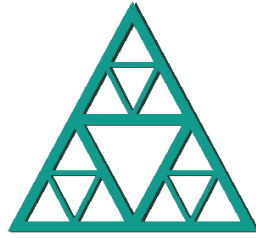


ECOLE NATIONALE DES PONTS ET CHAUSSEES



PROJET D'INITIATION A LA RECHERCHE 2021

Gestion de Microgrid

Thomas Béligon, Nouhayla Boubekri, Fanny Chaltiel, Charles-Auguste
Gourio, Simon Meunier, Alexandre Rigaud, Mathieu Roques, Michel Sénégas
sous la direction de
Olivier Beaude, Paulin Jacquot, Olivier Juan
EDF – R&D

ABSTRACT : Microgrids are small and intelligent network based on a local production and management of electricity. By creating a small grid of producers and consumers working together, one has the possibility to operate apart from the centralized network. That lead to a potential desynchronization and promote auto-consumption. The main idea of this project is to test the efficacy of such a Microgrid on a programmable simulation. Several actors such as an electrical farm or a data center are going to interact through the control of a microgrid manager. An important point is the conception of an efficient coordination algorithm that can stimulate specific reaction in order to have a better energetics result. For instance, an algorithm that try to maximize auto-consumption will probably send lower sale-prices for a solar farm if it transmits its produced power to a local neighbour. But the microgrid manager is not the only one to optimize as each actor can modify its consumption (in term of the power demand to the global network) by using a battery or choosing to sell heat or not. Another significant issue is the retrospective analysis of microgrid's global performance. That will lead us to considerate the monetarization of data outputs such as the global carbon emission or the cost of junction between the local and the centralized classic system.

KEYWORDS :

- Microgrid
- Auto-consumption
- Local actors
- Simulation
- Coordination Algorithm

Table des matières

1	Acteurs	7
1.1	Hypothèses générales pour les programmes d'optimisation	7
1.2	Charging station	7
1.3	Solar farm	8
1.4	Industrial consumer	8
1.5	Data center	8
1.6	Analyse des résultats	9
2	Données d'entrée-Métriques de sortie	12
2.1	Données d'entrée	12
2.1.1	Le clustering : méthode des K-Means	12

2.1.2	Interprétation	13
2.1.3	Scénarios	14
2.2	Métriques de sortie	16
2.2.1	Autonomie	16
2.2.2	Autonomie globale du Microgrid	16
2.2.3	Variantes de l'autonomie d'un Microgrid	17
2.3	Facture	18
2.3.1	Facture individuelle de la Solar Farm	18
2.3.2	Facture individuelle de la Charging Station	18
2.3.3	Facture individuelle du Data center	19
2.3.4	Facture individuelle du Industrial Site	19
2.3.5	Facture globale du Microgrid	19
2.3.6	Relativisation de la facture du Microgrid	19
2.4	Dimensionnement puissance : Raccordement	19
2.4.1	Solution de raccordement pour un Microgrid	20
2.4.2	Solution de raccordement relativisée pour un Microgrid	20
2.5	Emission C02	20
2.6	Comparer la performance des Microgrids	21
2.6.1	Monétarisation	21
2.6.2	Pondération des critères	22
2.7	Gain des acteurs en Microgrid	23
3	Coordinateur du microgrid	24
3.1	Rôle du coordinateur de la smart grid	24
3.2	Modèles de tarification initiaux	24
3.2.1	Prix affines	24
3.2.2	Variante proche de l'algorithme de départ	25
3.3	Améliorations l'algorithme initial	25
3.3.1	Minimisation du soutirage réseau $l(t)^+$	25
3.3.2	Minimisation des pics de consommation	26
3.3.3	Variante de boucle	26
3.3.4	Minimisation d'indicateurs	27
3.3.5	Utilisation de prix différents pour chaque acteur	28
3.3.6	Méthode de Newton	28
3.4	Condition d'arrêt de la boucle et nombre d'itérations	28
3.4.1	Nombre d'itérations fixé à l'avance	28
3.4.2	Arrêt dès qu'un indicateur à minimiser descend en dessous d'un seuil	28
4	Développement informatique des règles de coordination des acteurs	28
4.1	Fonctionnement informatique général du microgrid manager	28
4.2	Lancement des runs	29
4.3	Coordination des prix	29
4.4	Gestion des scénarios	31

4.5	Résultats observés	32
4.6	Pistes pour approfondir	32
5	Annexe : fiches des acteurs	34

Introduction

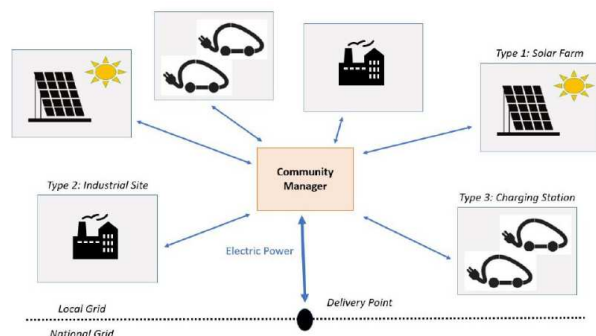
Les Microgrids (ou Smartgrids) sont des petits réseaux de production et de gestion d'électricité qui agissent à l'échelle locale. Leur importance ne cesse de grandir et cela pour deux raisons importantes :

- D'abord les microgrids s'appuient sur une production locale et souvent écologique d'électricité. À l'inverse des systèmes de production classiques et centraux qui génèrent de l'énergie dans des centrales pour la transporter sur de longues distances occasionnant des pertes (qui vont de 2 à 2.2% d'après RTE), une smartgrid va s'appuyer sur une production localisée (dans une zone dont la taille est de l'ordre d'une dizaine de km). De plus, les moyens de production utilisés sont le plus souvent écologiques (éoliennes, panneaux photo-voltaïques, ...)
- Ce type de micro-réseau est (partiellement) indépendant. En utilisant, en plus des moyens de production, des méthodes de stockage d'énergie (batteries, installations hydro-électriques), on peut alors essayer de répondre à des problématiques d'autoconsommation et gagner en indépendance sur le réseau central. Cette dernière n'est pas négligeable, surtout en cas de coupures massives ou de blackout massifs comme celui ayant eu lieu en avril 1988 au Canada (perte de synchronisme et effondrement du système électrique au Québec)

Dans le monde, la technologie de Smartgrid se développe petit à petit. Ainsi le gestionnaire de réseau californiens Pacific Gas & Electric a lancé en 2019 un ambitieux projet de construction de 20 microgrids. En France le développement est plus lent. On notera cependant le projet d'Engie Ineo qui vise à établir une dizaine de ces petits réseaux en métropole mais également dans des zone iso-

lées comme la Polynésie ou certains pays du territoire africain.

Notre projet vise à étudier le fonctionnement d'un microgrid à travers une simulation informatique. Pour cela nous allons imaginer un réseau fictif et non localisé (on aura la possibilité de le tester dans plusieurs régions de France). Un fonctionnement efficace nécessite en principe trois ingrédients principaux : un système de production indépendant, un système de stockage et enfin un outils de coordination intelligent qui peut récupérer les données de consommation et élaborer une grille de prix. Pour les acteurs, on considérera quatre consommateurs/producteurs différents. Ces derniers interagissent avec le manager du microgrid qui utilise un algorithme de coordination.



Un point essentiel est l'analyse des performances du micro réseau. Autrement, dis comment évaluer efficacement les résultats d'une simulation. Pour cela, il faut élaborer précisément la "facture" du microgrid et essayer de définir des critères relatifs à l'autonomie, au raccordement avec le réseau global mais également relatifs aux émissions de CO_2 .

Enfin, l'élaboration d'un microgrid efficace

s'est fait en collaboration avec 30 élèves qui ont put dans le cadre du cours "Optimisation et Énergie" proposé à l'École des Ponts nous fournir des résultats à la fois nombreux et intéressants ce qui a permis de tester nos différents algorithmes mais surtout les méthodes d'analyses post-simulations.

1 Acteurs

1.1 Hypothèses générales pour les programmes d’optimisation

Le travail de notre binôme consistait en la meilleure optimisation possible de chaque acteur, en fonction d’une grille de prix donnée à l’avance, et indépendamment des choix des autres acteurs. Nous avons, pour ce faire, commencé par formuler un certain nombre d’hypothèses, résumées ci-dessous :

On considère que tous les facteurs de la journée sont entièrement anticipés. Que ce soit au niveau de la demande du site industriel consumer ou du data center ou de la production solaire de la solar farm, ou encore des horaires de départ et d’arrivée des différents véhicules de la charging station. De plus, l’ensemble des prix pour chaque plage horaire est connu à l’avance.

Chacun des acteurs peut, à tout instant t , acheter de l’électricité au prix $price[price_purchase][t]$ ou vendre de l’électricité au prix $price[price_sale][t]$ et ce dans une quantité qui n’est limitée que par leur production ou la capacité de leur batterie.

Il est supposé qu’à tout instant t $price[price_purchase][t] \geq price[price_sale][t]$. Cela ne paraît pas être une hypothèse trop irréaliste. En effet, dans le cas contraire, tous les acteurs auraient intérêt à acheter et vendre de l’électricité sur le même créneau pour faire des bénéfices.

Les acteurs sont ici optimisés séparément et n’ont aucune interaction les uns avec les autres. Ces dernières se font au niveau de plusieurs tours de microgrid avec une modification à chaque tour de la grille tarifaire des prix qui dans l’idéal devrait à terme auto-réguler la consommation du microgrid.

1.2 Charging station

Pour la charging station, nous avons conservé les hypothèses de la fiche en annexe. A noter que pour $i \in 0, 1$ les voitures sont à charge lente et que pour $i \in 2, 3$ les voitures sont à charge rapide.

Les flux $l_i[t]$ sont les flux entrant ou sortant de la batterie i sans prise en compte du rendement. Concernant la modélisation, nous avons pris 2 variables pour modéliser chaque flux $l_i[t]$, sa valeur positive $prodvarsplus[t][i]$ et sa valeur négative $prodvarsmoins[t][i]$, qui correspondent respectivement à la partie positive du flux entrant dans la voiture i à l’instant t , et à la partie négative de celui-ci. La nécessité d’une différenciation des parties positive et négative vient de la prise en compte du rendement pour la batterie : la perte d’énergie engendrée par un flux entrant dans la batterie n’est pas la même que pour celle d’un flux sortant.

Nous avons également implémenté une variable booléenne *amende*[*i*], laquelle vaut 1 si la voiture *i* paye l’amende de la journée, et 0 sinon (pour rappel, l’amende a lieu dès qu’une voiture quitte la station avec moins de 25% de sa batterie remplie).

Enfin nous avons une dernière variable *sortieplus*[*t*] qui est la partie positive du $l_{tot}[t]$ (l_{tot} est la somme des l_i de chaque voiture). Cette dernière est utile pour le calcul de la fonction objectif, pour distinguer les temps où de l’électricité est achetée de ceux où elle est vendue, les prix à l’achat et la revente n’ayant aucune raison d’être identiques à chaque instant.

1.3 Solar farm

Pour la Solar farm, nous avons légèrement modifié les hypothèses de la fiche annexe.

Le tableau *sun* de la fonction *player* représente non pas l’énergie totale captée par les panneaux mais l’énergie que le soleil fournit par unité de surface de panneaux photovoltaïques en watt. Ainsi *sun*[*t*] de la fiche se traduit par $self.sun[t] * \frac{self.taille}{1000}$

Le flux $l[t]$ est le flux entrant ou sortant de la batterie sans prise en compte du rendement. Concernant la modélisation, nous avons pris 2 variables pour modéliser chaque flux $l[t]$ comme pour la charging station, sa valeur positive *prodvarsplus*[*t*] et sa valeur négative *prodvarsplus*[*t*], pour les mêmes raisons que celles énoncées précédemment.

Enfin nous avons une nouvelle fois eu recours à une dernière variable *sortieplus*[*t*], partie positive du $l_{tot}[t]$.

1.4 Industrial consumer

La modélisation dans les fiches du site industrial consumer est identique à celle de la solar farm à condition de modifier la variables *sun*[*t*] par - *demand*[*t*].

1.5 Data center

Pour ce qui est du data center, la modélisation était un peu particulière et a évolué au cours du projet. Après un peu de recherche, il est apparu que la première modélisation, telle qu’indiquée dans la fiche acteur, menait en fait à une résolution impossible. Nous avons donc modifié le principe d’équilibre offre-demande tel qu’il était théorisé dans la fiche, pour arriver à un acteur somme toute assez simple. Dans notre modèle, la seule manœuvre à laquelle a accès le data center, c’est de choisir ou non d’activer son système de revente de chaleur, lequel lui coûte de l’énergie puis lui rapporte du profit. Si à l’échelle du microgrid, cela en fait un acteur intéressant avec lequel jouer, à l’échelle de l’acteur en lui-même, l’optimisation individuelle est assez simple : à chaque instant, le programme détermine si il est plus rentable selon les prix d’activer ou non le système de revente de chaleur, et le cas échéant, il l’active

ou non, et retourne le coût engendré et le $l_{tot}[t]$.

En pratique, nous avons codé différents programmes pour chaque acteur. Le programme principal consistait en les fonctions `takethecorrectdecision`, une par acteur. Leur rôle est assez explicite : on lui donne en entrée tous les paramètres extérieurs (grille de prix, horaires pour charging station, etc) et elle nous retourne une fonction l_i optimisée, qu'elle inclut dans les paramètres de l'acteur (dans le self). Après cela nous avons dû coder une fonction `takeasimpledecision` pour chaque acteur aussi, sans utiliser le module d'optimisation PuLP cette fois-ci (il s'agit d'une optimisation basique et volontairement mauvaise, dans l'objectif final de pouvoir "noter" les participants, en prenant comme note basse ce programme, et comme note haute notre programme PuLP).

Après les premiers run, nous nous sommes assez vite rendus compte qu'il manquait un programme crucial pour les acteurs : un contrôle des contraintes. Plusieurs groupes (dont le groupe gagnant du premier run) n'avaient pas du tout respecté les contraintes, ce qui leur permettait d'atteindre un très haut score, mais qui n'avait rien de correct. Nous avons donc codé, pour chaque acteur, un programme de vérification des contraintes, assez large, qui vérifiait que le l_i proposé pour chaque acteur était plausible.

Enfin nous avons essayé de coder un programme d'optimisation globale, de sorte à optimiser tous les acteurs en même temps. Il s'agissait d'obtenir une optimisation meilleure qu'une simple optimisation séparée de chaque acteur, qui ne fonctionnait pas assez bien au niveau de l'autonomie de chaque acteur. Il a fallu prendre en compte les métriques de sortie, pour ce programme, et fusionner en quelque sorte toutes les optimisations de tous les autres acteurs. Nous n'avons malheureusement pas eu le temps d'aboutir entièrement pour ce programme.

1.6 Analyse des résultats

Au vu des résultats, plusieurs erreurs semblent subsister :

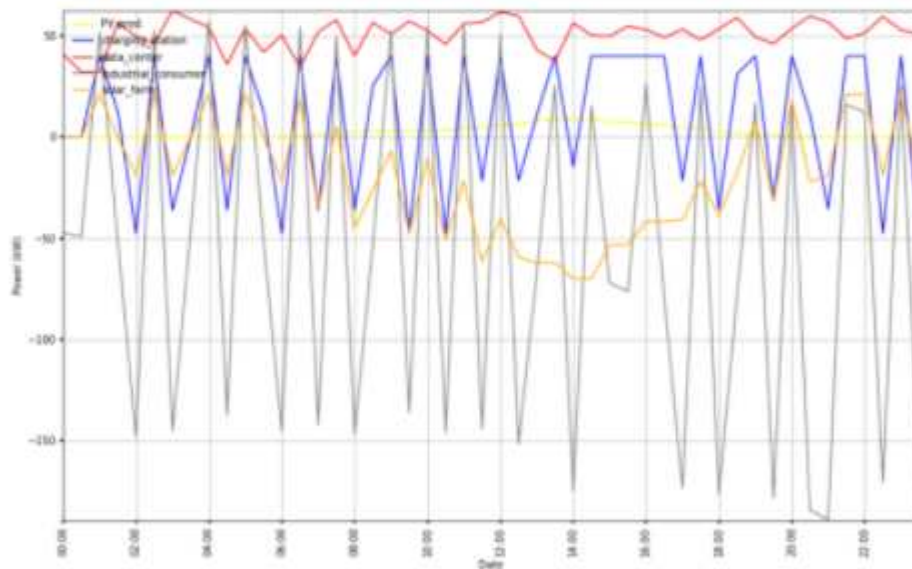
Pour la charging station, il y a un problème assez évident : on devrait avoir un palier à 0 au milieu de la journée, quand les voitures ne sont pas à la station, et ici ce n'est pas le cas. C'est peut être dû au fait que dans notre modèle, un acteur peut acheter et vendre au même moment, ce qui pousse peut être la charging station à acheter et vendre en même temps ici.

C'est cette condition qui pose problème :

$$lp+ = prod_vars_plus[t][i] + prod_vars_moins[t][i] == 0.0, constraintName$$

Une idée pour la résoudre est de créer une condition de plus où on impose $prod_vars_plus[t][i] == 0$ et $prod_vars_moins[t][i] == 0.0$.

**Team team_PIR ACTORS' load AT THE END of
price-coord. dyn.
(PV) region: grand nord**



Pour ce qui est du data center, le profil semble assez cohérent : il varie assez peu dans le temps, et cette variation est vraisemblablement celle attendue (c'est à dire qu'elle correspond à la variation liée à l'énergie d'activation du système de revente d'eau).

Pour industrial consumer, nous avons un gros problème : l'acteur revend de l'électricité, alors qu'il est censé être purement consommateur. Le problème est probablement simplement dû à une confusion dans l'entrée du programme : il s'agit probablement d'un signe - que nous avons cru devoir mettre pour transformer l'acteur en consommateur, alors qu'il était sans doute déjà présent dans le programme total, ce qui pose une grosse erreur globale, mais assez simple à corriger en réalité.

Quant à lui, Solar Farm semble avoir un profil cohérent : l'acteur produit beaucoup d'énergie en milieu de journée (donc quand le soleil est haut), et moins le reste du temps. Les variations parfois positives sont dues à la possibilité qu'à l'acteur d'acheter un peu d'énergie pour la stocker dans sa batterie, et donc de la revendre un peu plus tard, quand les prix lui seront plus favorables.

Ceci est le résultat du run que l'on a effectué.

Il a été lancé avec le programme d'optimisation de chaque acteur. Pour autant il est clairement évident au vu des courbes que nous avons un problème au niveau de l'autonomie, de l'indépendance. En effet, ce profil en dent de scie au niveau des acteurs nous amène à penser que notre stratégie n'est en rien optimale du point de vue de l'indépendance du Microgrid. On peut remarquer que les pics se situent tous au même niveau. C'est à dire que le comportement des 3 acteurs disposant d'une batterie (c'est à dire Industrial consumer, Solar Farm et Charging Station) est le même.

Ainsi que chaque acteur maximise son profit n'est pas la solution pour la maximisation de l'indépendance du Microgrid.

Solution :

Grille tarifaire différente pour chacun mais cette stratégie a ses limites. Afin d'observer et analyser le fonctionnement d'un Microgrid, il est nécessaire de proposer plusieurs scénarios de consommation d'une part, et de s'intéresser à un système d'évaluation des performances d'autre part. Les différents profils de consommations permettent de tester l'adaptation des Microgrid tandis que l'élaboration des métriques de sorties permet de dresser un bilan des comportements des Microgrids. Nous avons élaboré les entrées à partir de données réelles et avons cherché à mesurer les performances économique et énergétique des Microgrids en définissant plusieurs critères de sortie.

2 Données d'entrée-Métriques de sortie

2.1 Données d'entrée

Chargés de pourvoir les équipes de Microgrids de données d'entrées nécessaires pour qu'elles puissent gérer leurs Microgrids en toute lucidité, deux options s'offraient à nous : on pouvait soit leur communiquer les données relatives aux acteurs, fournissant les données du passé jusqu'au jour J-1, et les laisser décider de la gérance du Microgrid selon leur prévisions. Ou bien, en tant que responsables de la "conception entrées sorties", porter le chapeau du prévisionniste et proposer aux équipes quelques scénarios possibles de données d'entrée sur une journée : variations météorologiques, émissions CO2, productibilité de la Solar Farm...

2.1.1 Le clustering : méthode des K-Means

En se limitant aux données d'émissions de CO2 [1] et de production d'énergie solaire par les panneaux photo-voltaïques à notre disposition. On cherche à faire un travail de prévisionniste et présenter un nombre limité de scénarios types aux équipes du Serious Game. Pour cela, on recourt à une méthode de clustering, la méthode K-Means. C'est une méthode qui permet de regrouper en différents clusters les données d'entrées, en se basant sur un critère de minimisation de la somme des distances entre les échantillons de données et leurs centres de clusters associés. [2]

$$\min \sum_j \sum_{x_i \rightarrow c_j} D(c_j, x_i)^2$$

Avec :

c_j : Le centre du cluster

x_i : le ième point dans le cluster, de centre c_j

$D(c_j, x_i)$: La distance (euclidienne) entre le centre du cluster c_j et le point x_i .

$$D(c_j, x_i) = \sqrt{\sum_{j=1}^n (x_{in} - c_{jn})^2}$$

Nous avons eu quelques difficultés à trouver le nombre optimal de clusters à mettre en paramètre dans notre fonction. En effet, nous cherchions à avoir un nombre de clusters permettant de partitionner nos données, sans pour autant les fragmenter de manière excessive et nous empêcher de découvrir des similitudes intéressantes dans les données. D'autre part, nous ne voulions pas rester trop généraliste et jongler avec un nombre de clusters trop petit. Grâce à la *Elbow Method*, nous avons pu résoudre notre problème et déterminer le nombre optimal de clusters. Il s'agit du point à partir duquel la variance ne se réduit plus significativement sur le graphe obtenu en sortie. En pratique, c'est le point en coude des graphes ci-dessous.

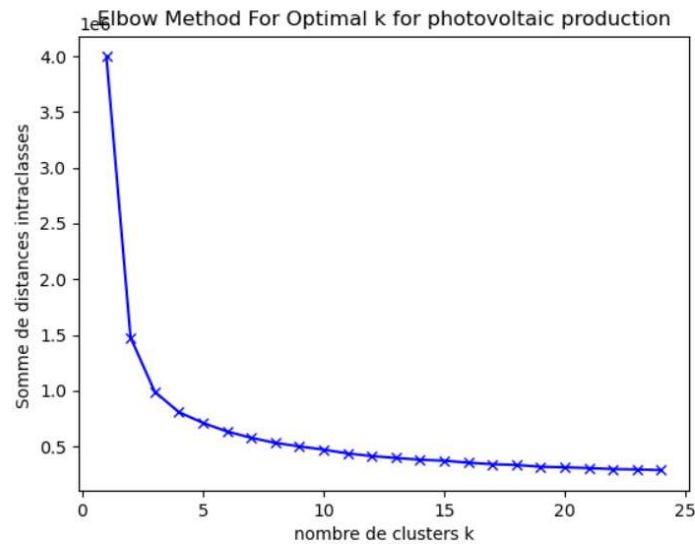


FIGURE 1 – Nombre de clusters optimal pour une répartition basée sur la production solaire

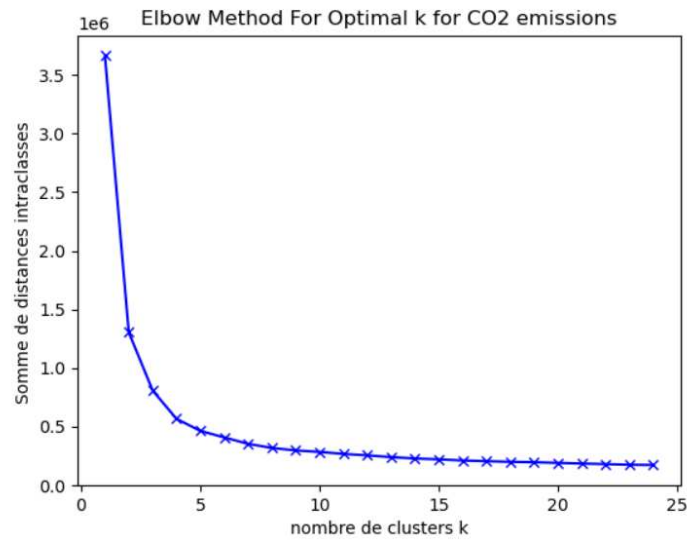


FIGURE 2 – Nombre de clusters optimal pour une répartition basée sur les émissions CO2 journalières

2.1.2 Interprétation

L'exécution du code python pour la méthode des K-Means nous a fait savoir que le nombre optimal de scénarios pour les émissions CO2 était de 4 de même que pour la production photo-voltaïque.

Ainsi, les scénarios communiqués aux différentes équipes seront donc du nombre de 16 puisque les données d'entrée relatives aux émissions de CO2 et à la production solaire sont considérées indépendantes.

2.1.3 Scénarios

Scénarios pour production photovoltaïque				Scénarios pour émission CO2			
Scénario 1	Scénario 2	Scénario 3	Scénario 4	Scénario 1	Scénario 2	Scénario 3	Scénario 4
0	0	0	0	58,0714286	26,1956522	14,8	45,5555556
0	0	0	0	57,25	23,5652174	14,7230769	43,375
0	0	0	0	60,4642857	23,0652174	13,4	44,5972222
0	0	0	0	59,5714286	22,7826087	13,4923077	44,4027778
3,11E-05	9,36E-05	4,65E-06	1,84E-04	59,8214286	22,8804348	13,5692307	44,1944444
3,71E-02	1,21E-01	4,39E-03	3,09E-01	59,8214286	23,1195652	13,7692308	44,7638889
8,04E-01	2,32E+00	1,12E-01	4,44E+00	60,8214286	23,5652174	13,9384615	46,3472222
3,60E+00	1,03E+00	4,12E-01	2,07E+01	60,7142857	24,6413044	13,8	47,875
9,53E+00	2,60E+01	1,63E+00	5,13E+01	61,2142857	25,4456522	13,5538462	47,2222222
2,05E+01	4,86E+01	6,35E+00	8,29E+01	60,4285714	25,2065217	13,4	46,6944444
3,60E+01	7,10E+01	1,33E+00	1,06E+02	58,75	24,7826087	12,8461539	46,0277778
4,81E+01	8,85E+01	1,72E+01	1,20E+02	58,75	24,3478261	12,4461539	45,3055556
5,64E+01	9,69E+01	1,88E+01	1,25E+02	58,5714286	23,8913044	11,9384615	44,9583333
6,05E+01	9,31E+01	1,92E+01	1,20E+02	59,5714286	23,5217391	11,5692308	45,2222222
5,64E+01	8,11E+01	1,91E+01	1,08E+02	60,25	23,6630435	11,4	45,4722222
4,50E+01	6,55E+01	1,64E+01	8,77E+01	60,1071429	24,076087	11,6615385	45,8611111
2,91E+01	4,67E+01	1,06E+01	6,32E+01	60,8571429	24,5434783	11,9846154	46,1805556
1,60E+01	2,79E+01	4,64E+00	3,72E+01	61,75	25,1304348	12,2307692	46,3611111
7,67E+00	1,39E+01	1,58E+00	1,71E+01	60,6071429	26,1413044	12,4769231	46,25
3,30E+00	6,56E+00	6,13E-01	8,93E+00	60,4642857	26,6086957	12,7384615	47,4166667
8,89401885e-01,	1,92E+00	1,62E-01	2,56E+00	60,1071429	26,9347826	13,0769231	47,3611111
4,14E-02	8,59E-02	1,74E-03	1,32E-01	58,4285714	26,7826087	13,4	47,0277778
0	0	0	0	59,4642857	25,8913044	13,6615385	46,0277778
0	0	0	0	59,3571429	26,0217391	13,7538462	45,75

FIGURE 3 – Scénarios en sortie après exécution de la méthode K-Means

En recueillant les coordonnées des centres de clusters en sortie, après exécution de la méthode des K-Means sur la base des données de production photo-voltaïque, puis sur la base des données d'émissions CO2 générées par la production d'électricité, on obtient les scénarios ci dessus, qu'on représente graphiquement ainsi :

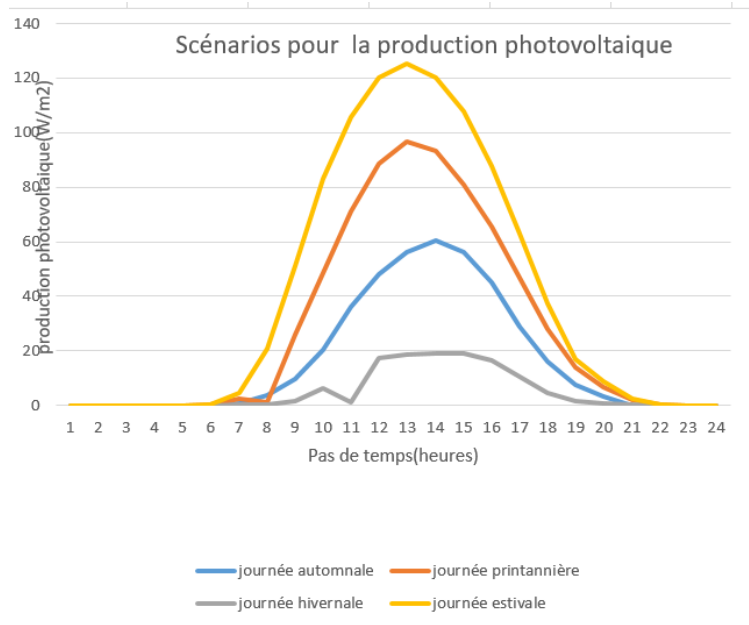


FIGURE 4 – Scénarios basés sur la production photo-voltaïque

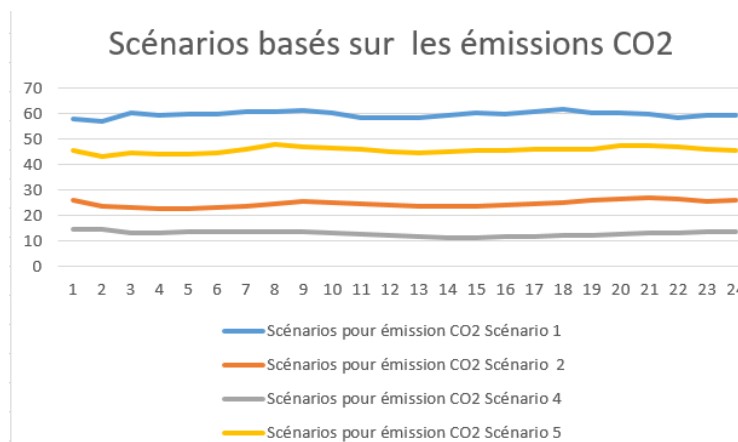


FIGURE 5 – 2Scénarios basés sur les émissions CO2

Sur la figure 4, on remarque que les 4 centres des clusters peuvent être interprétés comme des jours représentatifs des 4 saisons de l'année. En effet, la production photovoltaïque sur une journée peut être vue comme un indice des variations météorologiques.

De même, sur la figure 5 sont représentées 4 journées types pour les émissions CO2 générées par la production électriques. Sachant que les besoins en électricité et chauffage électrique varient selon les saisons, on peut se dire qu'ici encore, chaque scénario est représentatif d'une saison de l'année.

2.2 Métriques de sortie

Afin de comparer les performances des Microgrids, il fallait déterminer des critères d'évaluation. L'intérêt premier des Microgrids étant de maximiser l'utilisation d'énergies locales et de minimiser leur interaction avec le marché, nous avons donc défini un critère d'autonomie.

Par la suite, la question de la proportion d'énergie verte nous est apparue élémentaire. En effet, le Microgrid cherche à réduire son impact environnemental en produisant sa propre énergie avec une Solar Farm et en proposant des services de voitures électriques, limitant les émissions de CO_2 . Ainsi, nous avons réfléchi à un critère relatif aux émissions de CO_2 .

D'autre part, le Microgrid étant un système en marge du réseau général, nous nous sommes renseignés sur les systèmes de raccordement entre les installations électriques ce qui nous a amené à un troisième critère, le coût du raccordement.

Enfin, nous avons introduit une composante économique au fonctionnement du Microgrid en définissant un critère de facture globale. L'intérêt était d'observer comment les équipes valorisaient leurs échanges avec le marché.

L'objectif final de la détermination de ces critères était de trouver une formule globale permettant de les rassembler en un seul critère afin de faciliter l'évaluation du comportement d'un Microgrid, et de classer les différentes Microgrids du Serious Game selon leur performance.

2.2.1 Autonomie

Un Microgrid est un réseau électrique de petite taille, conçu pour fonctionner avant tout en autonomie et pouvoir assurer la relève en cas de panne du réseau, en fonctionnant de manière autonome. Il est alors primordial d'introduire une métrique de sortie qui reflète la dépendance du Microgrid avec le réseau, et qui est à minimiser pour un bon fonctionnement du Microgrid. Cette métrique est à interpréter comme le besoin résiduel qui doit être satisfait par le réseau électrique.

2.2.2 Autonomie globale du Microgrid

Pour numériser le niveau de dépendance de Microgrid, on considère la somme de la partie positive et de la partie négative des échanges du Microgrid avec le réseau sur une journée .

$$A_M = \sum_{t=1}^T \left(\sum_{i=1}^4 l_i(t) \right)^+ + \sum_{t=1}^T \left(\sum_{i=1}^4 l_i(t) \right)^-$$

2.2.3 Variantes de l'autonomie d'un Microgrid

On divise l'autonomie par le nombre d'acteurs pour obtenir une autonomie "moyenne" par acteur :

$$\frac{A_M}{\text{Nombre d'acteurs}}$$

En s'inscrivant dans une approche prévisionnelle, on peut imaginer un calcul où l'autonomie est divisée par la somme des besoins de consommation des acteurs, pour relativiser le besoin de consommation du Microgrid avec l'extérieur. On notera ces besoins "consommation prévisionnelle des acteurs".

$$\frac{A_M}{\text{Consommation prévisionnelle des acteurs}}$$

Où :

$$\text{Consommation prévisionnelle des acteurs} = \text{conso}_p^{CS} + \text{conso}_p^{DC} + \text{conso}_p^{SF} + \text{conso}_p^{IS}$$

Avec des consommations prévisionnelles des acteurs prises comme suit :

$$\text{conso}_p^{SF} = \frac{C_{\text{battery}} \times T}{2}$$

$$\text{conso}_p^{CS} = 25\% \times 4 \times C_{\text{battery}} \times T$$

$$\text{conso}_p^{DC} = \sum_{t=1}^T \left(l_{NF}(t) + \frac{C_{\text{battery}}}{2} \right) \times T$$

$$\text{conso}_p^{IS} = \left(\frac{C_{\text{battery}}}{2} + l_{NF,IS}(y) \right) \times T$$

Remarque : On travaille avec la moitié des capacités de batteries car on évalue la consommation moyenne.

En s'inscrivant dans une analyse a posteriori, on peut calculer cette autonomie relativisée avec les données réelles de consommation.

$$\frac{A_M}{\text{consommation réelle des acteurs}}$$

Avec :

$$\text{consommation réelle des acteurs} = \text{conso}_r^{CS} + \text{conso}_r^{DC} + \text{conso}_r^{SF} + \text{conso}_r^{IS}$$

$$\begin{aligned}
cons_o_r^{SF} &= \sum_{t=1}^T l_1^+(t)(t) \\
cons_o_r^{DC} &= \sum_{t=1}^T l_F(t) + l_{NF}(t) \\
cons_o_r^{CS} &= \sum_{k=1}^4 \sum_{t=1}^T l_{k,v2g}^+(t)(t) \\
cons_o_r^{IS} &= \sum_{t=1}^T l_3(t)
\end{aligned}$$

2.3 Facture

On commence par s'intéresser aux factures des acteurs un à un pour comprendre comment fonctionnent les transferts d'énergie à l'intérieur du Microgrid, et distinguer les consommateurs purs des consommateurs-producteurs. Ce qui nous aidera par la suite pour répartir le gain cumulé par le Microgrid sur les acteurs. Par la suite, on proposera une formulation de la facture totale du Microgrid, celle que nous retiendrons pour l'évaluation globale des performances du Microgrid en fin de journée .

- Les prix d'achat p_i^{buy} et de vente p_i^{sell} de l'électricité sont pris identiques pour tous les acteurs. On les notera respectivement p^{buy} et p^{sell} .
- $l_i^-(t)$ représente la quantité d'énergie produite au pas de temps t par un acteur i.
- $l_i^+(t)$ représente la quantité d'énergie consommée au pas de temps t par un acteur i .
- Le Data Center peut revendre de l'eau au prix p_{eau}^{sell} .
- L'Industrial Site est un consommateur pur.

2.3.1 Facture individuelle de la Solar Farm

$$F_{SF} = \sum_{t=1}^T (p^{buy}(t) \times l_1^+(t) - p^{sell}(t) \times l_1^-(t))$$

2.3.2 Facture individuelle de la Charging Station

$$F_{CS} = \sum_{k=1}^4 \left(\sum_{t=1}^T (p^{buy}(t) \times l_{k,v2g}^+(t)) - \sum_{t=1}^T (p^{sell}(t) \times l_{k,v2g}^-(t)) \right)$$

2.3.3 Facture individuelle du Data center

$$F_{DC} = \sum_{t=1}^T (p^{buy}(t) \times (l_F(t) + l_{NF}(t)) - p_{eau}^{sell}(t) \times l_{eau}(t))$$

2.3.4 Facture individuelle du Industrial Site

$$F_{IS} = \sum_{t=1}^T (p^{buy}(t) \times l_3(t))$$

2.3.5 Facture globale du Microgrid

$$F_M = \sum_{t=1}^T p^{buy}(t) \times \left(\sum_{i=1}^4 l_i(t) \right)^+ - \sum_{t=1}^T p^{sell}(t) \times \left(\sum_{i=1}^4 l_i(t) \right)^- - \sum_{t=1}^T p_{eau}^{sell}(t) \times l_{eau}(t)$$

2.3.6 Relativisation de la facture du Microgrid

- On divise la facture du Microgrid par le nombre d'acteurs pour évaluer une facture moyenne par acteur.

$$F_{M,moy} = \frac{F_M}{6}$$

- Comme pour l'autonomie, on adopte deux approches : prévisionnelle et réelle. Pour l'approche prévisionnelle on procède ainsi :

$$F_{M,prvision} = \frac{F_M}{consommation\ prévisionnelle\ des\ acteurs}$$

- Pour la facture relativisée réelle qui est calculée en fin de journée, on considérera la consommation réelle plutôt que prévisionnelle en dénominateur et travaillerons alors avec la formule ci dessous :

$$F_{M,relle} = \frac{F_M}{consommation\ réelle\ des\ acteurs}$$

2.4 Dimensionnement puissance : Raccordement

L'offre de raccordement intelligente permet aux producteurs d'énergies renouvelables de raccorder leurs installations au réseau le plus proche, en échange d'un prix variant suivant la plus grande valeur de l'énergie échangée avec le réseau sur une journée.

2.4.1 Solution de raccordement pour un Microgrid

On calcule, pour l'ensemble du Microgrid :

$$\max_t \left| \sum_{i=1}^4 l_i(t) \right|$$

Ci-dessous, une offre du coût de raccordement en fonction de la puissance maximale souscrite, envoyée par notre encadrant :

Puissance souscrite	Coût abonnement journalier TTC
6kVA	0.3386 €
9kVA	0.4146 €
12kVA	0.4855 €
15kVA	0.5517 €
18kVA	0.6182 €
24kVA	0.7525 €
30kVA	0.8206 €
36kVA	0.9248 €

2.4.2 Solution de raccordement relativisée pour un Microgrid

On effectue le même calcul en divisant par le nombre d'acteurs du Microgrid pour obtenir un critère de raccordement normalisé.

$$\frac{\max_t \left| \sum_{i=1}^4 l_i(t) \right|}{\text{Nombre d'acteurs}}$$

On peut envisager un calcul du rapport de la puissance maximale réelle sur la puissance maximale à la limite des fonctionnements de chaque acteur.

$$\frac{\max_t \left| \sum_{i=1}^4 l_i(t) \right|}{\sum_{i=1}^4 \max_t |l_i(t)|}$$

2.5 Emission CO₂

On recueille en données d'entrée [1] la quantité de CO₂ qui a été émise, afin de produire l'électricité nécessaire pour couvrir les besoins du Microgrid sur une journée. On prend donc en compte l'électricité achetée sur le marché, car l'électricité produite par le Microgrid est verte contrairement à l'énergie "grise" achetée.

On calcule le coût en CO₂ de la consommation du Microgrid :

$$E_M = \sum_{t=1}^T \left(\sum_{i=1}^4 li(t) \right)^+ \times g_{CO_2}(t)$$

Avec $g_{CO_2}(t)$ l'émission de CO₂ par kWh d'électricité produite en France au pas de temps t (en gCO_2eq/kWh)

Remarques :

- On n'a pas pu intégrer cette métrique dans les critères de performance du Serious Game à cause de la temporalité du projet. En sortie du Serious Game, on se limitera alors aux métriques de sortie : Autonomie, Facture et Raccordement.
- Selon les sites, les données d'émission CO_2 varient. En effet, les techniques de calcul de cet indicateur d'intensité carbone ne prennent pas toujours en compte les différentes énergies de production avec la même importance. Nous avons extrait les données du site *Eco2mix* qui se focalise sur un calcul intégrant uniquement les énergies primaires (charbon, pétrole, gaz naturel, nucléaire, énergie renouvelables), des centrales de production situées sur le territoire français. Elles ne prennent pas en compte les émissions de carbone générées lors de la construction des moyens de production ou lors du cycle d'extraction/ transformation/ transport des combustibles utilisés ni les échanges d'énergies relatifs aux imports/exports. L'ajout de ces échanges ferait varier le coût en CO_2 et modifierait donc le score du Microgrid sur ce critère.

2.6 Comparer la performance des Microgrids

2.6.1 Monétarisation

On veut harmoniser les unités des différentes métriques de sortie pour former un indicateur global du fonctionnement d'un Microgrid.

- La facture est évaluée en €.
- Le coût de raccordement est en €.
- Il faut monétariser le critère d'autonomie d'un Microgrid.
- Le CO₂ est en €.

Autonomie

- Première approche

On calcule le prix moyen :

$$p_{moy} = \frac{\sum_{t=1}^T p^{buy} + p^{sell}}{2\Delta t}$$

Puis l'autonomie devient :

$$A_M^e = p_{moy} \times A_M$$

- Deuxieme approche

On utilise les prix de vente et d'achat :

$$A_M^e = p^{buy} \times A_M^+ + p^{sell} \times A_M^-$$

2.6.2 Pondération des critères

On commence par ramener les deux grands critères : Autonomie et Coût à la même échelle. Pour cela, on normalisera en divisant chaque critère par la valeur maximale qu'il peut prendre.

1 - Normalisation des critères en évaluant leur valeur maximale :

$$\text{Facture maximale} : p_{max}^{buy} \times \left(conso_{max}^{SF} + conso_{max}^{CS} + conso_{max}^{DC} + conso_{max}^{IS} \right)$$

$$conso_{max,p}^{SF} = C_{battery} \times T$$

$$conso_{max,p}^{CS} = 100\% \times 4 * C_{battery} \times T$$

$$conso_{max,p}^{DC} = \sum_{t=1}^T l_{NF,DC}(t) + C_{battery,DC} \times T$$

$$conso_{max,p}^{IS} = c_{battery,IS} \times T_{tot} + \sum_{t=1}^T l_{NF,IS}(t)$$

$$\text{Raccordement maximal} : p_{max,tarriif} = 0.9248 \text{ €}$$

$$\text{Emissions maximales} : p_{CO_2} \times \left(conso_{max}^{SF} + conso_{max}^{CS} + conso_{max}^{DC} + conso_{max}^{IS} \right)$$

Le critère Coût = Facture + Emissions CO2 + Raccordement devient :

$$\frac{F_M + C_M + p_{racc}}{3(F_{max} + C_{max} + p_{racc,max})}$$

$$\text{Autonomie maximale non monétarisée} : A_{max,p}^{SF} + A_{max,p}^{CS} + A_{max,p}^{IS} + A_{max,p}^{DC}$$

$$\text{Solar Farm} : A_{max,p}^{SF} = C_{battery,SF}$$

$$\text{Charging Station} : A_{max,p}^{CS} = 4 \times C_{battery,CS}$$

$$\text{Industrial Consumer} : A_{max,p}^{IS} = \sum_{t=1}^T l_{NF,IS}(t) + C_{battery,IS}$$

Data center : $A_{max,p}^{DC} = \sum_{t=1}^T l_{NF,DC}(t) + C_{battery,DC}$

L'autonomie considérée devient alors :

$$\frac{A_M}{A_{max,p}^{SF} + A_{max,p}^{CS} + A_{max,p}^{IS} + A_{max,p}^{DC}}$$

2 - Mathématiques d'analyse de multi-critères

	Autonomie	Cout=Facture + Emissions C02 +Raccordement
Microgrid 1	A1	C1
Microgrid 2	A2	C2

On commence par attribuer un poids p_i à chaque critère (avec $\sum_{i=1}^2 p_i = 1$)

	Autonomie	Facture + Emissions C02 + Raccordement
Poids de critères	P1	P2

La performance globale d'un Microgrid, qui servira alors de base pour le classement des équipes est :

$$Performance_{globale} = p_1 \times A + p_1 \times C$$

Vu l'importance que prend le critère d'Autonomie dans l'évaluation des performances d'un Microgrid, on lui attribue le poids $p_1 = 1/2$ et le poids attribué à la métrique Coût, qui rassemble Facture, émissions CO_2 et Raccordement serait $p_2 = 1 - p_1 = 1/2$

2.7 Gain des acteurs en Microgrid

Facture du Microgrid avec la coordination des acteurs :

$$F_{M,c}$$

Facture du Microgrid sans coordination des acteurs :

$$\begin{aligned} F_{M,i} &= \sum F_{SF,i} + F_{DC,i} + F_{CH,i} + F_{IS,i} \\ &= \sum_{t=1}^T (p^{buy}(t) \times l_1^+(t) - p^{sell}(t) \times l_1^-(t)) \\ &\quad + \left(\sum_{k=1}^4 (\sum_{t=1}^T (p^{buy}(t) \times l_{k,v2g}^+(t)) \right. \\ &\quad \left. - \sum_{t=1}^T (p^{sell}(t) \times l_{k,v2g}^-(t)) \right) \end{aligned}$$

$$\begin{aligned}
& + \sum_{t=1}^T (p^{buy}(t) \times (l_F(t) + l_{NF}(t))) \\
& + \sum_{t=1}^T (p^{buy}(t) \times l_3(t))
\end{aligned}$$

Finalement, la quantification du gain des acteurs en Microgrid en terme de factures est disponible avec la différence :

$$G_{tot} = F_{M,c} - F_{M,i}$$

Reste alors à répartir le gain "équitablement" entre acteur. Pour cela, on peut se référer à la notion d'équité en théorie des jeux coopératifs, comme présentée par Shapley. [3]

3 Coordinateur du microgrid

3.1 Rôle du coordinateur de la smart grid

Le coordinateur du smart grid a pour objectif d'inciter les acteurs à consommer de l'électricité à certains instants plutôt qu'à d'autres, afin que la consommation électrique du smart grid vérifie certaines conditions ; qui peuvent être par exemple minimiser les coûts électriques des acteurs, accroître l'autonomie du microgrid, réduire les émissions de CO₂, etc.

Pour cela, le coordinateur met en place une tarification fictive qui est envoyée aux acteurs pour qu'ils décident de leur profil de consommation en fonction de ces prix fictifs, le coordinateur cherchera alors à fixer les meilleurs prix fictifs, c'est-à-dire ceux qui vérifient au mieux les contraintes fixées au préalable. (Le vrai prix payé par les acteurs sera le prix du marché heures pleines/heures creuses).

On fait le choix de fixer les prix et la consommation électrique à l'avance (et non à chaque pas de temps de manière dynamique). On commence par envoyer un prix de base aux acteurs, qui répondent par un profil de consommation que le coordinateur utilise pour proposer des meilleurs prix, qui sont renvoyés aux acteurs, et on procède ainsi par itérations successives. Cela pose la question de la méthode de choix des prix à chaque itération, ainsi que de la condition d'arrêt de la boucle. Plusieurs possibilités de mise à jour des prix seront proposées ci-dessous.

3.2 Modèles de tarification initiaux

3.2.1 Prix affines

Expression du prix La tarification affine s'exprime sous la forme :

$$\forall t, p(t) = p_0(t) + \mathbf{K} \sum_i l_i(t)$$

où $p_0(t)$ est le prix "idéal" si le microgrid est autonome, \mathbf{K} une constante de pénalisation strictement positive, et $l_i(t)$ est la puissance consommée par l'acteur i au temps t . Les prix sont identiques pour tous les acteurs, que ce soit à l'achat ou à la vente.



FIGURE 6 – Coordination classique

Justifications Le prix est affine en fonction de la surconsommation du microgrid. Il s'agit d'un système de tarification très simple, dont le but est d'améliorer l'autonomie du microgrid. Les prix sont les mêmes à l'achat et à la vente pour tous les acteurs car il n'est pas nécessaire de discriminer les acteurs. On augmente le prix si la demande dépasse l'offre, et on baisse le prix si l'offre dépasse la demande, ce qui encourage les acteurs à déplacer leur consommation (resp. leur production) sur les instants sur lesquels le prix est le plus bas (resp. le plus haut). À priori, il semble raisonnable de penser que si la constante \mathbf{K} est suffisamment grande, la consommation du smart grid convergera vers l'état d'autonomie optimal avec un grand nombre d'itérations. On observe cependant que ce choix de tarification a tendance à produire des pics de consommation.

3.2.2 Variante proche de l'algorithme de départ

Expression du prix

$$\forall t, \quad \begin{aligned} \nabla(t) &= \sum_i l_i(t) - \text{moy}_t(\sum_i l_i(t)) \\ p(t) &= p(t-1) + \varepsilon \nabla(t) \end{aligned}$$

Les prix sont toujours les mêmes pour tous, à l'achat ou à la vente. On a $\varepsilon > 0$.

Justification Ce choix de tarification présente des avantages similaires au premier, mais produit des profils de prix plus lisses, car c'est le changement de prix qui est désormais affine en fonction de la surconsommation. Cela permet de diminuer l'amplitude des pics de consommation constatés avec la précédente méthode.

3.3 Améliorations l'algorithme initial

3.3.1 Minimisation du soutirage réseau $l(t)^+$

L'idée serait de rajouter un terme avec une certaine pondération dans l'expression du prix telle que :

$$\forall t, \quad \begin{aligned} \nabla(t) &= \sum_i l_i(t) - \text{moy}_t(\sum_i l_i(t)) \\ p(t)_{\text{min soutirage reseau}} &= p(t-1) + \varepsilon \nabla(t) + \mathbf{K}' \sum_i l_i(t)^+ \end{aligned}$$

Effets attendus On espère que la surpondération quand on tire de l'énergie sur le réseau permettra d'inciter les acteurs à diminuer leur demande ou augmenter leur offre d'électricité sur le pas de temps là.

3.3.2 Minimisation des pics de consommation

L'idée serait de définir 4 plages horaires où les acteurs ont la priorité pour consommer et 4 autres où ils ont la priorité pour produire. Il faudrait ainsi envoyer 4 grilles de prix d'achats et 4 grilles de prix de vente différentes.

Cela fonctionnerait avec le principe de pénalités et subventions. Un acteur est pénalisé s'il consomme sur une plage horaire non adaptée et un acteur est subventionné (virtuellement) s'il fournit de l'énergie au microgrid au bon moment. On peut créer la fonction prix d'achat suivante :

$$\forall t, \quad \begin{aligned} \nabla(t) &= \sum_i l_i(t) - \text{moy}_t(\sum_i l_i(t)) \\ p_i(t) &= p(t-1) + \varepsilon \nabla(t) + \varepsilon_{i,2}(t) l_i(t) \end{aligned}$$

Explication des effets attendus On espère que les 4 acteurs vont faire en sorte de répartir leur pic de consommation sur 2 des plages horaires où elles sont incitées à le faire. Les valeurs de $\varepsilon_{i,2}$ sont normalisées pour donner un ordre d'idée relatif

Acteurs	Acteur 1	Acteur 2	Acteur 3	Acteur 4
$\varepsilon_{i,2}$ de 00h à 6h	1	0.5	0.5	1
$\varepsilon_{i,2}$ de 6h à 12h	1	1	0.5	0.5
$\varepsilon_{i,2}$ de 12h à 18h	0.5	1	1	0.5
$\varepsilon_{i,2}$ de 24h à 24h	0.5	0.5	1	1

FIGURE 7 – Valeurs de $\varepsilon_{i,2}$ en fonction du temps

3.3.3 Variante de boucle

On veut faire en sorte que chaque acteur soit au courant des actions des autres. Pour le moment tous les acteurs envoient leur grille de demande/production simultanément et on somme pour obtenir le profil de consommation total. Par conséquent, leur profil de consommation ne dépend pas du profil de consommation des autres acteurs au même tour. C'est pourquoi nous proposons d'effectuer notre algorithme de fixation des prix sur chaque acteur un à un pour affiner la coordination.

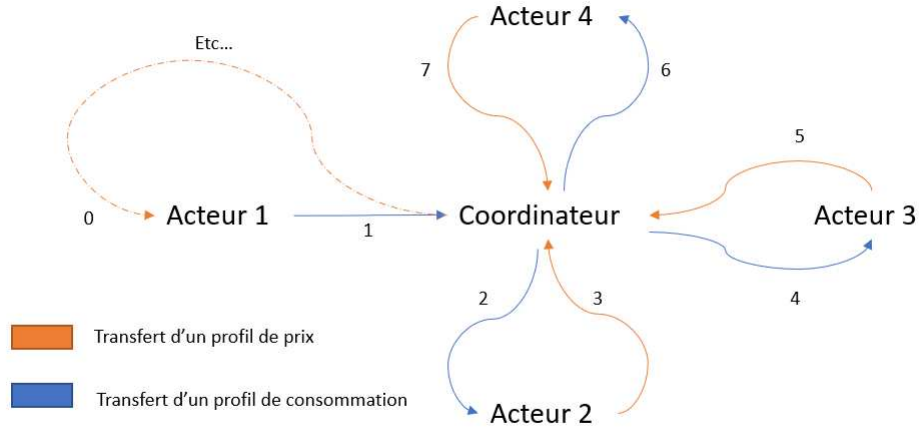


FIGURE 8 – Variante de boucle

3.3.4 Minimisation d'indicateurs

Minimisation de l'indicateur de dépendance On utilise ici l'indicateur de dépendance (défini dans la partie sur les données d'entrée et métriques de sortie) qui décroît avec l'autonomie du microgrid au réseau général. On a alors le problème suivant :

$$\text{Min. } A_M(\mathbf{p})$$

Il s'agit d'un problème de minimisation sans contrainte, que l'on peut résoudre en Python avec par exemple :

```
scipy.optimize.minimize
```

Note : Ce choix nécessiterait de ne plus écrire la boucle explicitement.

Minimisation des émissions de CO₂ Les émissions de CO₂ sont également définies dans la partie sur les métriques. Il s'agit également d'un problème de minimisation sans contrainte :

$$\text{Min. } C_M(\mathbf{p})$$

Cas général : Indicateur composé On peut définir une fonction objectif notée f formée de plusieurs indicateurs définis dans la partie sur les métriques (le plus raisonnable est que $f(\mathbf{p})$ soit une combinaison linéaire des indicateurs). On a alors à résoudre :

$$\text{Min. } f(\mathbf{p})$$

3.3.5 Utilisation de prix différents pour chaque acteur

Il est également possible de fixer des prix différents pour chaque acteur, afin de prioriser les acteurs dont l'impact serait le plus grand sur la consommation du microgrid.

3.3.6 Méthode de Newton

On suppose que l'autonomie optimale est parfaite, i.e. elle vérifie $\forall t, \sum_i l_i(t) = 0$ (en particulier, cela implique que le microgrid est autonome sur la journée). La consommation des acteurs dépend des prix fixés par le coordinateur, donc en notant \mathbf{l}_i le profil de consommation de l'acteur i , $\sum_i \mathbf{l}_i = F(\mathbf{p})$ avec \mathbf{p} le profil de prix. On doit donc résoudre l'équation $F(\mathbf{p}) = 0$. On pourrait donc utiliser des méthodes numériques classiques telles que la méthode de Newton.

3.4 Condition d'arrêt de la boucle et nombre d'itérations

3.4.1 Nombre d'itérations fixé à l'avance

Il est possible de fixer le nombre d'itérations à l'avance. Cela a l'avantage de toujours s'arrêter en un temps raisonnable. En revanche, il n'est pas certain que le profil de consommation finale soit satisfaisant.

3.4.2 Arrêt dès qu'un indicateur à minimiser descend en dessous d'un seuil

Il est également possible de fixer un seuil d'arrêt, qui garantit que le résultat est satisfaisant, mais pas l'arrêt de la boucle.

4 Développement informatique des règles de coordination des acteurs

4.1 Fonctionnement informatique général du microgrid manager

Comme expliqué dans la partie précédente, le microgrid manager a un rôle central dans le bon déroulé du serious game. C'est le code qui fait le lien entre les différents acteurs, qui définit le prix, les scénarios, qui calcule la facture de chaque joueur. En effet, chaque acteur de chaque équipe a créer un code prenant en entrée des prix et renvoyant un profil de consommation, et c'est le microgrid manager qui se charge de faire tout fonctionner et de renvoyer les résultats finaux.

D'un point de vue informatique, le microgrid manager va définir un scénario (ce qui est approfondi dans la partie 4.3), puis il va chercher à fixer un prix optimal (ce qui est expliqué dans la partie 4.2), puis une fois ce prix trouvé, il retournera les résultats (profils de

consommation des acteurs, leurs factures, mais aussi la consommation totale du microgrid et sa facture totale).

4.2 Lancement des runs

Regardons rapidement la dynamique de lancement d'une simulation informatique du microgrid. La première étape consiste à récupérer le code des acteurs et à le vérifier. Pour la partie transmission/récupération, on utilise Github avec d'une part un dépôt comprenant les codes des quatre acteurs et d'autre part un dépôt central contenant les mécanismes de coordination et de visualisation des résultats.

Le lancement d'un run commence par l'exécution de *checkout_code.py* qui pull depuis le dépôt des acteurs la dernière version du code de ces derniers. De plus ce programme réalise une vérification du bon fonctionnement des acteurs. On lance ensuite *run.py* qui se charge d'initialiser les scénarios, de les envoyer aux acteurs, de récupérer les listes de consommations et de renvoyer les prix. C'est ce même programme qui après plusieurs itérations du cycle : récupération des consommations/envois des prix se charge d'exécuter les programmes de visualisation des résultats.

4.3 Coordination des prix

Comme on a pu le voir dans la Partie 3.1, il est important que les acteurs répartissent bien leur consommation sur toute la journée. Pour ce faire, il faut augmenter le prix lorsque les acteurs consomment beaucoup, et ainsi les inciter à consommer sur un autre pas de temps. C'est pour cela que l'on a besoin de définir un schéma de coordination des prix.

Dans un premier temps, on a défini une relation affine pour définir le prix (comme expliqué dans la partie 3.2.1).

Informatiquement, cela revient à fixer un prix initial constant pour tous les pas de temps :

$$\forall t, \lambda(t) = \lambda^{(0)}$$

Ce prix initial correspond au prix auquel on aurait vendu l'électricité si la consommation était nulle. Ce prix est défini dans la fonction *self.initialize_prices()* appelée dans l'Algorithme 1 avant la boucle for et les différentes mise à jour du prix.

On envoie alors cette liste de prix aux acteurs (à l'aide de la fonction *self.send_prices_to_players(prices)*) et on leur demande de nous retourner leur consommation (*load_player*) pour de tels prix (que l'on récupère avec la fonction *self.get_microgrid_load()*). On a alors la consommation

total du microgrid à chaque pas de temps :

$$microgrid_load(t) = \sum_{acteur} load_player(t)$$

On peut ainsi redéfinir une nouvelle liste de prix de la manière suivante : $\lambda(t) = \lambda^{(0)} + K * microgrid_load(t)$, où K est une constante positive qui sert à inciter les acteurs à ne pas tous consommer au même pas de temps. Cette nouvelle liste de prix est calculée à l'aide de la fonction (*self.get_next_prices(iteration, prices, microgrid_load)*).

Ensuite, on réitère (on envoie aux acteurs les nouveaux prix, on reçoit leurs consommations et on peut ainsi calculer de nouveaux prix en rappelant la fonction *self.get_next_prices(iteration, prices, microgrid_load)* et on attend que les prix convergent. S'il n'y a pas de convergence après un nombre *self.n_max* d'itérations, on utilise les derniers prix calculés pour calculer les profils de consommations finaux des acteurs.

```

Fonction Prices_coordination (self)
prices = self.initialize_prices()
pour iteration in range(self.n_max) faire
    self.send_prices_to_players(prices)
    microgrid_load, player_loads = self.get_microgrid_load()
    prices, converged = self.get_next_prices(iteration, prices, microgrid_load)
    si converged alors
        | break
    fin
fin
finFonction

```

Algorithme 1 : Pseudo-code de la coordination des prix

Maintenant que l'on a vu comment coordonner les prix, on va chercher à complexifier la mise à jour des prix (dans la fonction *self.get_next_prices*). On ne va donc plus considérer un prix affine.

On définit un gradient de consommation comme un écart de la consommation à l'instant t par rapport à la moyenne de la consommation (comme on le voit dans l'algorithme 2). Lorsque le gradient au pas de temps t est positif, le prix va augmenter, lorsqu'il est négatif, il va diminuer. Pour s'assurer de la convergence des prix, on fait décroître le coefficient (α) dans l'algorithme 2) qui multiplie le gradient à chaque nouvelle mise à jour des prix. Cette nouvelle expression est justifiée et posée mathématiquement dans la partie 3.2.2.

L'Algorithme 2 explique comment ces prix sont calculés informatiquement.

```
Fonction get_next_prices(self, iteration, prices, microgrid_load)
old_prices = prices
grad = microgrid_load - np.mean(microgrid_load)
dp = 10-3 * 0.99iteration
new_prices = old_prices + dp * grad
return new_prices, False
finFonction
```

Algorithme 2 : Pseudo-code de la mise à jour des prix avec un gradient de consommation

4.4 Gestion des scénarios

Il faut maintenant élaborer un processus efficace de lecture des données d'entrées. Notre modélisation se base sur plusieurs scénarios représentants pour chaque acteur un profil de consommation spécifique. Revenons, pour tous les acteurs sur la structure des données. Pour le complexe industriel, on dispose d'un tableau de 90 scénarios de 48 demi-heures chacun. Ces derniers sont partagés entre 3 sites industriels « types ». En pratique on range ce tableau au format .csv dans un dictionnaire python nommé `scenario`. Pour accéder à un profil de consommation quelconques, il suffit donc de taper :

```
scenario['industrial_farm']['scenario_i']
```

Pour récupérer une liste de consommations à chaque pas de temps (un pas par demi-heure) Le procédé est le même pour le data-center. On dispose d'une liste de 10 scénarios de consommation différents qui peuvent s'appeler par la commande :

```
scenario['data_center']['scenario_i']
```

Les deux acteurs les plus délicats, à savoir la ferme solaire et la station de recharge ont par ailleurs des données d'entrées différentes de celles précédente. La ferme solaire par exemple à besoin de ressource solaire pour produire de l'énergie. Ainsi, la production varie fortement d'une région à une autre et il faut donc des scénarios adaptés à chaque type d'ensoleillement. Pour accéder à un scénario quelconque de la ferme, il faut en plus préciser la région d'étude.

```
scenario['solar_farm']['region_k']['scenario_i']
```

La commande ci-dessus nous renvoie donc une liste de production photovoltaïque pour une journée *i* dans la région *k*. Enfin, la station de recharge doit seulement prendre en compte les horaires de départ et d'arrivée des véhicules. On accède alors au scénario d'une journée par la commande

```
scenario['solar_farm']['scenario_i']['car_k']
```

Première équipe								
Itération	0	1	2	3	5	7	8	9
Total_bill	223.392	223.392	222.625	221.956	221.623	219.84	221.609	217.862
Deuxième équipe								
Itération	0	1	2	3	5	7	8	9
Total_bill	208.906	169.863	188.474	189.619	189.55	189.452	189.291	189.196
Troisième équipe								
Itération	0	1	2	3	5	7	8	9
Total_bill	239.664	239.664	239.664	238.227	237.309	237.978	238.799	235.696

TABLE 1 – Évolution des factures après plusieurs itérations pour différentes équipes (soumises à des scénarios différents)

qui nous renvoie un couple d'horaires pour la voiture k dans la situation i .

Par soucis d'efficacité, on choisit de récupérer ces données avant de lancer la simulation. Elles serviront ensuite au coordinateur qui en choisiras une au hasard pour chaque acteur du Microgrid considéré.

4.5 Résultats observés

Dans cette partie, on s'intéresse à l'efficacité de la mise à jour des prix sur l'évolution de la facture totale du microgrid.

La TABLE 1 regroupe les résultats de 3 équipes différentes soumises à des scénarios différents. On remarque que les factures des microgrid ont bien tendance à diminuer (ce qui est en partie l'effet recherché), même si ce n'est pas le cas à chaque itération (la facture totale de l'équipe 2 augmente entre les itérations 1 et 2).

4.6 Pistes pour approfondir

En ce qui concerne la coordination des prix, on a fait le choix d'essayer d'uniformiser le plus possible la consommation totale de la microgrid. Or ce n'est pas le seul choix possible. En effet, on aurait aussi pu :

- Chercher à maximiser l'autonomie en cherchant à réduire `microgrid_load` à chaque appel de la fonction `get_next_prices`, par exemple en augmentant le prix de vente lorsqu'un des acteurs consomment beaucoup pour inciter les autres acteurs à lui vendre de l'électricité et ainsi favoriser les échanges internes ;
- Minimiser le pic de consommation de la microgrid, c'est à dire minimiser $\max_t \text{microgrid_load}(t)$. Pour ce faire, on pourrait par exemple augmenter le prix au moment où ce maximum est atteint (`np.argmax(microgrid_load)`) et le diminuer sur les

autres pas de temps ; Une autre méthode plus compliquée à implémenter serait d'envoyer les prix à chaque acteurs les uns après les autres. En rajoutant des boucles itératives au sein de la fonction `get_next_prices`, on pourrait envoyer d'abord des prix à la ferme solaire qui adapterait sa consommation en réponse puis on recalculerait les prix sur la base de la décision de la ferme... Avec une telle méthode, les pics de consommations des acteurs se verraient déphasés ce qui améliorerait les performances du microgrid.

Conclusion

Pour conclure, nous avons plutôt bien avancé par rapport aux objectifs donnés, et avons réussi à proposer un code à peu près correct d'optimisation pure pour chaque acteur, en plus des codes de vérification et de choix simple. Les pistes potentielles d'amélioration sont évidemment celles du code final mélangeant les quatre acteurs en même temps, qui pourrait nous permettre d'atteindre un très bon niveau d'optimisation, et ce plus seulement au niveau du profit de chaque acteur, mais aussi de l'autonomie globale.

Pour comparer les performance des équipes de Microgrids, on s'est appuyé sur quatre critères principaux : Autonomie, Facture, Raccordement et Emissions CO2. Nous avons monétarisé ces critères, avant de les assembler dans une formule globale, après avoir assigné à chaque critère un poids, qui reflète son importance. C'est selon la valeur numérique obtenue pour ce critère qu'on pourrait ensuite classer les équipes de Microgrids entre elles.

Le modèle de coordination choisi pour les trois premiers runs à tendance à maximiser l'autonomie, mais crée des pics de consommation. Nous avons alors proposé plusieurs moyens pour minimiser le soutirage réseau ou pour minimiser les pics de consommation. On pourra également essayer de minimiser la dépendance et les émissions de CO2. Il faut cependant faire attention à ce que les modifications n'impactent pas l'objectif initial. Pour finir, il y a deux moyens d'imposer une condition de fin de boucle. Nous avons choisi d'utiliser un nombre d'itérations fixe, mais il serait également intéressant d'utiliser un seuil d'indicateur. Se pose alors la question de la convergence de l'algorithme. On ne se sait pas s'il va réussir à atteindre ce seuil.

Références

- [1] <https://www.rte-france.com/eco2mix/les-emissions-de-co2-par-kwh-produit-en-france>
- [2] <https://mrmint.fr/algorithme-k-means>
- [3] Georges Zaccour, *Valeur de Shapley et partage équitable des ressources*. L'Actualité économique, 64(1), 96–121.

5 Annexe : fiches des acteurs

1 Introduction & key notations

The following notations and principles apply to all groups to simplify their interactions:

- Each group is given a number $i \in \{1, \dots, N\}$ and is assigned a role (player) on the Smart Grid (Figure 1).
- You will play over discrete time steps $t \in \{0, 1, \dots, T\}$ representing a 24h day. Time intervals $[t, t + 1]$ span over the time increment $\Delta t = 30 \text{ min}$ (hence $T = 48$).
- Each player i on the grid has a **load profile** (i.e. electric power consumption) described as the vector

$$\mathbf{l}_i = (l_i(1), \dots, l_i(T)).$$

This vector expresses the electric power demand (kW) of player i over the interval $[t - 1, t]$ ¹, considering that $l_i(t)$ is non-negative (respectively non-positive) if power is imported (respectively exported) by the player. This load is constrained by a power limit $|l_i(t)| \leq p_i^{max}$ due to the design of the network.

- Each actor i on the grid has a **daily bill** described as the sum of the stage wise costs (i.e. cost per time interval) due to power exchanges, plus possibly a final cost:

$$b_i = \sum_{t=0}^{T-1} cost_i(t) + cost_i(T)$$

Note: costs (and daily bills in general) may be negative if you are making profit during stage t . This does not apply to all groups

- In each models, we distinguish quantities that are **non flexible**, i.e. which you cannot affect, from quantities that are **decision** variables, which are subject to your management policy.
- You may have additional variables according to your site. Parameters and data sample are available in the **data/** folder provided with this document. Each site has interactions within the Smart Grid according to Figure 1. All players want to manage their power flow at least cost to minimize their daily bill.

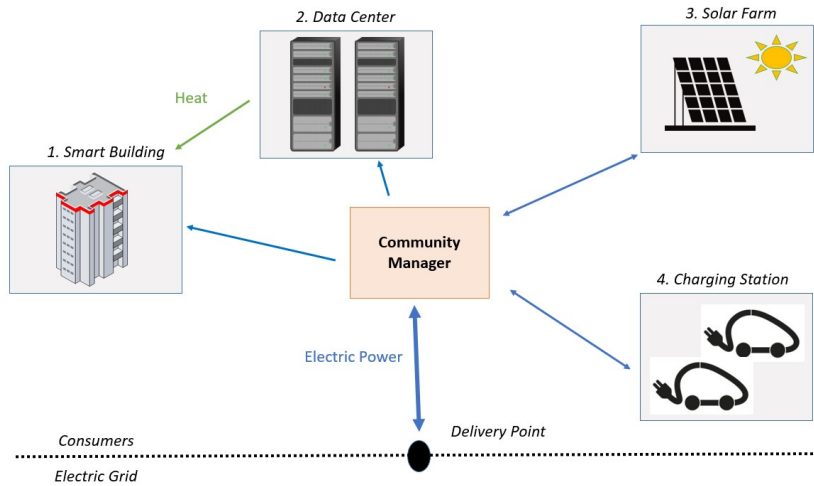


Figure 1: Schematic view of the Smart Grid

¹We adopt the following convention: for a time interval $[t, t + 1]$, decisions and state variables are indexed by t , while non-flexible (observed) variables are indexed by $t + 1$.

2 Modeling of an Industrial Site

2.1 Industrial demand

The electric load of an Industrial Site (IS) depends on its demand over the interval $[t - 1, t]$,

$$l_{\text{dem}}(t), t \in \{1, \dots, T\}. \quad (1)$$

Examples of daily power demand profiles are given on Figure 2. Observe that the power production is subject to randomness. Thus, l_{dem} cannot be controlled and is a **non-flexible** variable. Besides, we assume that the IS is equipped with a battery unit. Power may be charged or discharged according to the load

$$l_{\text{bat}}(t), t \in \{0, \dots, T - 1\}, \quad (2)$$

accounting for the power transferred to the battery over the time span $[t, t + 1]$. This load may be controlled to fulfill your objectives, and stands for a **decision** variable. The total load of the IS is then

$$l_i(t + 1) = l_{\text{bat}}(t) + l_{\text{dem}}(t + 1), \forall t \in \{0, \dots, T - 1\}. \quad (3)$$

In the case of the Industrial Site, power exchanges are mainly directed in order to buy power to other players or to the electric grid. You may also buy energy to store it and sell it later.

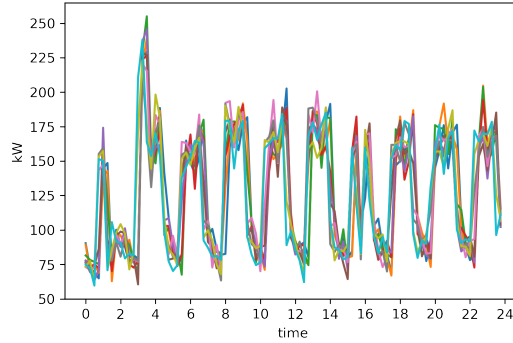


Figure 2: Examples of daily demand profiles. These profiles may not be representative of your site, data samples are available in the `data/` folder.

2.2 Model of the battery

Your Industrial Site is equipped with a battery which enables you to store energy. The purpose of the battery is to adapt your load l_i to the production of other players and to the prices that vary along the day. At time t , the available energy of the battery is

$$a(t), t \in \{0, \dots, T\}, \quad (4)$$

expressed in kWh. The battery stock evolves according to the load l_{bat} , and is ruled by the following dynamics equation:

$$\begin{cases} a(t+1) = a(t) + [\rho_c l_{\text{bat}}^+(t) - \frac{1}{\rho_d} l_{\text{bat}}^-(t)] \times \Delta t, \quad \forall t \in \{0, \dots, T-1\}, \\ a(0) = 0, \\ 0 \leq a(t) \leq c, \quad \forall t \in \{0, \dots, T\}, \\ |l_{\text{bat}}(t)| \leq p_i^{\max}, \quad \forall t \in \{0, \dots, T-1\}, \end{cases} \quad (5)$$

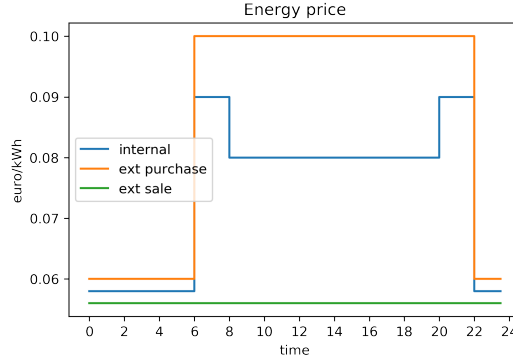
where

- $\rho_c = 0.95$ charge efficiency,
- $\rho_d = 0.95$ discharge efficiency,
- c capacity of the battery (kWh), see **data/** folder,
- p_i^{\max} maximum power of the battery (kW), see **data/** folder,
- $x^+ = \max(0, x)$, $x^- = \max(0, -x)$.

At the end of the day, the energy stock left in the battery is valued at the Smart Grid internal electricity price.



Figure 3: Example of battery with 9 modules.



delivery point is $l(t) = \sum_i l_i(t)$. Within the community, the total demand is $d(t) = \sum_i l_i^+(t)$ and the total supply is $s(t) = \sum_i l_i^-(t)$ so that $l(t) = d(t) - s(t)$. The local power consumption is $\min\{d(t), s(t)\}$. In this first class, we decide to apply fixed prices and a proportional sharing of the bill. The goal of the community manager is to maximize intra-community exchanges, hence prices set so that $\tilde{p}_e(t) \leq p_e(t) \leq \hat{p}_e(t)$.

- A buyer accounts for $\frac{l_i^+(t)}{d(t)}$ of the total purchase. The total purchase of power amounts to

$$purchase(t) = \underbrace{p_e(t) \times \Delta t \times \min\{d(t), s(t)\}}_{\text{internal}} + \underbrace{\hat{p}_e(t) \times \Delta t \times l^+(t)}_{\text{external}}$$

- A seller accounts for $\frac{l_i^-(t)}{s(t)}$ of the total sale. The total sale of power amounts to

$$sale(t) = \underbrace{p_e(t) \times \Delta t \times \min\{d(t), s(t)\}}_{\text{internal}} + \underbrace{\tilde{p}_e(t) \times \Delta t \times l^-(t)}_{\text{external}}$$

Note: $x^+ = \max\{0, x\}$, $x^- = \max\{0, -x\}$

- **NEW:** In the previous class, your policy could only vary with time. We provide you with more indicators which you can use to design a finer energy consumption policy

- $p_i^{buy}(t) = \frac{b_i(t)}{l_i^+(t) \times \Delta t}$ price at which you buy electric energy
- $p_i^{sell}(t) = \frac{-b_i(t)}{l_i^-(t) \times \Delta t}$ price at which you sell electric energy
- $p_g^{buy}(t) = \frac{purchase(t)}{d(t) \times \Delta t}$ price at which the community (the grid) buys energy
- $p_g^{sell}(t) = \frac{sale(t)}{s(t) \times \Delta t}$ price at which the community (the grid) sells energy
- Additionally, you can use the current and past values of your load $l_i(t)$ and your energy stock (if you have one) to design your policy

Note: with the current price system individual and community prices should be equal

2 Modeling of a Data Center

2.1 Electric load

The electric load of the data center has a **non flexible** part reflecting daily power usage for basic activities such as data storage, request exchanges, CPU/GPU computation tasks ... Although this load varies in time due to clients demand, it is not a control variable. In our case, we consider a TIER 1 data center [?] which implies a power density of about $250 \text{ W}/m^2$. We also consider a surface of 100 m^2 resulting in load profiles such as illustrated in figure 2

$$l_{IT} = (l_{IT}(1), \dots, l_{IT}(T))$$

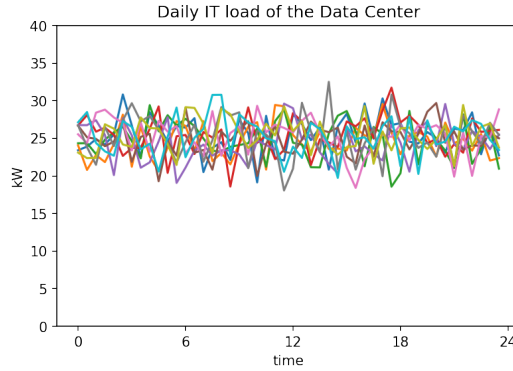


Figure 2

2.2 Cooling system

Data centers are operated 24/7 and should be protected from overheating to run efficiently and to avoid damaging processors. Your IT room is equipped with a cooling system absorbing air at about 25°C , refreshing it with water and re-introducing it at about 18°C

The heat rejected by the IT room equals the IT load $h_{IT}(t) = l_{IT}(t)$. The cooling system is activated at power cost $l_{CS}(t)$ characterized by the EER (Energy Efficiency Ratio) of the system and results in rejecting excess thermal power $h_r(t)$ characterized by a COP (Coefficient Of Efficiency) as expressed in equation 1

$$\begin{aligned} l_{CS}(t) &= \frac{h_{IT}(t)}{\Delta t \times EER} \\ h_r(t) &= COP \times l_{CS}(t) \times \Delta t \end{aligned} \tag{1}$$

For the cooling system we take $EER = 4$ and deduce the $COP = EER + 1$. The rejected power is under the form of air or water at temperature 35°C . In many applications, the power is typically thrown to nature. We assume that the cooling system is always working to ensure a safe usage of the data center. Consequently, the total non flexible electric load of the data center is

$$l_{NF}(t) = l_{IT}(t) + l_{CS}(t)$$

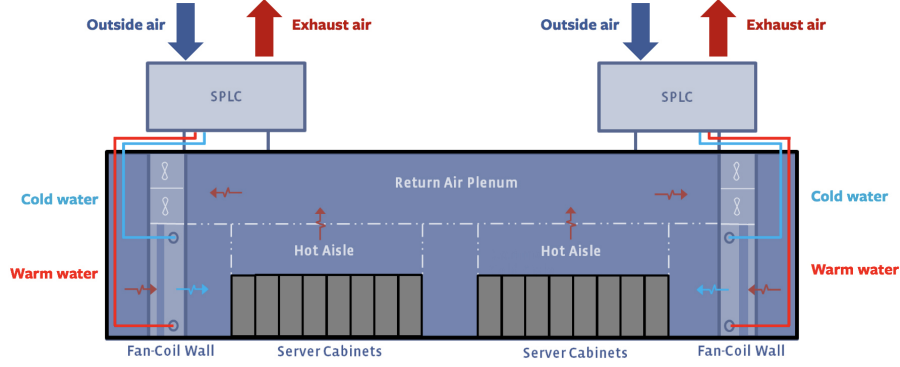


Figure 3: Air-air cooling unit for data centers. In this figure h_{IT} is the air rejected from the hot aisle and h_r corresponds to the exhaust air. In our case we will consider a water-air cooling system where the air exchanged on the roof is replaced with water. Figure credit from [?]

2.3 Heat cogeneration and storage

It is possible to valuate the rejected thermal power $h_r(t)$ in a heat network. First $h_r(t)$ must be treated by a heat pump to transform water at entry temperature $T_r = 35^\circ\text{C}$ into water at a comfort temperature $T_{com} = 60^\circ\text{C}$ that is suitable for domestic usages. Such transformation is operated at electric power cost $l_{HP}(t)$ and the thermal equations ruling the transformation are identical to the one of the cooling system resulting in a valuable thermal power $h_{DC}(t)$. The COP of this transformation is

$$COP = \frac{T_{com}}{T_{com} - T_r} \times e \quad (2)$$

- $e = 0.4$ thermal efficiency
- Note that temperatures must be expressed in Kelvin

The activation of the heat pump is **variable** and accounts for the flexible part of the electric load:

$$l_F(t) = l_{HP}(t)$$

The output of the pump $h_{DC}(t)$ is hot water at temperature 60°C which is suitable for domestic usage. It can be sold to your neighboring smart building at price $p_h(t)$ (€/kWh). The network allows a maximum heat transfer value of 10 kWh every half hour. Values $(p_h(t), h_{DC}(t))$ are computed at each stage as a supply and demand equilibrium (Figure 4). Heat transactions occur when such equilibria are possible. Supply curves $(p_{DC,t}(h))_{h=0..10}$ are subject to your decision.

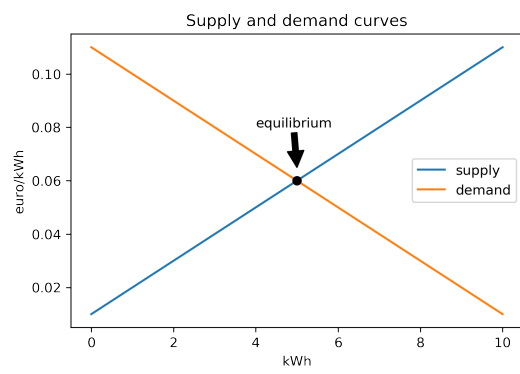


Figure 4

2 Modeling of a Charging Station

2.1 Electric load

We consider an urban charging station for domestic Electric Vehicles (EV) which can host 4 vehicles. The typical behavior of the vehicle owners is that they leave in the morning for work with average departure time of 7AM and come back at the end of the day around 6:30PM ("commuting" pattern). More precisely, both variables follow the respective gaussian distributions: $\mathcal{N}(7, 1)$ for departure and $\mathcal{N}(18.5, 1.5)$ for arrival. Arrivals and departures will induce the constraints of your charging problem (an EV can only charge when connected to the grid) and shape the daily load profile.

In an urban context, we consider vehicles such as the Renault ZOE Z.E. 40 which have a battery capacity of about 40 kWh with an autonomy of 300 km without charging. For urban usage in Europe electric vehicles typically drive 30 km/day. Thus, we will require that vehicles are charged up to 25% of their capacity at departure time (at 00:00 we assume that all batteries are empty). In order to reach that goal, you have to choose a policy for the load of each EV: $l_k(t)$, which are your **variables**. As a consequence, you will also obtain the load of the station $l_4(t)$ (which is also **variable**). If batteries are not charged with at least 25% of their capacity at departure, you will be fined 5 € per vehicle which did not meet the target. We will further assume that 2 of them are equipped with basic charging technology limiting their load at $p_{normal}^{max} = 3\text{kW}$ while the 2 others have more recent systems with $p_{fast}^{max} = 22\text{kW}$. Also, the aggregate charging power must not exceed the station power capacity, which is 40kW. In turn, the power limit of the set of EVs $p_{tot}(t)$ will vary over time according to vehicles arrivals and departures (see Figure 2).

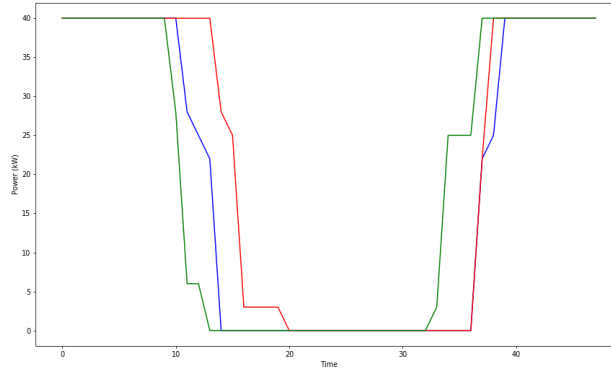


Figure 2: Maximal admissible charging power at the station as a function of time (3 scenarios). *It depends both on the presence of vehicles and on the station power capacity.*

2.2 Vehicle to Grid

When vehicles are parked at the station, you have the opportunity to take part in the Vehicle to Grid (V2G) operation i.e. discharge EV battery to the grid (or consume with a negative power $l_k(t) < 0$). This must be done under the constraint of a stock (of energy), i.e. the State-Of-Charge (SOC) of EV k , denoted by $a_k(t)$, must be between 0 and its capacity 40kWh at each instant t . Naturally, this stock can be managed only when the EV is connected to the grid (before departure and after arrival). Finally, observe that the SOC of EV battery k at the time of arrival t_k^{arr} is obtained based on the SOC at departure and on the amount of energy consumed when driving, assumed to be 4kWh (corresponding to a daily 30km trip): $a_k(t_k^{\text{arr}}) = a_k(t_k^{\text{dep}}) - 4$. Mathematically, this writes:

$$\begin{aligned} a_k(t+1) &= a_k(t) + \left[\rho_c l_{k,v2g}^+(t) - \frac{1}{\rho_d} l_{k,v2g}^-(t) \right] \times \Delta t \text{ when } k \text{ is connected to the grid} \\ a_k(0) &= 0 \\ a_k(t_k^{\text{arr}}) &= a_k(t_k^{\text{dep}}) - 4 \\ 0 \leq a_k(t) &\leq 40 \end{aligned} \tag{1}$$

with

- $\rho_c = 0.95$ charge efficiency
- $\rho_d = 0.95$ discharge efficiency
- $x^+ = \max(0, x)$, $x^- = \max(0, -x)$

and where the **variable is** $l_{k,v2g}$ for $k = 1, \dots, 4$.

Then, the amount of electric power exchanged between the vehicles and the grid $l_{v2g}(t) = \sum_{k=1}^4 l_{k,v2g}(t)$ is obtained. It is still subject to the charging station power capacity constraint:

$$|l_{v2g}(t)| \leq 40. \tag{2}$$

2 Modeling of a Solar Farm

2.1 Photovoltaic production

The electric load of a Solar Farm (SF) depends on its photovoltaic production over the interval $[t - 1, t]$,

$$l_{pv}(t), t \in \{1, \dots, T\}. \quad (1)$$

Examples of daily power production profiles are given on Figure 2. Observe that the power production is subject to randomness. Thus, l_{pv} cannot be controlled and is a **non-flexible** variable. Besides, we assume that the SF is equipped with a battery unit. Power may be charged or discharged according to the load

$$l_{bat}(t), t \in \{0, \dots, T - 1\}, \quad (2)$$

accounting for the power transferred to the battery over the time span $[t, t + 1]$. This load may be controlled to fulfill your objectives, and stands for a **decision** variable. The total load of the SF is then

$$l_i(t + 1) = l_{bat}(t) - l_{pv}(t + 1), \forall t \in \{0, \dots, T - 1\}. \quad (3)$$

In the case of the Solar Farm, power exchanges are mainly directed in order to sell power to other players or to the electric grid. You may also buy energy to store it and sell it later.

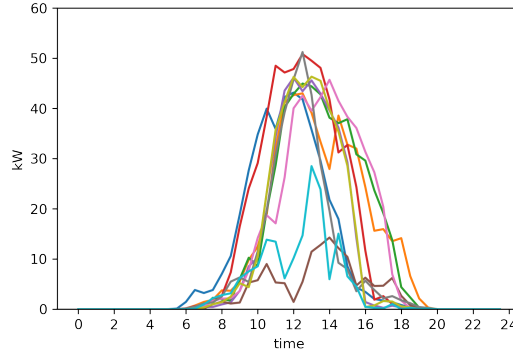


Figure 2: Examples of daily photovoltaic production profiles. These profiles may not be representative of your site, data samples are available in the **data/** folder.

2.2 Model of the battery

Your Solar Farm is equipped with a battery which enables you to store energy. The purpose of the battery is to adapt your load l_i to the demand of other players and to the prices that vary along the day. At time t , the available energy of the battery is

$$a(t), t \in \{0, \dots, T\}, \quad (4)$$

expressed in kWh. The battery stock evolves according to the load l_{bat} , and is ruled by the following dynamic equation:

$$\begin{cases} a(t+1) = a(t) + [\rho_c l_{\text{bat}}^+(t) - \frac{1}{\rho_d} l_{\text{bat}}^-(t)] \times \Delta t, \quad \forall t \in \{0, \dots, T-1\}, \\ a(0) = 0, \\ 0 \leq a(t) \leq c, \quad \forall t \in \{0, \dots, T\}, \\ |l_{\text{bat}}(t)| \leq p_i^{\max}, \quad \forall t \in \{0, \dots, T-1\}, \end{cases} \quad (5)$$

where

- $\rho_c = 0.95$ charge efficiency,
- $\rho_d = 0.95$ discharge efficiency,
- c capacity of the battery (kWh), see **data/** folder,
- p_i^{\max} maximum power of the battery (kW), see **data/** folder,
- $x^+ = \max(0, x)$, $x^- = \max(0, -x)$.

At the end of the day, the energy stock left in the battery is valued at the Smart Grid internal electricity price.



Figure 3: Example of battery with 9 modules. Image credit from [?]