.

2.4. Décentralisation de la résolution

Cette partie vise à expliquer comment (1) peut être décentralisé. Une démonstration complète peut être lu dans [14]-[9]. En bref, la contrainte (1b) est relaxée en utilisant un Lagrangien

TABEAU 1. Temps de calcul (s) et nombre d'itération sur différents cas pour un marché endogène

Cas			CPU		GPU	
Nom	N	B	temps	k	temps	k
Matpower 10ba	11	10	0.04	24	2.0	23
Matpower 69	49	69	33	4285	11	4234
Matpower 85	61	85	16	40	4.2	42
TestFeeder	57	905	540	18	14	19

TABEAU 2. Parameters for the simulation

Matpower 85		European TestFeeder	
Paramètres	valeur	Paramètres	valeur
k_{max}	4000	k_{max}	20000
j_{max}	5000	j_{max}	5000
ϵ_g	0.005	ϵ_g	0.05
ϵ_x	0.001	ϵ_x	0.005
ρ_g	1	ρ_g	1
ρ_l	ρ_g	ρ_l	ρ_g

La mise en œuvre du marché P2P utilise le code de [9], et la résolution GRD utilise l'OPF de [12]. Les lecteurs intéressés peuvent trouver les détails des algorithmes et des adaptations pour GPU dans [9]-[12]. Le consensus est réalisé en mettant directement à jour la fonction de coût et la variable duale sur le GPU. Les transferts entre le CPU et le GPU ne sont effectués qu'au début et à la fin de l'algorithme. Pour ces tests, le GRD résout un OPF à chaque itération du marché. Nous commençons par résoudre un marché P2P sans contraintes de réseau pour initialiser le marché endogène. La précision requise est de 10^{-3} . Les résultats pour $\rho = 10$ sont rassemblés dans le Tab. 1. avec un nombre maximum d'itérations de 10 000

Ces résultats montrent que le nombre d'itérations dépend fortement du cas étudié. Cela peut s'expliquer en partie par le fait que ρ est choisi de manière identique pour tous les cas (et peut être plus ou moins optimal selon le cas). Les deux implémentations ont des nombres d'itérations différents, probablement en raison de l'instabilité numérique de l'OPF qui entraîne des précisions différentes sur le CPU et le GPU [12]. Néanmoins, les deux marchés endogènes convergent vers le même optimum. Les résultats montrent que le GPU peut réduire de manière significative les coûts de calcul pour les plus grand cas.

Le temps relatif pour chaque bloc en fonction du processeur (CPU ou GPU) et du cas est représenté dans la Fig. 1. La résolution du marché sans contrainte de réseau est séparée du reste de l'initialisation. L'initialisation prend plus de temps sur le GPU en raison du transfert de mémoire entre le CPU et le GPU. À l'exception de Matpower 10ba sur GPU et de Matpower 69 sur CPU, le calcul DSO est dominant dans le temps de simulation. Cependant, ce calcul DSO, qui est à pleine précision à chaque itération, induit un calcul potentiellement inutile qui sera étudié dans la section suivante.

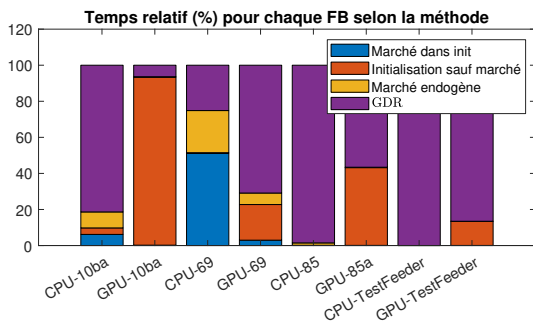


FIG. 1. Répartition des temps de calcul entre le GRD, le Marché, l'initialisation du marché ou du reste pour tous les cas sur CPU et GPU

4. COORDINATION MARCHÉ P2P ET GRD

4.1. Présentation

Dans cette section, nous étudierons l'impact de deux paramètres sur la convergence de cet algorithme. Tout d'abord, comme la résolution des OPF prend beaucoup plus de temps à calculer qu'une itération du marché, nous calculerons la partie GRD toutes les *step* itérations. D'autre part, il peut ne pas être utile de calculer un OPF très précis à chaque itération k du problème. Ainsi, le nombre d'itérations *iter* pour le calcul de l'OPF sera limité.

L'influence de ces paramètres sera analysé dans plusieurs études de cas. Tout d'abord, le cas Matpower 85 bus sera utilisé, puis la même analyse sera menée sur plusieurs dates de la base de données *European TestFeeder*. Ce cas comporte beaucoup plus de lignes que d'agents. Ainsi, le GRD et la coordination du marché doivent être modifiés pour empêcher le GRD de ralentir la résolution du marché. Les paramètres de l'algorithme peuvent être consultés dans le Tab. 2.

Cet article testera toutes les combinaisons de paramètres, de pas *step* = (1 : 1 : 12) et du nombre d'itérations *iter* = {1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000}. Ainsi, un total de $12 \cdot 12 = 144$ jeux de paramètres sera représenté sous forme de carte thermique. L'interpolation du graphe Matlab a été utilisée entre ces points. L'implémentation GPU sera utilisée pour calculer plus rapidement que le CPU.

4.2. Résultats sur des cas fixes

Cette section étudie l'impact des paramètres sur le nombre d'itérations globales k et le temps de calcul. Dans une simulation sur un seul matériel, le nombre d'itérations globales k n'a pas d'importance; seul le temps de calcul compte. Néanmoins, dans une implémentation distribuée (implémentation réelle ou simulation sur un cluster), k itérations globales impliquent k communication pour le marché, $k/step$ communication entre le GRD et le marché, et donc, $k/step$ calculs d'OPF.

Les figures Fig. 2 et Fig. 3 montrent le temps de calcul et le nombre d'itérations pour tous les jeux de paramètres sur le cas Matpower 85. Le temps de calcul dépend fortement du nombre d'itérations autorisées pour la résolution GRD. Néanmoins, en simulation, le *step* a peu ou pas d'impact sur le temps de calcul. Cependant ce paramètre influence le nombre d'itérations nécessaires pour converger, et donc le nombre de communications. Pour résumer la Fig. 2, le temps de convergence dépend directement et exclusivement de la valeur de *iter*. La Fig. 3 montre que pour un *step* trop élevé, le nombre d'itération nécessaire pour converger augmente. Cependant tous les jeux de paramètre

testés permettent une convergence en moins de 2000 itérations.

Les résultats moyens pour les 30 premières minutes du *Test-Feeder* européen diffèrent de ceux obtenus précédemment en raison de la taille différente du problème (moins d'agent et plus de bus). La figure 4 montre que le temps de calcul dépend du nombre d'itérations et de la taille du pas. Les simulations avec un *iter* plus faible et un *step* plus élevé sont plus rapides. Néanmoins, la Fig. 5 montre que de nombreux jeux de paramètres font que la simulation ne converge pas dans le nombre d'itérations autorisé (ici, 20 000). La zone où la simulation ne converge pas en 20 000 itérations n'est pas convexe. En effet, la simulation converge pour *step* = 7 et *iter* = 50 mais pas pour *step* = 7 et *iter* = 100 ou *step* = 7 et *iter* = 20. Il faut donc faire attention au choix des paramètres pour assurer la convergence.

4.3. Résultats sur des cas aléatoires

Dans cette partie, le réseau du cas Matpower 85 sera utilisé avec différents nombres d'agents N dans le marché P2P. Pour chaque nombre d'agents et chaque ensemble de paramètres, 20 configurations sont générées de manière aléatoire pour les fonctions coût des agents. Seule la valeur moyenne de ces 20 configurations est examinée par la suite. Le temps de convergence

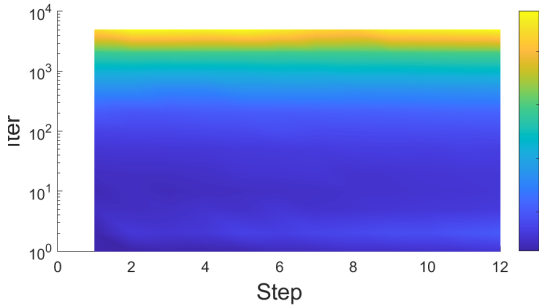


FIG. 2. Temps de calcul (s), cas Matpower 85

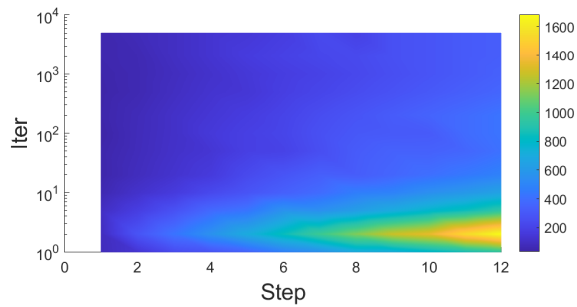


FIG. 3. Nombre d'itération k , cas MatPower 85

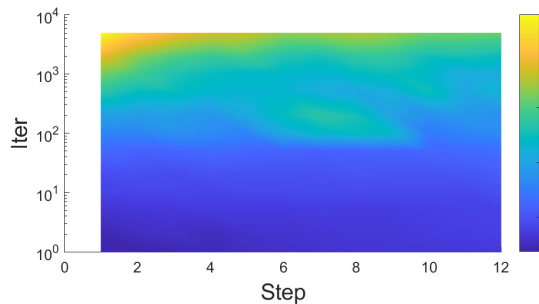


FIG. 4. Temps de calcul moyen (s), cas *TestFeeder*

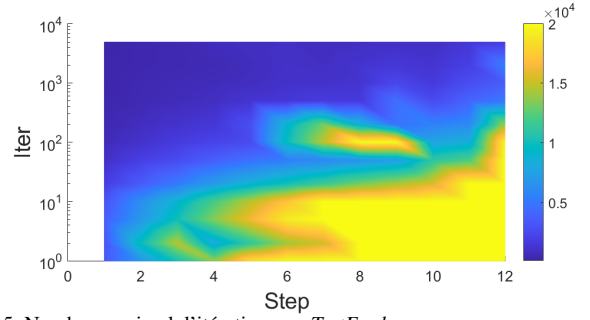


FIG. 5. Nombre maximal d'itération, cas *TestFeeder*

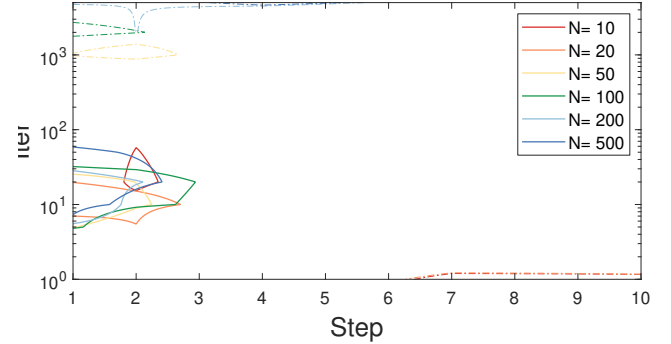


FIG. 6. Isolignes comprises dans les 10% pour meilleurs (ligne) et pire (pointillé) du temps de calcul selon le nombre d'agent

et les valeurs du nombre d'itérations pour les meilleures et les pires configurations sont présentés dans le Tab. 3.

Dans la Fig. 6, pour chaque nombre d'agents N , l'isoline correspondant à 110% du temps de simulation optimal est représentée par une ligne continue. De même, l'isoline correspondant à 90% du pire temps de convergence mesuré est représentée par des lignes pointillées. Quel que soit le nombre d'agents, la simulation est plus rapide pour ce type de cas avec *step* = {1, 2} et *iter* entre 10 et 20. En revanche, les paramètres conduisant au pire temps de simulation diffèrent selon le nombre d'agents. Pour un petit nombre d'agents, les pires cas sont pour un grand pas et un petit nombre d'itérations. Pour un grand nombre d'agents, les pires cas sont ceux avec un grand pas et un petit nombre d'itérations.

Après avoir étudié l'évolution du temps de convergence, la figure 7 montre l'évolution du nombre d'itérations globales nécessaires pour converger. Les meilleures et les pires configurations sont toujours situées dans la même zone. Comme prévu, moins d'itérations globales k sont rencontrées lorsque le GRD calcule un OPF précis (valeur *iter* élevée). La plupart des itérations se trouvent dans la zone inférieure droite, ce qui conduit à la non-convergence selon le critère d'arrêt de l'itération maximale, fixé à 4000.

TABLEAU 3. Valeur des meilleures et pires isolignes pour chaque nombre d'agent

taille (N)	10	20	50	100	200	500
min temps (s)	0.06	0.1	0.2	0.3	0.5	1.3
max temps (s)	0.8	0.7	0.7	1.8	3.5	12
min k	74	44	39	39	34	52
max k	4000	4000	4000	4000	4000	4000

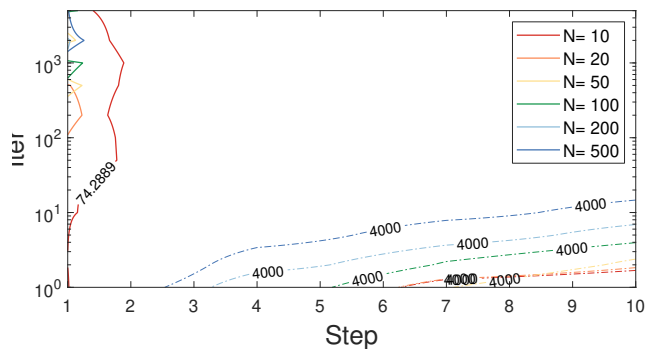


FIG. 7. Isolignes comprises dans les 10% pour meilleurs (ligne) et pire (pointillé) du nombre d'itération selon le nombre d'agent

5. CONCLUSION

Cet article étudie le partitionnement CPU-GPU d'un marché endogène. L'utilisation d'un GPU réduit le temps de calcul pour les cas les plus grand. Cela permet de réaliser des études paramétriques même sur ces cas. Dans les cas d'étude considérés, nous avons montré que plusieurs itérations du marché P2P pouvaient être effectuées entre chaque intervention du GRD. En outre, le nombre maximum d'itérations de l'OPF peut être limité. Cela permet, en simulation, de gagner du temps au prix d'un plus grand nombre d'itérations. Pour une mise en œuvre réelle, les résultats dépendront du délai de communication. En effet, il est crucial de s'assurer que l'augmentation du temps de communication due à l'augmentation du nombre d'itérations ne compense pas le gain en temps de calcul. Il convient de noter que les calculs du marché et du GRD sont effectués en série dans la simulation. Leur calcul peut être effectué simultanément dans une implémentation en temps réel, comme le montre l'article [15]. Cette étude pourrait être validée sur d'autres réseaux avec différents marchés P2P ou tailles dans le cadre de travaux futurs. L'influence de l'algorithme OPF doit également être étudiée dans les cas où une précision progressive ou un nombre limité d'itérations sont utilisés. Il convient de noter que la simulation avec $step = 1$ et $iter = 1$ converge dans tous les cas testés. Une mise en œuvre réelle avec ces paramètres impliquera que le marché et le GRD communiquent à chaque itération et mérite donc une attention particulière.

6. RÉFÉRENCES

- [1] RTE. Conditions and Requirements for the Technical Feasibility of a Power System with a High Share of Renewables in France Towards 2050
- [2] C. Bussar, et al. "Large-scale integration of renewable energies and impact on storage demand in a European renewable power system of 2050—Sensitivity study". *Journal of Energy Storage*, 6, 1-10. 2016
- [3] D. B. RicharGRDn, "Electric vehicles and the electric grid : A review of modeling approaches, Impacts, and renewable energy integration". *Renewable and Sustainable Energy Reviews*, 19, 247-254. 2013
- [4] V. A. Evangelopoulos, et al. "Heterogeneous aggregators competing in a local flexibility market for active distribution system management : A bi-level programming approach". *International Journal of Electrical Power & Energy Systems*, 136, 107639. 2022
- [5] T. Sousa, et al. "Peer-to-peer and community-based markets : A comprehensive review". *Renewable and Sustainable Energy Reviews*, 104. 2019
- [6] F. Capitanescu, et al. "State-of-the-art, challenges, and future trends in security constrained optimal power flow". *Electric power systems research*, 81(8), 1731-1741. 2011
- [7] P. N. Biskas, et al. "A decentralized solution to the security constrained DC-OPF problem of multi-area power systems." 2005 IEEE Russia Power

Tech.

- [8] R. C. Green, et al. "Applications and trends of high performance computing for electric power systems : Focusing on smart grid". *IEEE Transactions on Smart Grid*, 4(2), 922-931. 2013
- [9] B. Thomas, et al. "Hardware–software codesign for peer-to-peer energy market resolution". *Sustainable Energy, Grids and Networks*, 35. 2023
- [10] I. Araújo, et al. "Simultaneous parallel power flow calculations using hybrid CPU-GPU approach". *International Journal of Electrical Power & Energy Systems*, 105, 229-236. 2019
- [11] S. Shin, et al. "Accelerating optimal power flow with GPUs : SIMD abstraction of nonlinear programs and condensed-space interior-point methods". *Electric Power Systems Research*, 236, 110651. 2024
- [12] B. Thomas, et al. "Acceleration of a decentralized radial AC-OPF on GPU". *IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2024
- [13] T. Baroche, et al. "Prosumer markets : A unified formulation". In 2019 IEEE Milan PowerTech (pp. 1-6). IEEE.
- [14] T. Chernova, et al. "Peer-to-peer market with network constraints, user preferences and network charges". *International Journal of Electrical Power & Energy Systems*, 131, 106981. 2021
- [15] A. Dong, et al. "Convergence analysis of an asynchronous peer-to-peer market with communication delays". *Sustainable Energy, Grids and Networks*, 26, 100475. 2021
- [16] T. Baroche, et al. "Exogenous cost allocation in peer-to-peer electricity markets". *IEEE Transactions on Power Systems*, 34(4), 2553-2564. 2019
- [17] B. Thomas, et al. "GPU Optimisation of an Endogenous Peer-to-Peer Market with Product Differentiation". In 2023 IEEE Belgrade PowerTech.
- [18] S. Boyd, et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers". *Foundations and Trends® in Machine learning*, 3(1), 1-122. 2011
- [19] Q. Peng et al. "Distributed optimal power flow algorithm for radial networks, I : Balanced single phase case". *IEEE Transactions on Smart Grid*, 9(1), 111-121. 2016
- [20] R. D. Zimmerman et al. "MATPOWER : Steady-state operations, planning, and analysis tools for power systems research and education". *IEEE Transactions on power systems*, 26(1), 12-19. 2010
- [21] IEEE PES Distribution Systems Analysis Subcommittee Radial Test Feeders [Online], Available : <http://ewh.ieee.org/soc/pes/dsacom/testfeeders.html>