

Mémoire

Auteur : Pereira, Thomas

Promoteur(s) : Kiki, Gratien Jesugo Dieudonné; 2770

Faculté : Faculté des Sciences

Diplôme : Master en sciences et gestion de l'environnement, à finalité spécialisée

Année académique : 2024-2025

URI/URL : <http://hdl.handle.net/2268.2/23755>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

ULiège - Faculté des Sciences - Département des Sciences et Gestion de l'Environnement
UNI.LU – Faculty of Science, Technology and Medicine

ÉTUDE NUMERIQUE D'UN EMS COMMUNAUTAIRE MULTI- AGENT POUR LA GESTION DU SURPLUS DANS LES COMMUNAUTES D'ENERGIE

THOMAS PEREIRA

**MEMOIRE PRÉSENTE EN VUE DE L'OBTENTION DES DIPLOMES DE
MASTER EN SCIENCES ET GESTION DE L'ENVIRONNEMENT, A FINALITÉ SPECIALISÉE (ULIEGE)
MASTER EN DÉVELOPPEMENT DURABLE – FILIÈRE ENERGIE-ENVIRONNEMENT (UNI.LU)**

ANNEE ACADEMIQUE 2024-2025

REDIGÉ SOUS LA DIRECTION DE GRATIEN KIKI JESUGO DIEUDONNE ET FRANK SCHOLZEN

COMITÉ DE LECTURE :

ANNE WOUTERS

JONAS LAMBERT

Copyright

Toute reproduction du présent document, par quelque procédé que ce soit, ne peut être réalisée qu'avec l'autorisation de l'auteur et de l'autorité académique* de l'Université de Liège et de l'Université du Luxembourg.

*L'autorité académique est représentée par le(s) (co-)promoteur(s), membre(s) du personnel enseignant de l'Université de Liège et de l'Université du Luxembourg.

Le présent document n'engage que son auteur.

Auteur du présent document : PEREIRA Thomas,
thomas.pereira46@gmail.com.

Remerciements

Tout d'abord, je tiens à exprimer ma profonde gratitude à Anne Wouters et Gratien Kiki Jesugo Dieudonné. Merci à vous deux de m'avoir offert l'opportunité de participer au projet LOCOMOTRICE et de m'avoir accompagné tout au long de ce travail. Votre encadrement bienveillant, vos retours constructifs et votre disponibilité constante ont été déterminants dans l'aboutissement de ce mémoire. J'ai énormément appris à vos côtés et je vous en suis sincèrement reconnaissant.

Je remercie également Monsieur Frank Scholzen, co-promoteur de ce travail, pour sa disponibilité tout au long de ce mémoire.

Un grand merci à toute l'équipe BEMS pour leur accueil et leur aide précieuse pendant cette période de stage. Je tiens plus particulièrement à remercier les membres du projet LOCOMOTRICE, en dehors de mes encadrants principaux : Éric, Julien, Stéphane, Samuel et Jonas. Merci pour vos échanges, vos conseils et l'ambiance collaborative qui a grandement enrichi mon expérience.

Enfin, je remercie ma famille et mes amis pour leur soutien tout au long de mes études. Votre présence et vos encouragements m'ont permis d'aller au bout de ce projet avec persévérance.

Abstract

In the context of electrification of end-uses and increasing integration of renewable energies into the electricity mix, the optimized management of photovoltaic energy within energy communities through energy sharing emerges as a key lever. This thesis is part of the LOCOMOTRICE research project, which aims to develop and assess control strategies to maximize self-consumption within these communities. The study focuses on the algorithmic development of a central energy management system (EMS), designed to distribute surplus electricity in real time among community members by leveraging building flexibility. Algorithm performance is evaluated across several criteria, including responsiveness, computation time, operational efficiency, and system modularity. The objective of this research is to assess how algorithmic choices can impact the overall performance of collective surplus energy management. Results show a significant improvement in self-consumption with the chosen FIFO strategy. On average, 40% of the global photovoltaic surplus is utilized, validating the initial hypothesis that a central EMS can play a key role in optimizing energy flows in energy communities.

Résumé

Dans un contexte d'électrification des usages et d'intégration croissante des énergies renouvelables dans le mix électrique, la gestion optimisée de l'énergie photovoltaïque produite au sein des communautés d'énergie dans le cadre du partage d'énergie s'impose comme un levier essentiel. Ce mémoire s'inscrit dans le cadre du projet de recherche LOCOMOTRICE, qui vise à développer et évaluer des stratégies de pilotage pour maximiser l'autoconsommation au sein de ces communautés. L'étude porte sur le développement algorithmique d'un système de gestion d'énergie central, chargé de distribuer en temps réel le surplus d'électricité entre les membres d'une communauté en exploitant la flexibilité des bâtiments. La performance des algorithmes est évaluée selon plusieurs aspects, notamment la réactivité, les temps de calcul, l'efficacité opérationnelle, ainsi que la modularité de l'approche. L'objectif de cette étude est d'évaluer dans quelle mesure des choix algorithmiques peuvent influencer la performance globale d'une gestion collective de l'énergie excédentaire. Les résultats montrent une amélioration significative du taux d'autoconsommation avec la stratégie FIFO choisie. En moyenne, 40 % du surplus photovoltaïque global est valorisé, validant l'hypothèse de départ selon laquelle un

EMS central peut jouer un rôle clé dans l'optimisation des flux énergétiques des communautés.

Table des matières

Table des matières	5
Liste des abréviations	8
Liste des figures	9
Liste des tableaux.....	10
A. Introduction	11
B. Gestion de l'énergie dans les communautés d'énergie : Etat de l'art.....	13
1. Problématiques actuelles et rôle des EMS	13
1.1. L'autoconsommation.....	17
1.2. Le système de gestion d'énergie (EMS) dans les bâtiments intelligents	18
1.3. L'internet des objets (IoT)	21
1.4. L'optimisation en temps réel (RTO) : définition et fonctionnement	22
1.5. Principe et atouts du contrôle prédictif modélisé (MPC)	23
2. EMS dans le cadre des communautés d'énergie.....	25
2.1. Agents et systèmes multi-agents (MAS).....	26
2.2. Architectures des systèmes multi-agents	28
2.3. Structure hiérarchique EMS central – EMS locaux	30
2.4. Gestion des tâches	31
3. Problématique.....	32
C. Contexte d'étude : Projet LOCOMOTRICE	33
1. Présentation du projet LOCOMOTRICE	33
1.1. Architecture EMS central – EMS locaux	34
1.2. Système d'optimisation hiérarchique basé sur les MAS	36
1.3. Gestion des tâches et des réponses.....	37
D. Outils et méthodologie	38
1. Méthodologie	38
2. Outils.....	38

E. Développement algorithmique	40
1. Introduction	40
1.1. Description générale de l'EMS central.....	40
2. Modèles de fonctionnement de l'EMS central	41
2.1. Modèle 1 : Mise à jour périodique.....	42
2.2. Modèle 2 : Avec requêtes	44
2.3. Tableau comparatif des deux modèles	45
2.4. Considération des contraintes techniques et temporelles.....	47
2.5. Alternatives à l'allocation FIFO	49
3. Comment assurer l'allocation en temps réel ?	52
3.1. Priorité dans le traitement des réponses	52
3.2. Envoi successif des requêtes	53
3.3. Envoi simultané des requêtes.....	53
4. Choix du modèle (Modèle 2 : Avec requêtes)	54
4.1. Coordination centrale d'agents locaux homogènes et dynamiques	55
4.2. Le modèle multi-agent évènementiel	57
5. Choix des techniques de programmation.....	58
5.1. Programmation concurrente en Python	58
5.2. Méthodes de concurrence en Python	59
5.3. Evaluation de la performance des algorithmes de l'EMS central	60
5.4. Implémentation de l'algorithme	63
5.5. Illustration du comportement de l'algorithme dans différents cas-type ...	64
F. Résultats et discussion	69
1. Simulation et analyse de la gestion du surplus d'une CE par l'EMS central pendant une journée d'été.....	69
1.1. Valorisation du surplus de la CE	69
1.2. Relation entre le nombre d'EMS et la valorisation du surplus.....	70

1.3. Injection de la CE sur le réseau avec et sans EMS central	72
1.4. Impact sur l'autoconsommation	73
2. Discussion	74
3. Limitations.....	77
4. Conclusion.....	81
Bibliographie.....	82
Outils IA.....	88
Annexe : Script de l'algorithme de l'EMS central avec commentaires	89

Liste des abréviations

AsyncIO – Entrées / Sorties asynchrones

BEMS – Building Energy Monitoring and Simulation

CE – Communauté d'énergie

CER – Communauté d'énergie renouvelables

CPU – Processeur

CWaPE – La Commission wallonne pour l'Énergie

EMS – Système de gestion d'énergie

FIFO – Premier arrivé, premier servi

GIL – Verrou d'interprétation global

IA – Intelligence artificielle

IoT – Internet des objets

I/O – Entrées / Sorties

kW – Kilowatt

LOCOMOTRICE – Logiciel Open-source cocréée avec des COopératives citoyennes pour Maitriser et Opérationnaliser les actions des citoyens en faveur de la TRansition énergétique Intégrant les Communautés d'Energie

MAS – Systèmes multi-agents

MPC – Contrôle prédictif modélisé

PAC – Pompe à chaleur

PADE – Python Agent Development Environment

PID – Proportionnel, Intégral, Dérivé

PL – Programmation linéaire

PV – Photovoltaïque

RTO – Optimisation en temps réel

s – Secondes

SE – Système d'exploitation

SPADE – Smart Python Agent Development Environment

SPW – Service public de Wallonie

TRL – Niveau de maturité technologique

VE – Véhicule électrique

Win2Wal – Programme de financement Win to Wallonia

Liste des figures

Figure 1 : Figure de synthèse qui lie les enjeux à la solution proposée (source : auteur)	16
Figure 2 : Consommation de la CER avant et après implémentation des algorithmes pendant une journée d'été (Fonseca, 2022)	20
Figure 3 : Illustration d'un MPC (Li, 2022)	24
Figure 4 : Vue d'ensemble d'un système temps réel intégrant les concepts présentés (source : auteur)	26
Figure 5 : Architectures possibles : modèles classiques (a), (b) et pour les MAS (c), (d) (Kaspar, 2022)	28
Figure 6 : Architecture à deux niveaux RTO-MPC dans le cadre du projet LOCOMOTRICE (source : auteur)	35
Figure 7 : Modèle d'optimisation hiérarchique basé sur un MAS dans le cadre du projet LOCOMOTRICE (source : auteur)	36
Figure 8 : Schéma boîte noire de l'EMS central (source : auteur)	40
Figure 9 : Organigramme de l'EMS central avec mise à jour périodique (source : auteur)	42
Figure 10 : Organigramme de l'EMS central avec requêtes (source : auteur)	44
Figure 11 : Temps d'exécution en fonction du nombre d'EMS locaux des différentes approches (source : auteur)	61
Figure 12 : Régression log-log du temps d'exécution en fonction du nombre d'EMS locaux des différentes approches (source : auteur)	62
Figure 13 : Temps de réponse des différents EMS locaux, qui répondent aux requêtes de l'EMS central (source : auteur)	65
Figure 14 : Allocation du surplus en fonction des capacités des EMS locaux à activer leurs charges (source : auteur)	66
Figure 15 : Temps de réponse des différents EMS locaux, qui répondent aux requêtes de l'EMS central (source : auteur)	67
Figure 16 : Allocation du surplus en fonction des capacités des EMS locaux à activer leurs charges (source : auteur)	68
Figure 17 : Taux de valorisation en fonction du surplus disponible de la CE (40 simulations) (source : auteur)	69

Figure 18 : Corrélations entre taux de valorisation et nombre d'EMS pour 40 simulations (source : auteur)	70
Figure 19 : Évolution du taux de valorisation moyen selon le nombre d'EMS actif (avec écart-type) (source : auteur)	71
Figure 20 : Comparaison de l'injection de la CE sur le réseau avec et sans coordination par EMS central (source : auteur)	72
Figure 21 : Echelle TRL pour estimer le niveau de maturité (Mankins, 1995).....	74

Liste des tableaux

Tableau 1 : Tableau récapitulatif de l'optimisation à deux niveaux (Fonseca, 2022). 20	
Tableau 2 : Tableau comparatif entre l'approche FIFO et la programmation linéaire pour l'allocation du surplus	46
Tableau 3 : Comparaison des alternatives par rapport à l'allocation FIFO.....	51
Tableau 4 : Exemple de temps de réponse des EMS locaux.....	53
Tableau 5 : Caractéristiques des méthodes de concurrence	59
Tableau 6 : Temps d'exécution de l'algorithme d'allocation (seconde)	60
Tableau 7 : Pente (Exposant k) des différentes méthodes de programmation.....	62
Tableau 8 : Description générale de l'échelle TRL	74
Tableau 9 : Tableau récapitulatif des limitations identifiées.....	80

A. Introduction

À l'heure de la transition énergétique mondiale, marquée par l'électrification des usages et l'intégration croissante des énergies renouvelables, les modes de production et de consommation d'électricité doivent profondément évoluer. Une tendance à la décentralisation et à la transition vers des systèmes énergétiques distribués se met en place, visant à réduire les interactions avec le réseau public et à atteindre une autonomie énergétique locale.

L'importance de cette décentralisation devient de plus en plus évidente. Par exemple, le 28 avril 2025, une panne d'électricité majeure a affecté l'ensemble de la péninsule ibérique, plongeant l'Espagne, le Portugal, l'Andorre et certaines régions du sud-ouest de la France dans le noir pendant plusieurs heures. Cet incident a mis en lumière la vulnérabilité des réseaux centralisés.

Cette tendance à la décentralisation et à la production d'énergie renouvelables soulève de nouveaux défis, notamment celui de la gestion efficace des flux énergétiques afin de synchroniser la demande avec la production locale.

Dans ce contexte, le concept de communautés d'énergie se développe en Europe et en Wallonie, avec pour ambition de favoriser la production locale, le partage entre membres de la communauté et la consommation intelligente de l'énergie.

C'est dans le cadre du projet de recherche LOCOMOTRICE, financé par la Région wallonne, que s'inscrit le présent mémoire. Ce projet open-source, dédié à la conception d'un système de gestion de l'énergie (EMS), vise à développer des solutions concrètes et opérationnelles pour améliorer la gestion de l'énergie au sein des communautés d'énergie.

L'objectif de ce mémoire est de contribuer à ce projet de recherche en examinant comment et dans quelle mesure un EMS communautaire permettrait de garantir une exploitation efficace et réaliste du surplus photovoltaïque.

Ainsi, ce mémoire vise à concevoir, simuler et évaluer un système de gestion d'énergie centralisé pour valoriser le surplus au sein d'une communauté. Le taux de valorisation du surplus déterminera l'ampleur de l'augmentation de la consommation

locale de la communauté, car le surplus d'un participant peut devenir l'énergie consommée d'un autre, réduisant ainsi les échanges avec le réseau public.

Dans un premier temps, le cadre conceptuel et technologique des EMS sera présenté de manière générale, avant d'aborder leur adaptation au contexte des communautés d'énergie, en particulier les différents concepts associés aux EMS communautaires.

Le travail se poursuivra avec une présentation du projet LOCOMOTRICE. Ce projet servira de base pour développer un prototype permettant une gestion collective et automatisée de l'énergie excédentaire.

Ensuite, la méthodologie adoptée sera décrite, reposant sur la simulation informatique et les méthodes d'analyses des résultats.

La partie centrale du mémoire portera sur le développement algorithmique du système. Deux modèles de fonctionnement seront détaillés et comparés. Le choix du modèle sera justifié au regard de sa rapidité d'exécution et de son efficacité opérationnelle.

Enfin, une analyse des résultats de simulation sera réalisée en montrant l'impact de la stratégie choisie sur l'efficacité de la valorisation du surplus. Cette analyse sera complétée par une discussion sur les limitations du modèle et sur les perspectives d'amélioration du système.

B. Gestion de l'énergie dans les communautés d'énergie : Etat de l'art

1. Problématiques actuelles et rôle des EMS

La disponibilité croissante des installations photovoltaïques a apporté de nouveaux défis dans le marché de l'électricité, notamment en raison de l'intermittence de leur production. Le concept de communauté d'énergie (CE) permet aux prosumers (producteurs et consommateurs) d'exploiter et d'échanger l'électricité produite localement, réduisant ainsi leur recours à l'électricité fournie par le réseau public. D'après Mehta et al. (2022), cela peut permettre de réaliser d'importantes économies et d'augmenter la part d'énergie renouvelable dans le mix énergétique.

Au-delà des bénéfices techniques et économiques, ces communautés d'énergie renforcent une dynamique sociale en permettant aux citoyens de se réapproprier l'énergie et de mieux comprendre l'intermittence associée aux énergies renouvelables (Hemmerdinger, 2021). Cette prise de conscience améliore significativement l'acceptabilité des nouvelles installations énergétiques chez eux ou proche de chez eux, et favorise une adaptation proactive des comportements des citoyens. Cela génère un « double bénéfice » : d'une part, une hausse de la production d'énergie renouvelable, et d'autre part, un meilleur équilibre entre la demande et la production disponible (ECOinfos Energies Renouvelables, 2023).

Dans le contexte plus large de la transition énergétique, une tendance majeure vise non seulement à inciter les citoyens à réduire leur consommation d'énergie domestique, mais aussi à ajuster leur consommation en fonction de la disponibilité locale d'énergie renouvelable (Hemmerdinger, 2021). Les citoyens sont encouragés à produire eux-mêmes de l'énergie renouvelable, à stocker le surplus et à le partager avec d'autres, renforçant ainsi leur rôle actif dans la transition. Ces tendances s'expliquent par une volonté sociétale forte de réduire les impacts climatiques négatifs des émissions de carbone, par diverses réglementations gouvernementales, notamment des mécanismes de marché, tels que les Certificats Verts en Wallonie (Service public de Wallonie, 2024). Toutefois, ces mécanismes de marché ne sont pas toujours accessibles aux communautés d'énergie. Selon Scarcello (2022), les

prosumers restent souvent exclus des conditions avantageuses du marché en raison de leur inefficacité et de leur manque de fiabilité perçus quant à leur apport au réseau électrique.

L'inefficacité est renforcée par les problèmes de tension sur le réseau. En période d'ensoleillement intense, l'énergie produite par les systèmes photovoltaïques est rarement consommée instantanément. Dans une logique de réinjection, le surplus est injecté dans le réseau de distribution, provoquant des surtensions (Mehta, 2022). Ces surtensions représentent un véritable défi pour les gestionnaires de réseau qui doivent en permanence maintenir un équilibre. En cas de fréquence trop élevée, les onduleurs associés aux panneaux photovoltaïques se déconnectent automatiquement : les panneaux doivent alors cesser de produire, ce qui interrompt la génération d'énergie renouvelable.

À l'inverse, durant les périodes froides, la demande électrique augmente fortement en raison notamment de l'électrification des usages comme le chauffage ou les véhicules. Ce déséquilibre accentue les risques de sous-tension.

En parallèle, l'intermittence des sources renouvelables, comme le solaire ou l'éolien, renforce cette perception d'un manque de fiabilité (Scarcello, 2022). Leur production dépend entièrement des conditions météorologiques, ce qui rend leur contribution incertaine et difficile à planifier.

Cherchant à valoriser le surplus d'énergie photovoltaïque et à améliorer la gestion du réseau, de nouvelles stratégies sont donc nécessaires pour optimiser l'interaction entre prosumers et réseau électrique, tout en favorisant le partage d'énergie au sein des communautés.

Par conséquent, il faut éviter de réinjecter le surplus dans le réseau électrique, en consommant directement ou en stockant cette énergie résiduelle pour une utilisation ultérieure : c'est le principe de l'autoconsommation. L'autoconsommation de l'énergie photovoltaïque produite à l'échelle résidentielle est donc essentielle pour éviter les problèmes liés au réseau électrique (Mehta, 2022) et elle présente également des avantages économiques et environnementaux.

Ainsi, la recherche de méthodes pour augmenter l'autoconsommation est lancée. Il ne s'agit pas de consommer davantage, mais de mieux répondre aux besoins des utilisateurs. C'est ici que la flexibilité énergétique des bâtiments entre en jeu.

La flexibilité énergétique consiste à déplacer ou modifier la consommation d'appareils dits "flexibles" afin de l'adapter à la disponibilité d'électricité renouvelable. Par exemple, on peut préchauffer une maison avec une pompe à chaleur ou chauffer l'eau sanitaire avec un boiler lorsque l'énergie photovoltaïque est disponible, tout en garantissant le confort au moment de l'utilisation. De même, un véhicule électrique peut être rechargeé via une borne domestique lorsque la production photovoltaïque est suffisante, afin qu'il soit prêt à l'usage.

Enfin, lorsque la consommation directe n'est pas possible, les batteries permettent de stocker l'électricité pour une utilisation ultérieure, contribuant ainsi à une meilleure autoconsommation.

De plus, la flexibilité énergétique constitue également une réponse au problème de capacité du réseau électrique. En effet, lors des périodes de forte demande, le défi n'est pas seulement de produire de l'énergie, mais d'avoir suffisamment de puissance instantanée disponible. Selon l'IEA (2024), les charges flexibles peuvent être temporairement réduites ou déplacées, ce qui soulage le réseau sans nécessiter d'investissements coûteux dans de nouvelles infrastructures de production (International Energy Agency, 2024b). Ainsi, la flexibilité peut être considérée comme une ressource de capacité à part entière.

Par conséquent, la flexibilité permet de réaliser des économies sur la facture d'électricité, de contribuer à la transition énergétique en optimisant l'utilisation des ressources renouvelables et de renforcer la stabilité du réseau électrique.

Pour exploiter cette flexibilité, deux méthodes principales existent : manuelle ou automatisée. Les prosumers peuvent adapter eux-mêmes la consommation de leurs appareils, par exemple brancher leur véhicule électrique lorsqu'il y a du soleil, mais ces ajustements comportementaux ne sont pas toujours évidents à mettre en œuvre au quotidien. En effet, cette méthode n'est pas suffisamment fiable car les individus manquent de connaissances et ne manifestent pas la volonté de consacrer du temps à réfléchir et à planifier leur usage de l'énergie. C'est pourquoi il existe aujourd'hui

des solutions automatisées : les systèmes de gestion de l'énergie (EMS), qui permettent d'ajuster la consommation sans obliger l'utilisateur d'intervenir directement.

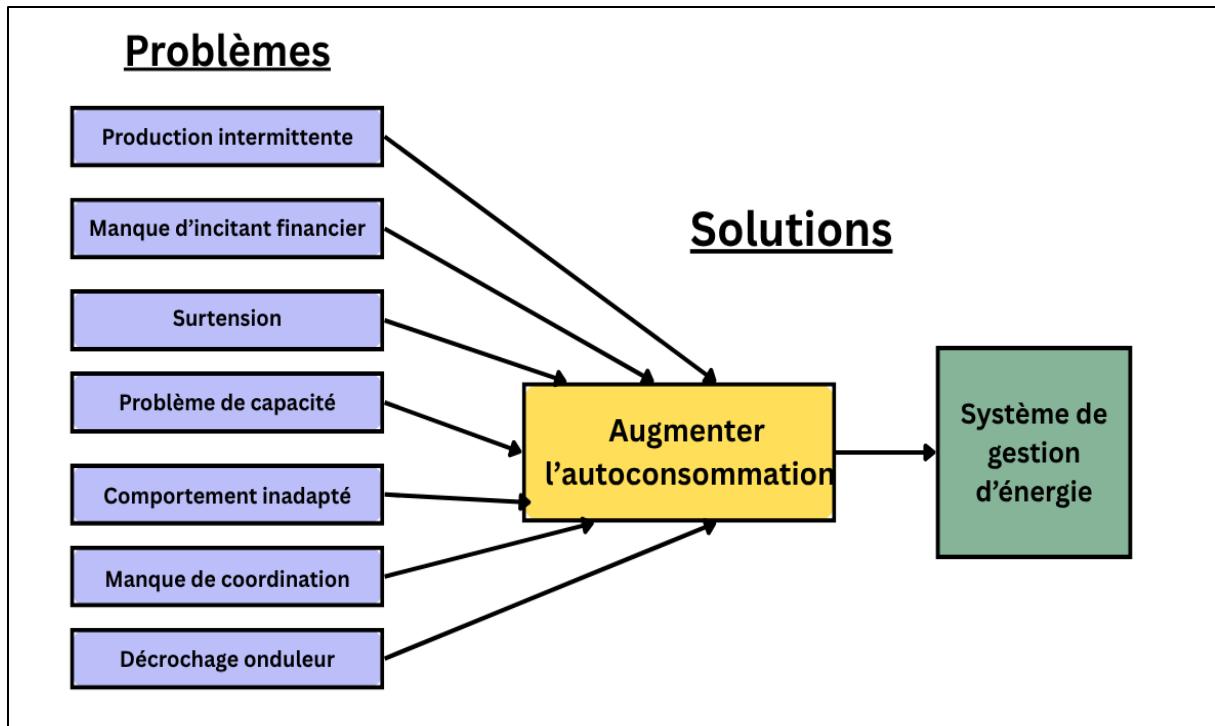


Figure 1 : Figure de synthèse qui lie les enjeux à la solution proposée (source : auteur)

La Figure 1 illustre les défis liés à la production d'énergie renouvelable et la rentabilité des communautés d'énergie tels que l'intermittence, les surtensions sur le réseau, la suppression des certificats verts, les problèmes de capacité liés aux pics de consommation et les limites comportementales et techniques de la gestion manuelle de l'énergie, qui soulignent l'importance de la mise en place d'un système de gestion de l'énergie (EMS) permettant une gestion automatisée de la consommation pour valoriser la production photovoltaïque.

Nous verrons par la suite ce qu'est un système de gestion d'énergie, le lien avec l'autoconsommation, ainsi que les différentes méthodes plus ou moins sophistiquées mises en œuvre pour y parvenir.

1.1. L'autoconsommation

Dans les foyers équipés de panneaux photovoltaïques (PV), une part significative de l'énergie produite n'est pas consommée sur place et est injectée sur le réseau, surtout en été. Avec les enjeux identifiés précédemment, il devient de plus en plus crucial d'augmenter le taux d'autoconsommation, c'est-à-dire la part de la production PV utilisée dans le même quart-horaire.

En Belgique, comme dans d'autres pays européens, la baisse des rémunérations pour l'électricité injectée incite les ménages à consommer un maximum de leur production PV sur site (SPW, 2023). Un prosumer résidentiel isolé en Wallonie autoconsomme annuellement entre 20 % et 40 % en moyenne de la production de son installation photovoltaïque lorsque la production annuelle des panneaux est plus ou moins équivalente à la consommation annuelle d'électricité de la famille (SPW, 2023) (Wallonie, 2025) (CWaPE, 2025) (Testachats, 2024). Ces données sont issues de mesures empiriques et de publications officielles.

Néanmoins, le taux d'autoconsommation n'est pas constant tout au long de l'année. En été, la production PV est très élevée en milieu de journée et dépasse souvent la consommation instantanée du ménage. Il en résulte un excédent important injecté sur le réseau et donc un taux d'autoconsommation relativement faible sur ces périodes. À l'inverse, en hiver, la production PV est faible et généralement inférieure aux besoins du foyer et toute l'énergie produite est alors consommée instantanément, ce qui se traduit par un taux d'autoconsommation très élevé durant les mois hivernaux (Testachats, 2024). Cela signifie qu'un ménage sans batterie peut autoconsommer moins de 20 % de sa production lors des journées estivales, alors qu'en période hivernale, la quasi-totalité (entre 80 - 100 %) de l'énergie PV produite sera utilisée instantanément. Sur l'année complète, ces effets se compensent partiellement et aboutissent aux moyennes annuelles de l'ordre de 20 –40 % mentionnées plus haut.

Dans le cadre des communautés d'énergie, l'autoconsommation collective de la communauté représente selon Demars (2023) : « *Une fourniture d'électricité effectuée entre un ou plusieurs producteurs et un ou plusieurs consommateurs finaux liés entre eux au sein d'une personne morale dont les points de soutirage et d'injection sont situés sur le réseau basse tension et respectent les critères,*

notamment de proximité géographique, fixés par arrêté du ministre chargé de l'énergie, après avis de la Commission de régulation de l'énergie ». Le taux d'autoconsommation collective désigne la part de la production d'énergie renouvelable locale consommée sur place, au sein d'une communauté d'énergie, rapportée à la production totale. En d'autres termes, c'est le pourcentage de l'électricité produite par la communauté qui est utilisé dans le même quart-horaire par ses membres, plutôt que d'être injectée sur le réseau. L'objectif des communautés d'énergie est de maximiser ce taux, afin de consommer localement la production et de minimiser les échanges avec le réseau public.

Pour ce faire, les systèmes de gestion de l'énergie (EMS) jouent un rôle central en pilotant les charges flexibles — un processus appelé gestion de la demande — afin de mieux les faire coïncider avec la production renouvelable locale.

1.2. Le système de gestion d'énergie (EMS) dans les bâtiments intelligents

Dans le contexte des bâtiments intelligents, un EMS désigne un système informatique conçu pour surveiller, gérer et optimiser l'utilisation de l'énergie au sein du bâtiment. Ce système centralise les données issues de différents capteurs et compteurs d'énergie et supervise les appareils électroménagers, comme les systèmes de chauffage, la production d'eau chaude et les équipements électriques. Il analyse les données recueillies par les capteurs et compteurs pour identifier des tendances de consommation et des opportunités d'économies d'énergie.

Une fois les données analysées et les opportunités d'économies identifiées, l'EMS peut mettre en œuvre des stratégies de contrôle. Son but est d'améliorer l'efficacité énergétique tout en sauvegardant le confort des utilisateurs (Kiki, 2024). Par exemple, un EMS a la capacité d'ajuster automatiquement le chauffage en fonction du nombre d'occupants ou des horaires prévus, et peut offrir des analyses pour identifier les éventuelles pertes d'énergie.

Enfin, il surveille en continu la consommation d'énergie et évalue l'efficacité des stratégies de contrôle mises en place. Il fournit également des rapports détaillés sur la consommation et les économies d'énergie.

Aux Etats-Unis, près de 40 % des émissions de CO₂ sont attribuables à la consommation d'électricité des bâtiments, dont 20 % pourraient être évités grâce à une gestion plus efficace (Mischos, 2023). L'utilisation des EMS est donc primordiale. L'EMS joue le rôle de cerveau d'un bâtiment intelligent, il surveille les flux d'énergie et met en œuvre des stratégies pour minimiser la consommation du réseau public et les coûts associés, tout en participant à la diminution des émissions.

Une étude menée en Finlande a démontré que l'intégration d'EMS domestique permettait de baisser la consommation d'électricité jusqu'à 30 % pendant les mois d'hiver. Cette réduction significative est principalement attribuée à l'optimisation du chauffage électrique, en ajustant la consommation en fonction des besoins réels, ainsi qu'une meilleure gestion des autres appareils électriques (Tuomela, 2021).

En Wallonie, une étude menée par Piron (2024) démontre numériquement qu'un EMS domestique pouvait augmenter l'autoconsommation d'électricité issue de la production photovoltaïque jusqu'à 50 %. Ce résultat démontre que l'EMS peut considérablement améliorer l'autoconsommation en optimisant l'utilisation locale de l'énergie renouvelable grâce à la flexibilité (Piron, 2024).

Une étude menée au Portugal propose une architecture à deux niveaux combinant un EMS local (dans chaque foyer) et un EMS communautaire (au niveau de la CER) en exploitant les charges flexibles (Fonseca, 2022).

L'optimisation locale consiste à planifier les charges flexibles (ex. machines à laver, chauffe-eau) en fonction de la production PV disponible dans chaque maison. L'objectif est d'augmenter l'autoconsommation individuelle en déplaçant les usages vers les périodes de production PV.

L'optimisation communautaire consiste à répartir de manière optimale les surplus PV entre les membres de la communauté. L'objectif est de maximiser la consommation locale de l'énergie produite au sein de la communauté d'énergie renouvelable (CER), en réduisant les injections vers le réseau.

Grâce à l'optimisation locale, les membres producteurs ont augmenté leur autoconsommation de +21 % en moyenne pendant un jour d'été.

Grâce à l'optimisation communautaire, jusqu'à 28 % de la consommation totale de la CER a été couverte par le partage d'énergie pendant une journée d'été. Sans coordination par l'EMS communautaire, cette part est significativement inférieure.

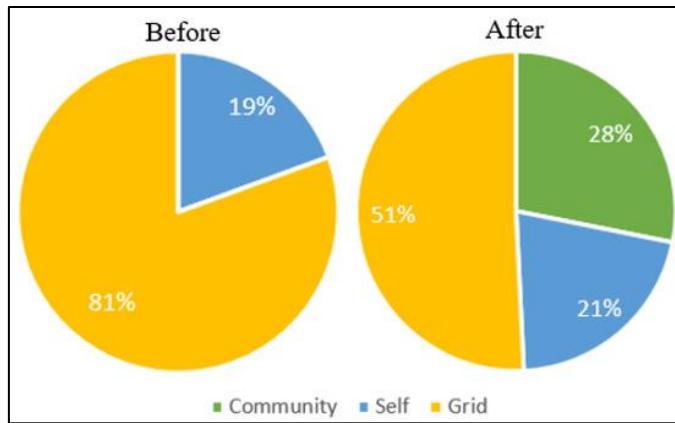


Figure 2 : Consommation de la CER avant et après implémentation des algorithmes pendant une journée d'été (Fonseca, 2022)

La Figure 2 illustre la répartition entre consommation et injection de la production PV au sein de la CER. Avant l'implémentation des algorithmes, plus de 80 % de la production PV était injectée sur le réseau. Après l'optimisation à deux niveaux, cette part d'injection est réduite à 51 %. Sur les 49 % autoconsommés collectivement, 21 % ont été autoconsommés individuellement par les prosumers, et 28 % ont été autoconsommés par la communauté grâce à une distribution optimale par l'EMS communautaire.

Le Tableau 1 résume les résultats de l'étude :

Tableau 1 : Tableau récapitulatif de l'optimisation à deux niveaux (Fonseca, 2022)

Aspect	Niveau local	Niveau communautaire
Échelle	Individuel (chez chaque membre)	Communautaire (entre membres)
Optimisation	Planification locale des charges	Distribution optimale des surplus PV
But	Augmenter l'autoconsommation individuelle	Augmenter la consommation du surplus par les autres membres
Gain	+21 % d'autoconsommation individuelle	+28 % de consommation communautaire

Ainsi, un EMS domestique pilotant les charges flexibles permet en général d'augmenter significativement le taux d'autoconsommation, avec un gain typique de 10 à 25 % par rapport à une installation photovoltaïque non optimisée. Cette amélioration dépend du profil de consommation, de la puissance PV installée et du nombre de charges flexibles pilotées (CUBE CONCEPTS, 2025) (Cieślak, 2022) (Pater, 2023).

En réalité, les expérimentations en conditions réelles sur les impacts des EMS sur la consommation d'énergie sont rares, très coûteuses et longues à organiser (Tuomela, 2021) ; ils ont principalement été démontrés par des simulations.

Pour implémenter les EMS et garantir leur optimisation en temps réel, encore faut-il que l'EMS ait accès aux données de consommation et de production. Pour ce faire, il est nécessaire d'introduire l'internet des objets, qui permet aux EMS de collecter des données et de piloter les appareils.

1.3. L'internet des objets (IoT)

L'internet des objets (IoT) fait référence aux capteurs, compteurs communicants et appareils connectés à un réseau qui communiquent en temps réel avec les EMS, de façon automatisée. Les IoT sont indispensables pour que les EMS puissent collecter les données et suivre en temps réel les flux énergétiques.

Pour la communication, les IoT utilisent soit des réseaux domestiques (par exemple, le Wi-Fi), soit des connexions filaires (par exemple, l'Ethernet). On peut également employer des réseaux longue portée (comme LoRa) dans certains cas, notamment dans des zones peu denses, bien que ceux-ci ne puissent transmettre qu'un volume de données faible avec un débit réduit.

Les compteurs intelligents mesurent la consommation d'énergie dans un bâtiment. Contrairement aux compteurs traditionnels, ces compteurs peuvent envoyer les données de consommation en temps réel ou à des intervalles définis à un EMS via des réseaux de communication (Wi-Fi, Ethernet, ou réseaux spécifiques comme LoRa).

Les onduleurs modernes sont souvent équipés de capteurs et de systèmes de communication qui leur permettent de transmettre des données de production aux EMS.

Pour connecter les appareils, on peut utiliser une prise intelligente, comme la Shelly Plug S, qui permet de rendre les appareils contrôlables et mesurables par les EMS (LogicMachine, s.d.). Par exemple, un chauffe-eau électrique connecté à une Shelly Plug S (via Wi-Fi) peut mesurer sa consommation électrique et transmettre ces données aux EMS en temps réel, afin de recevoir des ordres de ceux-ci (allumer, éteindre ou programmer des horaires).

Ainsi, les IoT communiquent en temps réel avec les EMS pour leur permettre d'optimiser la consommation selon des critères définis.

1.4. L'optimisation en temps réel (RTO) : définition et fonctionnement

L'optimisation cherche à modéliser, à analyser et à résoudre les problèmes réels en déterminant une ou plusieurs solutions satisfaisant un objectif tout en respectant d'éventuelles contraintes : il s'agit de minimiser ou maximiser une fonction, appelée la fonction objectif, afin d'atteindre la meilleure solution.

L'optimisation en temps réel (RTO) repose sur la collecte, l'analyse et l'exploitation immédiate des données. Le système RTO collecte et centralise les données en temps réel. Il nécessite des algorithmes capables de traiter des informations en continu et de proposer des solutions optimales quasi instantanément. Ces algorithmes doivent être robustes et rapides : le problème doit être résolu dans un délai court, de sorte que la solution soit applicable immédiatement. A chaque cycle (par exemple, toutes les 15 minutes), l'optimisation est de nouveau calculée avec les données collectées en temps réel. Ainsi, le RTO est utile lorsque le système nécessite des ajustements continus basés sur des données actualisées en temps réel.

Le RTO est un système visant à recalculer régulièrement les valeurs optimales pour améliorer la performance d'un système en résolvant un problème d'optimisation avec contraintes à l'état stationnaire (Jung, 2025). L'état stationnaire fait référence à une condition où les variables du processus ne changent pas, ce qui signifie que les paramètres du système sont considérés comme constants pendant la période d'optimisation.

Dans le contexte de la RTO, diverses techniques d'optimisation peuvent être mobilisées, comme la programmation linéaire, les métahéuristiques, le MPC ou encore les approches basées sur l'intelligence artificielle. Ce qui distingue la RTO, ce n'est pas tant la méthode utilisée, mais sa capacité à être exécutée de manière rapide et répétée en fonction des données en temps réel.

À l'inverse, une optimisation hors ligne est effectuée en amont, par exemple la veille pour le lendemain, et ne s'ajuste pas aux variations imprévues survenant en cours d'exécution. Elle peut être précise mais elle manque de réactivité.

Par exemple, un problème d'allocation de ressources (comme la répartition du surplus solaire entre différents consommateurs) peut être résolu hors ligne à l'aide d'une programmation pour générer un plan journalier, ou bien en ligne, en recalculant à chaque pas de temps une solution mise à jour en fonction de la production réelle, des prévisions météo et des besoins des utilisateurs.

Il est donc essentiel que les algorithmes choisis soient compatibles avec les contraintes de temps et la nature dynamique du système à gérer. Plusieurs méthodes d'optimisation existent. Cela nous permet d'introduire l'une des plus avancées : le contrôle prédictif modélisé.

1.5. Principe et atouts du contrôle prédictif modélisé (MPC)

Le contrôle prédictif modélisé (MPC) est une méthode sophistiquée couramment utilisée dans les systèmes de gestion de l'énergie des bâtiments pour le réglage optimal des systèmes de chauffage, climatisation et autres équipements. À chaque pas de temps, un problème d'optimisation est mis en œuvre pour identifier la meilleure séquence de contrôle (par exemple, l'énergie de chauffage à fournir) qui minimise un critère donné (consommation d'énergie, coût, énergie prélevée sur le réseau, etc.), tout en respectant les contraintes imposées (plages de température confortables, capacités des équipements, période de charge/décharge, etc.).

Le MPC fusionne les concepts de rétroaction (modifier le contrôle en fonction de l'état mesuré) et de planification prédictive, appelé contrôle à horizon glissant qui permet à l'EMS d'actualiser régulièrement ses prévisions et de s'adapter dynamiquement aux variations de la demande énergétique (Bianchini, 2019). Il s'appuie sur des projections de perturbations futures (comme la météo ou les

habitudes) et sur les exigences de confort pour prévoir et optimiser le comportement énergétique du bâtiment. Le calcul est ajusté au prochain pas de temps en incorporant les données en temps réel.

Contrairement à une régulation conventionnelle qui n'intervient qu'après avoir détecté une violation d'un seuil, comme les contrôleurs PID, le MPC a la capacité de commencer par exemple à refroidir un bâtiment par anticipation s'il prévoit une montée de la température extérieure (Bianchini, 2019). Historiquement limité par la puissance de calcul requise, le MPC a gagné en popularité grâce aux progrès en matière d'informatique (Bianchini, 2019), et est désormais une des méthodes les plus efficaces pour la gestion de l'énergie dans les bâtiments.

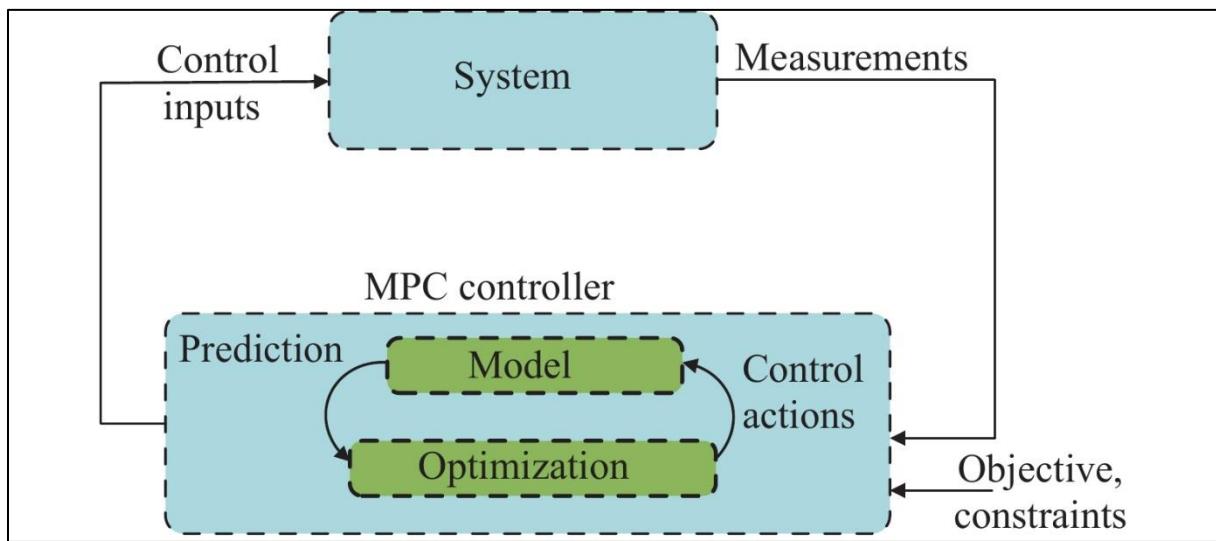


Figure 3 : Illustration d'un MPC (Li, 2022)

La Figure 3 illustre schématiquement le principe du MPC. À chaque étape de contrôle, le contrôleur MPC commence par mesurer l'état actuel du système. Ensuite, il détermine, à l'aide d'une optimisation, quelle commande appliquer, en trouvant les actions qui, sur un horizon de prédition de N pas de temps, offrent les meilleures performances prévues selon une fonction objectif donnée (Li, 2022). Les variables de commande déterminées pour le premier pas de l'horizon sont alors appliquées au système. Le système évolue ensuite vers un nouvel état, après quoi le cycle recommence.

Malgré ses avantages, le MPC nécessite un modèle fiable du bâtiment et des équipements. L'élaboration de ce modèle peut être complexe et des imprécisions

affectent la performance du contrôle. De plus, la résolution de l'optimisation à chaque pas de temps peut exiger une puissance de calcul non négligeable, surtout si le problème inclut de nombreuses variables.

Dans le cadre du partage d'énergie, il est nécessaire de disposer de systèmes capables de gérer plusieurs utilisateurs afin d'optimiser les flux énergétiques de la CE.

Nous aborderons par la suite les mécanismes d'optimisation énergétique collective, notamment les concepts liés à la gestion multi-utilisateurs dans le cadre des communautés d'énergie, en particulier la coordination entre les différents utilisateurs situés dans une même zone géographique, interconnectés par un réseau électrique et un réseau de communication.

2. EMS dans le cadre des communautés d'énergie

Les communautés d'énergie désignent un ensemble d'utilisateurs et de prosumers dans une zone locale (par exemple, un quartier) qui partagent de l'énergie (panneaux solaires, éoliennes locales, batteries communautaires, etc.) et optimisent collectivement leur consommation. Un EMS communautaire permet à ces communautés de gérer collectivement l'énergie de façon coordonnée.

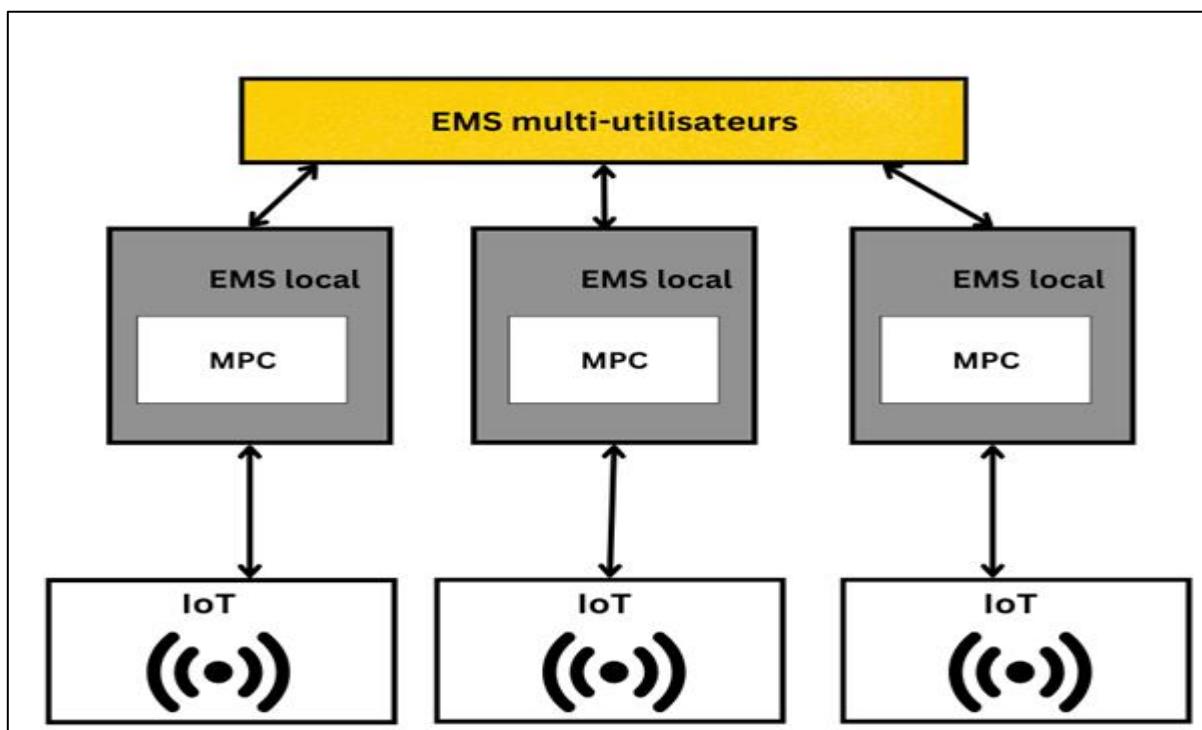


Figure 4 : Vue d'ensemble d'un système temps réel intégrant les concepts présentés (source : auteur)

La Figure 4 illustre l'organisation d'un système de gestion de l'énergie hiérarchique. Chaque utilisateur est équipé d'un EMS local, qui intègre un MPC pour optimiser sa consommation en temps réel en fonction des données collectées à travers les objets connectés (IoT). Ces EMS locaux sont coordonnés par un EMS central multi-utilisateurs, chargé d'assurer une gestion collective de l'énergie à l'échelle communautaire.

L'organisation de ces structures ainsi que les concepts liés aux systèmes multi-agents est détaillée en section suivante.

2.1. Agents et systèmes multi-agents (MAS)

Le système de contrôle basé sur les systèmes multi-agents (MAS) repose sur la coopération d'agents autonomes dans l'optique d'atteindre un but commun. Comme l'indiquent Wooldridge et al. (1995), les agents se distinguent par leur indépendance dans la prise de décisions, leur aptitude à interagir socialement, ainsi que leur capacité à réagir aux modifications de l'environnement. Les agents sont donc capables d'agir de manière autonome dans leur environnement. Les trois caractéristiques comportementales d'un agent sont donc :

- Réactivité : capacité à réagir aux changements de l'environnement,
- Proactivité : aptitude à adopter un comportement orienté vers la réalisation d'objectifs,
- Capacité sociale : sa capacité d'interagir avec d'autres agents via un langage de communication inter-agents. Dans un environnement multi-agent, tous les agents doivent être pris en compte pour atteindre les objectifs fixés.

Il est essentiel de trouver un équilibre entre réactivité et proactivité. Un bon équilibre entre les deux permet à un agent d'être à la fois adaptable et stratégique.

Dans un environnement multi-agent, chaque agent a son propre rôle (par exemple, un EMS local gère un bâtiment) et ils coopèrent pour atteindre un objectif commun (par exemple, maximiser l'autoconsommation collective). Le déroulement d'un système multi-agent consiste à « *orchestrer et coordonner des tâches réalisées par*

plusieurs agents autonomes », ces agents pouvant être des logiciels ou même des dispositifs IoT (Internet of Things) (Byrd, 2023). Les différentes formes d'interaction sont la collaboration et la coordination. La première s'intéresse à la manière de répartir le travail entre plusieurs agents et la seconde analyse la manière dont les actions des différents agents doivent être organisées dans le temps et l'espace de manière à atteindre l'objectif global (Ferber, 1997). Ce paradigme est utile lorsque les tâches sont distribuées, nécessitent de la collaboration et sont de nature asynchrone : chacun avec son propre rythme de calcul. En effet, la principale différence entre les systèmes multi-agents et les systèmes distribués ou concurrents réside dans le fait que chaque agent poursuit ses propres objectifs : il cherche à obtenir le meilleur résultat pour lui-même.

Dans le secteur de l'énergie, les MAS peuvent avoir plusieurs applications dont :

- la gestion de la demande, où les agents négocient pour ajuster la consommation grâce à leur flexibilité (Zeng, 2024);
- Les réseaux intelligents distribués et micro-réseaux, où les agents assurent l'équilibre production-consommation ainsi que la fiabilité du système (El Hafiane et al., 2024) ;
- le trading d'énergie pair-à-pair, où des ménages ou équipements, modélisés comme agents, échangent directement l'électricité en temps réel, selon les conditions économiques du marché (Tariq et al., 2025) ;
- la maintenance prédictive, où des outils basés sur des MAS permettent de prédire d'éventuelles défaillances et d'optimiser la gestion opérationnelle des infrastructures énergétiques. Zheng et al. (2025) présentent une méthode centrée sur le système pour l'optimisation de la maintenance prédictive, intégrant des agents intelligents capables d'apprendre et de s'adapter aux conditions changeantes des équipements. Chaque agent représente un composant spécifique du système. Ces agents interagissent entre eux et avec leur environnement pour collecter des données, apprendre des modèles de dégradation et prendre des décisions pour prédire les défaillances potentielles et optimiser les calendriers de maintenance (Zheng et al., 2025).

Après avoir posé les fondements théoriques des systèmes multi-agents, il convient d'examiner les différentes architectures possibles que ces derniers peuvent adopter

dans le cadre de la gestion d'énergie. La manière dont les agents sont organisés et interconnectés joue un rôle déterminant dans l'efficacité, la robustesse et la réactivité du système global.

2.2. Architectures des systèmes multi-agents

Généralement, les MAS sont associés à la décentralisation, permettant robustesse, modularité et tolérance aux pannes grâce à l'absence d'une unité centrale critique (Ferber, 1997). Toutefois, il est possible que certains MAS incluent une hiérarchie (Figure 4 (d)), dans laquelle des agents de rang supérieur orchestrent les agents de rang inférieur. Ces structures hiérarchiques démontrent sans ambiguïté la possibilité d'une autorité centrale au sein d'un MAS.

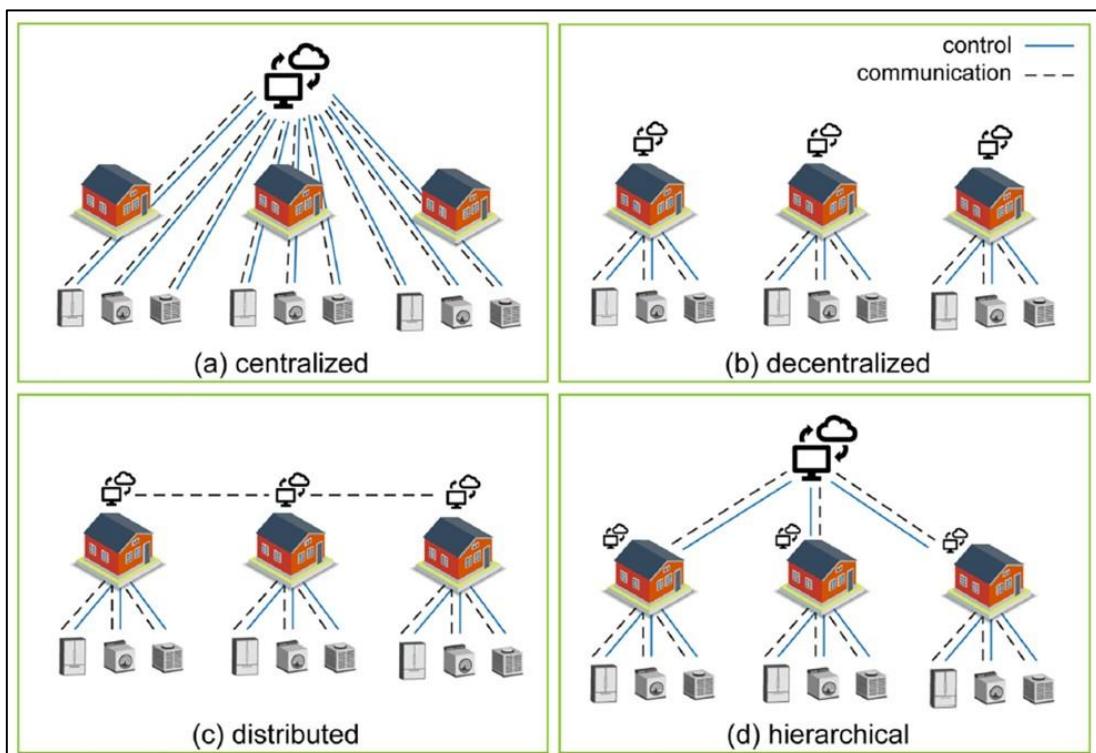


Figure 5 : Architectures possibles : modèles classiques (a), (b) et pour les MAS (c), (d) (Kaspar, 2022)

La Figure 5 présente différentes architectures de gestion de l'énergie au sein de groupes de bâtiments. Celles-ci ne se limitent pas à trois maisons, mais peuvent s'appliquer à n maisons par communauté d'énergie.

Les schémas (a) centralisé et (b) décentralisé représentent des architectures classiques. Dans le modèle centralisé, un contrôleur unique supervise l'ensemble

des bâtiments et de leurs équipements assurant à la fois la communication et le contrôle. En revanche, dans le modèle décentralisé, chaque bâtiment dispose de son propre contrôleur local qui gère uniquement ses équipements internes. Ce dernier ne permet pas la coordination entre bâtiments.

Les schémas (c) distribué et (d) hiérarchique illustrent des architectures communicantes, correspondants à des systèmes multi-agents.

Le modèle distribué repose sur plusieurs entités autonomes qui coopèrent entre elles via une communication entre les contrôleurs.

Le modèle hiérarchique combine un niveau de contrôle supérieur (agent central) qui supervise et coordonne plusieurs contrôleurs locaux autonomes de niveau inférieur, permettant une gestion à la fois locale (des agents locaux) et globale (de l'agent central).

Ces deux dernières structures sont particulièrement adaptées aux communautés d'énergie, car elles permettent une communication en ligne entre utilisateurs pour une meilleure flexibilité, évolutivité et réactivité dans un contexte multi-utilisateurs. Ces architectures multi-agents permettent aux utilisateurs (ou aux systèmes dans un environnement multi-agent) de partager des informations sur leur production et leur consommation, ce qui rend le système plus dynamique et améliore l'optimisation collective des ressources disponibles.

Plusieurs études récentes démontrent l'efficacité des stratégies multi-agents hiérarchique, avec une entité centrale, pour la gestion énergétique d'un groupe de bâtiments. Selon Saeed et al. (2023), un EMS hiérarchique (Figure 5 (d)), est proposé afin d'optimiser l'énergie pour un ensemble de 10 bâtiments, ce qui permet une diminution significative du gaspillage énergétique par rapport à une gestion individuelle. Dans ce système, chaque bâtiment intelligent dispose d'un compteur communicant (agent local) connecté à une plateforme qui sert d'agent central. La coordination est prédéfinie et gérée par la plateforme centrale qui décide des échanges d'énergie entre les bâtiments.

Dans les MAS distribués, la coordination est souvent le produit naturel des interactions directes entre agents, telles que la négociation ou l'utilisation de la théorie des jeux (Karavas, 2017). Malgré la grande robustesse des MAS distribués,

étant donné qu'une défaillance d'un agent n'entrave pas l'activité des autres (Figure 5 (c)), il reste parfois difficile de parvenir à une solution optimale à l'échelle globale (Ferber, 1997). Effectivement, les décisions sont établies uniquement sur la base d'informations limitées. Par conséquent, la collaboration entre agents s'avère plus complexe et potentiellement moins efficace que dans le cadre d'une structure centralisée. Dans le contexte des micros-réseaux, Karavas et al. (2017) ont prouvé qu'une structure multi-agent distribuée pouvait gérer efficacement plusieurs contrôleurs locaux sans la nécessité d'un contrôle centralisé. L'équilibre global du réseau est obtenu grâce à l'interaction des différents agents.

2.3. Structure hiérarchique EMS central – EMS locaux

Dans les bâtiments intelligents de grande taille ou dans des 'quartiers intelligents', on adopte souvent une architecture de contrôle hiérarchique combinant un EMS central et des EMS locaux (Saeed, 2023). L'EMS central supervise la stratégie globale du bâtiment ou de l'ensemble de bâtiments. Il agrège les données énergétiques, prend des décisions optimales à l'échelle macro (par exemple, répartition de l'énergie entre sous-systèmes) et coordonne les sous-ensembles (Saeed, 2023).

L'EMS local gère les équipements propres à un bâtiment (production PV locale, batterie, pompe à chaleur, etc.), tandis que l'EMS central coordonne l'ensemble des EMS locaux pour atteindre les objectifs collectifs (par exemple, augmenter la part d'énergie partagée dans une CE).

L'architecture EMS central – EMS locaux permet de combiner une supervision globale avec une autonomie locale. Cependant, à mesure que la complexité des interactions augmente (nombre croissant de prosumers, diversité des équipements, complexité de l'algorithme d'optimisation et de coordination), il devient nécessaire de disposer d'une structure distribuée et réactive.

Le paradigme multi-agent répond à cette exigence. Il permet de décomposer le système global en agents autonomes, chacun responsable d'un sous-système (par exemple, un bâtiment ou un équipement) tout en assurant une coordination globale. Chaque EMS local peut ainsi être modélisé comme un agent capable de percevoir son environnement, d'agir sur ses équipements et de communiquer avec d'autres agents via des échanges d'informations ou de décisions.

La structure correspond à une organisation hiérarchique, ce qui signifie qu'au lieu de tout gérer de manière centralisée, l'EMS central confie certaines décisions aux agents locaux tout en coordonnant l'ensemble, augmentant ainsi la souplesse et la réactivité du système. L'EMS central peut améliorer l'efficacité générale (par exemple, maximiser l'autoconsommation collective), pendant que les EMS locaux garantissent une efficacité locale et une continuité de service en cas de panne du système centralisé. Par conséquent, l'approche hiérarchique renforce la robustesse du système (les EMS locaux sont capables de fonctionner de façon autonome en cas de défaillance de la supervision) et favorise la modularité (il est possible d'ajouter ou de retirer des sous-systèmes sans avoir à revoir totalement le contrôle). Elle réduit la complexité individuelle en répartissant les tâches entre différents niveaux. En contrepartie, concevoir l'algorithme de coordination entre les différents niveaux peut être complexe en ce qui concerne la mise en place des communications (Berk Celik, 2019).

Pour assurer une bonne coopération entre les agents, il est essentiel de définir des mécanismes de communication et de coordination des décisions, notamment pour répartir les tâches, synchroniser les actions et gérer les priorités et la hiérarchie. Cela introduit la notion de gestion des tâches qui est essentielle pour piloter un EMS multi-agent.

2.4. Gestion des tâches

La gestion des tâches concerne principalement la manière dont les tâches sont distribuées et dont les réponses sont gérées et traitées (Remigio, 2020). La gestion des tâches est essentielle pour assurer que les ressources disponibles sont utilisées de manière efficace et que les objectifs opérationnels sont atteints dans les délais impartis. Dans les systèmes où plusieurs tâches doivent être coordonnées pour atteindre une optimisation en temps réel, la gestion des tâches devient cruciale (priorité, synchronisation). Par exemple, l'intégration des décisions de planification de la production dans un cadre d'optimisation dynamique permet de mieux coordonner les actions pour améliorer l'efficacité opérationnelle (Elshazly, 2021).

3. Problématique

Dans le contexte de la gestion collective de l'énergie excédentaire dans les communautés d'énergie, une question centrale émerge : comment un système de gestion multi-agent peut-il coordonner efficacement les EMS locaux afin d'optimiser la répartition en temps réel du surplus entre les membres de la CE ? L'objectif est de maximiser l'autoconsommation collective de la CE et de réduire au minimum l'injection d'électricité sur le réseau, tout en prenant en compte les contraintes techniques et temporelles des charges flexibles.

Cela soulève plusieurs interrogations : quelles architectures multi-agents sont adaptées ? Et comment concilier les objectifs d'optimisation globale avec l'autonomie locale des EMS locaux ? Quelles méthodes algorithmiques permettent de gérer cette répartition au sein de la CE efficacement ?

C. Contexte d'étude : Projet LOCOMOTRICE

Le projet LOCOMOTRICE s'inscrit précisément dans cette dynamique multi-agent en combinant supervision centrale et gestion locale intelligente.

Ce projet open-source se caractérise par son approche modulaire, distribuée et orientée utilisateur, intégrant des algorithmes avancés et une gouvernance participative. Il propose une solution concrète pour répondre aux défis soulevés précédemment.

1. Présentation du projet LOCOMOTRICE

Le projet LOCOMOTRICE, financé par le programme « Win²Wal » de la Région Wallonne, propose une plateforme logicielle open-source visant à renforcer la gestion collective des ressources au sein des communautés d'énergie. Il est axé sur la gouvernance participative, la mutualisation des ressources et le partage d'énergie, et il cherche à optimiser l'autoconsommation collective tout en assurant des bénéfices environnementaux, économiques et sociaux. Associé à des partenaires académiques, coopératives et acteurs locaux pour développer une solution modulaire innovante, il intègre intelligence artificielle et modélisation énergétique avec une démarche inclusive et adaptée aux besoins spécifiques des utilisateurs.

Le projet LOCOMOTRICE suit une approche structurée pour créer une plateforme logicielle destinée à améliorer la gestion collective des ressources énergétiques. Cette approche comprend l'élaboration d'un système de gestion multi-agent de l'énergie, c'est-à-dire une gestion à l'échelle locale (EMS local) ainsi qu'un système de gestion centralisé (EMS central). Ce dernier, cœur de la présente étude, intègre la modélisation d'un prototype et l'écriture d'un algorithme permettant de coordonner efficacement les EMS locaux grâce à une communication bidirectionnelle.

Le projet inclut également la création d'une interface utilisateur intuitive, accompagnée de modules de tarification et de monitoring afin de faciliter l'utilisation quotidienne par les membres de la CE. Le projet prévoit des tests en laboratoire suivis d'essais en conditions réelles dans des bâtiments tests afin de valider l'ensemble de la solution proposée et d'assurer sa pertinence aux besoins réels des communautés d'énergie.

Dans la section suivante, l'architecture EMS central – EMS local sera détaillée, ainsi que le fonctionnement de ce système multi-agent.

1.1. Architecture EMS central – EMS locaux

L'objectif est d'atteindre la solution optimale à l'échelle globale, c'est pourquoi l'architecture retenue pour la conception des modèles dans la suite du travail est celle d'un MAS hiérarchique. Au sein des communautés d'énergie, il est essentiel de centraliser les données de consommation et de production pour pouvoir prendre les meilleures décisions d'adaptation de la consommation à l'échelle communautaire. Cette architecture permet à un niveau central de coordonner plusieurs agents locaux tout en conservant l'aspect décentralisé. Ainsi, elle offre un meilleur équilibre entre coordination centrale et flexibilité locale. En effet, comme le souligne Wooldridge et al. (1995), les architectures hiérarchiques dans les systèmes multi-agents gèrent séparément ces deux aspects. Elles assurent la coordination à grande échelle tout en permettant à chaque agent local de s'adapter de manière autonome à son propre contexte. Cette organisation évite les problèmes liés à une gestion trop centralisée, souvent rigide, ainsi qu'à une gestion trop distribuée, qui peut manquer de coordination et d'optimisation globale. Ainsi, la gestion des ressources est facilitée : elles sont réparties entre différents niveaux, tout en permettant une bonne réactivité (Wooldridge, 1995).

Dans ce contexte, le modèle hiérarchique s'avère particulièrement adapté. Il permet de réaliser un pilotage centralisé avec une autonomie locale. Cette logique est mise en œuvre à travers une structure concrète basée sur un EMS central coordonnant plusieurs EMS locaux, que nous présentons à présent.

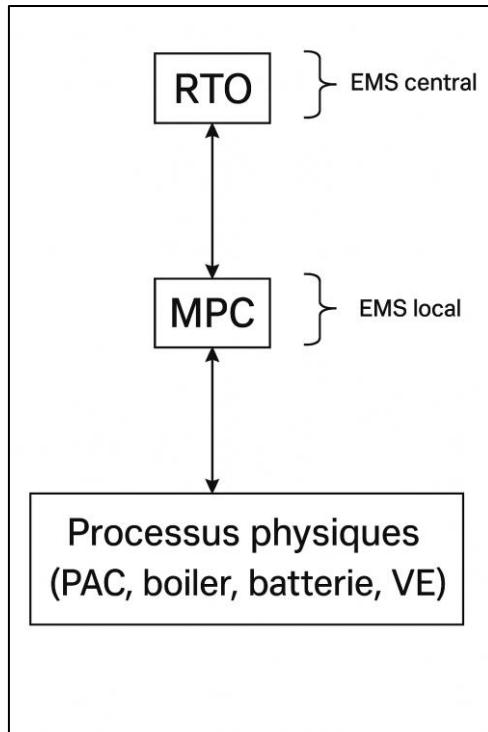


Figure 6 : Architecture à deux niveaux RTO-MPC dans le cadre du projet LOCOMOTRICE (source : auteur)

Dans le cadre du projet LOCOMOTRICE, l'EMS central est élaboré en tant qu'agent de haut niveau qui coordonne les sous-agents ou agents locaux de niveau inférieur, représentant les EMS locaux. Les agents locaux interagissent avec l'agent central et adaptent constamment leurs stratégies afin d'améliorer l'efficacité globale du système. L'EMS central agrège les données énergétiques provenant des EMS locaux, par exemple la surproduction photovoltaïque. Ce surplus d'énergie est ensuite réparti entre les EMS locaux en fonction des charges qu'ils peuvent déplacer (les charges flexibles).

Par conséquent, il s'agit d'une architecture à deux niveaux (Figure 6) qui combine une couche d'optimisation globale (RTO) avec une couche de commande dynamique locale (MPC). Le niveau supérieur (RTO) joue un rôle réactif. Il surveille le système en temps réel et déclenche l'optimisation lorsqu'il y a un surplus de production photovoltaïque.

Chaque EMS local intègre un MPC qui prend en charge les décisions d'activation des charges flexibles. À chaque réception d'un signal du RTO, il réinitialise l'optimisation. Cela permet une adaptation dynamique en temps réel (Xu, 2024).

Chaque EMS local réagit de manière autonome au signal reçu en optimisant la consommation de ses ressources flexibles (batteries, chauffe-eau, etc.) en fonction de ses contraintes locales.

1.2. Système d'optimisation hiérarchique basé sur les MAS

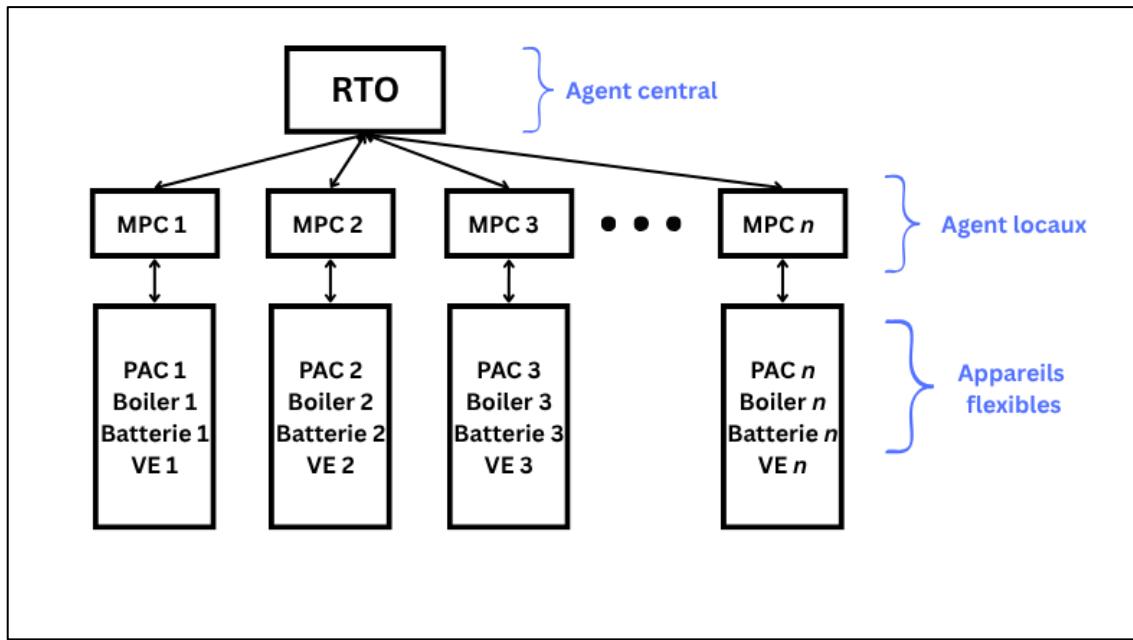


Figure 7 : Modèle d'optimisation hiérarchique basé sur un MAS dans le cadre du projet LOCOMOTRICE (source : auteur)

Chaque agent autonome communique via un réseau de communication. Cela permet d'optimiser efficacement les charges flexibles à différents niveaux du système (Zhang, 2025).

Le système proposé est structuré en trois couches (Figure 7) :

- La couche agent central : elle supervise le système global, agrège les données et coordonne les agents locaux. L'agent central envoie des propositions de déplacement de charges à la couche inférieure.
- La couche sous-agents ou agents locaux : composé de n agents locaux, elle réagit aux propositions de l'agent central afin d'optimiser les charges flexibles et transmet les instructions de contrôle aux appareils.
- La couche appareils flexibles et compteurs intelligents (IoT) : elle exécute les ordres de la couche agents locaux et fournit un retour d'information via le

réseau de communication. De plus, les compteurs intelligents transmettent les informations sur les charges nettes.

L'architecture multi-agent hiérarchique permet une coordination en temps réel du système, capable d'allouer efficacement les ressources disponibles en réponse aux variations des charges de la communauté (Zhang, 2025).

Nous allons à présent examiner comment l'architecture hiérarchique répartit les tâches entre les différents niveaux du système. Cette organisation multi-agent repose sur une distribution claire des rôles.

1.3. Gestion des tâches et des réponses

Avec une gestion efficace des tâches et des réponses correspondantes, l'EMS central peut coordonner les interactions avec les EMS locaux. Il collecte et traite les données en temps réel, et optimise les décisions de consommation de la communauté. La gestion des tâches et des réponses est au cœur de la coordination de l'EMS central.

Quand plusieurs tâches sont opérées en même temps, on parle d'exécution parallèle des tâches : une fois les tâches distribuées, chaque agent effectue sa part du travail en parallèle des autres. Dans notre système, tous les EMS locaux vont simultanément répondre au signal reçu : chacun fait ses calculs indépendamment.

Le système doit donc gérer ces réponses de manière concurrente, ce qui consiste à gérer et traiter plusieurs réponses qui peuvent se chevaucher dans le temps (Anderson, 2024). En coordonnant efficacement les tâches et les réponses, l'EMS central peut optimiser efficacement la distribution de l'énergie résiduelle.

D. Outils et méthodologie

1. Méthodologie

L'algorithme a été développé à partir d'une autoformation basée sur la documentation Python, ainsi que sur l'analyse de solutions similaires de gestion multi-agent. Des outils comme SPADE et PADE en Python ou la plateforme CrewAI illustrent concrètement comment déployer de tels MAS dans la coordination de services web ou d'agents IA, ainsi que les bonnes pratiques en systèmes distribués asynchrones mises en avant par Christie et al. (2022) (Christie, 2022). Des articles scientifiques et retours d'expérience seront mobilisés, notamment la plateforme Real Python (Anderson, 2024), des publications sur les systèmes multi-agents énergétiques pour l'architecture hiérarchique (Saeed, 2023) (Surmann, 2020), ainsi que des tutoriels et threads de développeurs Python pour les choix d'implémentation du logiciel (Lee, 2020) (Johnson, 2023).

Ces références montrent que les décisions techniques de ce projet sont fondées sur des bases théoriques et pratiques pour répondre aux exigences d'un EMS central multi-agent. Par conséquent, il s'agit d'un algorithme « maison », issu d'un processus de recherche par essais-erreurs, décrit dans la suite du travail.

2. Outils

L'implémentation de l'EMS sera réalisée dans un premier temps en Python. Il s'agira de développer un code fonctionnel et rapide, capable de résoudre efficacement la tâche définie. Des simulations seront ensuite effectuées afin d'analyser et de valider les résultats obtenus. L'analyse de la performance des algorithmes ainsi que l'analyse des résultats de simulation seront effectuées avec Excel pour traiter et visualiser les résultats.

À terme, les interactions entre l'EMS local et l'EMS central seront simulées avec le logiciel TRNSYS sur une CE virtuelle. Toutefois, cette étape n'a pas été réalisée dans le cadre du présent document. Le code Python pourra être intégré dans une architecture plus large, incluant :

- un système temps réel ;
- des échanges de données via des protocoles de communication ;

- une interface graphique utilisateur pour la visualisation.

E. Développement algorithme

1. Introduction

L'EMS central se charge de la répartition du surplus d'énergie photovoltaïque dans une CE. Son objectif est de maximiser l'autoconsommation et d'éviter les prélèvements d'électricité du réseau. Si aucun consommateur de la CE n'est en mesure de consommer le surplus de production, l'énergie est alors injectée dans le réseau.

La gestion du surplus d'énergie se fera en utilisant la flexibilité offerte par le stockage et les appareils domestiques pour minimiser les prélèvements sur le réseau et le pic d'énergie injecté dans le réseau. La gestion de la demande exploite la flexibilité des consommateurs en ajustant l'activité de leurs appareils en réponse à un signal.

L'adaptation de la consommation consiste à fixer des charges à des heures spécifiques en fonction des besoins des occupants. Son objectif est de réduire la demande du réseau pour maximiser la consommation locale et collective de l'énergie renouvelable, et ainsi diminuer les coûts associés à la fourniture d'électricité (Scarcello, 2022). Ainsi, l'objectif principal est d'allouer efficacement un surplus d'énergie aux EMS locaux. L'existence de plusieurs stratégies explique le fait que cette section se concentre sur une étude comparative de ces méthodes.

1.1. Description générale de l'EMS central

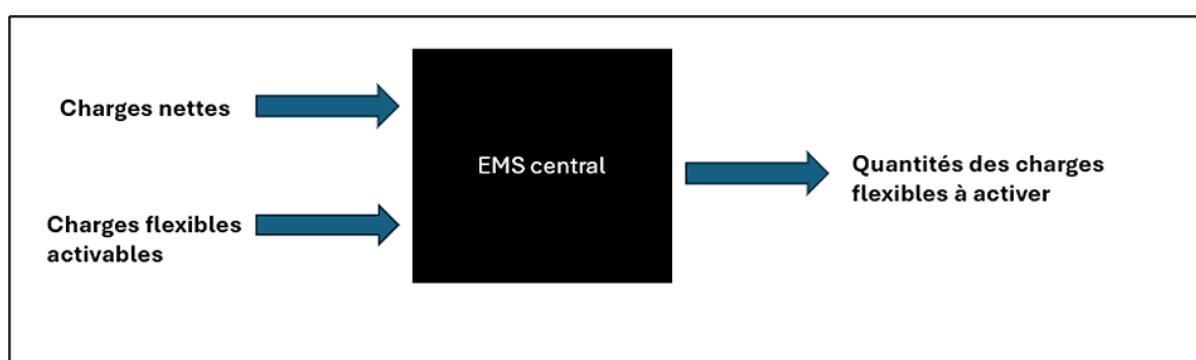


Figure 8 : Schéma boîte noire de l'EMS central (source : auteur)

L'EMS central joue un rôle de coordination entre les différents EMS locaux. Il ne commande pas directement les charges, mais il centralise les données de surplus et de flexibilité, et répartit le surplus disponible selon une logique déterminée. Le

schéma boîte noire sur la Figure 8 définit les entrées et les sorties de l'EMS central. Toutes les données sont échangées entre l'EMS central et les EMS locaux.

À partir des informations relatives aux charges nettes et aux charges flexibles activables des EMS locaux, l'EMS central procède à l'allocation du surplus d'énergie photovoltaïque, lorsqu'il est présent, afin que les EMS locaux puissent optimiser leur consommation grâce à leurs charges flexibles.

Un point important du design est la prise en compte des contraintes temporelles. Le pas de temps du cycle d'activation a été fixé à 15 minutes, c'est-à-dire que chaque itération porte sur un quart d'heure, après quoi un nouveau cycle démarre. Ce choix permet de donner aux équipements le temps nécessaire pour consommer le surplus alloué (par exemple, un ballon d'eau chaude disposera de 15 minutes de fonctionnement si de l'énergie lui a été attribuée), d'éviter des oscillations trop rapides des consignes (ce qui pourrait nuire aux appareils) et de s'aligner sur les résolutions temporelles couramment utilisées par les gestionnaires du réseau, opérant souvent par pas de 15 minutes.

Ainsi, à chaque quart d'heure, la situation est réévaluée avec les nouvelles conditions de production/consommation et un nouveau cycle de redistribution du surplus est initié. Cela évite une supervision trop fine par l'EMS central tout en permettant des ajustements suffisamment fréquents pour minimiser l'injection sur le réseau de distribution et ainsi maximiser l'utilisation de l'énergie excédentaire.

L'EMS central reçoit 2 entrées pour pouvoir répartir le surplus :

1. Les charges nettes transmises par les compteurs intelligents toutes les 15 minutes.
2. Les charges flexibles activables identifiées par les EMS locaux.

Deux modèles principaux ont été identifiés pour garantir le bon fonctionnement de l'EMS central. Ces modèles seront présentés ci-après, accompagnés de leurs avantages et inconvénients respectifs.

2. Modèles de fonctionnement de l'EMS central

Les deux modèles étudiés utilisent deux stratégies différentes. Le premier modèle utilise l'optimisation linéaire et le deuxième applique l'allocation FIFO (« first in, first

out » ou « premier arrivé, premier servi ») (GeeksforGeeks, 2025). La différence entre les deux modèles réside dans la façon dont ils traitent la flexibilité disponible (charges activables) et dans la gestion du surplus.

2.1. Modèle 1 : Mise à jour périodique

2.1.1. Fonctionnement général

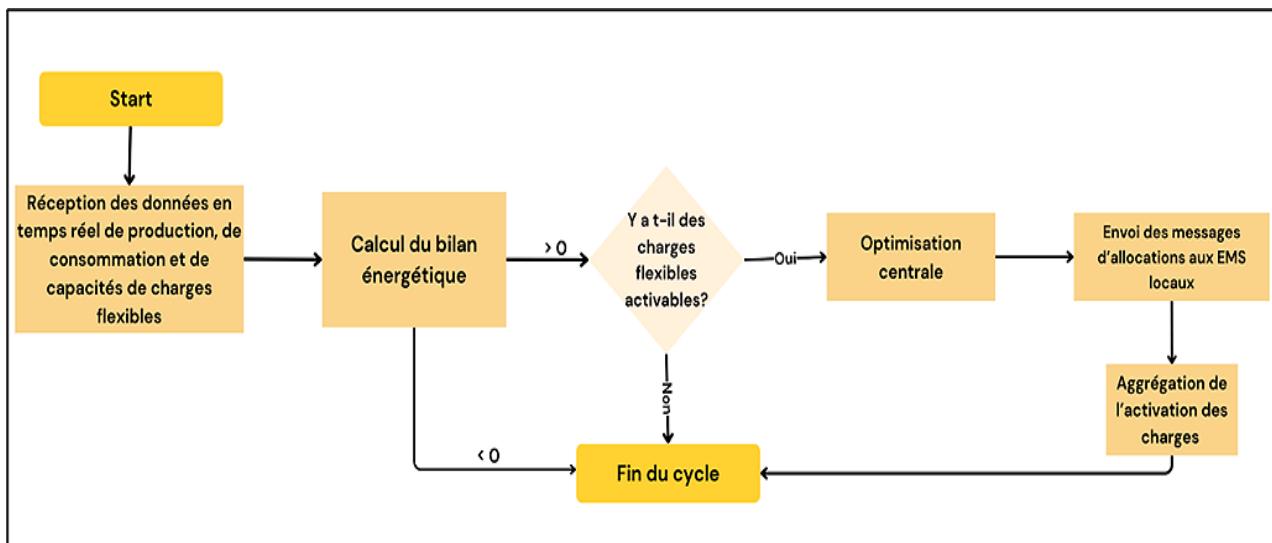


Figure 9 : Organigramme de l'EMS central avec mise à jour périodique (source : auteur)

Le modèle 1 repose sur un EMS central qui collecte de façon périodique les données des EMS locaux. L'EMS central reçoit périodiquement (toutes les 15 minutes) la charge nette et la flexibilité disponible de chaque EMS local (Figure 9). Il calcule le surplus global et distribue via l'optimisation centrale pour déterminer la meilleure allocation possible du surplus pour l'ensemble des EMS locaux sur cette période.

2.1.2. Formulation mathématique du problème d'optimisation centrale

Dans ce modèle, l'EMS central doit résoudre un problème d'optimisation. Le problème est modélisé comme un programme linéaire (PL), ce qui réduit la complexité des calculs. Il reste néanmoins suffisamment flexible pour intégrer des règles d'allocation, telles que des pondérations selon des critères techniques, économiques ou sociaux. Ce PL utilise des variables de décision représentant les quantités d'énergie attribuées à chaque EMS local.

1. Étape 1 : Identifier les variables du procédé (Amabile, 2021).

- Sets :

- U : Ensemble des EMS locaux $u \in U$
- H : Ensemble des pas de temps (ex : $h=1, 2, \dots, 96$ pour une journée par pas de 15 min)
- **Paramètres :**
 - $S(h)$: Surplus d'énergie total disponible au pas de temps h [kW]
 - $Pmax(u, h)$: Puissance maximale que l'EMS local u peut consommer à h (déclarée via une requête ou prédéfinie) [kW]
- **Variables :**
 - $x(u, h) \in [0, Pmax(u, h)]$: Puissance effectivement allouée à l'EMS local u à l'instant h

2. Étape 2 : Sélectionner la fonction objective.

L'EMS central cherche à maximiser la valorisation du surplus d'énergie, en allouant de la puissance aux EMS locaux de façon optimale :

$$\text{Max} \sum_{\substack{h \in H \\ u \in U}} x(u, h) \quad (1)$$

3. Étape 3 : Développer le modèle du procédé et les contraintes.

- Respect du surplus disponible à chaque pas de temps :

$$\sum_{u \in U} x(u, h) \leq S(h) \quad \forall h \in H \quad (2)$$

- Respect des limites locales (flexibilité maximale par EMS) :

$$0 \leq x(u, h) \leq Pmax(u, h) \quad \forall u \in U, \forall h \in H \quad (3)$$

La programmation linéaire est bien adaptée à ce contexte, car elle garantit, sous ses hypothèses, l'optimalité du résultat (autoconsommation maximale). Cette approche permet d'intégrer facilement de nombreux critères dans la fonction objectif (pondérations, prix de l'énergie, etc.) et d'obtenir une allocation optimale du surplus.

Cependant, elle repose sur des calculs synchrones et souvent intensifs. Ce schéma est rigide en temps réel. La charge computationnelle et la synchronisation des

données la rendent moins adaptée à une commande rapide. Ainsi, ce modèle présente une réactivité limitée.

2.2. Modèle 2 : Avec requêtes

2.2.1. Fonctionnement général

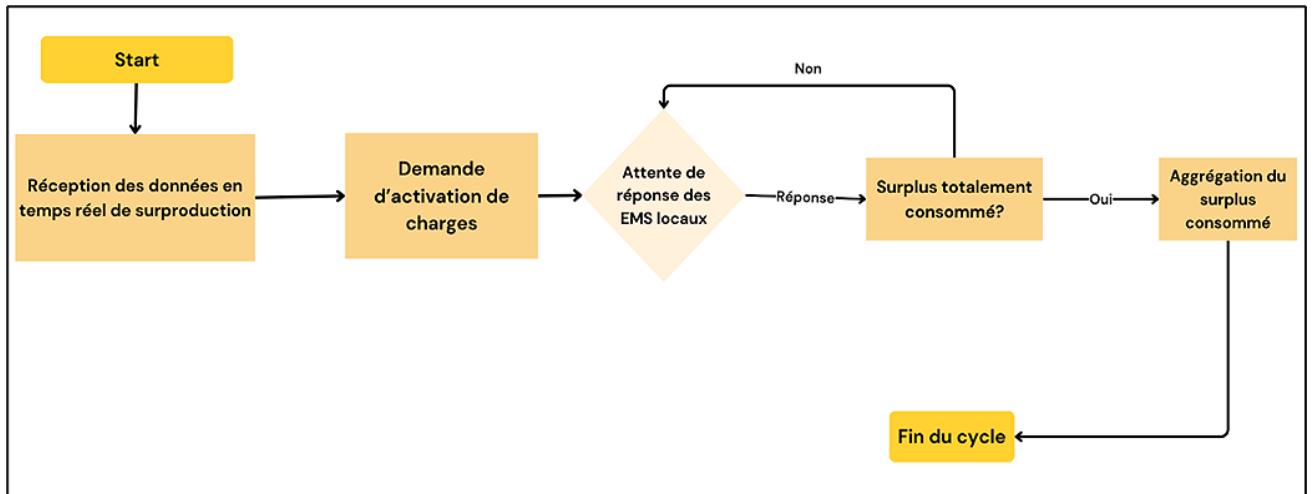


Figure 10 : Organigramme de l'EMS central avec requêtes (source : auteur)

Le modèle 2 est plus interactif. Dès qu'un surplus est détecté, l'EMS central envoie des requêtes aux EMS locaux pour connaître leurs capacités de consommation instantanée à partir des charges flexibles. Les EMS locaux répondent avec une quantité consommable. Au fur et à mesure que les réponses arrivent, l'EMS central alloue immédiatement le surplus disponible selon la règle FIFO (first-in, first-out), c'est-à-dire la première réponse reçue est traitée en premier. Chaque EMS local signale en temps réel sa flexibilité, et le surplus est distribué aux premiers répondants jusqu'à épuisement.

Dans ce fonctionnement, lorsque le surplus est détecté, l'EMS central envoie une requête aux EMS locaux (Figure 10). Ceux-ci répondent avec l'information nécessaire, ce qui permet une allocation en temps réel. Il s'agit d'un échange interactif : l'EMS central demande quelle quantité d'énergie chaque EMS local peut consommer à travers les appareils flexibles qu'il gère. Ensuite, il fait une allocation en fonction de l'arrivée des réponses.

Contrairement aux approches classiques où l'ensemble du système est modélisé dans un unique problème d'optimisation centralisé, ici, l'architecture repose sur une interaction dynamique entre l'EMS central et les EMS locaux.

2.2.2. Approche orientée événement

Le modèle 2 repose sur une logique événementielle. L'EMS central réagit à la détection d'un surplus en déclenchant immédiatement l'envoi de requêtes aux EMS locaux. Ces derniers répondent de manière asynchrone après avoir calculé le potentiel de déplacement de charges. Chaque EMS local agit de manière autonome, répond à son rythme, et peut être temporairement inactif sans perturber l'ensemble. Ce découplage des agents renforce la robustesse.

L'allocation se fait au fil de l'eau, selon une stratégie FIFO (premier arrivé, premier servi). Cela permet une réactivité optimale. Les EMS locaux les plus rapides sont servis en premier sans attendre une vision globale. L'EMS central n'est jamais bloqué, ce qui améliore la fluidité du système.

En contrepartie, ce fonctionnement peut mener à une allocation sous-optimale, car aucune priorisation n'est faite selon les besoins. Néanmoins, dans ce contexte distribué et dynamique, cette réactivité est souvent préférable à une recherche d'optimalité globale.

Par conséquent, ce modèle événementiel évite la nécessité de lourds calculs globaux et permet une allocation immédiate dès réception des premiers retours.

Il assure une grande réactivité et simplicité de mise en œuvre. La stratégie FIFO est facile à programmer et ne nécessite ni solveur complexe ni calcul intensif. Elle convient à des systèmes distribués où les décisions doivent être prises rapidement.

Cependant, l'allocation peut être sous-optimale. En effet, ce procédé ne prend pas en compte les priorités ou l'importance relative des demandes. Un EMS qui répond en premier reçoit prioritairement le surplus disponible, et ce, quel que soit son niveau de besoin. Cela peut conduire à une utilisation non optimale des ressources, car certaines demandes critiques tardives peuvent rester insatisfaites. Le résultat dépend fortement de l'ordre d'arrivée des réponses et ne garantit pas une maximisation globale contrairement à la PL.

2.3. Tableau comparatif des deux modèles

Nous allons maintenant comparer ces deux approches selon plusieurs critères (Tableau 2).

Tableau 2 : Tableau comparatif entre l'approche FIFO et la programmation linéaire pour l'allocation du surplus

Critère	Modèle 1 : Programmation Linéaire	Modèle 2 : FIFO
Architecture	Centralisée	Semi-distribuée
Nature	Optimisation mathématique	Heuristique simple
Objectif	Optimalité selon un objectif	Ordre d'arrivée
Flexibilité	Élevée	Faible
Type d'allocation	Allocation optimisée	Allocation directe
Temps de calcul	Plus élevé, dépend du modèle (variable continue/entière)	Faible
Rôle des EMS locaux	Passifs	Actifs
Réactivité	Lente	Rapide
Qualité de l'allocation	Optimale	Approximative

FIFO est un algorithme de gestion de files basé sur la priorité temporelle tandis que la programmation linéaire (PL) cherche une allocation optimale basée sur un objectif et des contraintes.

Dans le modèle 1 (PL), les EMS locaux jouent un rôle passif, se limitant à fournir des données de manière périodique. En revanche, dans le modèle FIFO, ils adoptent un comportement plus actif en initiant des requêtes vers l'EMS central dès qu'un besoin est identifié, au moment opportun. Cette différence modifie clairement la dynamique de coordination et le degré de centralisation.

Le modèle 1 (PL) repose sur une architecture hiérarchique centralisée, dans laquelle l'EMS central collecte les données des EMS locaux et détermine seul l'allocation optimale du surplus via une programmation linéaire.

Le modèle 2 (FIFO) s'inscrit toujours dans une architecture hiérarchique, mais avec une coordination partiellement distribuée. L'EMS central se contente d'ordonner les demandes des EMS locaux selon une logique FIFO, sans optimisation. Cela réduit la centralisation et améliore la réactivité au détriment de l'optimalité. Le modèle 2 opère selon un fonctionnement moins centralisé que le modèle 1.

2.4. Considération des contraintes techniques et temporelles

À l'issue de l'analyse des deux modèles de fonctionnement de l'EMS central, le modèle 2 (avec requêtes) a été retenu pour la suite de ce travail. Ce choix s'explique par sa meilleure réactivité, sa capacité à s'adapter à un système distribué dynamique et sa compatibilité avec une architecture événementielle. Le modèle 2 est particulièrement pertinent dans un contexte où les conditions locales (surplus, flexibilité, réponse des agents) évoluent rapidement et où une allocation immédiate est préférable à une optimisation centralisée plus lourde.

Ainsi, l'algorithme FIFO a été retenu pour coordonner l'utilisation du surplus photovoltaïque et satisfaire aux nombreuses contraintes opérationnelles des équipements pilotés. Cet algorithme simple, qui traite les demandes d'énergie dans leur ordre d'arrivée, s'avère efficace pour optimiser le surplus tout en respectant les contraintes techniques et temporelles des appareils (pompes à chaleur, ballons d'eau chaude, etc.).

D'un point de vue technique, de nombreux dispositifs électriques imposent des seuils de fonctionnement à respecter comme une puissance minimale en dessous de laquelle l'appareil ne peut pas opérer de façon stable (boiler électrique, PAC), ou encore une limitation du nombre de cycles marche/arrêt par heure afin de prévenir l'usure prématuée des équipements (pompe à chaleur). Un algorithme de gestion doit donc éviter de solliciter un appareil avec une puissance trop faible et de manière répétée.

Cette stratégie de répartition avec l'algorithme FIFO garantit que le premier EMS local ayant répondu obtient la totalité de la puissance qu'il peut techniquement absorber, plutôt qu'une fraction potentiellement inutilisable. En effet, certaines charges ont une puissance minimale d'activation. En fournissant la demande complète au premier répondant, on s'assure de respecter ces contraintes techniques.

Par exemple, une pompe à chaleur est un équipement sensible au cyclage excessif. Les démarrages et arrêts trop fréquents réduisent son efficacité énergétique et accélèrent l'usure du compresseur. Le rapport de l'Annexe 57 de l'IEA recommande de limiter le nombre de cycles de démarrage par heure pour éviter des coûts de maintenance accrus (International Energy Agency, 2024a). L'approche FIFO, en servant les demandes dans l'ordre et en maintenant la répartition tant que le surplus le permet, minimise ce risque. Une fois que la PAC a démarré, elle reste alimentée pendant toute la fenêtre temporelle, ce qui lui permet de fonctionner en continu et d'éviter les cycles courts (Brudermueller, 2023). L'étude de Fuhrmann et al. (2022) sur la gestion optimisée des systèmes de chauffage confirment l'importance d'imposer de telles contraintes pour éviter une régulation inefficace et l'usure due au cyclage.

De plus, un boiler nécessite une puissance suffisante, souvent proche de sa puissance nominale, pour qu'il puisse s'activer. Une allocation trop fragmentée ne permettrait pas de déclencher la résistance. Grâce à l'allocation séquentielle du FIFO, un EMS local peut recevoir la totalité de la puissance nécessaire pour atteindre ce seuil dans la limite du surplus disponible, garantissant un fonctionnement efficace du boiler. Cela évite une dispersion du surplus en petites fractions inexploitables.

Sur le plan temporel, l'algorithme FIFO traite également efficacement les contraintes de temps de cycle et de priorisation. L'algorithme FIFO ne réalloue pas en permanence le surplus d'un EMS local à un autre en fonction des nouvelles demandes. Une fois qu'un EMS local a reçu une allocation d'énergie, il conserve cette quantité pour la durée du cycle (15 minutes). Cette approche garantit que les EMS locaux peuvent planifier l'activation de leurs charges flexibles sans interruption brutale, ce qui est essentiel pour le bon fonctionnement de certains équipements. Cette logique évite par exemple d'avoir des cycles de chauffe interrompus qui réduiraient l'efficacité énergétique globale. L'allocation FIFO assure que le premier EMS à avoir répondu et reçu une part de surplus peut l'utiliser pleinement. Cela garantit une forme de stabilité temporelle dans l'optimisation de la consommation de l'énergie attribuée.

Le FIFO est un algorithme non-préemptif où une tâche (la consommation d'un surplus par un EMS) une fois lancée n'est pas stoppée avant son achèvement (GeeksforGeeks, 2025). Cela en fait un schéma simple mais adapté aux systèmes énergétiques contenant de l'inertie thermique. En somme, l'algorithme FIFO combine simplicité et respect des contraintes opérationnelles. Il nécessite peu de calculs, évite les conflits de ressources (il ne cherche pas à satisfaire tout le monde) et garantit une exploitation du surplus dans le respect des limites techniques (puissance, cycles) fixées par les équipements.

2.5. Alternatives à l'allocation FIFO

Plusieurs approches différentes de l'allocation FIFO peuvent être envisagées pour répartir le surplus photovoltaïque, chacune avec des principes différents. On pourrait considérer une logique d'allocation équitable, une méthode itérative d'optimisation, ou encore une répartition proportionnelle du surplus entre usagers. Il convient d'en discuter les avantages et limites, notamment en ce qui concerne la faisabilité d'implémentation et l'efficacité opérationnelle.

- Allocation équitable (Énergie Plus, 2022): cette approche vise à distribuer le surplus de manière égale entre tous les utilisateurs, indépendamment de leurs besoins spécifiques. Par exemple, dans une communauté de 10 foyers, chacun recevrait 10 % du surplus disponible à un instant t , quelle que soit sa consommation réelle à ce moment. Si ce principe semble juste sur le plan social, il présente d'importantes limites pratiques. En opérationnel, une allocation strictement égale peut mener à une sous-utilisation du surplus si certains participants n'ont pas de charge flexible disponible pour leur part d'énergie. Une répartition équitable pure nécessiterait de réserver une part du surplus à chaque acteur même si celui-ci ne peut l'utiliser, ce qui complexifie la gestion en temps réel de l'allocation. Ainsi, bien que simple à mettre en œuvre, cette stratégie présente une efficacité énergétique sous-optimale et une réactivité limitée en temps réel.
- Allocation proportionnelle : une stratégie plus pratique consiste à répartir le surplus proportionnellement à un certain critère, par exemple aux charges flexibles instantanées de chaque utilisateur. C'est d'ailleurs une méthode souvent retenue dans les schémas d'autoconsommation collective encadrés

(répartition du surplus au prorata de la consommation). Cette approche a l'avantage d'éviter certains gaspillages par rapport au partage égalitaire, car elle oriente plus d'énergie vers ceux qui peuvent l'absorber. Cependant, elle comporte aussi des limites. En temps réel, connaître et suivre précisément les demandes de chacun pour calculer les proportions peut exiger des calculs plus lourds. Ainsi, l'allocation proportionnelle améliore l'efficacité par rapport à l'égalité stricte, mais nécessite de bien choisir la métrique de proportion. Elle ne garantit pas non plus une optimisation globale du surplus à chaque instant et présente une réactivité limitée en temps réel.

- Approches itératives (Scarcello, 2022) : il s'agit ici de méthodes plus sophistiquées, souvent basées sur des algorithmes d'optimisation mathématique (par exemple méthodes de consensus en plusieurs passes). Le principe général est de calculer une allocation optimale du surplus qui maximise un certain objectif tout en respectant les contraintes de chacun. Une méthode itérative pourrait, par exemple, allouer d'abord le surplus à l'équipement le plus prioritaire, puis réallouer l'éventuel reste à la seconde priorité, et ainsi de suite, jusqu'à épuisement du surplus disponible. D'autres variantes reposent sur des enchères ou échanges successifs entre participants de la communauté pour ajuster l'allocation de manière quasi-optimale. Sur le papier, ces méthodes peuvent approcher près de l'optimum global. Cependant, leur mise en œuvre pratique est complexe : il est nécessaire d'avoir une capacité de calcul relativement rapide. Elles nécessitent une connaissance fine en temps réel de l'état des appareils et des demandes. De plus, les algorithmes itératifs peuvent souffrir de problèmes de convergence ou de latence. Ils n'arrivent pas toujours à une solution stable et il faut parfois plusieurs cycles d'ajustement avant d'y arriver. Enfin, d'un point de vue opérationnel, ces méthodes optimales risquent d'être peu robustes face à des imprévus. Ainsi, une donnée manquante peut les conduire à une solution sous-optimale. En résumé, les approches itératives offrent en théorie la meilleure utilisation du surplus, mais au prix d'une complexité élevée et de défis d'implémentation en temps réel (exigences de calcul et de communications rapides, maintenance logicielle complexe).

Le Tableau 3 ci-dessous résume les caractéristiques de chaque alternative :

Tableau 3 : Comparaison des alternatives par rapport à l'allocation FIFO

Méthode	Principe	Avantages	Limites	Réactivité
FIFO	Répond aux requêtes dans l'ordre d'arrivée	Simple, robuste, facile à implémenter	Biais temporel, pas toujours équitable ni optimal	Très bonne, allocation au fil de l'eau
Allocation équitable	Répartition égale du surplus entre tous les membres	Socialement juste, très simple à appliquer	Risque de gaspillage, manque d'adaptation aux besoins réels	Moyenne, doit connaître le nombre de participants prêts à consommer
Allocation proportionnelle	Distribution selon un critère (charge flexible disponible, consommation, etc.)	Meilleure efficacité énergétique, plus adaptée aux capacités de chacun	Besoin de données précises en temps réel, plus complexe à calculer	Faible, doit collecter et agréger les niveaux de charge
Approches itératives	Algorithmes d'optimisation par passes successives ou enchères	Optimisation théorique proche du maximum collectif	Très complexe, exigeant en calcul et communication, sensible aux imprévus	Faible, nécessite plusieurs cycles de calcul

En synthèse, si l'on compare ces alternatives à l'algorithme FIFO utilisé, on constate un compromis entre simplicité et optimalité. FIFO, bien que basique, offre une robustesse et une facilité d'implémentation tout en garantissant une utilisation efficace du surplus d'un point de vue technique. De plus, FIFO fonctionne avec des données incomplètes : il n'a pas besoin d'attendre que tous les EMS locaux répondent pour commencer l'allocation. C'est un atout fort pour les systèmes asynchrones ou réactifs.

Le choix d'un algorithme de gestion collective du surplus doit donc tenir compte non seulement de la performance attendue en théorie, mais aussi de la complexité

d'implémentation, de l'efficacité de la communication, de la réactivité et des contraintes techniques et temporelles des équipements.

3. Comment assurer l'allocation en temps réel ?

Ce chapitre examine comment assurer l'allocation en temps réel du surplus. Ces aspects s'appliquent aux modèles présentés précédemment. Deux questions principales y sont analysées :

- Quel effet si on considère une priorité dans le traitement des réponses ?
- Quel effet si les requêtes sont envoyées successivement ou simultanément ?

Pour répondre à ces questions, nous devons comprendre l'inhérence du processus en temps réel. Dans un système temps réel, la vitesse de résolution est aussi cruciale que la qualité des résultats obtenus.

3.1. Priorité dans le traitement des réponses

Faut-il attendre toutes les réponses pour optimiser l'allocation, ou traiter les informations dès leur arrivée ? Les réponses des EMS locaux peuvent être traitées selon différentes stratégies de priorité. Deux approches sont envisageables :

- Avec priorité : par exemple en privilégiant les EMS proposant la plus grande flexibilité disponible. Dans ce cas, l'EMS central doit attendre la réception de l'ensemble des réponses avant de procéder à l'allocation du surplus énergétique. Ce traitement s'apparente davantage à une logique d'optimisation centrale. Toutefois, il introduit un délai, car le système reste bloqué jusqu'à la réception des réponses les plus lentes.
- Sans priorité (FIFO) : les réponses sont traitées dès leur arrivée. L'EMS central alloue alors immédiatement une partie du surplus à l'EMS ayant répondu, sans attendre les autres. L'allocation est ainsi déclenchée de façon instantanée, dans l'ordre chronologique des retours. Cette approche permet de réduire l'attente inutile des réponses qui arrivent rapidement et d'améliorer la réactivité du système, même si elle peut entraîner une allocation sous-optimale du point de vue globale.

Une solution hybride (par exemple, attendre les réponses dans un délai donné) pourrait également être envisagée selon les contraintes du contexte.

3.2. Envoi successif des requêtes

L'envoi successif des requêtes peut s'inscrire dans une logique de proximité des EMS locaux. Par exemple, les EMS locaux les plus proches reçoivent les requêtes en premier. Ainsi, les requêtes sont envoyées successivement, en commençant avec l'EMS le plus proche, tandis que l'EMS le plus éloigné reçoit la requête en dernier. Prenons un exemple avec des temps de réponse choisis arbitrairement :

Tableau 4 : Exemple de temps de réponse des EMS locaux

EMS local	Temps de réponse
1	24
2	12
3	3

Dans le Tableau 4, les EMS sont classés selon leur proximité au site de production : l'EMS local 1 est le plus proche, l'EMS local 3 le plus éloigné.

L'EMS central envoie d'abord une requête à l'EMS local 1 pour lui demander s'il peut consommer le surplus. Il attend 24 unités de temps pour recevoir la réponse. Si le surplus n'est pas entièrement absorbé, il envoie ensuite une requête à l'EMS local 2 et attend 12 unités de temps supplémentaires, puis fait de même avec l'EMS local 3.

Suivant ce raisonnement :

- Réponse de l'EMS local 1 après 24 unités de temps.
- Réponse de l'EMS local 2 après $24 + 12 = 36$ unités de temps.
- Réponse de l'EMS local 3 après $36 + 3 = 39$ unités de temps.

Le temps d'attente moyen de l'EMS central est donc :

$$\text{Temps d'attente moyen} = \frac{(24 + 36 + 39)}{3} = 33 \text{ unités de temps}$$

3.3. Envoi simultané des requêtes

Reprendons l'exemple du Tableau 4, cette fois avec un envoi simultané des requêtes à tous les EMS locaux.

Les réponses sont reçues selon les temps de réponse individuels :

- EMS local 3 : 3 unités de temps
- EMS local 2 : 12 unités de temps
- EMS local 1 : 24 unités de temps

Le temps d'attente moyen est alors :

$$\text{Temps d'attente moyen} = \frac{(3 + 12 + 24)}{3} = 13 \text{ unités de temps}$$

Dans ce cas, l'envoi simultané permet de réduire considérablement le temps d'attente moyen de l'EMS central (13 unités contre 33 en envoi successif). En considérant que le traitement en temps réel favorise un temps d'attente minimal, l'envoi simultané apparaît comme la solution la plus efficace. Ainsi, le FIFO mise sur la rapidité et la simplicité, assurant une allocation en temps réel.

4. Choix du modèle (Modèle 2 : Avec requêtes)

L'EMS central alloue le surplus selon la stratégie « Premier arrivé, premier servi » (FIFO). C'est une approche simple, mais tout à fait légitime dans un contexte distribué avec des systèmes qui répondent à des requêtes.

Le déroulement d'un cycle d'une durée de 15 minutes peut être résumé en plusieurs étapes :

- L'EMS central évalue le surplus de production non consommé disponible à un instant donné et envoie une requête à l'ensemble des EMS locaux pour proposer ce surplus à la communauté.
- Chaque EMS local calcule, en fonction de ses propres contraintes et de la flexibilité de ses équipements, la puissance additionnelle qu'il peut absorber durant ce cycle (par exemple en chargeant un VE). Il répond à l'EMS central avec une demande de puissance qu'il est capable de consommer.
- À la réception des réponses, l'EMS central distribue le surplus disponible dans l'ordre d'arrivée des demandes. Concrètement, le premier EMS local à répondre se voit allouer la puissance qu'il a demandée (dans la limite du surplus total). S'il reste du surplus, l'EMS central passe au deuxième qui a

répondu et lui attribue à son tour ce qu'il a sollicité, et ainsi de suite jusqu'à épuisement du surplus ou satisfaction de toutes les demandes.

- Chaque EMS local sélectionné procède à l'activation de ses appareils flexibles pour consommer la puissance qui lui a été allouée.

4.1. Coordination centrale d'agents locaux homogènes et dynamiques

Dans le cadre du projet LOCOMOTRICE, les agents locaux (EMS locaux) communiquent avec un agent central afin d'atteindre l'objectif global. Les agents locaux ont tous le même rôle, on parle ainsi d'agents homogènes. Cela implique que tous les agents sont identiques du point de vue comportemental, leurs actions possibles étant définies par un même protocole commun (De Masellis, 2020). L'impact de ces actions ne dépend pas de l'identité précise des agents les réalisant, mais uniquement du nombre d'agents exécutant chaque action. Cela permet de simplifier fortement la modélisation et le contrôle du système avec une grande flexibilité, et une réactivité dynamique aux changements du nombre d'agents locaux (De Masellis, 2020). L'identité précise de l'EMS local n'influence pas le processus global, seulement le nombre total d'agents locaux capables de fournir une flexibilité énergétique à un instant donné compte.

Dans le modèle EMS développé, la dynamique signifie que le nombre des EMS locaux présents varie au cours du temps (De Masellis, 2020). Concrètement, à chaque étape de la simulation :

- un nombre variable d'EMS locaux (agents locaux) est considéré comme actif ;
- chaque EMS actif peut décider, à chaque cycle, de répondre ou non aux requêtes envoyées par l'EMS central, simulant ainsi une activation ou désactivation temporaire.

Cela reflète bien la réalité d'une CE où certains EMS locaux peuvent être momentanément indisponibles ou décider de ne pas participer à une demande de consommation, améliorant ainsi la robustesse. Les EMS locaux peuvent entrer ou quitter le système, signifiant que les membres rejoignent ou quittent la communauté d'énergie. Ainsi, contrairement à un système paramétrique où le nombre d'agents

serait fixe tout au long de l'évolution, le modèle permet des variations régulières dans le nombre d'agents locaux actifs.

L'architecture centralisée est limitée par le temps des calculs lorsque le nombre d'agents pris en compte augmente. Pour compenser l'augmentation de l'incertitude due à l'introduction d'un grand nombre d'agents, il est souhaitable de réduire le temps de la résolution. Or, la seule manière de réduire ce temps est de ne mettre à jour que les données des agents qui ont répondu assez rapidement. Cette résolution réduit le temps de résolution, mais introduit des écarts avec la solution optimale (Dong, 2022).

Dans ce contexte, l'EMS central lance simultanément un grand nombre de requêtes. Le coordinateur (agent central) peut recevoir et traiter les réponses des EMS locaux (agents locaux) dès qu'elles arrivent, sans attendre que toutes soient terminées, ce qui permet une gestion fluide et réactive. Chaque agent coopère en fournissant son information, et le coordinateur orchestre leur contribution pour atteindre l'objectif global, qui est l'autoconsommation collective.

Chaque EMS local exécute sa tâche indépendamment, et leurs réponses sont traitées dès qu'elles sont disponibles. Des études soulignent que cet asynchronisme dans les communications est un facteur clé d'indépendance et de robustesse des agents, ceux-ci pouvant « envoyer et recevoir des messages sans attendre de réponses immédiates », maintenant ainsi leurs activités sans blocage (Carrascosa, 2024). C'est le caractère asynchrone de l'EMS central : chaque sous-agent envoie son information quand il peut, et le centre fait au mieux avec ce qu'il a.

Chaque EMS local, considéré comme un sous-agent homogène, exécute la même tâche et communique sa réponse de manière asynchrone. Les systèmes multi-agents (MAS) homogènes présentent l'avantage d'être plus simples à implémenter, car il suffit de définir une seule fois le comportement type d'un agent, puis de le dupliquer pour chaque instance. Chaque agent a un identifiant unique, mais partage exactement la même logique comportementale (De Masellis, 2020).

Cette approche réduit le risque d'incohérences entre les agents et garantit une uniformité de comportement. Par exemple, si chaque EMS local suit exactement la même procédure pour répondre aux requêtes du système central (par exemple, le

calcul de sa flexibilité), on obtient une prise de décision cohérente à l'échelle du système. En effet, dans ce contexte, la diversité des comportements des agents locaux n'est pas nécessaire.

4.2. Le modèle multi-agent événementiel

Le modèle événementiel a été choisi en raison de son fonctionnement très proche de celui d'un système énergétique distribué. Plusieurs raisons justifient ce fonctionnement :

- Les interactions sont naturellement pilotées par des événements : dans un système d'énergie, beaucoup de choses se passent en réaction à des événements. Par exemple, un surplus d'énergie peut survenir soudainement (soleil intense, baisse soudaine de la demande de la communauté) ; c'est un événement déclencheur. L'EMS central doit alors en être informé et réagir en conséquence pour déclencher une redistribution. On évite ainsi le polling incessant (demander en boucle "et maintenant ?" à chaque EMS local), ce qui serait inefficace et passerait à côté de la spontanéité des phénomènes. Au lieu de cela, chaque agent est en attente d'événements et réagit dès que nécessaire, un peu comme un système de surveillance qui ne déclenche une alarme que s'il se passe quelque chose.
- Découplage entre agents : un argument majeur en faveur de la hiérarchie asynchrone est qu'elle découple les agents. D'après Christie et al. (2022), « *utiliser une communication asynchrone réduit le couplage entre agents* ». En pratique, cela signifie que chaque EMS local peut fonctionner à son propre rythme, indépendamment des autres. S'il met 2 secondes à calculer sa flexibilité alors que les autres répondent en 100 ms, l'EMS central n'est pas bloqué, il peut déjà traiter les premières réponses ou faire autre chose en parallèle (par exemple, gérer un autre événement). Ainsi le système global continue de fonctionner même si un agent local est lent.
- Coordination flexible : le mode événementiel asynchrone permet aussi d'envisager des interactions plus complexes qu'une simple requête-réponse unique. Par exemple, on pourrait implémenter un échange itératif : l'EMS central envoie une requête, reçoit des propositions partielles, renvoie une contre-proposition, et ainsi de suite, tout en restant non bloquant. Ce serait

utile si on voulait affiner l'allocation du surplus par ajustements successifs. Les événements conditionnels (si la production est plus élevée que la consommation, on déclenche l'allocation du surplus) non bloquants rendent cela possible sans engorger le système.

- Meilleure réactivité et scalabilité : le programme est plus dynamique aux changements d'état grâce à la boucle d'événements qui surveille en permanence les tâches, ce qui est important dans un contexte temps réel. Il est possible de modéliser avec un nombre important d'EMS locaux : ajouter plus d'agents n'alourdit pas proportionnellement l'algorithme s'il est bien conçu.

En somme, le modèle multi-agent événementiel asynchrone est naturellement adapté au cas étudié. Il apporte souplesse, rapidité et robustesse en épousant le caractère distribué du problème. Comme l'indiquent Christie et al. (2022), concevoir une application multi-agent revient à concevoir des agents autonomes qui se coordonnent via des messages asynchrones, car cela évite de dépendre de garanties lourdes d'ordonnancement (ordonner les tâches à exécuter) et limite les interférences entre agents. En synthèse, les choix techniques du projet (concurrence asynchrone des tâches, agents locaux homogènes, architecture hiérarchique) s'alignent avec les recommandations issues de la recherche.

5. Choix des techniques de programmation

5.1. Programmation concurrente en Python

Un programme concurrent, contrairement à un programme séquentiel, permet de gérer plusieurs tâches en chevauchant leur exécution (Anderson, 2024). Il n'a pas besoin d'attendre qu'une tâche soit terminée pour en commencer une autre. L'ordre dans lequel les tâches sont exécutées est donc variable, ce qui permet une utilisation efficace du temps en exécutant les tâches qui sont déjà prêtes.

Ceci est extrêmement utile lorsque les tâches sont liées aux entrées/sorties (I/O-bound), par exemple l'attente de réponses des EMS locaux, par opposition aux tâches liées au processeur (CPU-bound), qui requièrent des calculs intensif (Anderson, 2024). Étant donné que les tâches de l'EMS central ne requièrent pas de

calculs intensifs, mais consistent principalement à attendre les réponses des EMS locaux, le programme se concentrera sur la gestion efficace des tâches I/O-bound.

5.2. Méthodes de concurrence en Python

Il existe principalement deux méthodes pour implémenter la programmation concurrente adaptée aux tâches I/O-bound en Python. La première est le module *multithreading*, qui permet au programme de créer plusieurs threads, chacun représentant un fil d'exécution chargé d'exécuter des instructions. Grâce au multithreading, l'exécution des threads peut se chevaucher dans le temps donnant l'illusion d'une exécution parallèle (Anderson, 2024). Le système d'exploitation effectue une commutation de contexte entre les différents threads, ce qui permet au programme de progresser pendant que certaines tâches sont en attente d'opérations d'entrée-sortie. Toutefois, un seul thread peut être exécuté à la fois par le processeur (en raison du verrou d'interprétation global en Python, le GIL), ce qui peut entraîner un blocage des tâches (Anderson, 2024). Le multithreading est relativement simple à mettre en œuvre et améliore significativement la gestion des tâches en réduisant les temps d'attente inactifs (Anderson, 2024).

Une autre approche pour gérer des tâches concurrentes en Python est l'asynchronisme, notamment à travers la bibliothèque *asyncio*, qui permet de gérer plusieurs tâches sans bloquer le flux principal, le tout au sein d'un unique fil d'exécution (thread). Les tâches asynchrones sont représentées par des coroutines, fonctions spécialisées permettant l'exécution non-bloquante d'opérations I/O (Python Software Foundation, s.d.). Cette approche repose sur une boucle d'événements qui orchestre l'exécution de multiples tâches I/O-bound sans recourir à la création d'un thread par tâche. Elle nécessite toutefois une coopération entre les tâches, ces dernières devant signaler volontairement les moments où elles peuvent être mises en pause, sans intervention directe du système d'exploitation (Anderson, 2024). Ainsi, on ne perd pas de temps sur les tâches qui ne sont pas prêtes à s'exécuter immédiatement (Lee, 2020).

Le Tableau 5 ci-dessous reprend les différences entre les deux méthodes de programmation concurrentes liées aux tâches I/O-bound :

Tableau 5 : Caractéristiques des méthodes de concurrence

Module	Coordination des tâches	Gestion de l'attente	Tâches I/O-bound
Multithreading	Anticipée par le système d'exploitation (SE)	Gérée par le SE grâce à la commutation de contexte	Souvent bloquante
AsyncIO	Coopération entre tâches sans faire intervenir le SE	Gérée par les coroutines dans une boucle d'évènements	Non-bloquante

5.3. Evaluation de la performance des algorithmes de l'EMS central

Pour évaluer la performance de l'algorithme, on va comparer les différentes approches de programmation concurrente en mesurant le temps d'exécution de l'algorithme de l'EMS central en fonction du nombre d'EMS locaux (comparaison asymptotique). Comme il s'agit d'évaluer le temps d'exécution, on parle de complexité en temps de l'algorithme.

Tableau 6 : Temps d'exécution de l'algorithme d'allocation (seconde)

Nombre d'EMS	Threading manuel	Thread Pool	AsyncIO (coroutines)
1000	2.2	2.3	2.0
10000	4.7	4.5	2.2
50000	16.9	10.5	3.5
100000	34	22	5.2

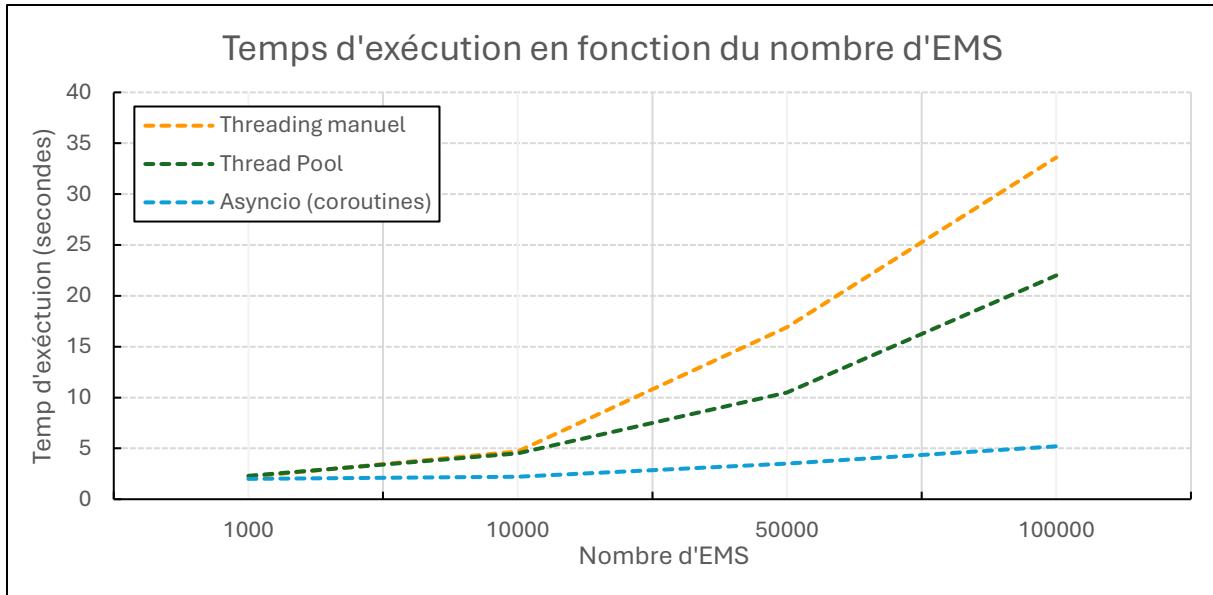


Figure 11 : Temps d'exécution en fonction du nombre d'EMS locaux des différentes approches (source : auteur)

Pour évaluer la complexité en temps de l'algorithme d'allocation en fonction du nombre d'EMS, on suppose que le temps d'exécution $T(n)$ peut être modélisé par une relation de la forme :

$$T(n) = C \cdot n^k \quad (4)$$

où :

- n est le nombre d'EMS (agents),
- C est une constante,
- k est l'exposant qui reflète la complexité algorithmique.

Pour cela, on applique le logarithme aux deux côtés de l'équation pour obtenir une forme linéaire :

$$\log(T(n)) = \log(C) + k \cdot \log(n) \quad (5)$$

Cette équation est de la forme :

$$y = a + kx \quad (6)$$

Avec :

- $y = \log(T(n))$
- $x = \log(n)$
- $a = \log(C)$

- k = pente de la droite, c'est l'indicateur de complexité.

Ensuite, on applique une régression linéaire entre $\log(n)$ et $\log(T(n))$ pour estimer k . Cela revient à ajuster une droite aux données pour calculer la pente.

Le coefficient k obtenu représente la croissance du temps d'exécution en fonction du nombre d'agents (Figure 12) :

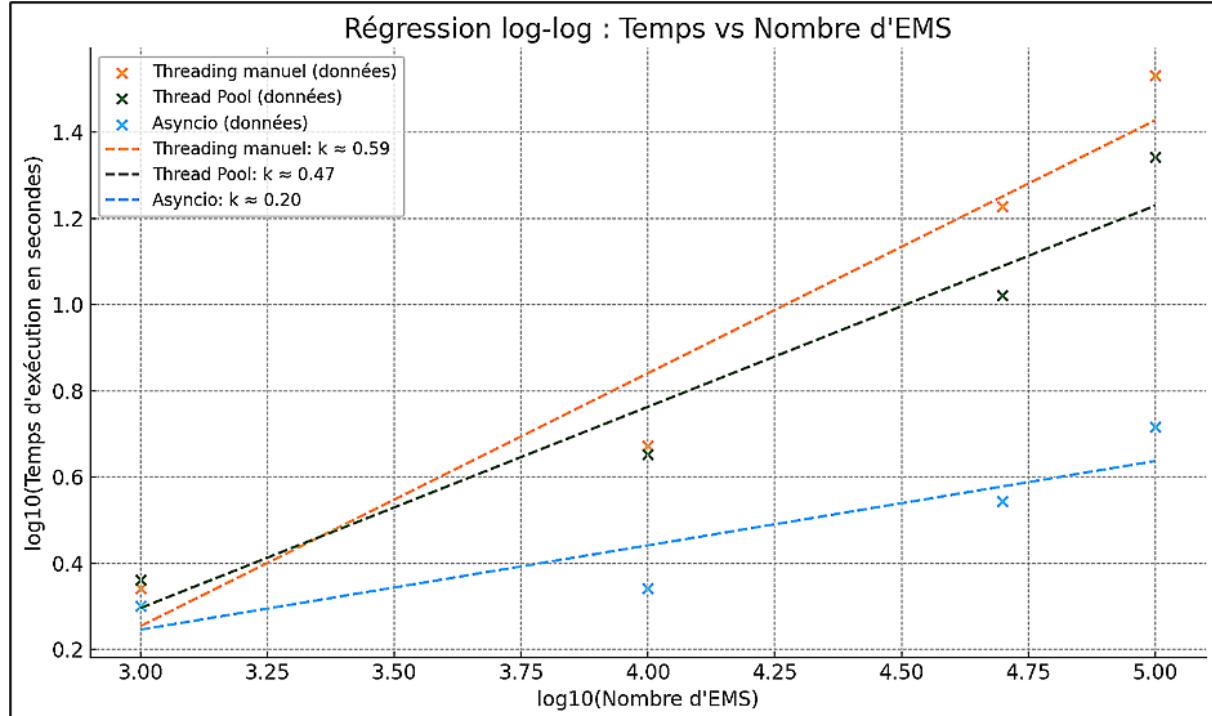


Figure 12 : Régression log-log du temps d'exécution en fonction du nombre d'EMS locaux des différentes approches (source : auteur)

Le Tableau 7 affiche de manière synthétique les résultats obtenus du graphique :

Tableau 7 : Pente (Exposant k) des différentes méthodes de programmation

Méthode	Exposant k
Threading manuel	0.59
Thread Pool	0.47
AsyncIO	0.20

Cela permet de conclure que la solution basée sur les coroutines asynchrones (AsyncIO) présente une excellente scalabilité et constitue l'approche la plus efficace

pour gérer un grand nombre d'EMS locaux, car elle présente la plus faible pente (exposant k) : le temps d'exécution augmente très peu avec le nombre d'EMS locaux croissant.

5.4. Implémentation de l'algorithme

L'algorithme coordonne la répartition optimale de la puissance flexible entre différents EMS locaux. Il agit en fonction d'un contexte dynamique : réponses asynchrones, disponibilité variable du nombre d'EMS locaux et de leur flexibilité, gestion du surplus en temps réel.

Ci-dessous nous présentons le pseudocode simplifié de l'algorithme :

Pseudocode de l'algorithme de coordination multi-agent

Entrées : Ensemble de charges nettes $L = \{l_i\}$ des EMS locaux

Sorties : Allocation de puissance flexible, surplus restant, statistiques de réponse

- 1: **Calculer** surplus $\leftarrow \sum l_i \in L$
- 2: **Si** surplus ≤ 0 **Alors**
- 3: **retourner** "Aucune allocation nécessaire"
- 4: **Sinon**
- 5: Envoyer requêtes parallèles aux EMS locaux actifs pour obtenir leur puissance flexible
- 6: **Tant que** surplus > 0 **et** réponses EMS restantes **existant Faire**
- 7: Attendre la réception de la prochaine réponse d'un EMS
- 8: **Si** réponse reçue **Alors**
- 9: allocation $\leftarrow \min(\text{réponse.puissance_flexible}, \text{surplus})$
- 10: surplus $\leftarrow \text{surplus} - \text{allocation}$
- 11: enregistrer (allocation, réponse.EMS_id, temps de réponse)
- 12: **Sinon** surplus = 0 **Alors**
- 13: annuler les requêtes restantes
- 14: **Fin Si**
- 15: **Fin Tant que**
- 16: **Fin Si**
- 17: **Calculer** le pourcentage de surplus utilisé
- 18: **retourner** allocations, surplus restant, temps de réponse

5.5. Illustration du comportement de l'algorithme dans différents cas-type

Dans cette section, nous illustrons le fonctionnement de l'algorithme à travers trois scénarios représentatifs :

- (0) aucun surplus disponible,
- (1) un surplus disponible entièrement réparti entre les EMS locaux,
- (2) un surplus disponible, mais seulement partiellement consommé.

Ces cas ne résultent pas d'une simulation aléatoire complète, mais sont construits de manière volontaire pour couvrir différentes situations possibles. Cela permet de vérifier la logique de l'algorithme selon la disponibilité du surplus et la flexibilité déclarée des EMS locaux.

À ce stade, le déséquilibre entre production et consommation n'est pas encore modélisé finement. En réalité, ce déséquilibre sera un facteur clé : plus les profils de consommation et de production sont désynchronisés, plus la coordination devient nécessaire pour maximiser l'autoconsommation collective.

Le nombre d'EMS locaux actifs varie selon les cas définis. L'algorithme reproduit cette dynamique et montre comment il alloue ou non le surplus en fonction des capacités déclarées par chaque EMS local.

5.5.1. Cas 0 : Pas de surplus disponible

Lorsque la somme des charges nettes de la communauté est inférieure ou égale à zéro, l'algorithme ne lance aucune requête et affiche le message suivant :

- Surplus calculé : -7 kW | Charge nette : [-5, 5, -9, 2, 0]
- Aucun surplus disponible, aucune requête lancée.
- Surplus total alloué : 0 kW

Ce comportement découle d'un paramétrage que j'ai volontairement défini afin de produire ces résultats.

5.5.2. Cas 1 : Epuisement du surplus (réutilisation à 100%)

Dans le cas où un surplus est disponible et totalement alloué aux EMS locaux, les résultats sont les suivants :

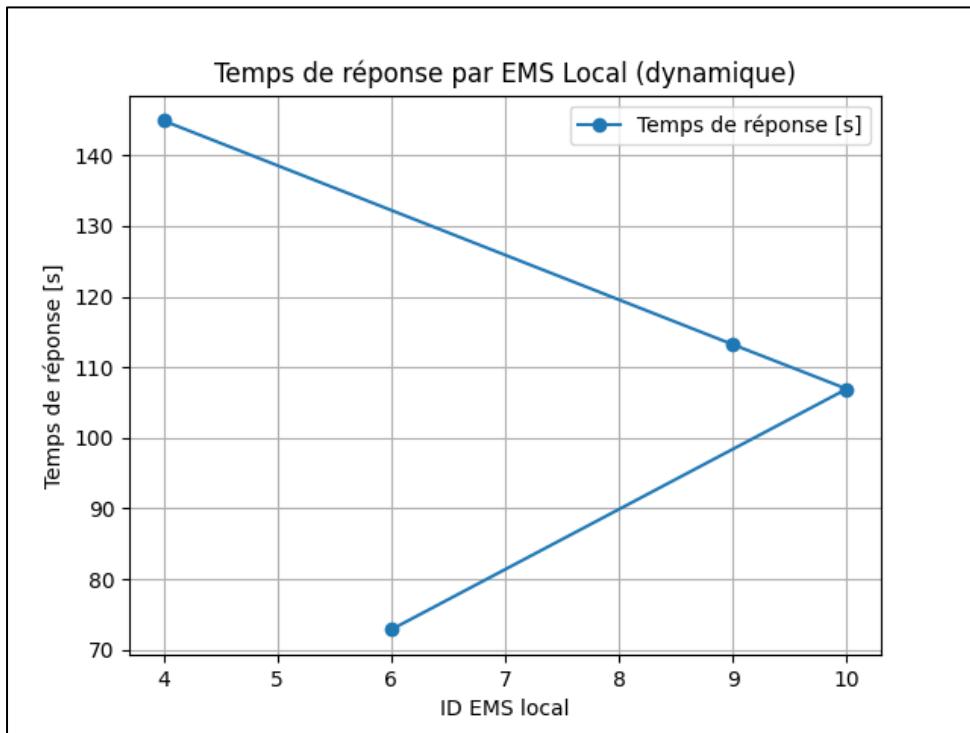


Figure 13 : Temps de réponse des différents EMS locaux, qui répondent aux requêtes de l'EMS central (source : auteur)

La Figure 13 montre les temps de réponse simulés des EMS locaux générés par l'algorithme. Le point le plus bas correspond à l'EMS local ayant répondu en premier (EMS local 6), tandis que le point le plus haut représente le dernier EMS ayant répondu, ici l'EMS local 4, avant que le surplus ne soit totalement alloué.

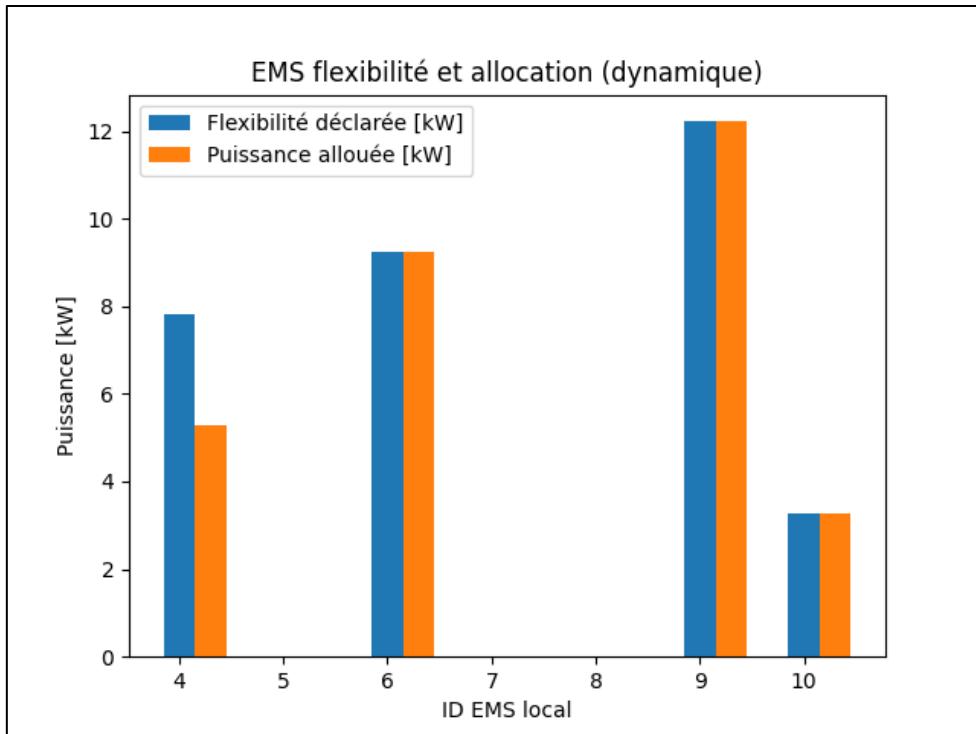


Figure 14 : Allocation du surplus en fonction des capacités des EMS locaux à activer leurs charges (source : auteur)

De même, la Figure 14 illustre l'allocation du surplus selon la stratégie « Premier arrivé, premier servi ». L'EMS local 6, ayant répondu en premier, reçoit le surplus en priorité, tandis que l'EMS local 4, ayant répondu en dernier, ne reçoit que le surplus restant, insuffisant pour satisfaire l'ensemble de sa consommation flexible disponible.

Finalement, l'algorithme affiche le message suivant :

- Surplus épuisé, annulation des tâches restantes.
- Temps total d'allocation : 144.917 secondes
- Surplus total alloué : 30.0 kW
- Pourcentage du surplus utilisé : 100.0 %

5.5.3. Cas 2 : Surplus restant (réutilisation partielle)

Si le surplus n'a pas été totalement alloué, l'algorithme répartit l'énergie en fonction des réponses des EMS actifs, capables d'activer des charges. Une fois que tous les EMS actifs ont répondu, il termine l'allocation et affiche les résultats suivants :

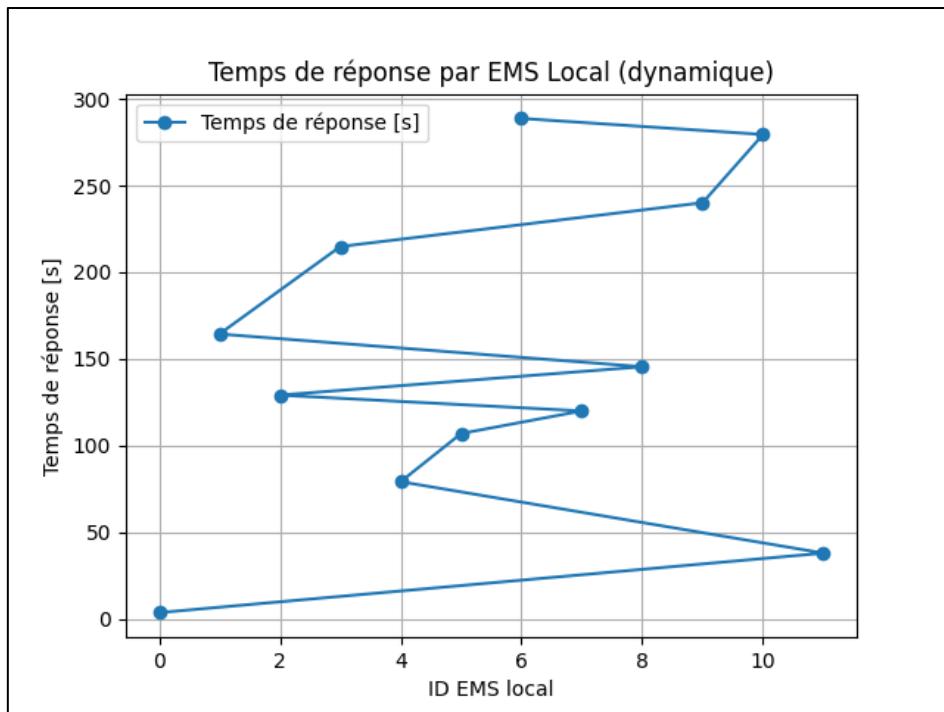


Figure 15 : Temps de réponse des différents EMS locaux, qui répondent aux requêtes de l'EMS central (source : auteur)

La Figure 15 montre le temps de réponse simulé de tous les EMS actifs. L'algorithme termine l'allocation dès que tous les EMS ont répondu aux requêtes.

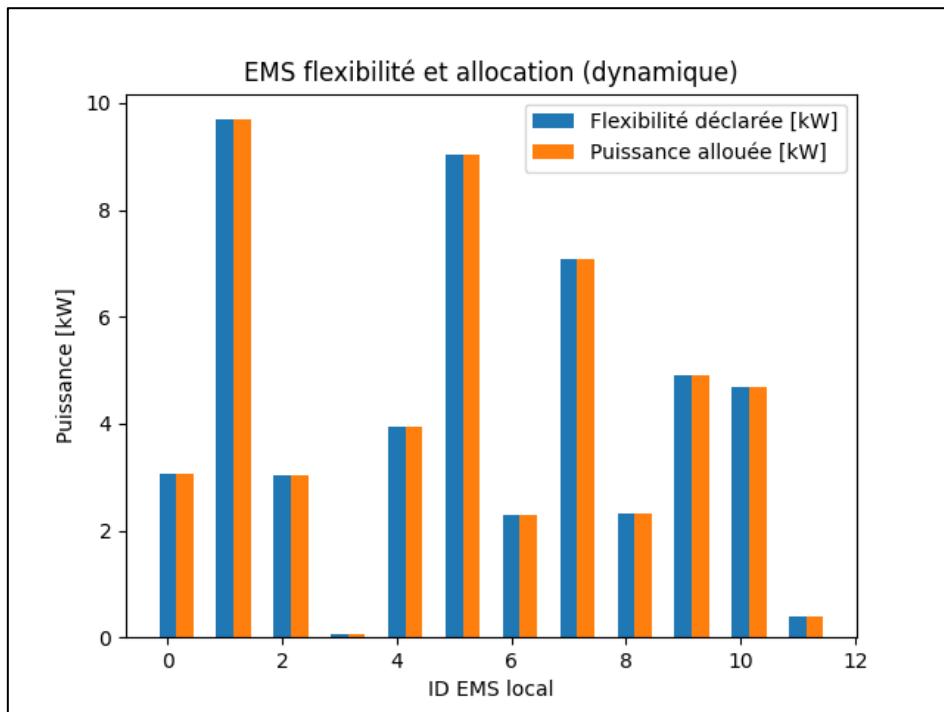


Figure 16 : Allocation du surplus en fonction des capacités des EMS locaux à activer leurs charges (source : auteur)

La Figure 16 montre que tous les EMS ont reçu la totalité de leur consommation flexible disponible. Le surplus est suffisant pour satisfaire l'ensemble des charges flexibles.

Le message affiché est le suivant :

- Temps total d'allocation : 288.710 secondes
- Surplus total alloué : 50.48 kW
- Pourcentage du surplus utilisé : 75.34 %

Ici, 75 % du surplus a pu être alloué. Les 25 % restants sont donc injectés dans le réseau.

F. Résultats et discussion

1. Simulation et analyse de la gestion du surplus d'une CE par l'EMS central pendant une journée d'été

Quarante cas ont été simulés avec un nombre aléatoire d'EMS locaux actifs (entre 5 et 30), des charges nettes comprises entre -5 kW (consommation maximale par consommateur) et 10 kW (production maximale par producteur), ainsi qu'une flexibilité disponible variant de 0 à 5 kW par membre n'étant pas en situation de surproduction. Ce cadre reflète bien une communauté d'énergie composée de foyers résidentiels. Les résultats ont été extraits et post-traités dans Excel afin de réaliser une analyse statistique. L'objectif est d'évaluer les relations entre le taux de valorisation du surplus, le nombre d'EMS actif, la valeur du surplus et son injection sur le réseau. Ici, nous faisons l'hypothèse qu'il s'agit d'une journée d'été, étant donné qu'un surplus est disponible dans tous les cas simulés.

1.1. Valorisation du surplus de la CE

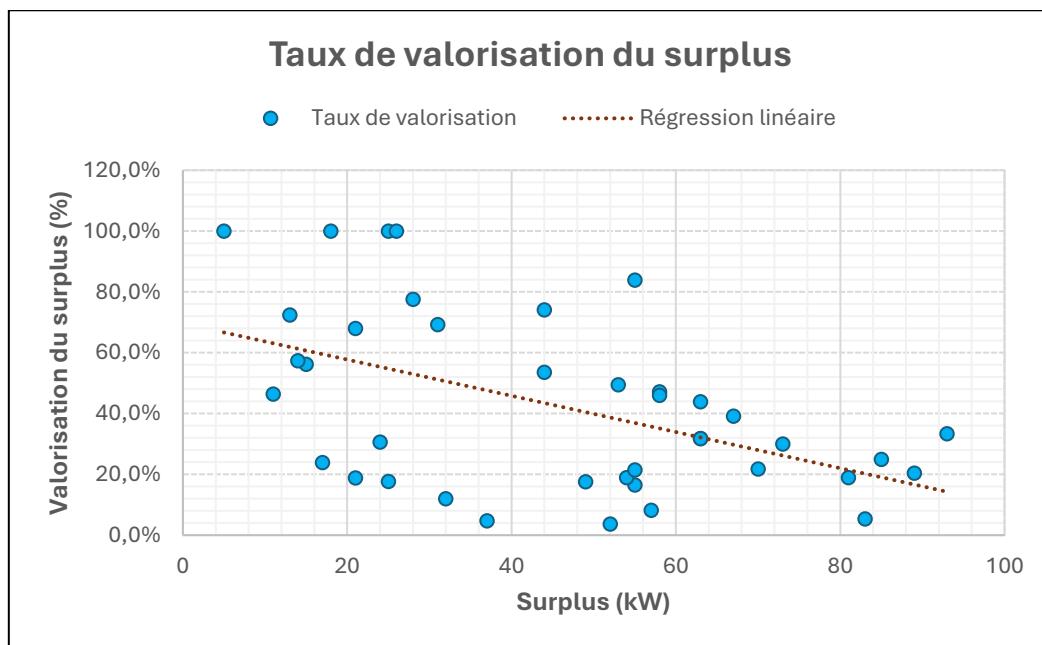


Figure 17 : Taux de valorisation en fonction du surplus disponible de la CE (40 simulations) (source : auteur)

La Figure 17 montre la variation du taux de valorisation en fonction du surplus disponible. Sans surprise, ce taux peut atteindre des niveaux élevés lorsque le surplus reste inférieur à 50 kW, car la quantité d'énergie à répartir est limitée. Entre 0 et 30 kW de surplus, certains cas atteignent une valorisation du surplus de 100 %.

De manière générale, on observe une tendance à la baisse de la valorisation lorsque le surplus augmente, comme l'indique la droite de régression linéaire descendante. Pour des valeurs de surplus proches de 100 kW, le taux reste limité à environ 20 %. Cela peut s'expliquer par le fait que, dans ces cas, une grande partie des EMS de la communauté sont eux-mêmes en situation de surproduction : ayant déjà optimisé leur consommation localement, ils ne disposent plus de charges flexibles à activer pour absorber le surplus, renforçant ainsi le déséquilibre.

1.2. Relation entre le nombre d'EMS et la valorisation du surplus

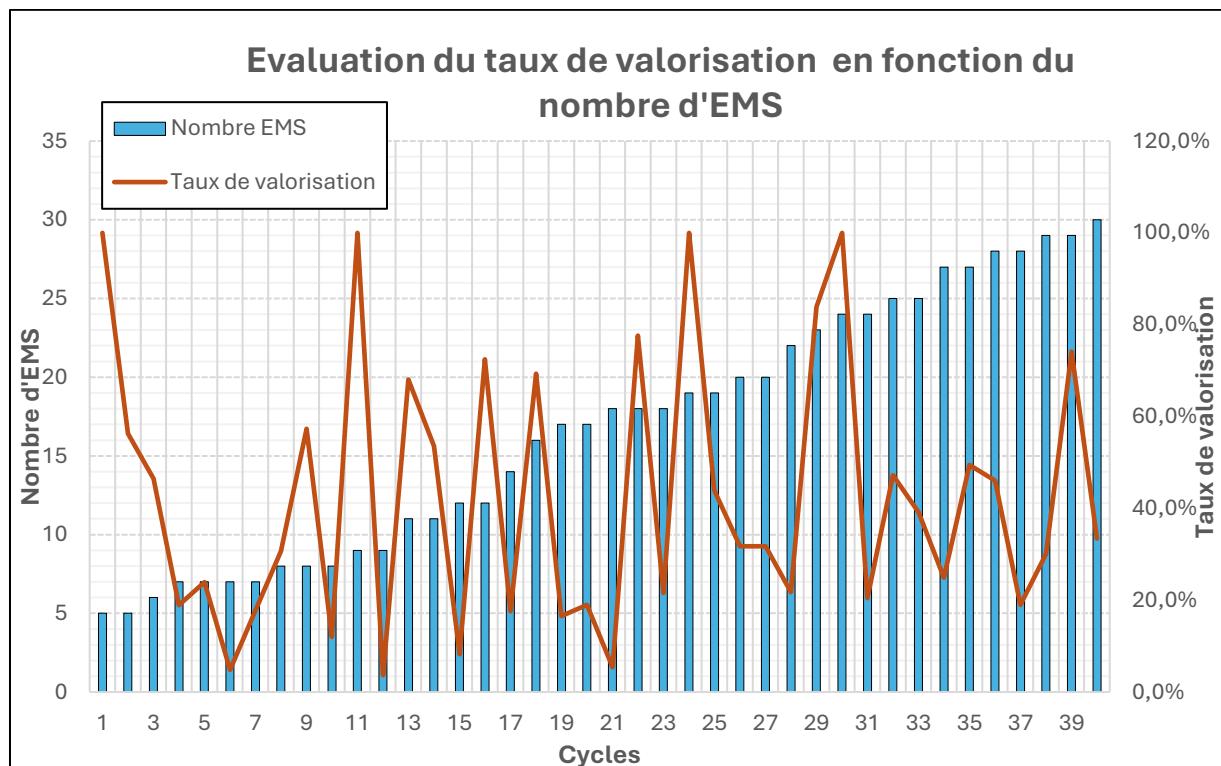


Figure 18 : Corrélations entre taux de valorisation et nombre d'EMS pour 40 simulations (source : auteur)

La Figure 18 présente le taux de valorisation en fonction du nombre d'EMS locaux pour les 40 cycles simulés. Le nombre d'EMS varie entre un minimum de 5 et un maximum de 30. On observe que, lorsque le nombre d'EMS est inférieur à 18

(jusqu'au cycle 21), le taux de valorisation peut descendre en dessous de 20 %. En revanche, à partir du cycle 23, où le nombre d'EMS augmente progressivement, les minima du taux restent systématiquement au-dessus de 20 %.

En effet, à partir de 18 EMS locaux (correspondant au cycle 23) les minima du taux restent au-dessus de 20 %. Cela suggère que plus le nombre d'EMS est élevé, plus le seuil minimal de la valorisation du surplus tend à augmenter, traduit par une meilleure capacité collective à absorber le surplus disponible. Une communauté plus large offre une meilleure capacité d'absorption du surplus ce qui augmente l'efficacité de l'algorithme.

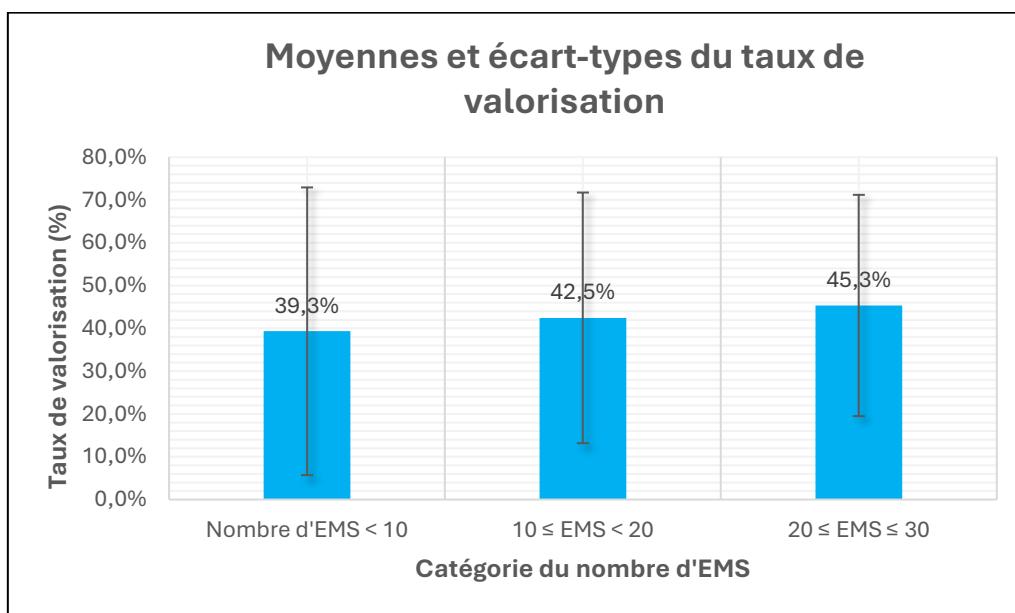


Figure 19 : Évolution du taux de valorisation moyen selon le nombre d'EMS actif (avec écart-type) (source : auteur)

Sur la Figure 19 le nombre d'EMS est classé en 3 catégories :

- Moins de 10 EMS
- Entre 10 et 20 EMS
- Entre 20 et 30 EMS

En moyenne, le taux d'utilisation observé est d'environ 40 %, toute classe confondue. En calculant la moyenne du taux pour chaque catégorie, on observe que la moyenne augmente avec le nombre d'EMS actifs. La catégorie contenant entre 20 et 30 EMS présente la moyenne la plus élevée, avec un taux d'utilisation supérieur à 45 %.

Néanmoins, les écart-types restent importants, ce qui reflète une forte variabilité entre les cas simulés, d'autant plus que l'analyse repose sur un échantillon de seulement 40 simulations (entre 10 à 15 cas pour chaque catégorie).

Pour les cas où le nombre d'EMS est inférieur à 10, le taux d'utilisation du surplus varie fortement, avec des valeurs allant de moins de 10 % à plus de 70 %. En revanche, lorsque le nombre d'EMS est compris entre 20 et 30, ce taux reste généralement supérieur à 20 %, avec des pics proches de 70 %.

Cette tendance confirme que l'augmentation du nombre d'EMS actif permet non seulement d'élever la moyenne du taux d'absorption du surplus, mais également de limiter les cas de sous-utilisation du surplus. Toutefois, le taux d'absorption dépend avant tout des profils de consommation énergétique des bâtiments et de la flexibilité des charges contrôlées. Ainsi, la simple augmentation du nombre d'EMS locaux actifs ne garantit pas nécessairement une meilleure utilisation du surplus sans prendre en compte ces facteurs.

1.3. Injection de la CE sur le réseau avec et sans EMS central

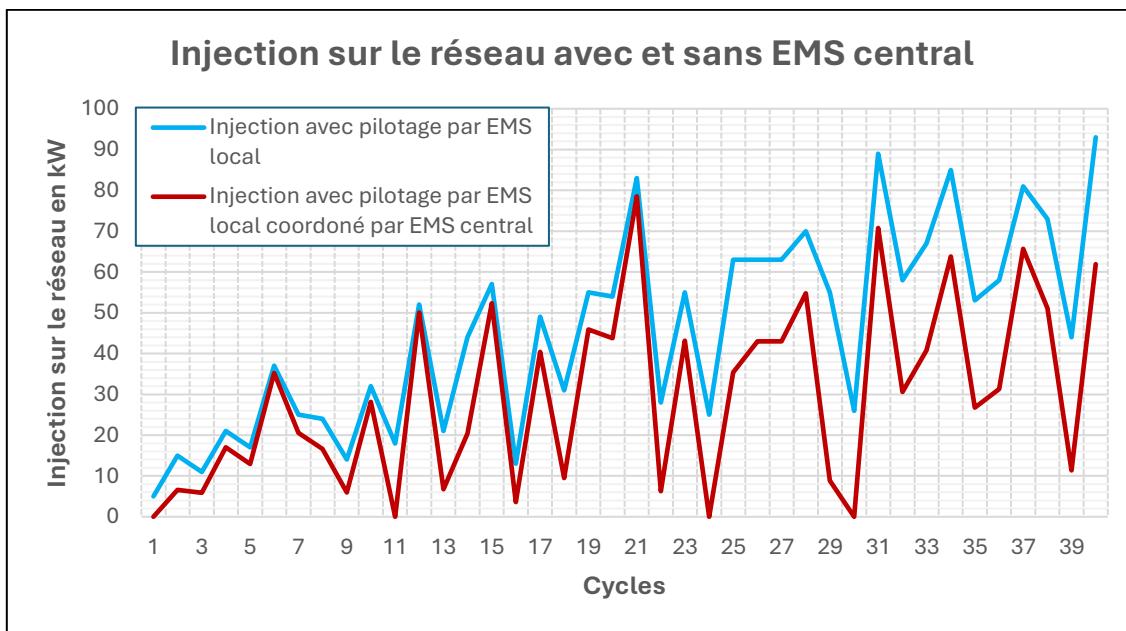


Figure 20 : Comparaison de l'injection de la CE sur le réseau avec et sans coordination par EMS central (source : auteur)

Sur la Figure 20, nous constatons que dans une situation sans coordination des EMS locaux par un agent central, l'intégralité du surplus (ligne bleue) généré par la

communauté est injectée sur le réseau. Les simulations sont numérotées dans l'ordre croissant du nombre d'EMS locaux. Ainsi, à partir de la simulation n°30 jusqu'à n°40, la communauté compte environ 20 à 30 EMS, ce qui correspond aux cas les plus étendus. On observe que pour les communautés les plus larges, des pics d'injection proches de 100 kW apparaissent.

En revanche, lorsque la communauté est pilotée par des EMS locaux coordonnés par un EMS central, on observe une réduction significative de l'injection sur le réseau. Une partie du surplus est alors répartie entre les EMS locaux selon leurs capacités de flexibilité. Ainsi, pour les communautés les plus larges, les pics d'injection sont contenus sous les 70 kW, contre les 90 kW sans coordination.

Cette comparaison illustre clairement l'intérêt d'un algorithme de coordination centralisée pour limiter les injections de puissance sur le réseau, contribuant ainsi à une meilleure autoconsommation collective au sein de la communauté.

1.4. Impact sur l'autoconsommation

Les résultats présentés représentent la distribution du surplus pendant une journée d'été, ainsi l'autoconsommation d'un prosumer se trouve plus proche des 20 % que des 40 %. Comme la production photovoltaïque est déjà optimisée localement par l'EMS domestique, nous pouvons estimer que le taux d'autoconsommation est augmenté d'au moins 10 %, en supposant que les membres disposent de plusieurs charges flexibles (CUBE CONCEPTS, 2025) ; (Cieślak, 2022); (Pater, 2023). Ainsi, l'autoconsommation individuelle d'un prosumer peut être estimée à 30 %.

Dans ce cas précis, les 70 % restant de la production PV en été des prosumers sont injectées sur le réseau. En intégrant l'EMS centralisé, capable de redistribuer ce surplus entre les membres de la communauté, environ 40 % de ce surplus peut être valorisé grâce à la coordination par l'allocation FIFO.

Ainsi, sur ces 70 % de surplus, 40 % sont valorisés par l'EMS central ce qui permet d'augmenter l'autoconsommation collective de la communauté de 28 % grâce à la redistribution du surplus. Ce partage s'explique par l'absorption du surplus par les consommateurs de la CE qui ne possèdent pas de panneaux PV ou qui ne produisent pas assez, mais dont la consommation est pilotée par la structure EMS central – EMS local.

L'autoconsommation collective de la CE est alors de 30 % (autoconsommation individuelle optimisée) + 28 % (redistribution du surplus) = 58 % pendant une journée d'été.

2. Discussion

L'échelle TRL (Technology Readiness Level) permet d'estimer la maturité d'une technologie, de l'observation de principes fondamentaux (TRL 1) à l'exploitation d'un système complet en conditions réelles (TRL 9), comme illustré sur la Figure 21.

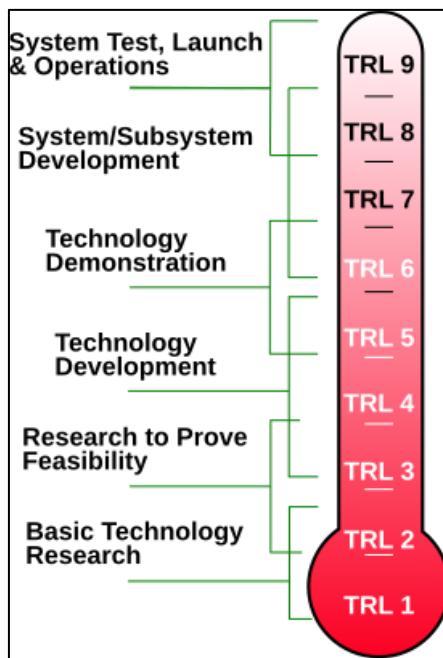


Figure 21 : Echelle TRL pour estimer le niveau de maturité (Mankins, 1995)

Le Tableau 8 ci-après décrit les différents niveaux TRL utilisés pour classer le degré de maturité d'une technologie.

Tableau 8 : Description générale de l'échelle TRL

TRL 1	Principes de base observés
TRL 2	Formulation du concept technologique
TRL 3	Preuve de concept expérimentale
TRL 4	Validation de la technologie en laboratoire
TRL 5	Validation de la technologie dans un environnement pertinent
TRL 6	Démonstration de la technologie dans un environnement pertinent
TRL 7	Démonstration du prototype du système en environnement opérationnel

TRL 8	Système complet et qualifié
TRL 9	Système réel prouvé en environnement opérationnel

Dans le cadre de ce travail, l'EMS central développé repose sur une preuve de concept implémentée et validée par simulation, sans expérimentation en laboratoire. Il peut donc être positionné au TRL 3, correspondant à une preuve de concept expérimentale.

Cela signifie que les principes ont été définis (TRL 2) et que leur faisabilité a été démontrée dans un environnement simulé, mais pas encore validé en conditions expérimentales réelles (TRL 4). Cette évaluation permet de situer le niveau d'avancement du développement et d'identifier les étapes futures nécessaires à une mise en œuvre réelle.

L'approche multi-agent retenue pour mettre en œuvre ce système a démontré son efficacité dans la gestion du surplus au sein d'une CE. Nous avons montré numériquement que le taux moyen de valorisation du surplus atteint environ 40 %, ce qui témoigne de la pertinence de la coordination centrale dans une architecture hiérarchique.

Cette décomposition en agents permet d'éviter une optimisation entièrement centralisée, souvent coûteuse en ressources de calcul. Grâce à cette architecture, la charge computationnelle de l'EMS central reste faible, ce qui permet des décisions rapides. En outre, cette structure hiérarchique et événementielle favorise une réactivité élevée, réduit les échanges inutiles et évite les temps d'attente liés à une planification globale.

L'un des principaux atouts de cette approche réside dans sa capacité à monter en échelle (Surmann, 2020). Étant donné que chaque EMS local prend ses décisions de manière autonome sur la base des consignes reçues, il est possible d'élargir la communauté sans modifier l'architecture globale du système. De nouveaux équipements flexibles (micro-cogénérations, appareils électriques pilotables, etc.) peuvent ainsi être intégrés progressivement, sans entraîner de surcharge computationnelle pour l'EMS central.

La stratégie d'allocation séquentielle adoptée (algorithme FIFO) garantit que le premier EMS local à répondre reçoit la totalité de la puissance qu'il peut absorber, ce qui permet de respecter les contraintes techniques de certains appareils. Ce choix évite des répartitions trop fragmentées qui rendraient l'activation des charges inefficace ou impossible.

Un autre aspect déterminant est la gestion temporelle. Le pas de temps a été fixé à 15 minutes, ce qui correspond au pas de temps généralement utilisé par les gestionnaires de réseau. Cela offre un bon compromis entre réactivité et stabilité. Ce fonctionnement cyclique (réévaluation toutes les 15 minutes) permet de suivre l'évolution de la production et de la consommation en temps réel en maintenant une coordination simple et robuste.

Les résultats montrent que le nombre d'agents (EMS locaux actifs) n'est pas le seul facteur déterminant de l'autoconsommation. Il faut aussi que ces EMS disposent de charges flexibles disponibles et activables.

Ainsi, le nombre d'EMS actif, bien qu'important, ne suffit pas à garantir une autoconsommation élevée. La disponibilité réelle d'activer des charges flexibles de chaque EMS est un facteur tout aussi déterminant. Il faut donc une bonne synchronisation entre surplus et flexibilité disponible.

On constate que dans certains scénarios favorables, le système parvient à une autoconsommation de 100 % du surplus disponible. Ce cas extrême apparaît typiquement lorsque le surplus de production est faible et que les charges flexibles disponibles sont bien alignées pour consommer ce surplus. Cela montre que l'algorithme permet, dans des conditions optimales, d'absorber la totalité de l'excédent de production. Autrement dit, aucune énergie n'est alors injectée sur le réseau externe, toute la production étant consommée à l'échelle de la communauté. Bien que cette situation ne soit pas généralisable, elle illustre l'efficacité maximale du système : lorsque les ressources le permettent, la redistribution par l'EMS central peut entièrement éviter l'injection sur le réseau et ainsi diminuer la dépendance au réseau public.

Toutefois, en moyenne, une partie du surplus continue d'être injectée sur le réseau (environ 60 % du surplus d'après les résultats des simulations, puisque 40 % sont

valorisés). Ce reste à injecter s'explique par les cas où la production excédentaire dépasse les capacités instantanées des charges flexibles agrégées. Néanmoins, même dans ces situations, la réduction des injections sur le réseau grâce à l'EMS central n'est pas négligeable. On observe que le système permet de réduire les pics d'injection. Ainsi, au lieu de renvoyer la production non utilisée au réseau, une partie est absorbée par les EMS locaux. Cette absorption est bénéfique pour la stabilité du réseau de distribution. Comme l'ont noté d'autres auteurs, utiliser localement l'énergie produite plutôt que de l'exporter réduit la sollicitation du réseau et les risques de congestion tout en valorisant l'énergie renouvelable produite (Denysiuk, 2020).

Nos résultats démontrent concrètement la possibilité de matérialiser une diminution des échanges avec le réseau et une meilleure résilience énergétique locale. Il convient de souligner que ces performances ont été obtenues en considérant exclusivement les flux physiques d'énergie. L'étude s'est focalisée sur l'aspect technique de la maximisation de la valorisation du surplus, sans intégrer les dimensions économiques. Autrement dit, le système a été évalué sur sa capacité à réduire les surplus injectés indépendamment des prix de l'électricité ou des mécanismes de facturation. Cette approche « physique » permet d'évaluer l'efficacité énergétique intrinsèque du concept, c'est-à-dire la maximisation de l'autoconsommation de la communauté, sans qu'il soit nécessaire de prendre en compte les flux financiers. Les résultats encourageants, avec une valorisation du surplus de 40% forment une base solide suggérant que le concept est viable pour réduire les échanges avec le réseau.

En résumé, cette architecture événementielle multi-agent démontre une forte robustesse, une capacité de montée en charge, et une bonne performance dans la valorisation du surplus, tout en limitant les injections sur le réseau. Elle constitue une approche particulièrement adaptée aux exigences du projet LOCOMOTRICE.

3. Limitations

Bien que la méthode suggérée ait prouvé son efficacité, plusieurs limitations ont été identifiées, indiquant des axes d'amélioration pour les recherches à venir. La première concerne le caractère « premier arrivé, premier servi » de l'algorithme, qui induit un biais en faveur de l'EMS local le plus réactif. En effet, dans la mise en

œuvre actuelle, le participant qui répond le plus rapidement à la requête du central est assuré de recevoir en priorité tout le surplus qu'il est capable d'absorber. Or, cette rapidité de réponse peut dépendre de facteurs techniques (latence de communication, rapidité de calcul, marque ou modèle). Ainsi, un EMS local disposant de faibles charges, mais répondant en quelques millisecondes pourrait être systématiquement servi avant un autre EMS potentiellement capable de consommer davantage de surplus, mais répondant avec un léger décalage. Cela pose un problème d'équité entre participants : le plus rapide profitera plus souvent du surplus, tandis que les plus lents pourraient être frustrés de ne rien recevoir ou que peu s'il reste de l'énergie après l'allocation aux premiers. Sur le plan collectif, cette dynamique pourrait même conduire à une utilisation sous-optimale du surplus dans certaines situations.

Plusieurs approches peuvent être envisagées pour atténuer ce biais de priorité temporelle. Une idée serait d'introduire un mécanisme de priorité tournante entre EMS locaux : au lieu d'un pur FIFO, on pourrait alterner l'ordre de service d'un cycle à l'autre ou réserver par moments la priorité à ceux qui ont été désavantagés aux cycles précédents.

Cette équité mérite d'être explorée, d'autant plus que dans un cadre communautaire réel, l'acceptation du système par les membres du partage d'énergie peut dépendre de la perception d'une justice dans l'accès au surplus.

La seconde limitation notable réside dans la variabilité élevée des performances observées selon les cas de figure, mise en évidence par des écart-types importants autour de la moyenne de 40 % de valorisation. En d'autres termes, si certaines simulations atteignent des taux très élevés (près de 100 %), d'autres restent largement en dessous de la moyenne, ce qui indique une hétérogénéité des résultats. Cette variabilité s'explique par la diversité des situations testées : profil de production renouvelable plus ou moins concentré et disponibilités des charges flexibles différentes, et donc une synchronisation plus ou moins favorable entre surplus et flexibilité.

Dans le cadre de cette étude, l'optimisation porte principalement sur la gestion des charges flexibles au sein des EMS locaux. Cependant, il peut arriver qu'un surplus d'énergie subsiste après cette allocation. Pour améliorer encore la valorisation de ce

surplus, l'intégration d'une batterie communautaire pourrait être envisagée. Cette solution permettrait de stocker l'énergie excédentaire, offrant ainsi une flexibilité supplémentaire et réduisant davantage l'injection vers le réseau.

De plus, il pourrait y avoir des marges d'améliorations en rendant la coordination plus intelligente. Par exemple, l'algorithme actuel ne tient pas compte de l'anticipation. Ainsi, chaque cycle traite le surplus instantané sans planifier sur un horizon plus long. Intégrer une certaine prédition via des prévisions de production/consommation ou l'état futur des charges pourrait améliorer l'allocation. Par exemple, si une charge importante devient disponible juste après la clôture d'un cycle, le surplus aura déjà été distribué et l'opportunité d'une activation plus pertinente sera manquée.

De même, l'ajout d'une priorité des types de charges pourrait aider. Par exemple, on pourrait décider de servir en priorité des charges dont la flexibilité est limitée dans le temps (par exemple, un VE qui est sur le point de s'en aller) par rapport à d'autres qui peuvent attendre.

Dans notre cas, l'absence de signal tarifaire dans la logique de contrôle signifie que chaque EMS local opère simplement en réponse à la disponibilité d'un surplus physique. Il sera donc instructif d'évaluer, dans des études futures, comment l'introduction de signaux de prix ou de mécanismes de facturation pourrait influencer les décisions des EMS locaux et la répartition du surplus, potentiellement pour encore améliorer la rentabilité du système et des communautés d'énergie. L'intégration des aspects économiques constituerait une évolution potentielle de l'algorithme.

La discussion des résultats reste incomplète tant que l'on n'a pas examiné la viabilité financière du système : répartition des gains d'autoconsommation et par conséquent la réduction de la facture électrique. Il faudra vérifier que maximiser l'autoconsommation s'aligne avec l'optimisation des coûts pour les utilisateurs, ce qui n'est pas toujours trivial si, par exemple, le tarif de l'électricité partagée est faible comparé à l'électricité prélevée. Des travaux futurs pourront explorer un couplage de notre algorithme avec un mécanisme de tarification interne à la communauté en s'appuyant par exemple sur des technologies comme la blockchain pour enregistrer les échanges. Certains travaux antérieurs ont d'ailleurs intégré ces considérations économiques, notamment à travers l'usage de signaux de prix dans les algorithmes

d'optimisation, comme le montrent les études de Díaz-Londoño et al. (2018), Karavas et al. (2017), Mehta et al. (2022), ou encore Denysiuk et al. (2020).

Le Tableau 9 ci-dessous récapitule les limites identifiées :

Tableau 9 : Tableau récapitulatif des limitations identifiées

Limitation	Description	Conséquence	Pistes d'amélioration
Biais temporel (FIFO)	Répartition selon l'ordre d'arrivée des réponses	Favorise les EMS les plus rapides, crée une injustice	Introduire une priorité tournante ou un équilibrage en prenant en compte les allocations passées
Variabilité des performances	Forte dispersion des résultats selon les cas testés	Résultats instables et difficilement généralisables	Mieux caractériser les cas, tester plus de cas
Absence d'anticipation	Réponse au surplus sans vision prospective	Manque d'optimalité si des charges pertinentes apparaissent juste après	Intégrer des prévisions de production/consommation et de disponibilité des charges
Pas de hiérarchisation des charges	Toutes les charges flexibles sont considérées égales	Moins bon couplage avec les contraintes temporelles des charges	Ajouter des priorités en fonction de la contrainte de temps associée à chaque charge
Pas de signal économique	Les EMS réagissent sans incitation financière	Décisions potentiellement incohérentes avec les coûts réels	Tester l'introduction de signaux de prix
Viabilité économique non évaluée	Répartition des gains pas encore analysée	Incertitude sur la rentabilité réelle pour la CE	Coupler le modèle à un système de tarification communautaire

4. Conclusion

En synthèse, la discussion met en avant que l'approche algorithmique multi-agent proposée réussit à augmenter significativement l'autoconsommation collective, validant ainsi le concept d'une coordination du surplus par un EMS central communiquant avec des EMS domestiques. Cette coordination relativement simple offre des bénéfices en réduisant les injections sur le réseau et en valorisant mieux l'énergie renouvelable locale, ce qui s'aligne avec les objectifs des communautés d'énergie.

Potentiellement, des améliorations sont envisageables pour perfectionner le système : traiter le biais de priorité induit par le « premier arrivé, premier servi », affiner la coordination avec une anticipation pour uniformiser les performances, et intégrer les aspects économiques. Ce travail ouvre donc la voie à une optimisation plus globale (économique, équitable) de la gestion collective de l'énergie, où la flexibilité locale des utilisateurs est mise à contribution de manière orchestrée pour maximiser l'autoconsommation et, par conséquent, la rentabilité des communautés. Les résultats obtenus sont encourageants. Ils montrent qu'avec une architecture hiérarchique EMS central – EMS locaux, il est possible d'atteindre en moyenne un niveau d'utilisation locale de l'énergie excédentaire de 40%. Le défi sera de faire évoluer cette solution en conservant sa simplicité et son caractère distribué, tout en y ajoutant les garanties d'équitabilité et d'incitation financière.

Bibliographie

- Amabile, L. B.-P. (2021). Optimizing the self-consumption of residential photovoltaic energy and quantification of the impact of production forecast uncertainties. *Advances in Applied Energy*, 2. doi:10.1016/j.adapen.2021.100020
- Anderson, J. (2024). *An Intro to Threading, Multiprocessing and Async IO in Python*. Récupéré sur Real Python: <https://realpython.com/python-concurrency/>
- Berk Celik, R. R. (2019). Electric energy management in residential areas through. *HAL open science*. Récupéré sur <https://hal.science/hal-02131022/document>
- Bianchini, G. C. (2019). Demand-response in building energy management: Model predictive control implementation. *12(9)*, 1710. doi:10.3390/en12091710
- Brudermueller, T. K. (2023). Large-scale monitoring of residential heat pump cycling using smart meter data. *Applied Energy*, 350, 121734. doi:10.1016/j.apenergy.2023.121734
- Byrd, C. (2023). *What are multi-agent workflows?* Consulté le avril 2025, sur Temporal Technologies: <https://temporal.io/blog/what-are-multi-agent-workflows>
- Carrascosa, C. P. (2024). Asynchronous consensus for multi-agent systems and its application to Federated Learning. *Engineering Applications of Artificial Intelligence*, 135. doi:10.1016/j.engappai.2024.108840
- Christie, S. H. (2022). Mandrake: multiagent systems as a basis for programming fault-tolerant decentralized applications. *Autonomous agents and multi-agent systems*, 36(1). doi:10.1007/s10458-021-09540-8
- Cieślak, K. J. (2022). Multivariant Analysis of Photovoltaic Performance with Consideration of Self-Consumption. *Energies*, 15(18), 6732. doi:10.3390/en15186732
- Crew AI. (s.d.). *Tasks*. Consulté le avril 2025, sur CrewAI Concepts: <https://docs.crewai.com/concepts/tasks>

CUBE CONCEPTS. (2025). *Augmenter l'autoconsommation photovoltaïque*. Consulté le mai 2025, sur CUBE CONCEPTS: <https://cubeconcepts.de/fr/augmenter-lautoconsommation-photovoltaïque>

CWaPE, C. w. (2025). *Évaluation du cadre relatif aux communautés d'énergie, au partage d'énergie et à l'autoconsommation*. Rapport CD-25b20-CWaPE-0109, Commission wallonne pour l'Énergie. Récupéré sur <https://www.cwape.be/sites/default/files/cwape-documents/2025.02.20-0109-Rapport%20%C3%A9valuation%20CE%20et%20partage.pdf>

De Masellis, R. G. (2020). Logic-based specification and verification of homogeneous dynamic multi-agent systems. *Autonomous Agent Multi-Agent Systsems*, 34(34). doi:10.1007/s10458-020-09457-8

Demars, P.-T. (2023). *Autoconsommation collective, principe et état des lieux en France en 2021*. Consulté le mai 2025, sur Encyclopédie de l'énergie: <https://www.encyclopedie-energie.org/autoconsommation-collective-principe-etat-lieux-france-2021>

Denysiuk, R. L. (2020). Multiagent system for community energy management. In *Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020)* (pp. 28–39). SCITEPRESS. doi:10.5220/0008914200280039

Díaz-Londoño, C. R. (2018). Smart charge of an electric vehicles station: A model predictive control approach. *IEEE Conference on Control Technology and Applications (CCTA)*, (pp. 54–59). doi:10.1109/CCTA.2018.8511498

Dong, A. (2022). *Algorithmes asynchrones pour la gestion décentralisée des réseaux électriques soumis aux aléas de communication*. Thèse de doctorat, École normale supérieure de Rennes. Récupéré sur <https://theses.hal.science/tel-03783740>

ECOinfos Energies Renouvelables. (2023). *Autoconsommation collective : principe et état des lieux en France*. Consulté le mai 2025, sur Les Energies Renouvelables: <https://www.les-energies-renouvelables.eu/conseils/autoconsommation/autoconsommation-collective-energie-panneau-solaire-photovoltaïque/>

Elshazly, H. E. (2021). Towards enabling I/O awareness in task-based programming models. *Future Generation Computer Systems*, 121, 74–89. doi:10.1016/j.future.2021.03.009

Énergie Plus. (2022). *Partages d'énergie*. Consulté le mai 2025, sur <https://energieplus-lesite.be/theories/consommation-energetique/partages-denergie/>

Ferber, J. (1997). Les systèmes multi-agents : un aperçu général. *Technique Et Science Informatiques - TSI*, 16, 79-1012. Consulté le mai 2025, sur https://www.researchgate.net/publication/242623967_Les_systemes_multi-agents_un_apercu_general

Fonseca, T. F. (2022). Flexible Loads Scheduling Algorithms for Renewable Energy Communities. *Energies*, 15(23), 8875. doi:10.3390/en15238875

Fuhrmann, F. S. (2022). Model-predictive energy management system for thermal batch production processes using online load prediction. *Computers & Chemical Engineering*, 163. doi:10.1016/j.compchemeng.2022.107830

GeeksforGeeks. (2025). *GeeksforGeeks*. Récupéré sur FCFS – First Come First Serve CPU Scheduling: <https://www.geeksforgeeks.org/first-come-first-serve-cpu-scheduling-non-preemptive/>

Hemmerdinger, T. (2021). *Institut Paris Région*. Récupéré sur Quand les citoyens se saisissent de la transition énergétique: <https://www.institutparisregion.fr/environnement/developpement-durable/quand-les-citoyens-se-saisissent-de-la-transition-energetique/>

International Energy Agency. (2024a). *Flexible heat pumps – Task 4 report: Integration of flexible heat pumps in energy systems*. Consulté le mai 2025, sur Heat Pumping Technologies (IEA HPT) Annex 57: <https://heatpumpingtechnologies.org/annex57/wp-content/uploads/sites/69/2024/03/task-4-reportfinal-2.pdf>

International Energy Agency. (2024b). *More efficient and flexible buildings are key to clean energy transitions*. Consulté le mai 2025, sur

<https://www.iea.org/commentaries/more-efficient-and-flexible-buildings-are-key-to-clean-energy-transitions>

Johnson, R. (2023). *DataRodeo*. Consulté le avril 2025, sur Optimizing Python Workflows with asyncio for Asynchronous Programming: <https://www.datarodeo.io/python/optimizing-python-workflows-with-asyncio-for-asynchronous-programming/>

Jung, H. K. (2025). Two-stage dynamic real-time optimization framework using parameter-dependent differential dynamic programming. *Computers & Chemical Engineering*, 192. doi:10.1016/j.compchemeng.2024.108896

Karavas, C.-S. A. (2017). A Game Theory Approach to Multi-Agent Decentralized Energy Management of Autonomous Polygeneration Microgrids. *Energies*, 10(11). doi:10.3390/en10111756

Kiki, G. (2024). *EMS local – Développement du modèle et de l'algorithme*. Rapport scientifique, Université de Liège.

Lee, M. (2020). *Multi-threading vs Event Loop in Python*. Consulté le avril 2025, sur DEV Community: <https://dev.to/mervynlee94/multi-threading-vs-event-loop-in-python-1h4h>

Li, S. L. (2022). Model controlled prediction: A reciprocal alternative of model predictive control. *IEEE/CAA Journal of Automatica Sinica*, 9(6), 1107–1110. doi:10.1109/JAS.2022.105611

LogicMachine. (s.d.). *Shelly IoT device integration via MQTT*. Consulté le juin 2025, sur <https://kb.logicmachine.net/integration/shelly/>

Mankins, J. C. (1995). Technology readiness levels. *NASA Advanced Concepts Office*.

Mehta, P. T. (2022). Solar PV sharing in urban energy communities: Impact of community configurations on profitability, autonomy and the electric grid. *Sustainable Cities and Society*, 87. doi:10.1016/j.scs.2022.104178

Mischos, S. D. (2023). Intelligent energy management systems: a review. *Artificial Intelligence Review*, 56, 11635–11674. doi:10.1007/s10462-023-10441-3

PADE. (2019). *PADE design aspects*. Consulté le avril 2025, sur <https://pade.readthedocs.io/en/latest/user/desenvolvedores.html>

Palanca, J. (2020). *SPADE*. Consulté le avril 2025, sur SPADE-MAS: <https://spade-mas.readthedocs.io/en/latest/readme.html>

Pater, S. (2023). Increasing Energy Self-Consumption in Residential Photovoltaic Systems with Heat Pumps in Poland. *Energies*, 16(10), 4003. doi:10.3390/en16104003

Piron, M. (2024). *Modélisation et optimisation de la consommation énergétique d'un logement type en Wallonie à l'aide d'un EMS*. Mémoire de Master, Université de Liège. Récupéré sur https://matheo.uliege.be/bitstream/2268.2/21163/4/PIRON_Maxence_Memoire_2024

Python Software Foundation. (s.d.). *asyncio — Asynchronous I/O*. Consulté le avril 2025, sur Documentation Python: <https://docs.python.org/3/library/asyncio.html>

Remigio, J. E. (2020). Production scheduling in dynamic real-time optimization with closed-loop prediction. *Journal of Process Control*, 89, 95–107. doi:10.1016/j.jprocont.2020.03.009

Saeed, M. E. (2023). Energy management system in smart buildings based coalition game theory with fog platform and smart meter infrastructure. *Scientific Reports*, 13(2023). doi:10.1038/s41598-023-29209-4

Scarcello, L. (2022). Cascade computing model to optimize energy exchanges in prosumer communities. *Helion*, 8(2). doi:10.1016/j.heliyon.2022.e08902

Service public de Wallonie. (2024). *Régime d'octroi de certificats verts kECO*. Consulté le mai 2025, sur Énergie Wallonie: <https://energie.wallonie.be/fr/regime-d-octroi-de-certificats-verts-keco.html?IDC=10276&IDD=135639>

SPW. (2023). *Photovoltaïque : fini la compensation, vive l'autoconsommation*. Consulté le mai 2025, sur Energie Wallonie: <https://energie.wallonie.be/fr/20-10-2023-photovoltaïque-fini-la-compensation-vive-l-autoconsommation.html?IDD=172713&IDC=8187>

Surmann, A. R. (2020). Agent-based bidirectional charging algorithms for battery electric vehicles in renewable energy communities. *Proceedings of the 9th DACH+ Conference on Energy Informatics*, 3. doi:10.1186/s42162-020-00122-8

Testachats. (2024). *Comment augmenter l'autoconsommation avec vos panneaux solaires ?* Consulté le mai 5, 2025, sur <https://www.test-achats.be/maison-energie/energie-renouvelable/dossier/augmenter-autoconsommation-production-photovoltaïque>

Tuomela, S. d. (2021). Impacts of home energy management systems on electricity consumption. *Applied Energy*, 299, 117310. doi:10.1016/j.apenergy.2021.117310

Wallonie. (2025). *Photovoltaïque : s'informer sur le tarif prosumer.* Consulté le mai 2025, sur Le site officiel de la Wallonie: <https://www.wallonie.be/fr/demarches/photovoltaïque-sinformer-sur-le-tarif-prosumer>

Wooldridge, M. &. (1995). Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2), 115–152. doi:10.1017/S0269888900008122

Xu, J. (2024). Efficiency-Oriented Model Predictive Control: A Novel MPC Strategy to Optimize the Global Process Performance. *Sensors*, 24(17), 5732. doi:10.3390/s24175732

Zeng, P. X. (2024). Demand Response Strategy Based on the Multi-Agent System and Multiple-Load Participation. *Sustainability*, 16(2), 902. doi:10.3390/su16020902

Zhang, Y. C. (2025). A system-centred predictive maintenance re-optimization method based on multi-agent deep reinforcement learning. *Expert Systems with Applications*, 274, 127034. doi:10.1016/j.eswa.2025.127034

Outils IA

- ChatGPT : Reformulation de certaines phrases et outil d'assistance au codage Python.
- Rabbit AI : Recherche d'articles scientifiques.
- Grok : Synthèse d'articles scientifiques et recherche avancée.

Annexe : Script de l'algorithme de l'EMS central avec commentaires

```
#importer les modules nécessaires
import asyncio
import random
import matplotlib.pyplot as plt
import datetime
import time

# Table de paramètres de simulation
MAX_DELAY      = 10      # délai max de réponse (s)
min_delay      = 1       # délai min de réponse (s)
MIN_FLEX        = 0       # puissance flexible min (kW)
MAX_FLEX        = 5       # puissance flexible max (kW)
MIN_ACTIVE_EMS = 5       # EMS min par cycle
MAX_ACTIVE_EMS = 30      # EMS max par cycle
MIN_NET_LOAD   = -5      # charge nette min (kW)
MAX_NET_LOAD   = 10      # charge nette max (kW)

# Événement et liste partagée pour les nouvelles données
new_data_event = asyncio.Event()
latest_loads  = [] #C'est une liste globale

#La coroutine productrice (producer_de_net_loads) écrit dans la liste à
chaque cycle de 15 min
#La boucle principale (main) lit cette même liste (loads = latest_loads)
dès qu'on réveille l'Event.

async def producer_de_net_loads(interval: int = 900):
    #Produit de nouvelles charges nettes toutes les 15 minutes, notifie la
    boucle principale via `new_data_event`.
    global latest_loads
    while True:
        n = random.randint(MIN_ACTIVE_EMS, MAX_ACTIVE_EMS)
        latest_loads = [random.randint(MIN_NET_LOAD, MAX_NET_LOAD) for _ in
range(n)]
```

```

        new_data_event.set()
        await asyncio.sleep(interval)

# Simulation asynchrone des réponses des EMS locaux avec délai et
flexibilité

async def ems_local_response(query_id, net_load, max_delay=MAX_DELAY):
    #Simule la réponse d'un EMS local :
    #Si net_load > 0 (besoin local), pas de flexibilité (0)
    #Sinon, propose une flexibilité aléatoire entre MIN_FLEX et MAX_FLEX

    delay = random.uniform(min_delay, max_delay)
    await asyncio.sleep(delay)
    if net_load > 0:
        flexible_power = 0.0
    else:
        flexible_power = round(random.uniform(MIN_FLEX, MAX_FLEX), 2)
    return delay, query_id, {'EMS_id': query_id, 'puissance_flexible':
flexible_power}

# Classe RTOCordinatorDynamic
# Ce coordinateur agit comme l'agent manager qui supervise l'allocation de
la
# puissance flexible pour réduire le surplus.

class RTOCordinatorDynamic:
    def __init__(self):
        self.response_times = []
        self.flexibilities = []
        self.allocations = []
        self.remaining_surplus = 0
        self.initial_surplus = 0

    # Réception des charges nettes et calcul du surplus.
    # Le surplus est défini comme la somme des charges nettes.

    def receive_data(self, net_loads):
        self.net_loads = net_loads
        self.remaining_surplus = sum(net_loads)

```

```

        self.initial_surplus    = self.remaining_surplus
        # Remise à zéro de l'historique
        self.response_times.clear()
        self.flexibilities.clear()
        self.allocations.clear()
        print(f"[RTO] Surplus calculé: {self.remaining_surplus} kW | Charge
nette: {net_loads}")

        # Attribution dynamique des tâches aux EMS locaux :
        # - Vérifie si un surplus existe.
        # - Sélectionne le nombre d'EMS locaux actifs (IDs de 0 à
num_active_emss).
        # - Envoie des requêtes asynchrones pour obtenir la puissance flexible.
        # - Traite les réponses dès leur arrivée
        # - Alloue la puissance flexible jusqu'à épuisement du surplus.
        # - Annule les tâches restantes si le surplus est épuisé.

    async def query_and_allocate_parallel(self, num_active_emss):
        print("[RTO] Lancement dynamique des requêtes...")
        if self.remaining_surplus <= 0:
            print("[RTO] Aucun surplus disponible, aucune requête lancée.")
            return

        # Création des tâches en passant la charge nette correspondante
        pour chaque EMS actif
        tasks = [
            asyncio.create_task(emss_local_response(i, self.net_loads[i]))
            pour i in range(num_active_emss)
        ]
        start = time.perf_counter()

        # Traitement des réponses au fur et à mesure qu'elles arrivent
        for future in asyncio.as_completed(tasks):
            try:
                delay, qid, resp = await future
            except Exception as e:
                print(f"[EMS] Erreur tâche : {e}")

```

```

        continue

    # Enregistre le délai de réponse des agents locaux
    self.response_times.append((qid, delay))

    if resp and self.remaining_surplus > 0:
        flex      = resp['puissance_flexible']
        allocated = min(flex, self.remaining_surplus) # Allocation
        de la valeur minimale entre la flexibilité et le surplus
        self.flexibilities.append((qid, flex))
        self.allocations.append((qid, allocated))
        self.remaining_surplus -= allocated # Calcul du surplus
        restant après allocation
        print(f"[EMS {qid}] délai={delay:.2f}s | Flex={flex} kW |
        Alloué={allocated} kW")
    elif not resp: # Gestion des cas où l'agent local n'a pas
        répondu
        print(f"[EMS {qid}] pas de réponse")
    else: # Gestion des cas où le surplus est épuisé
        print(f"[EMS {qid}] surplus épuisé, pas d'allocation")

        if self.remaining_surplus <= 0: # Si surplus épuisé ->
            annulation des tâches restantes
            print("[RTO] Surplus épuisé, annulation des tâches
            restantes.")
            for t in tasks:
                if not t.done(): t.cancel()
            break

        elapsed = time.perf_counter() - start
        print(f"[RTO] Temps total d'allocation: {elapsed:.3f}s")

    # Visualisation des résultats :
    # - Graphique en barres pour comparer la flexibilité déclarée et la
    puissance allouée.
    # - Graphique linéaire pour visualiser les temps de réponse.
    # - Affiche le surplus total alloué et le pourcentage de surplus

```

utilisé.

```
def plot_results(self):  
    # Extraction des données pour les graphiques  
    ems_ids, flex_vals = zip(*self.flexibilities) if self.flexibilities  
else ([], [])  
    _, alloc_vals      = zip(*self.allocations)      if self.allocations  
else ([], [])  
    ids_resp, resp_v = zip(*self.response_times) if self.response_times  
else ([], [])  
  
    timestamp = datetime.datetime.now().strftime("%Y%m%d_%H%M%S")  
  
    # Graphique en barres flexibilité vs allocation  
    fig1, ax1 = plt.subplots()  
    ax1.bar(ems_ids, flex_vals, width=0.3, label='Flex [kW]')  
    ax1.bar([i+0.3 for i in ems_ids], alloc_vals, width=0.3,  
label='Alloc [kW]')  
    ax1.set_title('Flexibilité vs Allocation')  
    ax1.legend()  
    fig1.savefig(f"allocation_barres_{timestamp}.png")  
    plt.close(fig1)  
  
    # Graphique linéaire du temps de réponse  
    fig2, ax2 = plt.subplots()  
    ax2.plot(ids_resp, resp_v, marker='o')  
    ax2.set_title('Temps de réponse EMS')  
    fig2.savefig(f"allocation_temps_{timestamp}.png")  
    plt.close(fig2)  
  
    # Calcul de l'utilisation totale du surplus  
    total = sum(v for _, v in self.allocations)  
    pct   = round(total/self.initial_surplus*100, 2) if  
self.initial_surplus>0 else 0  
    print(f"[RTO] Surplus total alloué: {total} kW ({pct} %) - fichiers  
générés à {timestamp}")
```

```

# Fonction principale du script
# Simule le flux complet du système MAS :
async def main():
    rto = RTOCoordinatorDynamic()
    # Démarré le producteur de net_loads en tâche de fond
    producer = asyncio.create_task(producer_de_net_loads(900))

    try:
        while True:
            try:
                await new_data_event.wait()
                new_data_event.clear()
            except asyncio.CancelledError:
                break

            loads = latest_loads
            n = len(loads)
            rto.receive_data(loads)
            await rto.query_and_allocate_parallel(n)
            rto.plot_results()

    except KeyboardInterrupt:
        print("[RTO] Arrêt demandé, fin du service.")
    finally:
        producer.cancel()

# Point d'entrée du script et démarrage de la boucle
if __name__ == "__main__":
    try:
        asyncio.run(main())
    except KeyboardInterrupt:
        print("[RTO] Script interrompu.")

```