A Distributed Game Theoretic Approach for Optimal Battery Use in an Energy Community

Kirsch Nicolas, Energy Science and Technology

Master Thesis
Electrical and Electronic Section

Lausanne, academic year 2023–2024
Submitted on: 02.02.2024

Supervised by:

Dr Tomasz. Gorecki
Giulio Salizzoni
Prof. Maryam Kamgarpour

sycamore lab
SYSTEMS CONTROL AND MULTIAGENT OPTIMIZATION RESEARCH

csem

Abstract

With the recent rise of decentralized energy resources like solar panels or batteries, energy communities (EC) have grown in popularity. Within these communities, several members manage and share local energy resources with the objective of minimizing the amount of energy sourced from the grid. However, without a collaborative management of loads, some of the locally produced electricity could be unused and sold back to the grid, hindering the economic and environmental benefits of ECs. A possible way to address this would be to manage the loads of the community centrally, but this could present computational and privacy bottleneck. In this thesis, I will present a distributed algorithm based on game theory which provides optimal battery use to all the members of an EC so that the consumption of local resources is maximized. The first part of the thesis will describe the algorithm designed to solve such distributed optimization problems. A key component of this work is the transition from a standard centralized game theoretic resolution approach to a distributed one. Then, its integration in a receding horizon control framework will be showcased. The performance of this approach is compared to other EC management methods, and it is shown that it matches the performance of a centralized optimizer and performs better than an EC with completely individual energy management systems. The algorithms robustness to several imperfect solar forecasting scenarios is also evaluated, with minimal uncertainty induced increases in cost.

Acknowledgement

The accomplishment of this masters thesis would not have been possible without the continuous and precious support of my supervisors Dr Tomasz Gorecki and Giulio Salizzoni. Their comments and recommendations were always relevant and useful for the project. For this, I sincerely thank them.

Tomasz, thank you for your onboarding and for sharing your expertise in real life applications of energy system control. I really enjoyed working under your supervision. I always knew that if I had any doubt I could ask you and receive a clear explanation.

Giulio, thank you for sharing with me your sound and precise theoretical knowledge.Your supervision was crucial in navigating and applying game theoretic concepts, and without your guidance, writing the theoretical proofs required would have been a much more challenging task. I hope that the work we have done together will be useful for the rest of your thesis.

I would also like to thank Prof Maryam Kamgarpour for welcoming me in the Sycamore lab to work on this project. Introducing me to this subject allowed me to delve deeper on topics which have greatly interested and challenged me.

A warm appreciation goes to Team 567 of the Centre Suisse d'Electronique et de Microtechnique for hosting me during the four months of my Master's thesis. The insights gained from observing your team's work and the informal exchanges were invaluable for the project and have significantly contributed to my personal and professional growth. I am grateful for the experience, which has helped shape my path towards a career in research.

Contents

List of Figures

1 Introduction

To achieve a successful and long-lasting energy transition, significant efforts are necessary to transform how energy resources are being utilized. The traditional view of energy systems with a unilateral power flow from large scale power plants towards end-user consumption cannot grant the flexibility required to account for new renewable sources [1].

End-users need to be active in future energy systems, rather than being passive and fixed loads needing to be accommodated for. In that direction, the development of distributed energy resources is required. Distributed energy resources (DER) are smaller generation units that are located on the consumer's side. Residential solar panels or energy storage systems are examples of DER. They are particularly interesting because they allow for a better control of the individual energy demand, and thus can be useful for demand response programs reducing the stress on the grid [2]. Thanks to DER, traditional consumers can become prosumers, which means they still demand energy, but can also generate some too. These DER have therefore enabled a shift in paradigm when looking at energy systems, with flexibility and decentralisation as core components.

Combined with DER technologies, the creation of energy communities (EC) is particularly relevant and has been gaining momentum worldwide [3]. An energy community is a localized group of individuals, businesses, or organizations that collaboratively generate, manage, and share energy resources, often incorporating renewable energy technologies and decentralized energy production. It can either be a group of consumers sharing a common DER (a building with rented apartments and a PV installation on the roof) or a group of independent prosumers. In an energy community, when the demand of one of the users is greater than its production, it can directly use the energy generated in excess by another prosumer. That way, the amount of energy exchanged with the grid is significantly reduced, potentially smoothing peaks in demand during the day.

EC are often constructed around a community manager, whose role is to organise and overview the different physical and economical flows to and from the members of the EC. Figure 1 represents these different flows for a community. Supposing a horizon $T \in \mathbb{R}$, the community managers receives the solar load $s \in \mathbb{R}^T$ from the central solar installation and dispatches it to the community members. For a community with N members, each member i receives a share of this local resource $p_{local,i} \in \mathbb{R}^T$, buying it at a cost of $c_{local} \in \mathbb{R}$. This corresponds to the annualised investment cost and maintenance costs of the PV installation. If this energy is not sufficient to meet the members' demand at each timestep, they will also consume electricity from the grid, $p_{import,i} \in \mathbb{R}^T$ at a price $c_{import} \in \mathbb{R}$. The sum of all these grid import corresponds to the imports of the community $p_{import} \in \mathbb{R}^T$. If on the other hand there is too much solar production at some point compared to the demand, this excess load p_{excess} will be sold back to the grid at a price $c_{feed-in} \in \mathbb{R}$. The local cost c_{local} is often designed such that it is greater than $c_{feed-in}$ and smaller than c_{import}. That way both consuming and selling within the community is encouraged.

In the optic of optimizing energy consumption within an EC, two types of loads are introduced for each member: $e_i \in \mathbb{R}^T$ is the initial default load, and $x_i \in \mathbb{R}^T$ is the load after optimization

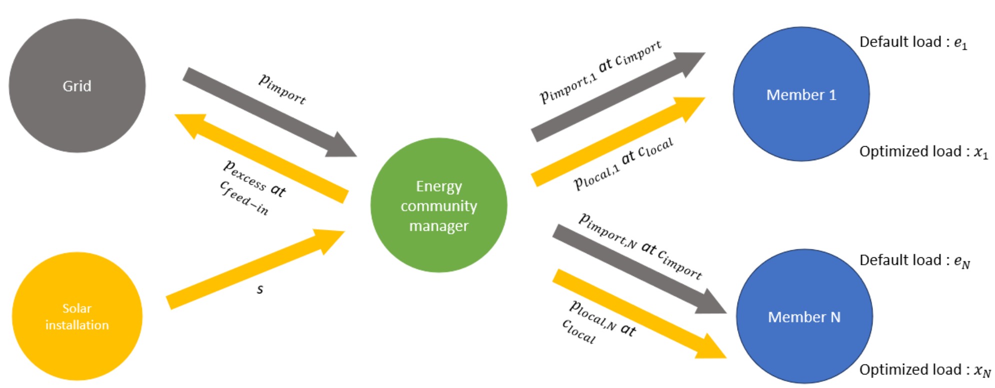


Figure 1: *Operation of a one building energy community with a solar installation for renters*

Current energy communities application are still relatively simple. Still, as there are several energy sources, controllers are needed to operate the energy systems optimally. These controllers can for example plan the optimal activation time of shiftable loads. For prosumer communities, the energy control is often done at a household level, using home energy management system (HEMS). These systems are successful in optimizing individual consumption, but they can lead to poor global performance within the energy communities, because of a lack of coordination between prosumers. If the HEMS do not account for the behavior of all the prosumers in the community, all the optimal loads could overlap, causing a suboptimal global load [4]. For example, it would make sense for a HEMS to plan the shiftable loads to be activated at peak PV production. However, if all prosumers adopt this behavior, the total energy demand at that time could be significantly greater than the available solar energy, resulting in a bigger need for grid electricity. On the other hand, there could be solar energy available at other timesteps that would not be consumed.

An energy management tool with a holistic view of the system is thus necessary to reach optimal operation [2]. A possible method to do so could be by aggregating all the users in a central optimization problem. However, the limits of a centralized optimization approach become evident in contexts where privacy, fairness and equity are significant concerns within an energy community. A centralized approach indeed often necessitates the sharing of detailed and sensitive data, raising privacy and security issues. Also, a centralised approach can raise computational challenges when the number of participants increases, limiting its efficiency for large scale applications [5]. Finally, it can be complicated to design a single cost function that guarantees an equal distribution of local resources among community members. Overall, these challenges could motivate consumers and prosumers to opt out of energy communities, hindering their benefits.

Given that fully decentralised and fully centralised energy control within communities have both advantages and drawbacks, it would be interesting to adopt a midway approach to manage energy communities, capitalising on the benefits of the two approaches described previously. Ideally, such a methodology could be distributed, but considering information coming from all the community members to ensure optimal planning. That way, it would be possible to to maintain the privacy and to preserve computationally efficient optimization while also avoiding the pitfall of individual control.

The framework described above could be modeled using game theory. Indeed, the optimal load scheduling of a community can be seen as game where each member tries to optimize its own benefits (for example minimize their electricity costs) with respect to the other members action. In these games, each player has its own objective function but it is influenced by the actions of the other players. No player can compute its optimal load without accounting for the actions taken by others, in contrast to fully decentralized optimization.

A game theoretic approach can also provide a more balanced optimal solution between players, ensuring that no member would be tempted to leave the community. Several game theory algorithms also are able to compute the optimum in a decentralized fashion, reducing privacy and computation concerns.[6]. A complete description of game theoretic applications to EC is included in section 2.3

The objective of this thesis is therefore to answer the following questions: **How can distributed game theory concepts be utilized to maximize self consumption and minimize the energy costs for small scale energy communities? How does such approach compares to other methods like centralised or individual control?**

2 Literature review

The section examines the existing literature related to energy community management. It starts by analyzing real-world energy community operations to gain insights into the current state of the field. Subsequently, it delves into more advanced research methodologies. Finally, the section introduces game theory in more detail and explores its practical applications for energy management.

2.1 Energy communities - state of the art

Although it is not a new concept, ECs have been growing steadily in recent years. It is especially true in Europe, where nearly 4000 citizen-led energy initiatives were recorded in 2023 [7]. This has been fostered by the transposition of the EU directives [[8],[9]] to national laws between 2020 and 2022 [7]. These energy communities are very diverse, sharing different primary energy depending on their geographical context (hydropower, geothermal installations...). They however all have the same central idea of sharing renewable energy sources within the community before trading it back with the grid.

In Switzerland, a solar energy community is called a “Regroupement de Consommation Propre” (RCP). It can regroup one or multiple buildings and DER, but it has only one connection point to the main grid. For the electrical distribution system operator, a RCP is seen as single consumer [10].

In most cases, current ECs do not have a management system acting on the physical flows. The management is rather done ex-post and acts on the electricity tariffs for the different members of the community. To do so, the energy community manager has access to consumption readings for each members and can therefore calculate how to invoice all members [11].

On top of the consumption readings, the community manager needs to know the price at which the local energy is billed, c_{local}. In Switzerland, there are two options to set the local energy price:

- Fixed estimation: The local tariff is set as a specific percentage of what the member would pay if it was not in the EC [10]. In the case of tenants, it can be at most 80% of the cost of electricity on the grid. For example, $p_{local} = 0.8 * p_{grid}$
- Effective cost: The local tariff is computed based on the actual costs related to the solar installation. It needs to account for the annualised investment costs, the operation and maintenance costs, and the service costs. The electricity sold back to the grid also influences the local cost [10]

The second method ensures a more precise estimation of the internal price and often yields stronger savings for consumers. It is however more complicated to calculate and relies on some uncertain estimations. In many cases, the first method is preferred.

Using the calculated local cost and the grid cost for electricity, the community manager can calculate the invoices for each members. This is a complex task because the energy price should represent as faithfully as possible the consumption behavior of the different members. Otherwise, members could feel wronged and leave the community. Several methodologies are currently used :

- **Bill sharing**: The first, most simple approach is to aggregate the total amount of energy consumed by a community over a horizon and to calculate how much it costs [10]. To calculate the total cost, the total amount of local energy is multiplied by the cost of local energy and the amount imported from the grid for the entire community is multiplied by the grid price. Using this, an average cost of electricity per kWh can be calculated. Multiplying it by individual consumption yields the amounts every member has to pay. The following equation describes the calculation to get $C_i \in \mathbb{R}$, the total cost for member i. $p_{i,tot} \in \mathbb{R}$ is the amount of energy consumed by member i, $p_{import,tot} \in \mathbb{R}$ is the total amount of energy imported from the grid, and $p_{local,tot} \in \mathbb{R}$ is the total amount of local energy consumed locally:

$$C_i = \frac{c_{import} * p_{import,tot} + c_{local} * p_{local,tot}}{p_{import,tot} + p_{local,tot}} * p_{i,tot}. \tag{1}$$

- **Time of use**: This method is similar to the previous one, but the time periods are much more granular (every 15 minutes in many applications). In this way, it is possible to capture whether the members consume during local generation time and give them a lower price per kWh if it is the case. The average price per kWh is here calculated at each timestep, and varies depending on the amount of local energy used at that time. The equation below shows the calculation to compute C_i for a member i, using the variables introduced with figure (1) and with $p_{import} \in \mathbb{R}^T$, the total local consumption of the community:

$$C_i = \sum_{t=1}^{T} \underbrace{\frac{c_{import} * p^t_{import} + c_{local} * p^t_{local}}{p^t_{import} + p^t_{local}}}_{\text{ToU tariff}} * x^t_i. \tag{2}$$

The time of use method allows for a more faithful distribution of energy costs as it can account for when each user consumes, while the bill sharing method only considers how much each user consumes. However, time of use requires more complex metering system to ensure the granularity of the readings. Still several companies like Climkit [12] or NeoVac [13] already offers such solutions for ECs.

In Europe, there are also some pioneering approaches based on peer-to-peer (P2P) energy trading. In such a system, EC members can buy and sell energy directly from other members depending on their needs [14]. The local trading price can be defined in several ways and can be varying through time, and the different members can then trade energy within the community as it would be done on a larger market. These exchanges are either directly between community members or through an organising entity which ensures that all trading goes smoothly. Entrnce is an example of a company offering trading services within a community [15]. With it, members can trade internally and with the grid to ensure redundancy in case of low local production.

Overall P2P energy trading is quite new, but it can be an interesting way of managing ECs, as it can act directly on the power flows, as opposed to the ex-post methods presented above.

To summarise the state of the art, most real-life applications of energy community management relies on simple ex-post methods, only influencing the energy cost for the different members. However, there are some innovative P2P based projects which are disrupting this by enabling a direct influence on the way energy is used within the ECs.

2.2 Research approaches for managing energy dispatch

Given the complexity of optimally managing energy in a community, this topic has been an active area of academic research. Numerous research projects have emerged with the aim of improving upon the methodologies discussed in section 2.1. This section provides a comprehensive exploration of the various methodologies documented in academic literature, shedding light on the innovative approaches and insights that have been developed to address the challenges posed by EC management.

2.2.1 Ex-post methods

The simplest approaches in the literature manage the energy communities by updating the prices each members pay after the energy flows have taken place. These methodologies enables a post-event breakdown of the total cost between the community members.

Locally generated energy can sometimes be considered as traded freely within the community and the only cost is the cost of imported electricity. For example, with the bill sharing presented in [16] each members gets the same cents per kWh tariff based on the percentage of consumption covered by imports for the entire community. The purchasing price for all member would be $c_{buy} = c_{grid} * \frac{E_{import}}{E_{total}}$ with $c_{buy} \in \mathbb{R}$ the local tariff, $c_{grid} \in \mathbb{R}$ the standard grid price, $E_{import} \in \mathbb{R}$ the amount of electricity imported from the grid and $E_{total} \in \mathbb{R}$ the total consumption of the energy community. This is similar to what is currently done in real-life applications. In [17] a percentage of the total cost for the community is assigned to each member based on their contribution to the local generation effort. This is done by comparing the optimal cost with and without them. That way, the authors argue the cost repartition is the fairest, because those who are most useful in providing local energy will have to pay the least. [18] finds the allocation of locally generated energy to different consumers minimizing the total costs of the EC. Each member is then billed what he would need to import from the grid given this new allocation of local resources. To avoid unfair solutions (a member with no solar energy for example) the optimization problem is constrained so that each member has a minimal amount of self-sufficiency rate (ratio of personal local energy consumption over total personal energy consumption)

In some other cases, peer to peer trading can be designed to allow community members to buy and sell their solar energy. Still described in [16], the mixed market rate approach fixes a unique price for buying and selling inside the community. For it to be interesting for all, it is designed as the average between the cost of electricity from the grid and the price at which the grid buys back excess PV production. Thus, if at time t one member has excess production and another is in deficit, the energy can be traded internally, making the seller earn more and the buyer save money.

The advantage of ex-post methods is that they are relatively easy to implement and they can be successful in redistributing the benefits generated by regrouping prosumers in energy communities. Thanks to these, it is possible to ensure satisfactory economic performance within the energy community, ensuring no members would want to leave. However, given that they do not influence the energy flows, they do not grant an optimal use of local resources. This limitation has been addressed by the papers in the following sections, which provides the optimal loads in advance to fully utilize the DERs.

2.2.2 Optimal planning methods

To improve on the performances of ex-post methods several papers look into day-ahead or real-time optimization. These methods would require accurate forecasts to work, but would ensure that power flows are as close to optimum as possible. The objective is to anticipate the best consumption patterns for all members of the EC to maximize local resource consumption and therefore minimize the energy costs.

Doing so is possible by acting on the energy price through the day. By creating preferential tariffs periods, it is possible to motivate shifts in consumption habits. With DERs, these preferential periods can be matched with periods of high local generation. [19] for example minimizes EC energy cost by jointly acting on local energy cost and prosumer load profiles. The paper looks at an energy community where the only contact with the main grid is an Energy Sharing provider (ESP). Each prosumer buy and sell energy only to and from this ESP, with no interactions between prosumers. The time dependent prices at which energy is sold and bought is calculated using the ratio between internal energy generation and demand at each timesteps. The selling price is proportional to this ratio while the buying price is inversely proportional, ensuring that the more local energy is produced, the more people will want to consume at that time. By iteratively updating each prosumer net loads and the prices, this approach converges to an optimal load dispatch, reducing costs for all members of the community.

Rather than using the energy price as a catalyst for prosumer consumption modification, some other approaches seek to modify loads directly. This modification can take several forms, either by installing energy storage to decouple consumption and demand, or by modifying the time of use of certain appliances, such as dishwashers or washing machines. The solution for these papers is to optimize the control of these appliances to minimize the energy consumption.

Overall, the methodology is fairly similar between the different papers. First, the energy communities are modeled. This includes modeling each prosumer's demand, the solar generation and the economic and physical flows between all the agents in the model. A global optimization problem is then formulated, the aim of which is to minimize the total cost of electricity for the community. The decision variables are the batteries and shiftable loads of each prosumer. This often enables day-ahead planning. A rule-based control layer is then used in real time to compensate for errors and inaccuracies.

The differences then lie mainly in the optimization algorithm. [20] uses Mixed Integer Non-linear Programming MINLP to optimize, [21] uses constrained nonlinear programming CNLP. Other papers also use metaheuristic optimization approaches. [22] optimizes using greedy randomized adaptive search procedure and [23] uses Particle Swarm Optimization with adaptive weights.

Some of the studied papers had slightly different approaches which stood out from the standard load based energy community optimization methods. [24] for example solves this problem with a hierarchical approach. The lowest level is a fully decentralised optimization: each consumer finds its control minimizing its own objective function. Afterwards, the outcomes are improved through the implementation of a peer-to-peer energy sharing algorithm. This algorithm selects pairs of households that can achieve more significant reductions in operational expenses by collaborating in an optimized process. That way, the paper argues that the consumers benefit from functionning at a community level, while avoiding the pitfalls of fully centralised control.

In [25], the objective is to avoid having to generate a complex model to find the optimal planning of shiftable loads in a peer to peer energy trading context. Instead, a fuzzy Q learning artificial intelligence algorithm is used to train a lookup table in advance. That way, when in operation, to find the best action, the algorithm only has to search for the optimal response (buy energy, sell energy, charge the battery, discharge the battery) given a set of current conditions (grid price, local price, state of charge). That way, the author argues that the approach is useful and scalable, because it can be used in real-time with no tedious modeling of the community.

Lastly, [26] focuses on uncertainty management. Using long term forecasting, many scenarios are generated for renewable generation and loads. Optimal day ahead plans are then generated for each scenario using improved particle swarm optimization. All these scenarios are then joint and adjusted using short term forecasting, which is more precise. Finally, real time rule based control mitigate the final differences between forecasts and reality.

The paper's focus on uncertainty management is very important, and including some considerations about is key for this work. Indeed, control and optimization methods are based on a lot of forecasting information: solar potential forecasting, prosumer typical load forecasting and so on. The previous papers are based on deterministic approaches, which are successful but will have trouble bridging the gap with real-life applications. Taking these uncertainties into account makes for more robust algorithms.

2.3 Game theory

The literature also focuses on formulating the problem of EC management as a game. To better understand these papers, it's important to take a closer look at the basic concepts of game theory.

In game theory, a game consists of a set of players interacting according to given rules. All the players are interdependent and their interactions will have an impact on each other and on the entire game. A general definition can be given as:

> *The subject of game theory are multi-agents decision making situations, where the result for a player does not only depend on his own decisions, but also on the behaviour of the other players.*[27]

A game is defined by the following elements:

- **The players**
- **The strategies**: these are the actions each players can chose. They can either be discrete (buy or sell an asset) or continuous (buy a specific amount of an asset).
- **The objective function**: these are objectives each players seeks to optimize. All objective functions are influenced by strategy choices of other players.
- **The information structure**: what does each player know about the strategies of other when making a decision.

There exists cooperative and non-cooperative games. In cooperative games the different players can exchange information, negotiate and form coalition to maximize a shared objective. In non-cooperative games on the other hands each player can be seen as adversary. They each aim at optimizing individual objective functions, by acting rationally and taking into account the responses expected from other players.

Let Γ be a non cooperative game defined for the players set $\mathcal{N} = \{1, \ldots, N\}$. For each player i, let $K_i \subset \mathbb{R}^n$ be his set of acceptable strategies, $x_i \in K_i$. The vector $x = x_1 \times x_2 \times \ldots \times x_N \in K$, with $K = K_1 \times K_2 \times \ldots \times K_N \subset \mathbb{R}^{nN}$ represents the decision of all the players in the game. The vector $x_{-i} = x_1 \times \ldots \times x_{i-1} \times x_{i+1} \times \ldots \times x_N$ is the strategies of all players but player i and belongs to the set $K_{-i} = K_1 \times \ldots \times K_{i-1} \times K_{i+1} \times \ldots \times K_N \subset \mathbb{R}^{n(N-1)}$.

The objective function of each player i is a function of their own strategy x_i and the strategy of all the other players x_{-i}. This function can be defined as $J_i(x_i, x_{-i}) : \mathbb{R}^{nN} \to \mathbb{R}$. Each player try to find the x_i that minimizes $J_i(x_i, x_{-i})$.

The solution of a non-cooperative game is called **Nash Equilibrium**. It corresponds to a state where each player has chosen a strategy and no one can improve its objective function by unilaterally modifying their strategy. Mathematically, a point x^* is a Nash Equilibrium if:

$$J_i(x_i^*, x_{-i}^*) \leq J_i(x_i, x_{-i}^*), \quad \forall x_i \in K_i, \quad \forall i \in \mathcal{N}. \tag{3}$$

For cooperative games, there is no explicit and unique concept like Nash equilibrium to define a game solution. As information can be shared between players, finding the optimum is often twofold. First the optimal coalitions within the players are defined. This means that different combinations of players between which they can exchange information are trialled and the one the coalitions set with the best outcome is kept. It is often shown that the optimal coalition is the grand coalition, namely the one including all players, The benefits are then allocated within coalitions members so that each has an incentive to stay in the community.

It is therefore clear that the EC problem translates particularly well into a game theory problem. Each member of the community can be a player, with their strategies being their load profiles and their objective function the price they pay for electricity. Each member wants to find the optimal load profile strategy, but need to account for the actions of others to be successful (by accounting for their influence on energy price for example).

For cooperative games applications, [28] looks at the optimal scheduling of multiple microgrids by solving a central MINLP problem and allocating the costs using the Shapley value, which is a metric of player contribution to coalitions. [29] and [30] both attempt to optimally control distributed flexible loads and battery storage by forming coalitions, and use the concept of nucleolus to allocate coalition benefits between the members. [31] presents a method of allocating costs which can be used in real-time, addressing the major drawback of cooperative games, which have computation times growing exponentially with number of players.

Several non-cooperative games based methods are also described in the literature. [32] for example formulates the problem of micro-storage use in large scale energy systems as a non cooperative game. In this game each agent can modify his battery use to attempt and minimize his energy costs. The set of consumers is sufficiently large that the total demand at a time influences the price at that time. All the player influence each other because of this fact. The small size of individuals compared to the entire set also helps them in finding the Nash Equilbrium, as they prove that because of it, the search for NE can be simplified to the resolution of one central optimization problem. They apply this approach to the grid of the United Kingdom and show that this game theoretic induced collaboration allows for peak shaving and cost reductions.

This framework can also be complexified to better capture prosumer behaviors. In [33] there is indeed a combination of two games. First, the set of prosumers is divided between buyers and sellers at each timestep, based on whether they have surplus or deficit production with respect to their needs. Then, a game is created between buyers, and another game is created between sellers. In the buyers' game, the strategies are to decide which sellers to buy electricity from. In the sellers' game the goal is to maximize their profits by acting on the price at which they sell energy. It is an iterative approach, where at each iteration, each game is updated by taking into account the optimal strategies of the other game, until the strategies are no longer modified. By combining this framework with demand response measurements, the authors show that energy savings are far more interesting than with the ex-post methods presented in section 2.2.1

2.3.1 Decentralised game theory

To take full advantage of the game-theoretic framework for EC management, the Nash Equilibrium should be computed in a decentralized or distributed way. As mentioned above, this reduces privacy risks and computation time problems, as the optimization is performed simultaneously for all consumers. Given this information sharing structure, decentralized games are rather non-cooperative than cooperative.

Several algorithms can be used to solve games in a decentralized way. These are iterative algorithms, where each iteration comprises at least one round of individual optimization for each player and one round of interaction between players. The following are those most frequently mentioned in the literature:

- **Mean Field Games**: Unlike centralized resolutions, where each player acts according to the strategies of all the other players, here the players react to a statistical aggregation of the strategies. Each player solves an individual optimization problem, where the impact of other players' strategies is encapsulated in a shared parameter z. Optimization problems are therefore simpler and can be solved independently by each player. Often, in addition to the players, there is an aggregator whose role is to retrieve the results of individual optimizations, update the shared parameter with a learning rate η accordingly and finally broadcast it to all players. It has a purely "virtual" role and does not act on physical flows. The diagram below shows an iteration of the mean field algorithm:

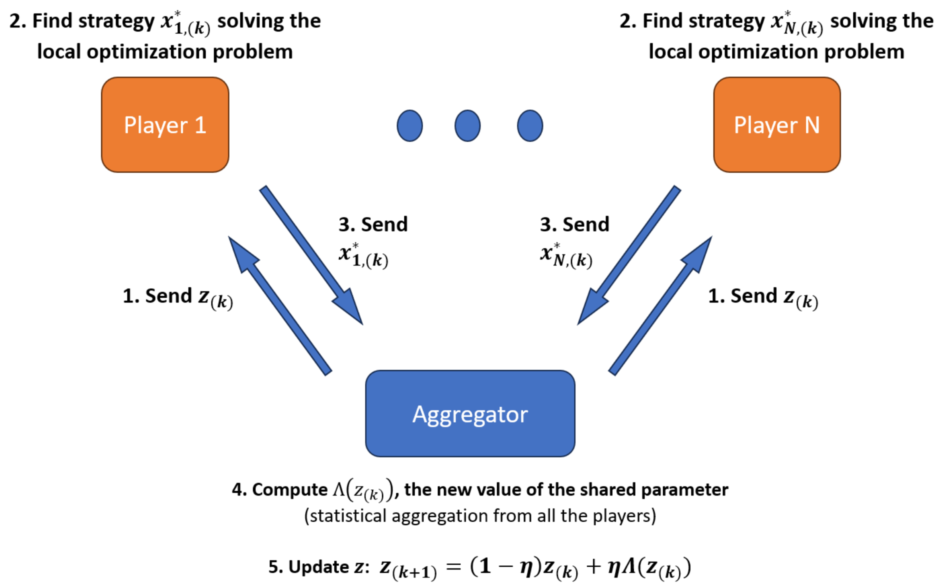


Figure 2: *One iteration of a mean-field game optimization algorithm*

In the literature, several papers present an application of mean field games to Energy Communities. In [5],[6] [34], the shared value is the price of electricity to be purchased from the grid. Thus, during an iteration, all consumers optimize individually with the version of the price vector from the last iteration. The optimal loads are then sent to the aggregator, which calculates the new value of the electricity price based on the total community consumption. The aggregator then updates it, using a learning rate to ensure convergence with many players. In [6], it is shown that the Mean field approach is also robust to PV and consumption forecasting uncertainties.

In [35] and [36], the shared parameter z is the mean consumption load of all the players over the horizon. This average load is used to compute the expected price at each timestep, which will be used by each players to plan accordingly. This method is applied in both cases to the control of plug-in electric vehicles. Both papers once again assume that the number of players is sufficiently large to influence the cost of electricity. The example used in both papers is indeed the control of a fleet of 10^7 electric vehicles.

On the whole, these methods are successful with many players [34]. The less impact each individual has on total consumption, the more the statistical aggregation allows convergence towards an optimal solution. Current real life ECs are however rarely composed of enough members for mean field games to be useful in practice.

- **Stackelberg Games**: In this type of game, there is an active aggregator who influences the system. It is said to be the leader of the game, and the consumers/prosumers are the followers. Unlike mean field games, here the aggregator does not broadcast a statistical aggregation of all the players' actions, but uses the information from all the players to perform an optimization of its own on an element it influences. It then sends the updated values of this parameter to all players until convergence. In such games the solution is called a Stackelberg equilibrium, and it is a point where neither the players nor the aggregator can improve the situation unilaterally.

An iteration of a decentralized Stackelberg game algorithm is shown in the diagram below.

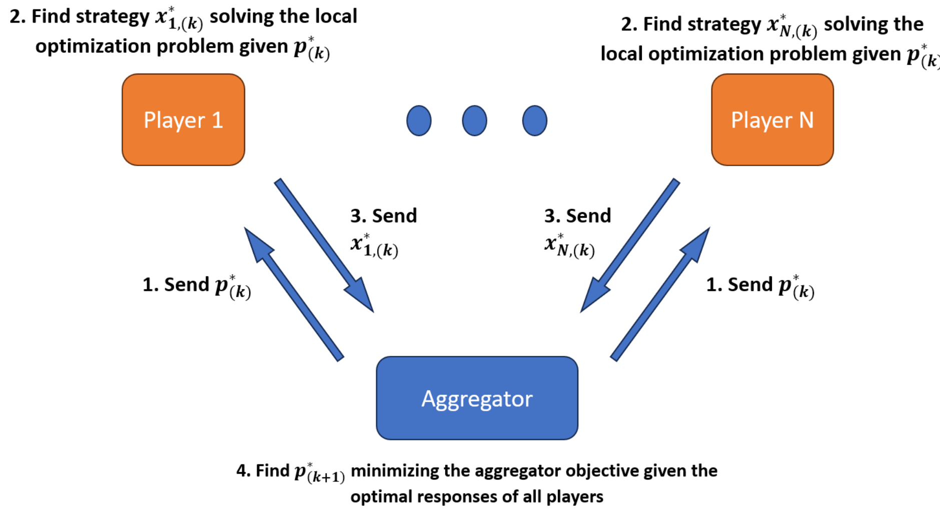


Figure 3: *One iteration of a Stackelberg game optimization algorithm*

In the literature, there are several applications of Stackelberg games for decentralized EC optimization. Overall, the papers have a similar methodology. They generally differ in two areas: the choice of decision variable for aggregator optimization and the type of optimization performed at the aggregator and prosumer levels.

For example, [4] presents a version of a Stackelberg game where the aggregator has a centralized battery to manage flows between the grid and the consumers. Optimization at leader level is therefore based on the scheduling of this battery. This optimal battery planning is then broadcasted to all consumers, along with the sum of the loads of the other cosumers. Each consumers then optimize their shiftable loads and send them back to the aggregator. All these optimization problems are solved using genetic algorithms. Similarly, in [37], the leader is a central storage facility aiming at maximising its profits and the followers are the consumers willing to minimize the cost of external energy supply.

In [38], the objective of the aggregator optimization is to minimize the cost of buying local energy from prosumers by acting on the local price per kWh. This is sent to the players along with the total load of the other players. In turn, each prosumers optimize the amount of energy consumed. [39] presents a two layers Stackelberg game to control clusters of microgrids. In the first layer, the leader is the cluster agent and each microgrid is a follower, and in the second layer, each microgrids are leaders and end users are followers. This seems to show that Stackelberg games can be applied to a great range of population sizes, as they can be scaled up to find optimal solutions for larger consumer groups.

2.3.2 Literature review summary

As seen above, a lot of different papers have addressed the problem of energy community management. Table 1 in the annex summarizes all the methodology differences. Overall, the game theoretic approaches stand out because of their ability to be computed in a distributed setting. However, the existing distributed algorithms are not applicable to small scale energy communities with no active coordinators. Moreover, they have only been tested in day-ahead planning and not implemented with real-time controllers. The following sections will first present a new distributed game theoretic algorithm inspired by the ones existing in the literature. This algorithm will then be use in a receding horizon control framework, underlining its value for real-time energy control of energy communities.

3 Distributed game optimizer

After looking at the current literature on energy dispatch in EC, game theoretic approaches seems like they have a strong potential. Such methods are also compatible with distributed resolution, making it even more valuable given the limitations of centralised resolution.

However, existing versions of these methodologies either necessitate large energy communities to fully leverage mean field games or rely on an active central player when implementing Stackelberg games. Designing a distributed algorithm to optimize loads within a small-scale energy community, with a passive central agent solely responsible for information sharing, appears to be a promising approach with minimal prior exploration.

To design such an algorithm, the first step is formulating the problem at hand as a game. Let this game be called $\Gamma = \{\mathcal{N}, \mathcal{X}, \{J_i\}_{i\in\mathcal{N}}\}$, with $\mathcal{N} = 1, \ldots, N$ the players set. The set $\mathcal{X} \subset \mathbb{R}^{TN}$ is the game strategies set, defining the allowed loads for each members of the community. The objective function J_i for each player i should represent as well as possible the goal of the community members in reality. Building on section 2.1, a straightforward way of defining the objective would be as the cost each player has to pay for electricity consumption under a time of use pricing regime. For player i, the objective function would be:

$$J_i(x_i, x_{-i}) = \sum_{t=1}^{T} \underbrace{\frac{c_{import} * p^t_{import} + c_{local} * p^t_{local}}{p^t_{import} + p^t_{local}}}_{\substack{\text{ToU tariff} \\ \text{Depends on all the players}}} * x^t_i + J_{shifting,i}(x_i), \tag{4}$$

where $J_{shifted,i} : \mathbb{R}^T \to \mathbb{R}$ is the inconvenience cost of having to change consumption load. With this objective function, each player would try to minimize the cost of electricity, which would be influenced by the decisions of all the other community members through the time of use tariff. Indeed, both p_{local} and p_{import} are the total energy consumed locally and from the grid respectively and therefore both depend on the players' strategy as well as the strategies selected by all the other community members.

Even if this objective function definition is the most faithful to real-life applications in energy communities, it might be complex to use for optimization problems. The time of use tariff requires a division by the total consumption of the community at a time step to be computed. Indeed, the total consumption is the sum of imported and local consumption, and can also be seen as the sum of a players consumption and the consumption of all the other players: $p^t_{import} + p^t_{local} = x^T_i + \sum_{j\neq i} x^t_j$. This reformulation underlines that when solving a problem minimizing a time of use objective function, there will be a division by the decision variable. This often makes the resolution complex and should thus be avoided. For this reason, even if a time of use game appeared interesting for its proximity with real EC, another game needs to be defined to be used for the problem at hand.

3.1 Available solar game definition

As mentioned in the previous section, a game based on the Time of Use tariffs is harder to study analytically because of the nonlinear objective functions. Finding an alternative way to formulate the problem at hand without dividing by the decision variable is thus necessary. A possible reformulation of the problem can be, that instead of having the actions of all the players influence the cost of the energy at different time steps, the other players could impact the amount of energy each player could source locally.

Figure 4 represents a possible situation for player i. The orange curve represents the total solar energy available for the entire community, the green curve is the total amount of energy consumed by all the members but member i and the red curve is the amount of local energy consumed by all the other players. They are respectively denoted $s \in \mathbb{R}^T$, $x_{-i} \in \mathbb{R}^T$ and $p_{local,-i} \in \mathbb{R}^T$. In that case, player i would want its local consumption, $p_{local,i} \in \mathbb{R}^T$ to be as close as possible to the light orange area, which is the local production remaining after the consumption of all the other members. Given that local energy would be always cheaper than grid energy, player i would minimize its costs by having a local consumption as close to what is available to him as possible.

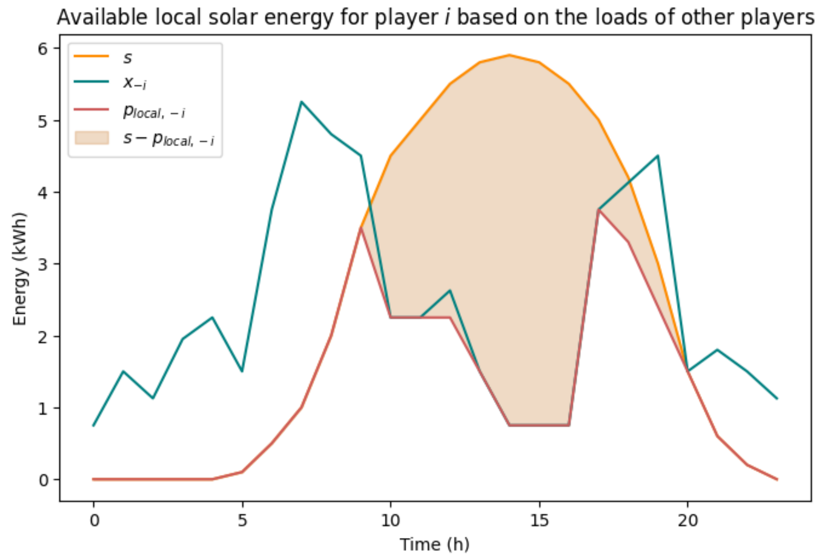


Figure 4: *Available local solar energy for player i based on the loads of other players*

Each player decision is now twofold: they can decide their total load (x_i) and decide how much local energy they consume ($p_{local,i}$). For clarity, $y_i = (x_i, p_{local,i}) \in \mathbb{R}^{2T}$ is introduced to encapsulate the decisions variables of each player.

The strategy sets containing all possible y_i also need to be defined. For all players, the total loads and local consumptions need to satisfy the following constraints:

- Positive local and total consumption: $x_i \geq 0$ and $p_{local,i} \geq 0$.
- Total player consumption has to be greater or equal to local player consumption: $x_i \geq p_{local,i}$.
- Given an initial consumption profile $e_i \in \mathbb{R}^T$, the total optimal consumption has to be equal to the initial one: $x_i^\top \mathbf{1} = e_i^\top \mathbf{1}$.

Given these constraints, the strategy set $\mathcal{Y}_i$ of player i, can be defined as:

$$\mathcal{Y}_i = \{(x_i, p_{local,i}) \in \mathbb{R}^{2T}_{\geq 0} | x_i^\top \mathbf{1} = e_i^\top \mathbf{1} \, , \, p_{local,i} \leq x_i\}. \tag{5}$$

From player i's perspective, the vector $y_{-i} \triangleq (y_1, \ldots, y_{i-1}, y_{i+1}, \ldots, y_N) \in \mathcal{Y}_1 \times \ldots \times \mathcal{Y}_{i-1} \times \mathcal{Y}_{i+1} \times \mathcal{Y}_N \subset \mathbb{R}^{T(N-1)}$ denotes the strategies of all the other players. Defining the set $\mathcal{Y} = \mathcal{Y}_1 \times \mathcal{Y}_2 \times \ldots \mathcal{Y}_N \in \mathbb{R}^{2NT}$, the vector $y = (y_1, y_2, \ldots, y_N) \in \mathcal{Y}$ contains the strategies of all the agents in the game. Note that the constraint $x_i^\top \mathbf{1} = e_i^\top \mathbf{1}$ is used in this project to represent the energy budget constraint. The following work would however still be correct if this constraint was replaced by x_i being required to be in any compact and convex set.

With these strategy sets, we need to design the objective each player would minimize. It should include a term proportional to the scale of the difference between each player's local consumption and the amount of local energy available for it. That way, the community would be motivated to consume more local energy. It is also important for the objective functions to include an evaluation of the inconvenience caused by changing consumption profile. This can be done using the function $J_{shifted,i}$ introduced earlier. The objective function each player tries to minimize is:

$$J_i(y_i, y_{-i}) = (p_{local,i} - (s - \sum_{j \neq i} p_{local,j}))^\top B (p_{local,i} - (s - \sum_{j \neq i} p_{local,j})) + {p_{local,i}}^\top D p_{local,i} + J_{shifted,i}(x_i). \tag{6}$$

Here, $s - \sum_{j \neq i} p_{local,j}$ represents the amount of local energy remaining given the other community members local consumptions. Both $B \in \mathbb{R}^{T \times T}$ and $D \in \mathbb{R}^{T \times T}$ are diagonal scaling matrices. The term including matrix D is necessary for there to be a unique solution. This will be shown in section 3.4.2. The matrices B and D are assumed to be the same for all community members.

The objective function is quadratic so that it penalizes strongly big deviations between local consumption and available resources. It can be rewritten more compactly as:

$$J_i(y_i, y_{-i}) = (\sum_{j=1}^{N} p_{local,j} - s)^\top B(\sum_{j=1}^{N} p_{local,j} - s) + p_{local,i}{}^\top D p_{local,i} + J_{shifted,i}(x_i). \tag{7}$$

It is now possible to have a new formulation of the game. For a players set $\mathcal{N}$, the game Γ is defined as $\Gamma = \{\mathcal{N}, \mathcal{Y}, \{J_i\}_{i \in \mathcal{N}}\}$. In this game, each player wants to solve the following minimization problem:

$$\begin{aligned} y_i^* = \arg\min \quad & J_i(y_i, y_{-i}) \\ \text{s.t.} \quad & y_i \in \mathcal{Y}_i, \end{aligned} \tag{8}$$

where the dependence of the optimal solution on the other players actions is in the objective functions.

3.2 $J_{shifted,i}$ definition

To further analyse the game at hand, an explicit definition of $J_{shifted,i}$ is required. The idea behind this term in the objective function is to represent the inconvenience of having to change the amount of electricity used at a certain time. It can also be seen as a simplistic evaluation of the degradation costs involved in charging and discharging a battery to store energy in periods of high solar power generation and use it later. This cost should therefore be correlated to the scale of change between the initial and the optimized load. To ensure continuous differentiability, it is preferable to formulate $J_{shifted,i}$ as a quadratic function. It is thus defined as:

$$J_{shifted,i}(x_i) = (x_i - e_i)^T R(x_i - e_i), \tag{9}$$

where $R \in \mathbb{R}^{T \times T}_{>0}$ is a diagonal matrix with scaling factor in its diagonal elements. It is assumed to be the same matrix for all players. All its diagonal elements are strictly positive so R is positive definite. It is a tuning parameter defining how strong the inconvenience of shifting consumption is.

3.3 Convexity of the objective functions and strategy sets

The individual objective function for the different players is now:

$$J_i(y_i, y_{-i}) = (\sum_{j=1}^{N} p_{local,j} - s)^\top B(\sum_{j=1}^{N} p_{local,j} - s) + (x_i - e_i)^T R(x_i - e_i) + p_{local,i}{}^\top D p_{local,i}. \tag{10}$$

This function is the sum of three terms, each one being convex. Indeed, $(x_i - e_i)^T R(x_i - e_i)$ is quadratic in x_i and is therefore convex, $(\sum_{j=1}^{N} p_{local,j} - s)^\top B_i(\sum_{j=1}^{N} p_{local,j} - s)$ and $p_{local,i}{}^\top D_i p_{local,i}$ are quadratic in $p_{local,i}$ and thus are convex.

So $J_i(y_i, y_{-i})$ is a convex function in y_i. Furthermore $J_i(y_i, y_{-i})$ is continuous in all variables and continuously differentiable in y_i

The strategy sets are defined in (5). They are the intersection of hyperplanes and half-spaces. Thus, the strategy sets are convex and compact.

3.4 Nash Equilibrium and variational inequalities

As mentioned in section 2.3, the objective when studying a game is often to find the strategies y^* for each players satisfying:

$$J_i(y_i^*, y_{-i}^*) \leq J_i(y_i, y_{-i}^*), \quad \forall y_i \in \mathcal{Y}_i\,, \quad \forall i \in \mathcal{N}. \tag{11}$$

In words, the objective is to find a combination of player strategies from which no player can improve by unilaterally changing its strategy. This point is the Nash Equilibrium and can be interpreted as a stable solution of the game studied. If there is no Nash equilibrium to a game, there is always at least one agent that has an incentive to change strategy, making it impossible to reach a stable solution. Providing theoretical proofs that a Nash Equilibrium exists is therefore useful. Furthermore, it is desirable to show that this solution is unique, as this avoids having to compare different Nash equilibria.

Both existence and uniqueness can be proven using the concept of *variational inequality (VI)*. This concept is presented in the next section. All the theorems and propositions used come from the work done in [40].

3.4.1 Variational inequality

Given a set $K \subset \mathbb{R}^n$, and $F\colon K \to \mathbb{R}^N$, the variational inequality problem $VI(K, F)$ consists of finding an $x \in K$ such that:

$$(z - x)^\top F(x) \geq 0\ , \quad \forall z \in K. \tag{12}$$

The set of solutions of $VI(K, F)$ is denoted by $SOL(K, F)$. The concept of variational inequality can be applied to some optimization problems. Given a continuously differentiable and convex function $f : \mathbb{R}^n \to \mathbb{R}$ with derivative $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ and a convex set $K \subset \mathbb{R}^n$, having $x^* \in SOL(K, \nabla f)$ is equivalent to having x^* being the optimal solution of the minimization problem:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & x \in K. \end{aligned} \tag{13}$$

From the definition of VI, $x^* \in SOL(K, \nabla f)$ is equivalent to $\nabla f(x^*)^\top(y - x^*) \geq 0$. Given that f is a convex function, it satisfies:

$$f(y) \geq f(x) + \nabla f(x)^\top(y - x), \quad \forall x, y \in K. \tag{14}$$

so $f(y) \geq f(x^*) + \nabla f(x^*)^\top(y - x^*) \geq f(x^*)$. The solution x^* is therefore a minimizer of the problem in (13). It is then possible to prove that x^* being a solution to the optimization problem implies $x^* \in SOL(K, \nabla f)$, by first assuming it does not ($\nabla f(x^*)^\top(y-x^*) < 0$) and reaching a contradiction. ■

3.4.2 Existence and uniqueness of the Nash Equilibrium

Focusing back to the game at hand, let us define the problem $VI(\mathcal{Y}, F_\Gamma)$, where $F_\Gamma : \mathbb{R}^{2NT} \to \mathbb{R}^{2NT}$ is the pseudo-gradient defined as $F_\Gamma = (\nabla_{y_1} J_1; \nabla_{y_2} J_2; ... : \nabla_{y_N} J_N)$. Proposition 1.4.2 in [40] states that with the conditions mentioned here, a vector y^* is a Nash equilibrium if and only if $y^* \in SOL(\mathcal{Y}, F_\Gamma)$.

To prove this equivalence, first recall that at NE, the strategy of each player i, y_i^*, satisfies $J_i(y_i^*, y_{-i}^*) \leq J_i(y_i, y_{-i}^*)\ \forall y_i \in \mathcal{Y}_i$. This means that for each player i , y_i^* is a minimizer of $J_i(y_i, y_{-i}^*)$.
In section 3.4.1, it was demonstrated that y_i^* being a minimizer is equivalent to it belonging to the set $SOL(\mathcal{Y}_i, \nabla_{y_i} J_i)$. By definition of a VI problem, this means that $\nabla_{y_i} J_i(y_i^*, y_{-i}^*)^\top(y_i - y_i^*) \geq 0$ $\forall i \in \mathcal{N}$.
As this is true for all players, all the inequalities can be summed into the single inequality $\sum_{i=1}^N \nabla_{y_i} J_i(y_i^*, y_{-i}^*)^\top(y_i - y_i^*) \geq 0$. This can be rewritten in a more compact form as:

$$(\nabla_{y_1} J_1(y_1^*, y_{-1}^*); \nabla_{y_2^*} J_2(y_2^*, y_{-2}^*); ... : \nabla_{y_N^*} J_N(y_N^*, y_{-N}^*))^\top(y - y^*) \geq 0.$$

Recalling the definition of F_Γ, this is equivalent to $(y - y^*)^\top F_\Gamma \geq 0$. Therefore, y^* is a NE of the game, if and only if $(y - y^*)^\top F_\Gamma \geq 0$. As this is the definition of the set $SOL(\mathcal{Y}, F_\Gamma)$, the equivalence is proven. ■

Thus, to study the Nash equilibrium of the game at hand, it is possible to equivalently study the corresponding VI problem. To find out whether a Nash equilibrium exists or is unique, it is sufficient to check the properties of $SOL(\mathcal{Y}, F_\Gamma)$. Corollary 2.2.5 in [40] states that for a compact set K, and a continuous map F, $SOL(K, F)$ is non-empty and compact.

Given what has been proven in section 3.3, these conditions are met for F_Γ and $\mathcal{Y}$. $SOL(\mathcal{Y}, F_\Gamma)$ is thus non-empty and compact. There exists therefore a Nash Equilibrium to the game at hand. ■

We can also prove that the Nash equilibrium is unique by demonstrating that $SOL(\mathcal{Y}, F_\Gamma)$ is a singleton. As Theorem 2.3.3 in [40] states, this is the case if and only if the map F_Γ is strictly monotone.

A possible way to assess the monotony of F_Γ is by analysing its jacobian JF_Γ defined as $JF_\Gamma := (\frac{\delta F_i}{\delta x_j}) \in \mathbb{R}^{2NT \times 2NT}$. If JF_Γ is positive definite, then F_Γ is strictly monotone, as demonstrated in proposition 2.3.2b in [40].

Therefore, by showing that JF_Γ is positive definite, it is possible to conclude that there is a unique Nash equilibrium.

As a initial step, it is useful to look at the explicit value of F_Γ. Given that:

$$\begin{aligned} F_\Gamma = \Bigg(&\frac{\partial J_1}{\partial x_{1,1}}, ..., \frac{\partial J_1}{\partial x_{1,T}}, \frac{\partial J_1}{\partial p_{local,1,1}}, ..., \frac{\partial J_1}{\partial p_{local,1,T}}, ..., \\ &\frac{\partial J_N}{\partial x_{N,1}}, ..., \frac{\partial J_N}{\partial x_{N,T}}, \frac{\partial J_N}{\partial p_{local,N,1}}, ..., \frac{\partial J_N}{\partial p_{local,N,T}} \Bigg), \end{aligned} \tag{15}$$

and given J_i as it is defined in (10), the first derivative is equal to:

$$\frac{\partial J_i}{\partial x_{i,t}} = 2R_{t,t}(x_{i,t} - e_{i,t}), \tag{16}$$

and

$$\frac{\partial J_i}{\partial p_{local,i,t}} = 2B_{t,t} \left(\sum_{j=1}^{N} p_{local,j,t} - s \right) + 2D_{t,t} p_{local,i}. \tag{17}$$

Each row in the jacobian of F_Γ corresponds to the derivative of the first order derivatives computed above with respect to all the players decision variables. The first row of JF_Γ is for example the derivative of $\frac{\partial J_1}{\partial x_{1,1}}$ with respect to all players decision variables:

$$\begin{aligned} JF_{\Gamma,1*} = \Bigg(&\frac{\partial^2 J_1}{\partial x_{1,1}^2}, ..., \frac{\partial^2 J_1}{\partial x_{1,T} \partial x_{1,1}}, \frac{\partial^2 J_1}{\partial p_{local,1,1} \partial x_{1,1}}, ..., \frac{\partial^2 J_1}{\partial p_{local,1,T} \partial x_{1,1}}, \dots, \\ &\frac{\partial^2 J_1}{\partial x_{N,1} \partial x_{1,1}}, ..., \frac{\partial^2 J_1}{\partial x_{N,T} \partial x_{1,1}}, \frac{\partial^2 J_1}{\partial p_{local,N,1} \partial x_{1,1}}, ..., \frac{\partial^2 J_1}{\partial p_{local,N,T} \partial x_{1,1}} \Bigg). \end{aligned} \tag{18}$$

Given the first order derivative in (16), taking the derivative of this expression will only yield a non-zero value if the differentiation is done with respect to the same decision variable $x_{i,t}$. So for all players i:

$$\frac{\partial^2 J_i}{\partial x_{i,t}^2} = 2R_{t,t}. \tag{19}$$

Differentiating the first order derivative in (17), will yield a non zero value if differentiating with respect to the players $p_{local,i,t}$ at the same timestep:

$$\frac{\partial^2 J_i}{\partial p_{local,i,t}^2} = 2B_{t,t} + 2D_{t,t}. \tag{20}$$

It will also be non zero when differentiating with respect to any other players $p_{local,j,t}$ at the same timestep t:

$$\frac{\partial^2 J_i}{\partial p_{local,j,t} \partial p_{local,i,t}} = 2B_{t,t}. \tag{21}$$

Any other element of the jacobian will be equal to 0. The jacobian JF_Γ is therefore:

$$JF_\Gamma = 2 * \begin{pmatrix} \mathcal{A} & O_{NT\times NT} \\ O_{NT\times NT} & \mathcal{B}_N \end{pmatrix}, \tag{22}$$

where:

$$\mathcal{A} = I \bigotimes R. \tag{23}$$

$I \in \mathbb{R}^{N\times N}$ is the identity matrix and $\bigotimes$ represents the Kronecker product. Also, $\mathcal{B}_N \in \mathbb{R}^{NT\times NT}$ is equal to:

$$\mathcal{B}_N = \begin{pmatrix} B+D & B & \dots & B \\ \vdots & \vdots & & \vdots \\ B & \dots & B & B+D \end{pmatrix}. \tag{24}$$

Having a strictly monotone F_Γ is equivalent to having JF_Γ being positive definite. So by proving that the matrix in (53) is positive definite, it is possible to show that F_Γ is strictly monotone. Given the definitions of $\mathcal{A}$ and $\mathcal{B}_N$, JF_Γ is a block diagonal matrix. It is positive definite if and only if $\mathcal{A}$ and $\mathcal{B}$ are positive definite. For $\mathcal{A}$, given that R is positive definite, it is a diagonal matrix with only strictly positive values so it is positive definite.

$\mathcal{B}_N$ is positive definite if and only if for any non zero $z = (z_1, ...z_N)$, with $z_i \in \mathbb{R}^T \, \forall i \in 1, ..., N$, $z^\top \mathcal{B}_N z > 0$.

$$z^\top \mathcal{B}_N z = \sum_{i=1}^{N} (z_i^\top B z_i + z_i^\top D z_i) + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} z_i^\top B z_j. \tag{25}$$

For all i, $k_i \in \mathbb{R}^T$ is defined as $k_i = \sqrt{B} z_i$. $z^\top \mathcal{B}_N z$ is now:

$$z^\top \mathcal{B}_N z = \sum_{i=1}^{N} z_i^\top D z_i + \sum_{i=1}^{N} k_i^\top k_i + 2 \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} k_i^\top k_j = \sum_{i=1}^{N} z_i^\top D z_i + || \sum_{i=1}^{N} k_i ||_2^2. \tag{26}$$

Given that a norm cannot be negative, $|| \sum_{i=1}^{N} k_i ||_2^2 \geq 0$. However, it could be possible that the different k_i within the norm cancel each other out, resulting in the norm being equal to 0. Nonetheless, $\sum_{i=1}^{N} z_i^\top D z_i = \sum_{i=1}^{N} \sum_{j=1}^{T} \, d_j z_{i,j}^2 > 0$. So $z^\top \mathcal{B}_N z > 0$ for any non zero z, and $\mathcal{B}_N$ is positive definite.

Since $\mathcal{A}$ and $\mathcal{B}_N$ are positive definite, the jacobian JF_Γ is positive definite too. Thus, the game Γ admits a unique Nash equilibrium. ■

Without the term ${p_{local,i}}^\top D p_{local,i}$ in the objective function, it would have only been possible to state that $z^\top \mathcal{B}_N z \geq 0$. The jacobian matrix JF_Γ would then be positive semidefinite, providing no guarantee about the uniqueness of the Nash Equilibrium. In practice, D can be arbitrarily small, limiting its impact on performance, while theoretically ensuring the uniqueness of the NE.

For the Nash Equilibrium to be a relevant outcome for the community, the sum of the local consumption $p_{local,i}$ for all the players at the Nash equilibrium should be smaller or equal the available solar energy at every timestep. Otherwise the NE would be unachievable, with a total local consumption greater than what can be produced locally. The following proof ensures that the total consumption is indeed always lower or equal to the local production at every timestep. Suppose, that this is not true and the sum of the local consumptions at a certain time step t, $\sum_{j=1}^{N} p_{local,j,t}^{NE}$ is bigger than the available solar energy $s_t \geq 0$. This implies that:

$$B_{t,t}\left(\sum_{j=1}^{N} p_{local,j,t}^{NE} - s_t\right)^2 > 0.$$

For any agent i for which $p_{local,i,t}^{NE} > 0$, there always exists a $\tilde{p}_{local,i,t} < p_{local,i,t}^{NE}$ such that:

$$B_{t,t}\left(\sum_{j\neq i} p_{local,j,t}^{NE} + \tilde{p}_{local,i,t} - s_t\right)^2 < B_{t,t}\left(\sum_{j\neq i} p_{local,j,t}^{NE} + p_{local,i,t}^{NE} - s_t\right)^2.$$

Notice that $\tilde{p}_{local,i,t}$ would always respect the constraint of being smaller than $x_{i,t}$, since $\tilde{p}_{local,i,t} < p_{local,i,t}^{NE} \leq x_{i,t}$. Thus, $p_{local,i,t}^{NE}$ could not have been the optimal solution for agent i. This proves that $\sum_{j=1}^{N} p_{local,j,t}^{NE}$ is smaller or equal to s_t. ■

3.5 Algorithm design

With the proofs of existence and uniqueness of the Nash Equilibrium done in the previous section, it is now interesting to look at how to design an algorithm converging towards this point. Our objective is to design an algorithm that converges to the NE in a distributed way. That way, the limitations of centralized resolution, namely loss of privacy, computational limits and possible unfairness between members could be mitigated. To do so, the algorithm should converge without each agent knowing the exact strategies of the other agents in the game.

Looking at the objective function of the different players in (10), it can be observed that the agents do not actually need to know the exact strategies to find their optimal response. In fact, the only information required is the aggregate strategy of all the other agents. By knowing the sum of the local consumption of everyone except themselves, each player can compute their strategy.

This property of the game Γ can be utilised to design a distributed algorithm to find the NE. By introducing a term which would encapsulate the strategies of all the players, and by broadcasting it to everyone instead of the exact strategies, it would be possible for each player to solve its individual optimization problem. Each player would need to send their optimal strategy to a central coordinating agent, which would compute the aggregate strategy and send it back to everyone.

In the current game, let the shared parameter $z \in \mathbb{R}^T$ be an estimation of the community load of local energy consumed. After sending an initial estimation of their local consumption, each player would receive z, and would update their strategies accordingly. However, as z would include an estimation of each players own consumption, it cannot be used directly in each players individual optimization problems. Instead, each players should compute player specific versions of z. For player i, this specific version would be z_{-i} and would be calculated by subtracting its initial estimation of local consumption to z. The objective function from player i defined in (10) becomes:

$$J_i(y_i, z_{-i}) = (p_{local,i} + z_{-i} - s)^\top B(p_{local,i} + z_{-i} - s) + (x_i - e_i)^T R(x_i - e_i) + p_{local,i}{}^\top D p_{local,i}. \quad (27)$$

Each player would therefore find their optimal strategy by solving:

$$\begin{aligned} y_i^* = \arg\min \quad & J_i(y_i, z_{-i}) \\ \text{s.t.} \quad & y_i \in \mathcal{Y}_i. \end{aligned} \quad (28)$$

These optimal strategies could be computed for all players through a two-step iterative algorithm. Given a current iteration step k, each player would first compute their optimal action by solving (28) given $z_{(k)}$, the value of z at iteration k. By collecting each agents optimal $p_{local,i(k)}^*$, the central coordinating agent would then compute the resulting estimation of the community local load $\Lambda(z_k)$ as:

$$\Lambda(z_{(k)}) = \sum_{i=1}^{N} p_{local,i(k)}^*. \quad (29)$$

This would then be used to update the value of z for the next iteration:

$$z_{(k+1)} = (1-\eta)z_{(k)} + \eta\Lambda(z_{(k)}), \tag{30}$$

where $\eta \in (0,1)$ is the learning rate. It is useful to avoid having too big steps between iterations which could hinder convergence. Figure 5 shows the interactions between the different players and the central coordinator for one iteration. Such iterations would go on until the updates on z become smaller than a convergence criteria ϵ_{stop}.

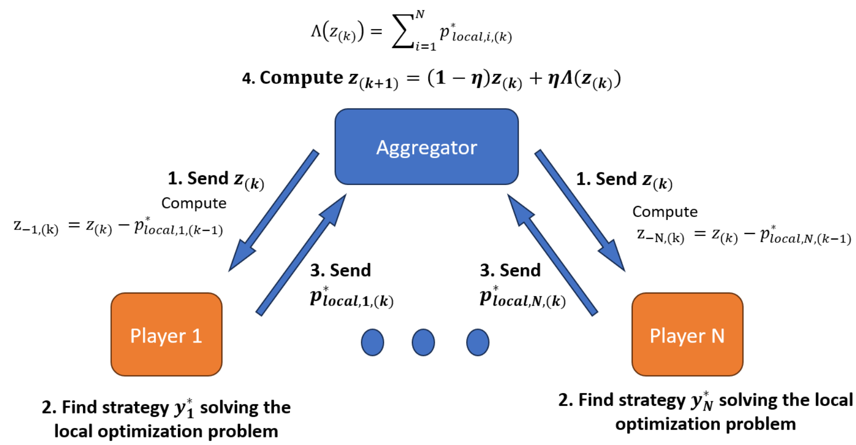


Figure 5: *Exchanges between players and central node during one iteration*

To start the algorithm, $z_{(1)}$ would be initialized as the community consumption of local energy when no optimization is done (using each players default loads e_i). Each users local consumption is initialized similarly with their individual default loads. Algorithm 1 summarizes the steps of the approach described above. The framework of this algorithm is inspired by the algorithm in [6]. In this work the algorithm was however designed for a mean field game, so some changes were made to it to be useful for the load game approach instead. The major change is changing the shared parameter from the estimated price over the horizon to the estimated local consumption of the community.

Algorithm 1 Distributed load game resolution

initialize:
$z_{(1)} \leftarrow min(\sum_{i=1}^{N} e_i, s)$, $z_{(1)} \in \mathbb{R}^T$
$p_{local,i,(0)} \leftarrow min(e_i, s/N)$ for $i = 1...N$, $p_{local,i,(0)} \in \mathbb{R}^T$
$k \leftarrow 1$
repeat
for $i \in N$ **do**
$z_{-i,(k)} \leftarrow z_{(k)} - p^*_{local,i,(k-1)}$
Solve local optimization problem to compute $(y^*_{i,(k)})$
end for
$z_{(k+1)} \leftarrow (1-\eta)z_{(k)} + \eta * (\sum_{i=1}^{N} p^*_{local,i,(k)})$
$k \leftarrow k+1$
until $||z_{(k+1)} - z_{(k)}|| \leq \epsilon_{stop}$

3.6 Convergence of the algorithm

The algorithm presented in the previous section is simple and easy to use. It would also be desirable to provide theoretical guarantees that it converges to the Nash Equilibrium. However, the convergence of algorithms similar to Algorithm 1 is not well understood, even for centralized resolution frameworks. This algorithm is indeed similar to the Non-linear Jacobi type Algorithm described in [41], albeit it is distributed instead of centralized. This algorithm is described as at best a good heuristics, but there are no proofs for its convergence. Still, [41] states that for Nash Equilibrium problems with no constraints depending on players actions, it is possible to prove that if the algorithm converges, it converges to the Nash Equilibrium. Section 3.6.2 will prove this for the game at hand.

To ensure that our approach converges to the NE, a secondary algorithm with proven convergence behavior needs to be added. The idea is that after some set number of iterations, if the primary algorithm has not converged, it would switch to this secondary algorithm which is certain to converge, even if it could take some time. That way, if the faster primary algorithm does not converge, the secondary algorithm still enables us to reach the NE. For this research, the secondary algorithm used is projected gradient descent. Section 3.6.1 describes this approach and provides proof that it converges to the NE.

3.6.1 Secondary algorithm: projected gradient descent

This section describes the secondary algorithm used and provides proofs that it converges to the Nash Equilibrium. As mentioned above, the algorithm selected is the projected gradient descent. This algorithm has been designed in [35], but it used the mean load of the players as a shared parameter. Here the total local load is used instead.

Before describing how it works, the projection operator needs to be introduced. Given a convex set $K \subset \mathbb{R}^n$, the projection operator $\Pi_K(x) : \mathbb{R}^n \to K$ is defined as:

$$\begin{aligned} \Pi_K(x) := \arg\min_{z} \quad & \tfrac{1}{2}(z-x)^\top(z-x) \\ \text{s.t.} \quad & z \in K. \end{aligned} \tag{31}$$

The action of the projection operator can be interpreted as mapping any $x \in \mathbb{R}^n$ to the point $\bar{x} \in K$ which has the minimum Euclidean distance to x.

This operator is often used in optimization to iteratively converge to a fixed point. Given a map $F\colon K \to \mathbb{R}^n$, and a starting point $x_{(0)}$ at iteration $k=0$, the next iterations are obtained by solving $x_{(k+1)} = \Pi_K(x_{(k)} - \eta F(x_{(k)}))$. Here η is the step size.

This algorithm runs until a point x^* satisfying $x^* = \Pi_K(x^* - \eta F(x^*))$ is found. This corresponds to the optimal solution. Moreover, for any $\eta > 0$, finding a x^* satisfying the above condition is equivalent to finding a x^* such that $x^* \in SOL(K, F)$.

To prove that, theorem 1.5.5b in [40] can be used. It states that for a non empty convex subset K of $\mathbb{R}^n$, the vector $x^* = \Pi_K(x)$ is the only vector in K satisfying:

$$(z - x^*)^\top(x^* - x) \geq 0,\ \forall z \in K. \tag{32}$$

Therefore,

$$\begin{aligned} x^* = \Pi_K(x^* - \eta F(x^*)) &\Leftrightarrow (z-x^*)^\top(x^* - (x^* - \eta F(x^*))) \geq 0\ \forall z \in K \\ &\Leftrightarrow (z-x^*)^\top F(x^*) \geq 0\ \forall z \in K. \end{aligned} \tag{33}$$

From the definition of VI in (12), this last term is the condition for $x^* \in SOL(K, F)$. Thus, $x^* \in SOL(K, F)$ is indeed equivalent to $x^* = \Pi_K(x^* - \eta F(x^*))$. It is also good to recall that x^* being a solution to the VI problem is also equivalent to it being a Nash Equilibria for the corresponding game. The question of computing the Nash equilibria of the game is thus reduced to the computation of fixed points of the operator $x \to \Pi_K(x - \eta F(x))$.

For the algorithm to converge to this fixed point, this map needs to be a contraction. This means that it should satisfy:

$$||\Pi_K(x - \eta F(x)) - \Pi_K(z - \eta F(z))||_2^2 \leq ||x - z||_2^2\ , \quad \forall x, z \in K. \tag{34}$$

Using the properties of the projection operator and by making some assumptions on the map F the above inequality can be proven. First, as stated in theorem 1.5.5d of [40], the projection operator is a non expansive operator. This means that for any two vectors x and $z \in \mathbb{R}^n$:

$$||\Pi_K(z) - \Pi_K(x)||_2 \leq ||z - x||_2\ , \quad \forall z, x \in \mathbb{R}^n. \tag{35}$$

Then, the assumptions on the map F are the following. Let $L, \mu \in \mathbb{R}_{>0}$ be such that $\forall x, y \in K$:

- $(F(x) - F(z))^\top (x - z) \geq \mu ||x - z||_2^2$ (F is strongly monotone).
- $||F(x) - F(z)||_2 \geq L||x - z||_2$ (F is Lipschitz continuous).

Using (35) we can first state that:

$$||\Pi_K(z - \eta F(z)) - \Pi_K(x - \eta F(x)||_2^2 \leq ||z - \eta F(z) - (x - \eta F(x))||_2^2. \quad (36)$$

Writing out the 2-norm squared on the right hand side gives:

$$||z - \eta F(z) - (x - \eta F(x))||_2^2 = ||z - x||_2^2 + \eta^2 ||F(z) - F(x)||_2^2 - 2\eta(F(z) - F(x))^\top (z - x). \quad (37)$$

Then, thanks to the assumptions on the map F, we know that $\eta(F(x) - F(z))^\top (x - z) \geq \eta\mu||x - z||_2^2$ and $\eta^2 ||F(x) - F(z)||_2^2 \geq \eta^2 L^2 ||x - z||_2^2$. Thus:

$$||\Pi_K(z - \eta F(z)) - \Pi_K(x - \eta F(x)||_2^2 \leq ||z - \eta F(z) - (x - \eta F(x))||_2^2 \leq (1 + \eta^2 L^2 - 2\eta\mu)||z - x||. \quad (38)$$

To satisfy (34), $0 < (1 + \eta^2 L^2 - 2\eta\mu) < 1$ which is equivalent to having a step size η satisfying $0 < \eta < \frac{2\mu}{L^2}$. If this condition is satisfied for a map F following the assumptions mentioned above, a projected gradient descent approach can be used to converge to the Nash Equilibrium of the game. ■

Focusing back on the game at hand, the map used would be F_Γ and the set on which the strategies would be projected would be $\mathcal{Y}$. The projection operator would be $\Pi_{\mathcal{Y}}(y) : \mathbb{R}^{2TN} \to \mathcal{Y}$. Given the definition of F_Γ, the assumptions presented above hold for this map. Annex B proves the strong monotony and Lipschitz continuity of F_Γ. So, by using the following iteration update, the projected gradient descent algorithm would be sure to converge to the Nash Equilibrium, provided the step size η satisfies $0 < \eta < \frac{2\mu}{L^2}$.

Although this algorithm would converge to the NE in a centralised manner, it is possible to modify it so that it can be solved in a distributed way. To solve the algorithm, each player i only needs to know his strategy set $\mathcal{Y}_i$ as well as $\nabla_{y_i} J_i$. Recalling the gradients of J_i in (16) and (17), each player does not need to know the specific strategies of the other players, only the total sum of their local consumption. Instead of having one iteration update for all the players, each player could update their own strategies y_i by projecting on their own strategy sets $\mathcal{Y}_i$.

As for the primary algorithm, it is possible to introduce the shared parameter z and each players version z_{-i}. As defined before, z still represents the community consumption of local energy. The objective function used for each player would also be the one defined in (27). Each iteration would be composed of two steps: first the players would use $z_{(k)}$ to update their strategies using projected gradient descent. Then, each player would send their new optimal local consumption to compute $z_{(k+1)}$. Contrarily to the primary algorithm, the update on z would not depend on a learning rate, as the step size is already present in the projection computation. Algorithm 2 summarizes the steps of the secondary algorithm.

Provided η is well chosen, this secondary algorithm is sure to converge to the NE [35] and provides insurance that the NE will still be found, even if the primary algorithm fails to converge.

3.6.2 Convergence behavior of the primary algorithm

Although the convergence guarantee of the algorithm are not proven, it is possible to prove that if the algorithm converges, it converges to the Nash Equilibrium. The following work proves this for an unconstrained case. Extending this to the constrained problem would be a good direction to improve this project theoretically.

This is the case if at convergence, no players would be motivated to change their strategies. In other words, the algorithm converge to the Nash equilibrium if the strategies of each players converge to fixed points too. At each iteration k, every players select a new optimal strategy by solving:

$$\begin{aligned}(p^*_{local,i(k)}, x^*_{i(k)}) &= \\ &argmin_{(p_{local,i}, x_i)} (p_{local,i} + z_{(k)} - p^*_{local,i(k-1)} - s)^\top B(p_{local,i} + z_{(k)} - p^*_{local,i(k-1)} - s) \\ &\quad + p_{local,i}{}^\top D p_{local,i} + (x_{i(k)} - e_i)^\top R(x_{i(k)} - e_i). \quad (39)\end{aligned}$$

Algorithm 2 Distributed projected gradient descent

initialize:
$z_{(1)} \leftarrow min(\sum_{i=1}^{N} e_i, s)$, $z_{(1)} \in \mathbb{R}^T$
$p_{local,i,(0)} \leftarrow min(e_i, s)$ for $i = 1...N$ $p_{local,i,(0)} \in \mathbb{R}^T$
$k \leftarrow 1$
repeat
for $i \in N$ **do**
$z_{-i,(k)} \leftarrow z_{(k)} - p^*_{local,i,(k-1)}$
Solve $y_{i,(k+1)} \leftarrow \Pi_{\mathcal{Y}_i}(y_{i,(k)} - \eta\nabla_{y_{i,(k)}}(J_i(y_{i,(k)}, z_{-i,(k)})))$
end for
$z_{(k+1)} \leftarrow (\sum_{i=1}^{N} p^*_{local,i,(k)})$
$k \leftarrow k + 1$
until $||z_{(k+1)} - z_{(k)}|| \leq \epsilon_{stop}$

Assuming the algorithm has converged means that $z_{(k)}$ reached a constant value. Calling it z, $z_{(k+1)} = z_{(k)} = z$. However, as $z_{(k)} = \sum_{i=1}^{N} p^*_{local,i(k)}$. Having $z_{(k)}$ converging to a fixed point is not enough to say that it is the Nash equilibrium. Indeed, even if z does not change anymore ($||z_{k+1} - z_k|| \leq \epsilon_{stop}$), it is possible that the different $p_{local,i}$ vary, with their modifications cancelling each other in the sum.

As z is now constant, the actions of a player do not influence the other players anymore. Each player now tries to solve individual optimization problems. To ensure a NE has been reached, $p_{local,i(k)}$ also needs to converge to a fixed point for each of these individual optimizations. Let this point be $\bar{p}_{l,i}$. It is defined such that for all players i:

$$\begin{aligned}(\bar{p}_{l,i}, x^*_{i(m)}) = argmin_{(p_{local,i},x_i)}(p_{local,i} + z - \bar{p}_{l,i} - s)^\top B(p_{local,i} + z - \bar{p}_{l,i} - s) \\ + p_{local,i}{}^\top D p_{local,i} + (x_i - e_i)^\top R(x_i - e_i).\end{aligned} \tag{40}$$

To see if the algorithm convergence cause $p_{local,i}$ to converge towards $\bar{p}_{l,i}$, it is first needed to see what the value of $\bar{p}_{l,i}$ is. To do so, the derivative of the objective function J_i with respect to $p_{local,i}$ needs to be analysed. For readability, the vector $b = z - \bar{p}_{l,i} - s \in \mathbb{R}^T$ is introduced. The objective function becomes:

$$\begin{aligned}J_i(p_{local,i}, x_i) &= (p_{local,i} + b)^\top B(p_{local,i} + b) + p_{local,i}{}^\top D p_{local,i} + (x_i - e_i)^\top R(x_i - e_i) \\ &= p_{local,i}{}^\top B p_{local,i} + p_{local,i}{}^\top B b + b^\top B p_{local,i} + b^\top B b + p_{local,i}{}^\top D p_{local,i} + (x_i - e_i)^\top R(x_i - e_i),\end{aligned}$$

so

$$\frac{\partial J_i}{\partial p_{local,i}} = 2Bp_{local,i} + 2Dp_{local,i} + Bb + (b^\top B)^\top. \tag{41}$$

As $(b^\top B)^\top = B^\top b = Bb$ because B is a diagonal matrix, $\frac{\partial J_i}{\partial p_{local,i}} = 2Bp_{local,i} + 2Dp_{local,i} + 2Bb$. Substituting back for b:

$$\frac{\partial J_i}{\partial p_{local,i}} = 2Bp_{local,i} + 2Dp_{local,i} + 2B(z - \bar{p}_{l,i} - s). \tag{42}$$

The optimal value of $p_{local,i}$ is the one satisfying $\frac{\partial J_i}{\partial p_{local,i}} = 0$, and if it converged, it is also equal to $\bar{p}_{l,i}$. So, with $p_{local,i} = \bar{p}_{l,i}$:

$$\frac{\partial J_i}{\partial p_{local,i}} = 0 \Leftrightarrow 2B\bar{p}_{l,i} + 2D\bar{p}_{l,i} + 2B(z - \bar{p}_{l,i} - s) = 0$$

$$\Leftrightarrow B\bar{p}_{l,i} + D\bar{p}_{l,i} - B\bar{p}_{l,i} = B(s - z)$$

$$\Leftrightarrow \bar{p}_{l,i} = D^{-1}B(s - z).$$

So, there exists a fixed point $\bar{p}_{l,i}$ satisfying equation (40), and it is equal to $\bar{p}_{l,i} = D^{-1}B(s - z)$.

Now that the existence of such a point has been proven, it is possible to assess whether $p_{local,i}$ converges to it for a fixed value of z. To do so, it is interesting to look at how the solution at an iteration depends on the optimal solution at the previous iteration. Looking at the derivative of the objective function defined in (39) with respect to $p_{local,i}$:

$$\frac{\partial J_i}{\partial p_{local,i}} = 2Bp_{local,i} + 2Dp_{local,i} + 2B(z - p^*_{local,i(k-1)} - s). \tag{43}$$

When this derivative is equal to 0, the optimum is reached, and $p_{local,i} = p^*_{local,i(k)}$. So

$$2(B + D)p^*_{local,i(k)} + 2B(z - p^*_{local,i(k-1)} - s) = 0. \tag{44}$$

Let $\bar{p}_{l,i} + d_{i(k)} = p^*_{local,i(k)}$, where $d_{i(k)} \in \mathbb{R}^T$ represents the difference between the optimal value of $p_{local,i}$ at iteration k and the fixed point $\bar{p}_{l,i}$. The above equation becomes:

$$2(B + D)(\bar{p}_{l,i} + d_{i(k)}) + 2B(z - \bar{p}_{l,i} - d_{i(k-1)} - s) = 0, \tag{45}$$

so

$$\begin{aligned}
&(B + D)(\bar{p}_{l,i} + d_{i(k)}) = Bd_{i(k-1)} + B\bar{p}_{l,i} + B(s - z)\\
&\Leftrightarrow (\bar{p}_{l,i} + d_{i(k)}) = (B + D)^{-1}Bd_{i(k-1)} + (B + D)^{-1}B\bar{p}_{l,i} + (B + D)^{-1}B(s - z)\\
&\Leftrightarrow d_{i(k)} = (B + D)^{-1}Bd_{i(k-1)} + (B + D)^{-1}B\bar{p}_{l,i} - \bar{p}_{l,i} + (B + D)^{-1}B(s - z)\\
&\Leftrightarrow d_{i(k)} = (B + D)^{-1}Bd_{i(k-1)} - (B + D)^{-1}D\bar{p}_{l,i} + (B + D)^{-1}B(s - z).
\end{aligned}$$

Given that $\bar{p}_{l,i} = D^{-1}B(s-z)$, $-(B+D)^{-1}D\bar{p}_{l,i} = -(B+D)^{-1}DD^{-1}B(s-z) = -(B+D)^{-1}B(s-z)$. The last two terms on the right hand side therefore cancel out, leaving:

$$d_{i(k)} = (B + D)^{-1}Bd_{i(k-1)}. \tag{46}$$

Given that B and D are both strictly positive diagonal matrices, $(B + D)^{-1}B$ is a diagonal matrix with all diagonal elements strictly positive and smaller than 1. So

$$|d_{i(k)}| = |(B + D)^{-1}Bd_{i(k-1)}| < |d_{i(k-1)}|. \tag{47}$$

So at every iteration, the difference between the optimal $p_{local,i}$ and the convergence fixed point $\bar{p}_{l,i}$ decreases geometrically. So for a fixed value of z, $p^*_{local,i(k)}$ converges to $\bar{p}_{l,i}$.

Therefore, if the algorithm converges to a fixed value of z, it has converged to the Nash Equilibrium of the game at hand. ■

3.7 Theoretical guarantees of the algorithm

The proposed algorithm holds the following theoretical guarantees. Given a compact and convex strategies set $\mathcal{Y}$ and individual objective functions J_i defined as in (10) that are convex and continuously differentiable for each player i:

- The game Γ admits a unique Nash Equilibrium.
- If algorithm 1 converges it converges to the Nash Equilibrium.
- Algorithm 2 ensures that if the primary algorithm does not converge, the overall approach will still reach the Nash Equilibrium.

Building onto these theoretical guarantees, it is now interesting to assess the behavior of the algorithm in practice. The following section presents the simulation setup used to evaluate the algorithm.

3.8 Simulation setup

To assess the performance of the load game approach, the simulation setup in which it will be tested first needs to be defined. First, the strategy set for each player is defined as in (5) to represent a situation where consumption can be pushed from one time to the other but cannot be increased or decreased. The inconvenience cost of having to change consumption load $J_{shifted,i}$ is also defined as in section 3.2.

The initial loads of the community members are constructed from a common base load $e_{base} \in \mathbb{R}^T$ representing a standard residential load with morning and evening consumption peaks. The differences between members are then induced by adding some normal noise to e_{base}. The noise is sampled from a normal distribution with a mean of 0 and a standard deviation of 0.2. The solar energy available s is initialized as a standard solar load, with peak production during the day and no production during the night.

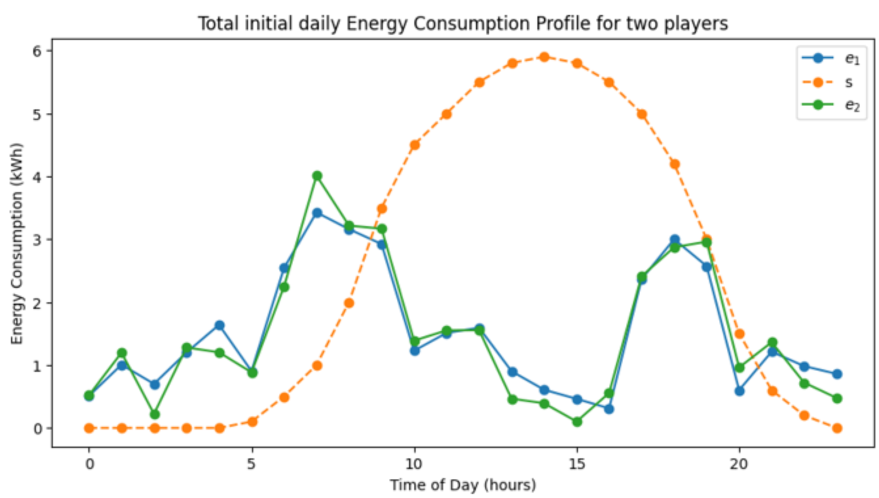
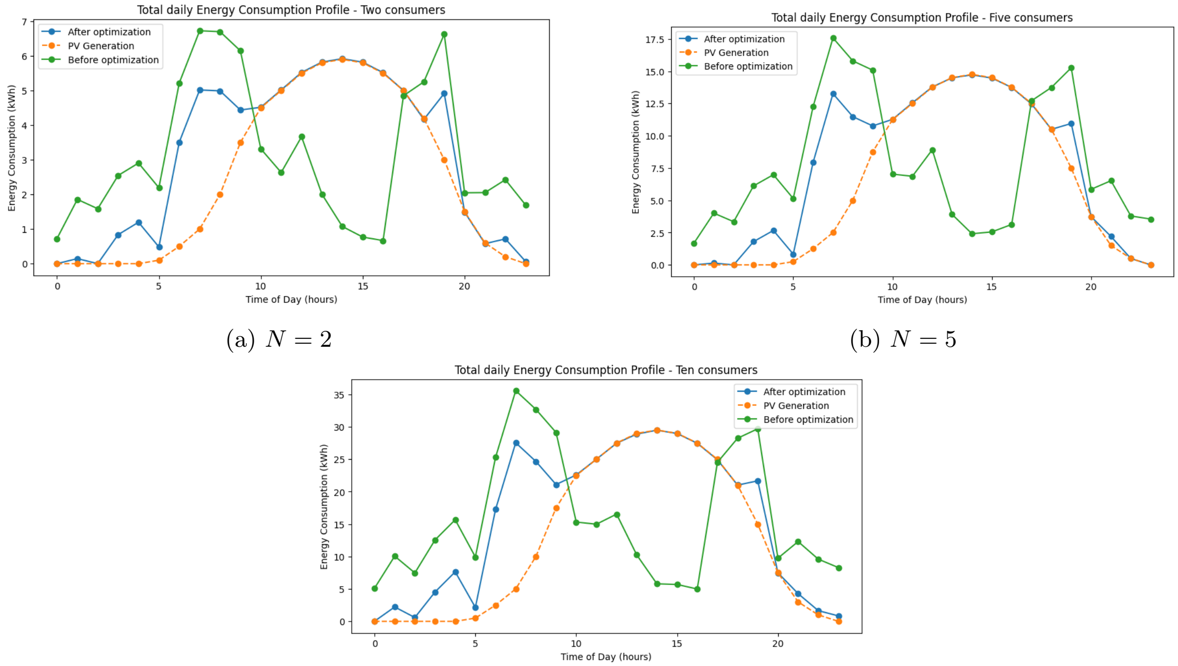


Figure 6: *Default loads and solar production for a two-member energy community*

Figure (6) shows the initial consumption and solar loads for a two members energy community. The two consumption loads follow the same pattern with some slight variations between them. Note that the absolute value of the consumption is not particularly relevant to this analysis. What matters is the relative scales of the amount of consumption and production within the energy community. For this reason, the simulated solar installation is scaled with respect to the total community consumption. This scaling is such that the solar energy can be used to cover most of the demand, but not its entirety.



(a) $N = 2$

(b) $N = 5$

(c) $N = 10$

Figure 7: *Optimal loads for varying community sizes*

3.9 Optimization results for varying community sizes

With the simulation environment defined it is now possible to look at the results of load optimization using algorithm 1. Figure (7) shows the initial and optimal loads for different small scale communities, composed of 2, 5 and 10 members. The initial load is in green and the optimal one in blue. For all community sizes, the results are similar, with the optimal load having shaved morning and evening peaks and an increase in consumption during the day. More specifically the optimal load follows the available solar load and consumes the entirety of the available local solar energy. The game approach thus seems to allow the community to reduce the energy imported from the grid and maximise its self-consumption. There is still some energy imported from the grid, because the local production is not enough to cover the entire demand.

Upon seeing these results at the community level, it is interesting to look at the loads of the individual members. Figure (8) shows the loads of all the community members in the case where $N = 5$. All the loads share the same trend, with a majority of their consumption during daytime. Still, there is some slight variations from one member to another, induced by the differences in their respective initial conditions. Nonetheless, these differences are not an issue when aggregating all the loads as in the plots in figure (7). The algorithm thus successfully outputs for each player the optimal load, accounting for the initial conditions and decisions of the entire community.

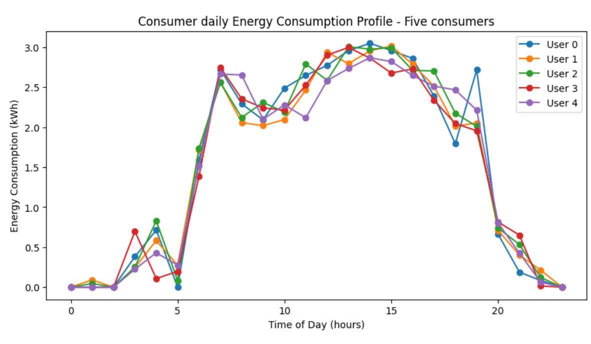
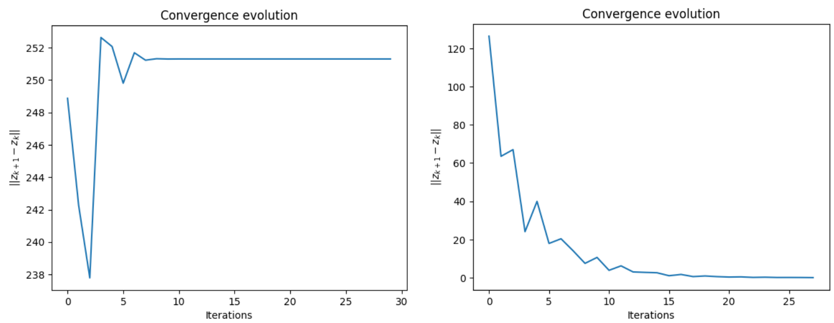


Figure 8: *Individual optimal loads for a five members community*

3.10 Convergence behavior

We have seen that the convergence of the primary algorithm is not theoretically guaranteed. Still, it can be useful to empirically evaluate its convergence behavior for varying community sizes. To do so, the convergence of the algorithm was studied for $N = 5$ and $N = 10$. For both sizes, different values of the learning rate η were tested.

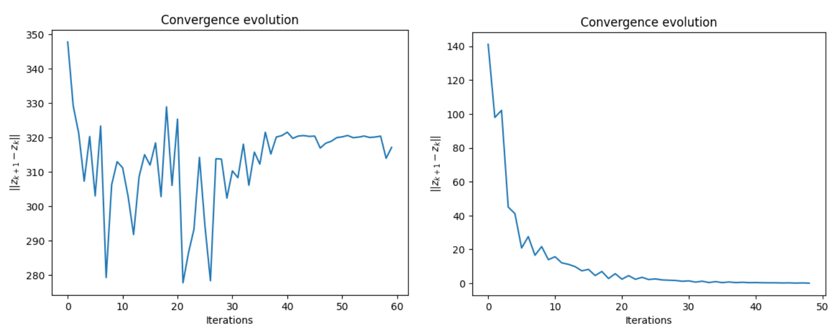
Looking at an EC with $N = 5$, the algorithm does not converge for all values of the learning rate. Indeed, figure (10a) shows that with $\eta = 0.8$, the primary algorithm does not reach a fixed point and iterates until it reaches the maximum number of iteration possible. However, when the learning rate is reduced to $\eta = 0.4$ as in figure (10b), the algorithm converges. With this simulation setup, the algorithm converges if the learning rate is sufficiently small.



(a) $\eta = 0.8$ (b) $\eta = 0.4$

Figure 9: *Convergence of the algorithm for a EC of five with different learning rates*

To confirm this, the convergence for a community with $N = 10$ is shown in figure (10). Once again, there is no convergence for a larger value of η, but by making it sufficiently small, convergence can be witnessed. Note however that compared to $N = 5$ where convergence occurred for $\eta = 0.4$, the learning rate needs to be significantly smaller for convergence with $N = 10$.



(a) $\eta = 0.5$ (b) $\eta = 0.2$

Figure 10: *Convergence of the algorithm for a EC of ten with different learning rates*

For all the experiments tested, we were able to make the algorithm converge by making η smaller than a threshold value. It appeared that this threshold was inversely proportional to the size of the community. For the bigger communities tested, this caused the convergence to take more iterations, and thus have a higher computational cost.

On the limited experimental work done, the algorithm was made to converge by using a sufficiently small learning rate. As demonstrated in section 3.6.2, if it converges, it converges to the NE for unconstrained strategies. Here the strategies are constrained, but each agent's individual strategy also converge when z converges. For all the experimental setups trialled, we were thus able to make the load game algorithm converge to the Nash Equilibrium by choosing the right learning rate.

4 Integration in a control framework

The potential of the available solar game has been demonstrated on a simple optimization problem in the previous section. It is now interesting to evaluate its performance when integrated in an optimal control framework. The section first presents the control environment used to assess this, before evaluating the resulting controller behavior by performing some sensitivity and uncertainty analysis.

4.1 Control framework: NRGMaestro™

To analyse the use of the available solar game algorithm for control, the tool used is NRGMaestro™. NRGMaestro™ is a smart energy management software designed by the "Centre Suisse d'Electronique et de Microtechnique" (CSEM) to model energy systems of varying size, from individual households to entire communities. It models energy systems as directed graphs between devices and loads. By solving a Mixed Integer Linear Programming (MILP) problem it finds the optimal way to control the devices. For this thesis, the optimal control is the optimal charging and discharging power of the batteries over the control horizon. Figure (11) shows a simple example of an energy system modelled by NRGMaestro™. In this example, given the battery specifications, consumer load and forecasted solar power generation, NRGMaestro™ would output the optimal battery use sequence to minimize costs. A more complex community could be modelled on top of this simple example by adding other consumer nodes with their own storage and load.

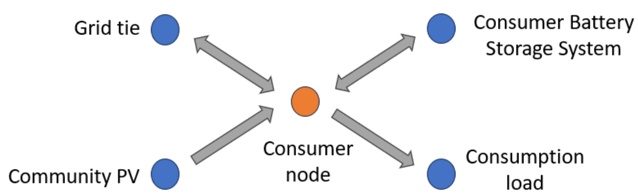


Figure 11: *Simple one-consumer NRGMaestro™ network*

NRGMaestro™ has been selected because it allows for receding horizon control and because it provides precise battery models. Receding horizon is a control method used for model predictive control in which the control horizon of the problem updates over time. This allows for real-time control and is therefore an important specification for the control framework to have to be applicable in real-life projects.

To implement this, the solar forecasts and the initial consumption loads of the players are seen as cyclical. This means that their values are identical at $t = 1$ an $t = T + 1$. That way the data can be extended to allow for a complete optimization at each timesteps of the control horizon. At $t = 0$, the optimal open loop profile is computed using the provided initial conditions and the data from $t = 0$ to $t = T$. Only the optimal values at the first optimization timestep are used to update the energy system following its dynamics. This updated state then serves as the new initial conditions for another open loop optimization, looking to find the optimal loads from $t = 1$ to $t = T + 1$. This is repeated until the entirety of the control horizon has been covered.

Regarding battery modelling, the setup until now assumed that consumption could be shifted freely during the day, without accounting for the physical reality of such actions. Using NRGMaestro™, the constraints on load changes can be defined more accurately, accounting for the dynamics and the physical limits of batteries. The optimal loads reached with NRGMaestro™ are more realistic. For each player i, the added constraints are:

- Battery dynamics: $SoC_i^{t+1} = SoC_I^t + \eta_{char} p_{char,I}^t + \frac{1}{\eta_{disc}} p_{disc,i}^t$.
- State of charge constraints: $SoC_{min} \leq SoC_i^t \leq SoC_{max}$.
- Charge and discharge constraints: $0 \leq p_{disc,i}^t \leq p_{disc,max}$ and $0 \leq p_{char,i}^t \leq p_{char,max}$.

Where, $SoC_i \in \mathbb{R}^T$ is the state of charge of the battery, $p_{disc,i} \in \mathbb{R}^T$ is the discharge power of the battery and $p_{char,i} \in \mathbb{R}^T$ the charging power. The charging and discharging efficiencies are respectively $\eta_{char} \in \mathbb{R}$ and $\eta_{disc} \in \mathbb{R}$.

Thanks to the use of NRGMaestro™, the inconvenience cost $J_{shifting,i}$ can also be defined more precisely. Instead of being an arbitrary cost proportional to the displacement, it can now be defined as an estimation of the cost of using the battery. This can represent an estimation of the impact of using the battery on its expected lifetime.

To be compatible with resolution in NRGMaestro™, the individual optimization problem have to be slightly modified. First, with the inclusion of the battery, the total consumption decision variable is not directly x_i, but the charging and discharging power $p_{char,i}$ and $p_{disc,i}$. Still these are related with: $x_i = e_i + p_{char,i} - p_{disc,i}$. The strategy vector is now defined as $y_i = (p_{char,i}, p_{disc,i}, p_{local,i}) \in \mathbb{R}^{3T}$. The objective functions also need to be reformulated. Indeed, NRGMaestro™ is designed to minimize the cost of the energy system designed. For each player i, the objective function thus becomes:

$$J_i(y_i) = {p_{local,i}}^\top c_{local} + (x_i - p_{local,i})^\top c_{import} + (\eta_{char} p_{char} + \frac{1}{\eta_{disc}} p_{disc})^\top C_{cyc} (\eta_{char} p_{char} + \frac{1}{\eta_{disc}} p_{disc}), \tag{48}$$

where the prices $c_{local} \in \mathbb{R}^T$ and $c_{import} \in \mathbb{R}^T$ are now vectors of length T and $C_{cyc} \in \mathbb{R}^{T \times T}$ is a diagonal matrix scaling the cost of using the battery. The individual objective problems are now:

$$\begin{aligned} \arg\min_{y_i} \quad & J_i(y_i) \\ \text{s.t.} \quad & x_i = e_i + p_{char,i} - p_{disc,i} \\ & p_{local,i} \leq x_i, \; p_{local,i} \leq s - z_{-i} \\ & SoC_i^{t+1} = SoC_I^t + \eta_{char} p_{char,i}^t + \frac{1}{\eta_{disc}} p_{disc,i}^t, \; \forall t = 1, \dots, T-1 \\ & SoC_{min} \leq SoC_i^t \leq SoC_{max} \\ & 0 \leq p_{disc,i}^t \leq p_{disc,max}, \; 0 \leq p_{char,i}^t \leq p_{char,max} \end{aligned} \tag{49}$$

The influence of the other players is now in the constraints of the optimization problem and not in the objective function. This makes the game at hand a Generalised Nash Equilibrium Problem [41]. Doing so means that the proof of NE uniqueness is no longer valid. Implementing the load game algorithm in NRGMaestro™ thus allows for receding horizon control and better battery modelling, at the exepnse of theoretical guarantees of uniqueness of Nash Equilibrium.

To integrate NRGMaestro™ in the load game algorithm, the only step that is modified is the resolution of the individual optimization problem. A separate energy system is modelled for each of the agents, which is composed of a battery, their load, a connection to the grid and a solar production load. The local optimisation problems are then restated as independent NRGMaestro™ problems. Each player i has its solar load defined as the local load minus the total local consumption of the other players: $s - z_{-i}$.

Looking back at the objective functions in (10), the individual problems are not linear but quadratic. Furthermore, there are no discrete variables in the problem. Using a MILP solver is thus not applicable here and a Quadratic Programming solver is instead used.

4.2 Open and closed loop behavior

The load game distributed algorithm was integrated in a control framework as described in the previous section. It was first tested in open loop, resulting in the behavior displayed in figure (12) for a EC with two members. Looking at the green curve representing the optimal load of the community, we can see that even with more constraints, the algorithm enables the community to consume the entirety of the available solar energy.

It is now also possible to look at the evolution of the battery state of charge along the horizon. The battery is discharged at the beginning of the simulation when demand is higher than available solar, reducing imports from the grid. Then during the day the battery is charged using the remaining

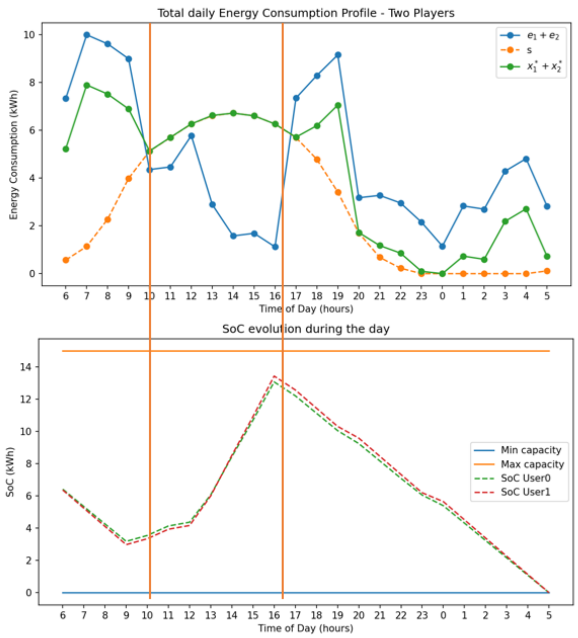
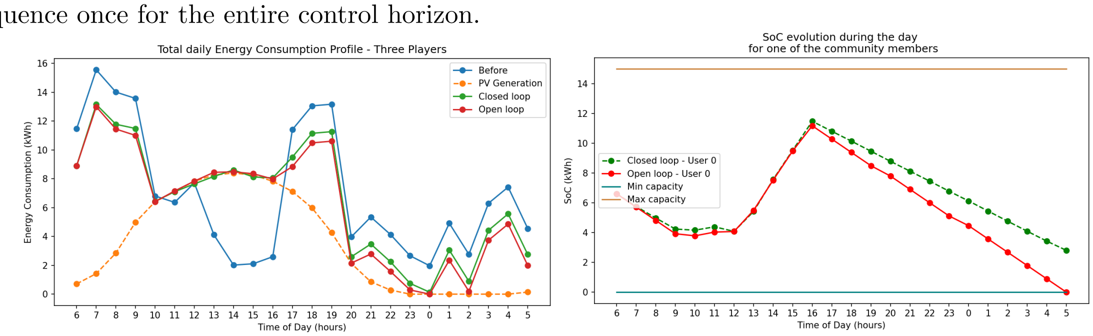


Figure 12: *Open loop optimal loads and battery use in a two members EC. The vertical lines emphasize the timing match between peak solar production and battery charging period*

solar energy after supplying the demand of the community. The stored energy is then discharged continuously until the end of the horizon, reducing the dependence on the grid. This battery behavior represents an optimal control approach, aligned with the goals of maximizing self-consumption and minimizing reliance on grid imports.

Building upon these results in open loop it is interesting to assess the algorithm behavior when used in closed loop. Closed-loop control involves iteratively adjusting the control inputs based on the current system state through feedback mechanisms and solving optimization problems at each timestep, ensuring dynamic adaptation. In contrast, open-loop control computes the entire control sequence once for the entire control horizon.



(a) Optimal loads

(b) Optimal battery use

Figure 13: *Optimal loads and battery use in closed and open loop for a two members EC*

Figure (13) shows the differences between open loop and closed loop results. As with open loop, the optimal closed loop load consumes the entirety of the solar energy available, showing that the algorithm preserves the wanted behavior in closed loop. The main difference between open loop and closed loop occurs during the discharge at the end of the horizon period. The open loop use of the battery results in a complete discharge by the end of the horizon whereas the battery remains charged in closed loop. This is because close loop accounts for the fact that there will be some need for stored energy in the next day, while open loop does not see further than the end of the horizon.

The above results show that the algorithm designed in this project can be successfully implemented in closed loop receding horizon control, and could thus be used for real time applications.

4.3 Impact of the scale of local energy available

The previous section showed that the algorithm can be used successfully in a receding horizon framework. Still, for now it has only been tested in a arbitrary environment. It is important to evaluate how the performance of this approach is sensitive to specific parameters in the model, to see how resilient it can be in varying conditions.

The most important parameter influencing performance is the quantity of solar energy locally produced. To assess its impact, the other parameters first need to be set. The following list of parameters are used for all the following analysis.

Battery parameters:

- Capacity: 15kWh
- Charging and discharging efficiency: 95%
- Maximum charge and discharge rate: 7.5kW
- Starting SoC: 7.5kWh (50%)

Economical parameters:

- Grid electricity: 0.2 CHF/kWh
- Local electricity: 0.1 CHF/kWh
- Feed-in tariff: 0.05 CHF/kWh

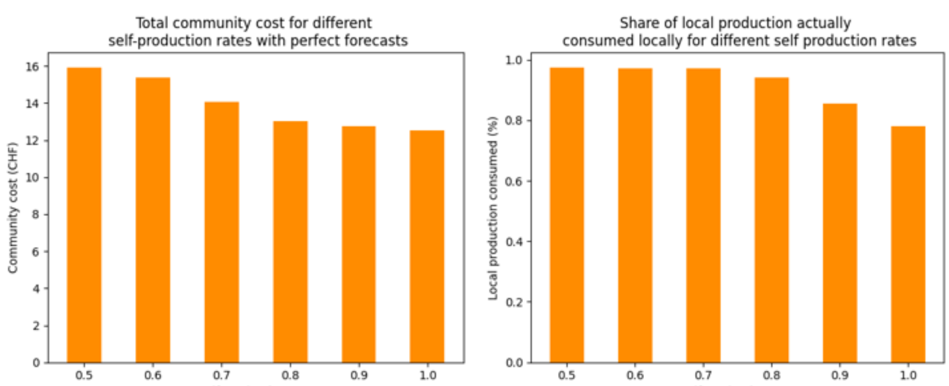
Community parameters:

- Size: 3

With these parameters fixed, it is now needed to quantify the amount of available solar with respect to the energy demand. To do so, the parameter $\gamma \in \mathbb{R}$ is introduced. It represents the self production rate, which is the share of total demand over a horizon which can be produced locally over the same period. It is defined as:

$$\gamma = \frac{s^\top 1}{x^\top 1}. \tag{50}$$

The higher the γ, the more solar energy is available. Note that γ represents aggregate values of the parameters over the horizon, but do not assess their time granularity. Having $\gamma = 1$ does not necessarily mean that no electricity is imported from the grid. This depends on when the consumption takes place.



(a) Costs

(b) Local energy consumed

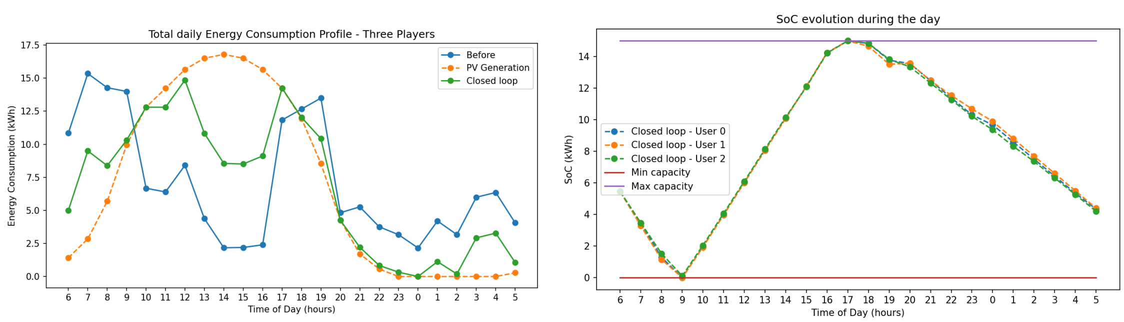
Figure 14: *Total community costs and share of local resources consumed locally for different self production rates*

Using γ, the effect of the quantity of solar energy has been analysed. To do so the optimal closed loop control sequence has been simulated for different values of γ ranging from 0.6 to 1 and the total community costs have been calculated each time. The costs are calculated ex-post, by looking at the local and total consumption loads of each players. The imported loads are computed as the differences between them and the cost for each player can then be calculated using the economical parameters presented above. These are summed to get the community costs.

Figure (14a) displays the results. As expected, the more solar energy is available the lower is the cost for the community. This is because a greater share of the demand can be met using the cheaper local resources. However, for γ greater than 0.8, the cost reaches a plateau, after which increasing the amount of local energy does not reduce the cost.

To understand why this is the case, figure (14b) shows the share of available local energy really consumed locally for different values of γ. For values of γ where the cost was decreasing, the entirety of the available solar energy is consumed. The amount of solar energy thus increases, leading to cost savings. However, for $\gamma = 0.8$ and onwards, this percentage gradually decreases as γ increases. As the share of an increasing value decreases, the absolute amount of solar energy consumed remains nearly constant, explaining that no additional savings are induced by increasing γ.

The optimal loads and battery use with $\gamma = 1$ are presented in figure (15). Looking at the loads, it is clear that not all the PV energy available is consumed by the community. There is also still some dependence on the grid at the beginning of the horizon and at the end. It can be surprising to have this apparently suboptimal result, but this can be explained when looking at the SoC profiles. Indeed, without using the entire solar energy available, the battery of each player already reaches its maximum capacity. If more PV energy were consumed earlier on to match the available resources at that time, the maximum capacity would have been reached earlier on, causing the same amount of local resources not to be consumed. Of all the possible loads maximising self-consumption, the resulting loads are the one minimizing battery use cost.



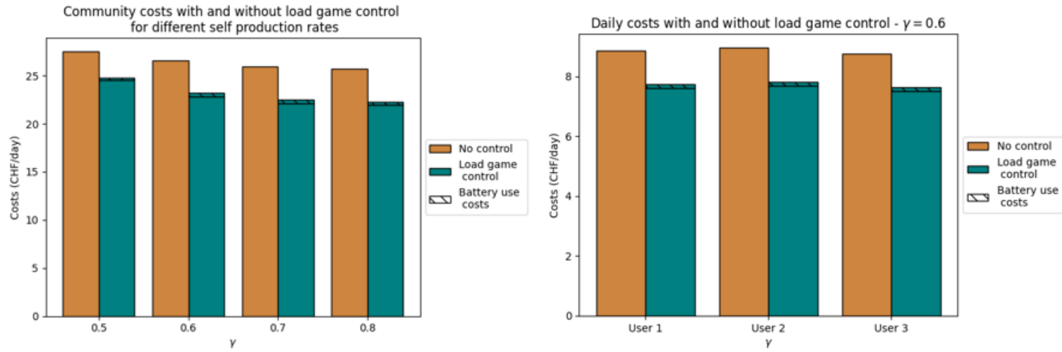
(a) Optimal loads

(b) Optimal battery use

Figure 15: *Optimal loads and battery use in closed and open loop with* $\gamma = 1$

By analysing the controller behavior for different values of γ it can be seen that using the load game algorithm as an optimizer offers the best possible performance given the hardware specifications at hand.

Figure (16a) shows the savings induced by adopting load game control compared to the initial loads for different values of γ. The dashed section of the bar represents the costs of using the battery. Even when accounting for these costs, the load game control offers better economical performances for the community than the default loads for all the values of γ tested. The more solar energy there is the bigger the savings are, until the point where the physical limits of the batteries are reached. Figure (16b) represents the costs for each agent in the community when $\gamma = 0.6$. For the simulation setup trialled, it appears that the savings for the community are evenly distributed within the community, ensuring that no member is tempted to leave. Adopting a load game controller thus seems interesting for energy communities, because it leads to important and even cost saving for all its members.



(a) Community level

(b) Agent level

Figure 16: *Community costs with and without load game control*

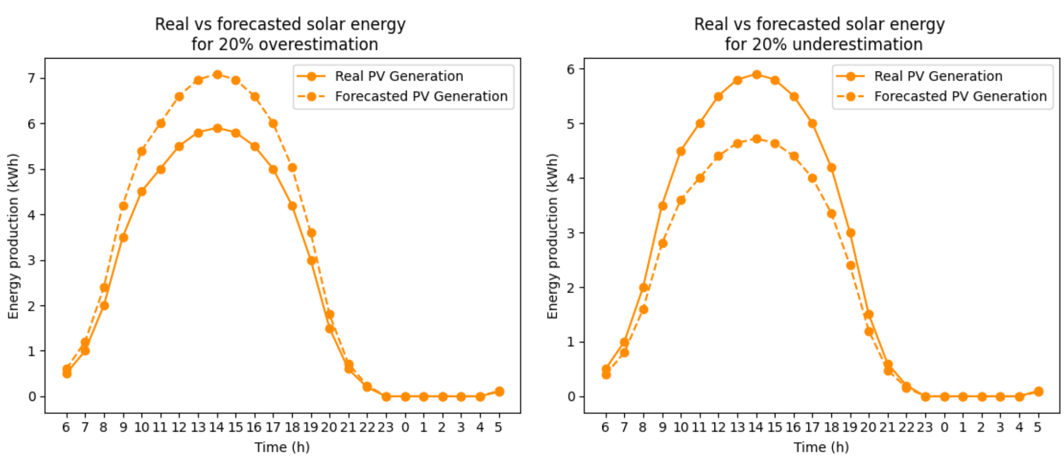
4.4 Uncertainty analysis

The previous results are promising, but they are coming from a setup with perfect forecast. It was indeed assumed that the solar load over the control horizon was perfectly known when deciding on the control input. However forecasts are never perfect and often entail some strong uncertainty. Evaluating whether the performance of the algorithm is robust to such forecasting errors is therefore critical to assess its worth for real-life applications.

To model this error in forecasting, each solar installation is now composed of two loads: the real solar load and the expected one. The control decisions will be made using the expected loads, but the updates to the system state will be based on the real solar loads.

For example, assume that at a timestep the optimal action is to charge all the agents' batteries with 5kWh because the forecast expected 15kWh of solar production. If in reality there is only 12kWh of solar production, then the state update will be that all the batteries will be more charged by 4kWh and not 5kWh.

For simplicity, the studied uncertainties are constant over- or underestimations. Along the entire horizon, the forecast will be constantly $X\%$ higher or lower than the real solar power available. During the night, as the real power is 0kWh, so will be the forecasted power. Figure (17) shows the exact and forecasted for scenarios with 20% over- and underestimations.



(a) 20% overestimation

(b) 20% underestimation

Figure 17: *Exact and forecasted solar loads for different uncertainty scenarios*

To account for the physical limitation of the battery, there is a difference made between the updates due to overestimation and underestimation errors at the studied timestep. The updates on the system state are done as follows:

- Overestimation: $SoC_t = min(Capacity, SoC_t + \Delta s)$.
- Underestimation: $SoC_t = max(0, SoC_t + \Delta s)$.

Here $Capacity \in \mathbb{R}$ is the maximum capacity of the battery and $\Delta s \in \mathbb{R}$ is the difference between real and forecasted solar energy felt by each community member. These updates are done before stepping to the next timestep in the closed loop process.

To avoid initial condition bias, the simulation runs for 48h hours, and only the last twenty four hours are kept for analysis.

4.4.1 Nominal prices

The effect of uncertainty is first assessed using the price scheme described in section 4.2. The total community cost is compared for different scenarios ranging from 30% underestimations to 30% overestimation. This analysis also includes a perfect forecast scenario to use as benchmark. This is done for $\gamma = 0.6$ and $\gamma = 0.8$.

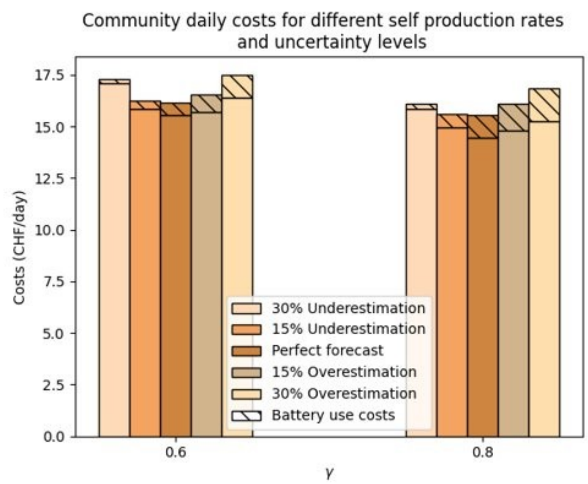


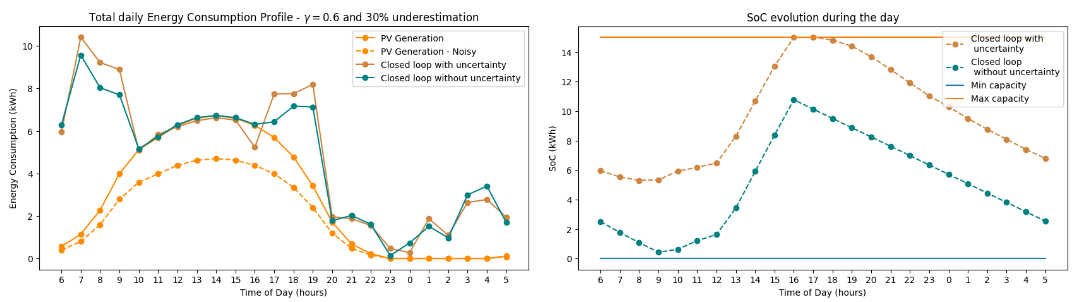
Figure 18: *Impact of imperfect forecasts on community costs for different self-production rates*

Figure (18) shows the community costs for the different scenarios studied. The first thing worth noticing is that the cost of importing energy (without accounting for battery use) is higher for all uncertainty scenarios than for perfect forecast, underlining the adverse impact that noise in forecast can have. For both values of γ, this increase is stronger for underestimation than overestimation. This is because as the expected solar load is smaller, the incentive to shift consumption away from the initial load is weak. Therefore, the resulting loads still entail strong peaks in period of low solar production, requiring a bigger amount of costlier imported electricity.

This behavior can also be understood by looking at the battery costs. These are indeed the lowest for underestimation scenarios and gradually decrease as the underestimation increases. This behavior occurs for both values of γ and underlines the fact that with underestimating forecast, the optimal loads are reached with a smaller use of the battery and are therefore closer to the default, non-optimized consumption loads.

The loads of the entire community and SoC evolution of one of its member for perfect forecasts and 30% underestimations are represented in figure (23). As expected, the morning peak is higher for the imperfect forecast. As less solar energy is expected to charge the battery later on, less energy can be used at the beginning of the day. This is also seen in the SoC, with the perfect forecast nearly fully discharging the battery in the morning while the imperfect one remains around 40% capacity.

As the real amount of PV produced is greater than what is anticipated, the battery still gets charged and even reaches maximum capacity around 4pm. By not discharging the batteries deeply enough in the morning, they do not have enough available storage space to benefit from the entirety of the local energy. Afterwards, both loads are comparable, with the notable difference that the perfect forecast batteries operate at a lower state of charge, leaving room to accommodate the entire solar production of the next day. Underestimation errors in the forecast thus have a negative impact on performance both by increasing imports and reducing local consumption.



(a) Loads

(b) SoC

Figure 19: *Optimal loads and state of charge using perfect forecasts and forecast with 30% underestimation*

Going back to figure (17) for overestimation errors, the cost of energy increases, and so does the cost of battery use. As the expected solar production is higher, the optimal control action is to use the battery more to discharge during low production period because the expected resources available are sufficient to recharge afterwards. However as the real PV production is smaller, the batter will charge using the entire available resources, but will also charge the battery directly with imported energy too. As the charging and discharging efficiencies are not equal to 1, this incurs some direct losses and thus a larger amount of imported electricity is required overall. Overestimation errors thus have negative impact on performance because they increase battery use costs and energy costs.

The above analysis provide interesting insights on the qualitative impact of uncertainty on performance, but when looking quantitatively at the costs increases caused by uncertainty, they seem relatively low. In the worst cases, when uncertainty reaches 30%, the cost does not increase by more than 10%. This is in part due to the fact that the local and import costs are not very different.

Using a load game distributed optimizer seems to ensure that under nominal conditions, forecasting errors only cause small and acceptable losses in the performance of the controller.

4.4.2 Peak prices

To complement the previous analysis, the impact of uncertainty should be assessed in a context where making errors is particularly problematic. An example of such context is periods of peak demand, during which the import costs can rise significantly, making any extra unplanned imports very costly for the community. To imitate such an environment, the cost of importing electricity is now set at 2CHF/kWh, The value selected is voluntarily extreme, to make the results clearer.

The same scenarios are simulated again in this new environment. Figure (20) shows the resulting costs. As the cost of importing electricity has increased tenfold compared to the nominal conditions, the battery use costs are now negligible. Intuitively, the cost for perfect forecast has also increased more for $\gamma = 0.6$ than $\gamma = 0.8$, because even with perfect knowledge, it is more dependent on the more expensive grid energy.

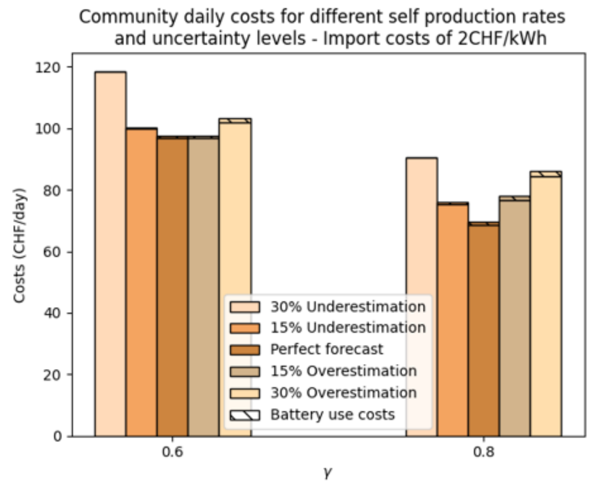


Figure 20: *Impact of imperfect forecasts on community costs for different self-production rates with peak tariffs*

For overestimation, errors cause increase in costs which are comparable in scale to the nominal case for $\gamma = 0.6$. However, the increase in cost for all other scenarios is much stronger than in the nominal case, reaching close to 30% for the worst cases.

Furthermore, when looking at underestimations for both $\gamma = 0.6$ and $\gamma = 0.8$, there seems to be a nearly exponential increase in cost with underestimation. The cost augmentation is indeed vastly superior when passing from 15% to 30% than from perfect forecast to 15%.

Uncertainty has a non-negligible impact on performance which gets even worse with the scale of the error. While for nominal conditions the controller could be left as it is to handle noise, a change is required to mitigate its impact in peak tariff regime.

4.4.3 Battery buffer

A possible way to reduce the impact of uncertainty would be to introduce a buffer zone within the battery. By motivating the battery state of charge to remain approximately between 20% and 80%, there would be some room available to better account for the deficit or excess of solar energy. If the battery operates far from maximum capacity and the controller underestimates the available PV, it would still be possible to use the excess production to charge the battery more.

Introducing such a concept would however sacrifice some part of the performance. Indeed, providing an incentive to not use the full depth of the battery equates to artificially reduce the size of the battery. As it was shown earlier, the physical specifications of the batteries are the limiting factor and smaller batteries leads to worse performances. The use of a buffer thus leads to a tradeoff between deterministic performance and robustness to forecast errors.

For this project, the buffer bounds have been set to 20% and 80%. These bounds ensure that there is enough room for handling noise while still making the battery deep enough for good performance. In further work, these bounds could also be refined to find the optimal point in the tradeoff.

The buffer constraints are also designed as soft constraints. This means that breaking them is possible but is costly and thus not interesting when optimizing. If breaking them leads to a benefit outweighing the penalty, it is still possible to have a state of charge outside of the buffer zone.

To model this soft constraint, a new variable $SoC_{safe} \in \mathbb{R}^T$ is introduced. It is defined such that each timestep, $SoC_{safe,t} \in [0.2, 0.8]$. The penalty cost caused by a deviation from the buffer zone is then calculated as:

$$J_{i,SoC}(y_i) = (SoC - SoC_{safe})^\top M (SoC - SoC_{safe}), \tag{51}$$

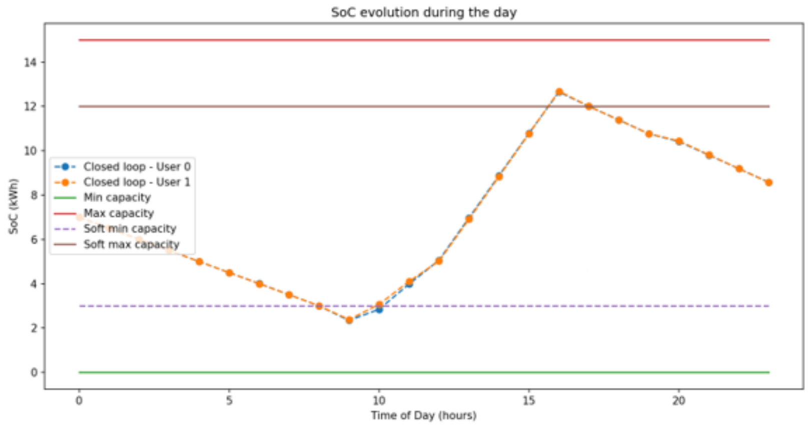


Figure 21: *State of Charge evolution with buffer soft constraint*

where $M \in \mathbb{R}^{T \times T}$ is a diagonal scaling matrix. The penalty cost is thus computed as the squared distance between SoC and SoC_{safe} at each timestep. That way, when SoC is within the buffer zone, the penalty cost is zero (because $SoC_{safe} = SoC$ is possible), and when it is outside, the further away the costlier it is.

The diagonal value of M is empirically tuned to get the behavior displayed in figure (21). With this value, the state of charge is mostly in the safe zone, but the penalty is sufficiently low to allow for some deviation if it could improve performance. To get this behavior, $m_{i,i} = 0.001$.

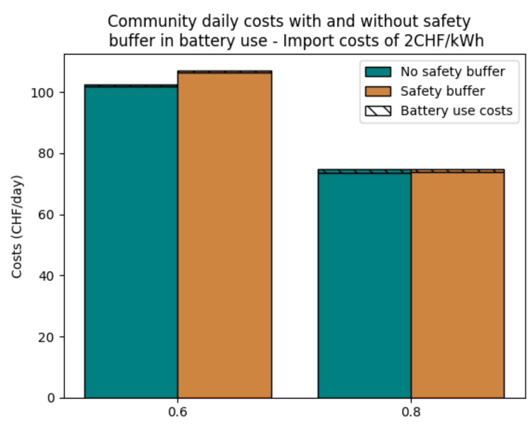
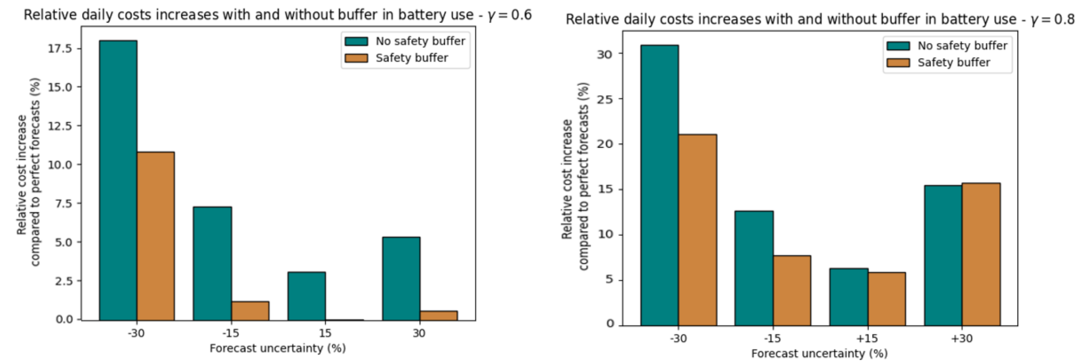


Figure 22: *Community cost with perfect forecast with and without battery buffer*

Figure (22) shows the community costs with and without buffer when using perfect forecasts. For $\gamma = 0.6$ the cost is slightly bigger when using the buffer, but the increase is not massive, being lower than 10%. For $\gamma = 0.8$, the increase is even lower, barely visible on the scale of the figure. This means that the sacrifice induced by the use of a buffer zone is nearly marginal. Integrating a sefety zone in the battery use is therefore not so detrimental to performance with perfect forecast, and if it yields a more robust behavior to noisy forecasts, it should be adopted in the controller design.

The difference in the impact of uncertainty on performance with and without buffer is presented in figure (23). For $\gamma = 0.6$ and $\gamma = 0.8$, the percentage increase compared to perfect forecast is displayed for the different uncertainty scales, with and without safety buffer.



(a) $\gamma = 0.6$

(b) $\gamma = 0.8$

Figure 23: *Uncertainty induced cost increases with and without safety buffer in the battery*

For $\gamma = 0.6$, errors in forecast still induce cost increases, but at a much smaller scale. The worst case is for forecasts underestimating by 30%, but even in this scenario the cost increase is only of approximately 11% compared to more than 17.5% without using the safety buffer. For all the other scenarios including a safety buffer is even more beneficial, decreasing by at least five time the excess costs caused by forecasting errors. With 15% overestimation using a buffer even seems to nearly cancel out the extra costs.

For $\gamma = 0.8$, the effect of the safety buffer is also beneficial for underestimation errors. In both underestimations scenarios the inclusion of a safety buffer reduces the extra cost by around 30%. Recalling that the deterministic performance was close to being the same with and without buffer these percentage reduction also translate directly to around 30% reduction in the absolute value of the costs induced by uncertainty, underlining the benefits of using a buffer.

For overestimation errors the use of the buffer seems less interesting. The cost is indeed only slightly lower with 15% overestimation and it is even higher for 30% overestimation. This can be explained by the fact that with $\gamma = 0.8$ the amount of available solar energy is huge. Expecting 30% more in this context is a bigger error than for $\gamma = 0.6$. Therefore the expected benefits of using the battery at full capacity is very rewarding. The penalty of going out of the safety zone is completely outweighed by the prospect of consuming more of the less expensive expected local production. With this much expected solar production, the effect of the soft buffer constraints vanishes. The resulting battery behavior thus becomes similar with and without the buffer constraints. This can also be seen in the deterministic scenario, explaining why the cost with the buffer is the same as without it. By tuning the scaling matrix M differently, it could be possible to mitigate this effect and have a valuable use of the buffer even in these conditions.

Given the low impact on deterministic performance and the appealing excess cost reduction in environments with imperfect forecasts, the inclusion of a safety region in the battery thus seem promising. In both nominal and peak tariff conditions, managing an energy community using load game based controller is as a possible way to guarantee sound performance even when using imperfect forecasts.

4.5 Load game control: conclusion

The previous sections have shown that the distributed load game algorithm introduced in this project can be successfully integrated in a receding horizon control framework. The algorithm was indeed implement in the smart energy management software NRGMaestro™. Even with the more realistic and strict modelling constraints of NRGMaestro™, it managed to compute optimal control sequences both in open and closed loop. By assessing how the amount of solar energy available influences the performance of the controller, it was also shown that the load game algorithm maximised the possible self-consumption of a community given the physical limitations of the batteries used.

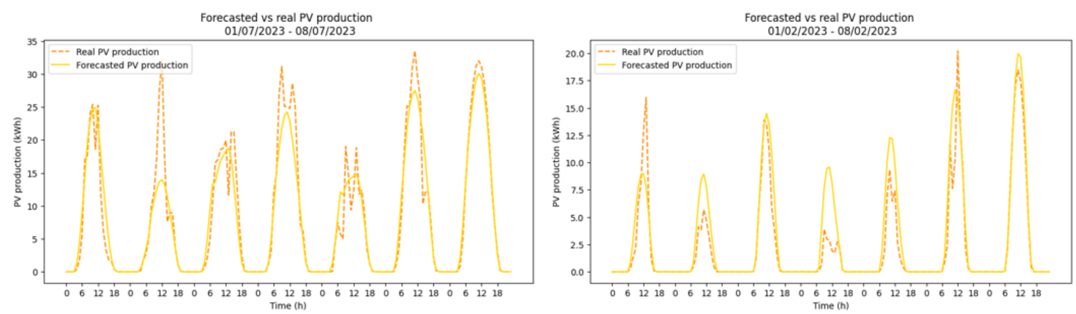
The impact of imperfect forecasts has also been studied. It was demonstrated that under nominal pricing conditions, although errors in forecasts qualitatively influence the cost for communities, the scale of the increases remain relatively low and are therefore not too worrisome. For peak tariff conditions, the effect of uncertainty was handled by implementing a safety region in the battery range, leaving some room for unexpected solar production excesses or deficits. By using this region, the controller managed to mitigate some of the extra costs caused by uncertainty.

Designing a controller around the load game optimizer thus ensures a good behavior with respect to noise, both in nominal and peak tariff conditions.

5 Application to real-life data

Building on the results presented in the previous section, it is interesting to evaluate if the proposed algorithm keeps the same performance when applied to real data. Having an algorithm that performs well with toy data does not necessarily guarantees it will be the case in real life applications.

Instead of using simulated solar loads, the simulation environment is now built around real PV energy recordings from a PV installation close to Zurich. The dataset used contains forecasted and real solar power time series for several years and locations, with quarter hourly granularity. One installation is selected from the dataset, and two weeks of data are sampled. For statistical significance, one week is sampled in winter, from February 1st 2023 to February 8th 2023, and the other week in summer, from July 1st 2023 to July 8th 2023. Doing so ensures that the algorithm is tested on several varying patterns. The real and forecasted solar loads are then scaled to the size of the energy communities using γ. For the winter week, $\gamma = 0.6$ and for the summer week $\gamma = 0.8$, to represent the fact that more energy is produced in summer.



(a) Summer

(b) Winter

Figure 24: *Real and forecasted solar loads in summer and winter*

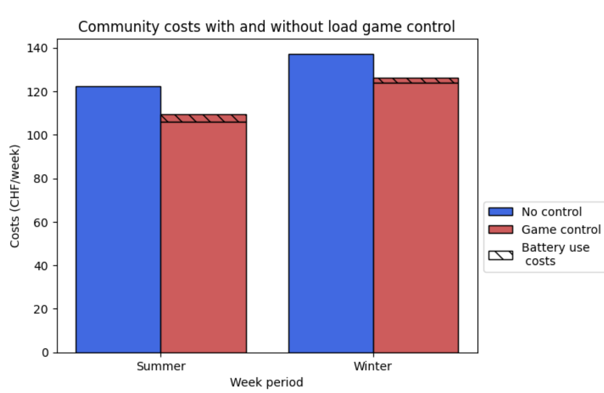
Figure (24) shows the resulting loads. These loads are less smooth than the toy data which we decided to use until now. There is also greater inter-day variations too. Regarding the difference between real and forecasted loads, there is a combination of over- and underestimations of different percentages. The impact of uncertainty can thus be expected to be manageable as with the toy data in section 4.4.

5.1 Load game results

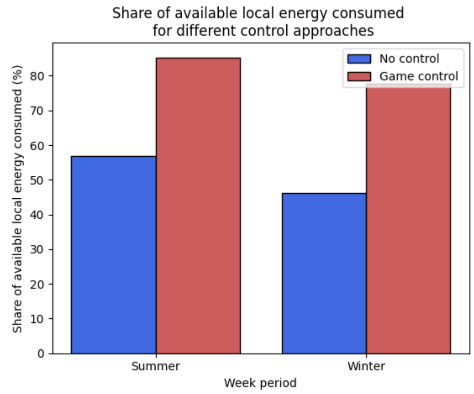
The load game algorithm is used to find the optimal battery control sequence over both one week control horizons. The environment is set back to its nominal case, with an import cost of $0.2CHF/kWh$.

The results are presented in figure (25). For both periods, implementing load game control is economically beneficial, reducing the costs by around 10%, even when accounting for battery use costs. Intuitively, the difference between default costs and optimal costs is even greater in the summer, as there is more cheap local energy. The incentive to use the battery is thus greater, as shown by the bigger battery cost.

It is also interesting to note that these costs reductions are based on noisy forecasts, with sometimes important errors, like on July 4th. Even with such imperfections, the load game algorithm still manages to provide some savings, underlining its value in uncertain environment.



(a) Costs



(b) Local energy consumed

Figure 25: *Total community costs and share of local resources consumed locally*

The cost difference between default and optimal control might seem relatively low, but this originates from the small difference in price between imported and local energy under nominal conditions. Looking at the share of available resources consumed locally with and without load game control in figure (25b), the impact of the algorithm becomes clearer. The share indeed increases by more than 20% in summer and winter. In both case it does not reach 100% because of the physical limitations of the battery but it is still strongly improved compared to the reference amount.

In the context of energy communities aiming to maximize local energy utilization, the adoption of a load game control framework thus proves to be a highly valuable approach.

5.2 Controller benchmark

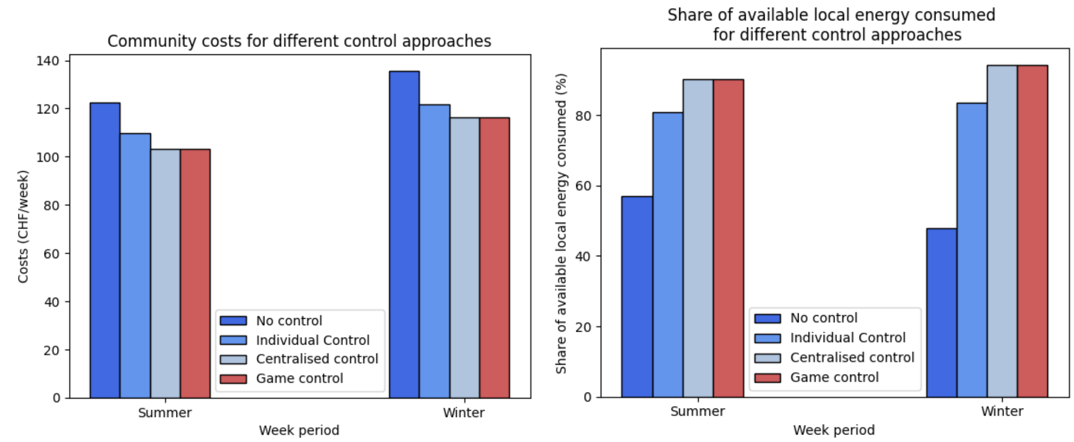
In the pursuit of optimizing energy management within communities, it becomes imperative to assess and benchmark the effectiveness of the load game algorithm with other existing approaches. While the earlier findings suggest positive outcomes associated with the utilization of a load game approach, it remains imperative to subject these results to a comparative analysis with other methodologies.

It was previously mentioned that other ways to manage energy communities could be categorized between individual control and centralized control. By comparing the load game approach to these two, it will be possible to identify any added value brought forward by the algorithm designed in this project. The different methodologies are summarised below:

- **Individual control**: Each agent optimizes its load individually, without accounting for the decision and strategies of the other agents. No information is shared between the agents.
- **Centralised control**: The community manager solves a centralised problem including each agents loads as decision variables. All the agents behaviors are public and shared.
- **Load game**: Each agent solves an individual problem, taking into account the aggregate strategy of the community. Only partial information in the form of total local consumption is shared.

The different approaches vary in the amount of information shared between the agents, with the load game approach being a middle ground between the two other. By having no information exchange, individual control risks reaching sub-optimal results because of a lack of coordination between agents. For the load game to be interesting, it thus needs to outperform the individual control, in terms of community costs and self-consumption rate.

Compared to the centralized approach, the load game shares a minimal amount of information, and thus better preserves privacy. By being distributed, it also avoids the computational pitfalls of centralized resolution. A satisfactory result would thus be that the load game matches the performance of centralized control, while providing the other benefits mentioned here.



(a) Costs (b) Local energy consumed

Figure 26: *Total community costs and share of local resources consumed locally for different control approaches*

Figure (26) compares the results for the three control approaches when used on the solar data described in section 5. For comparison fairness, individual and centralized control have also been implemented in NRGMaestro™, with the same design parameters as the one described in section 4.2. The performance of the default loads are also shown for reference.

First, both for costs and share of local energy used all control strategies provide an improvement to the performance of the initial loads. Then, it is also clear that individual control is the worse of all the strategies. Both in summer and winter using individual control is the strategy resulting in the highest cost of energy for the community. It also leads to a higher share of local resources being wasted by not being consumed within the community.

The results for load game control and centralized control appear identical. Both control methodologies yield the same costs for the community and the same share of local resources consumed locally. The load game approach however reaches this results while requiring less information sharing.

By looking at the resulting loads on a specific day, it is possible to understand better what causes the difference in performance between approaches. Figure (27) shows the resulting loads on February 2nd 2023 with perfect forecasts. The individual control load has a peak in consumption at 9am, which outmatches the solar energy available at that moment. This peak is the result of the lack of communication between agents. As seen by individuals, the available resources at that time are sufficient to match their needs so the load is shifted to this time. However, as all agents do the same, the community consumption, which is the sum of each agent consumption is now greater than the available resources, causing imports from the grid. As the loads were all shifted at that time, there is less demand between 10am and noon, resulting in more of the local energy not being consumed, explaining the results in figure (26).

The consumption profiles of load game and centralized control are nearly identical. As opposed to individual control, they lead to a community load of exactly the local production at 9am and then consume more of the available local resources between 10am and noon. This explains their better performance. After 4pm, all the loads are highly similar. As there is no solar production anymore, these loads just corresponds to the discharge profile of the battery minimizing battery use costs.

Thus, by comparing the methodology developed in this project to other already existing methods, the potential of load game control for EC energy management has benn emphasized. It indeed outperforms individual control by allowing for some coordination between members and matches the results of centralized control, while requiring significantly less sharing of personal information between members.

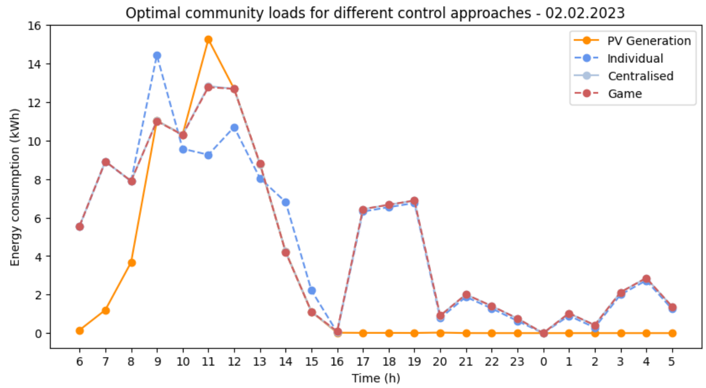


Figure 27: *Optimal loads for different control approaches - 02.02.2023*

6 Conclusion and future work

In conclusion, this project has designed a distributed algorithm for battery control in energy community which maximizes self consumption and minimizes grid dependence. To do so, the first step was to formulate the energy dispatch problem as a game, with the possible battery use as the strategy choices available for each player. The interaction among participants was represented by considering their impact on the available local energy for each player. This modeling resulted in the game having a unique solution, the Nash Equilibrium. An iterative algorithm was designed to converge to this point distributively, only sharing the expected total local consumption at each iterations between player. By adding a safety secondary algorithm, theoretical guarantee of convergence to the Nash Equilibrium was ensured, even if from the empirical evidence it seemed possible that the primary algorithm could always converge.

The algorithm was then successfully integrated in a receding horizon optimal control framework, maintaining its performance in closed loop. By analysing the sensitivity of the algorithm to the amount of local resources available it was demonstrated that given some hardware specifications, the load game approach provided the best possible control sequence to maximise self-consumption. The resilience of load game control to errors in forecasts was also underlined. The adverse effect of uncertainty on community prices was indeed minimal under nominal conditions, and was mitigated in peak tariff periods by introducing a safety buffer in the battery range of use.

The load game approach also performed well on real life data, improving self-consumption in the winter and in the summer over varying solar production profiles. It outperformed individual control thanks to its better collaboration between agents and matched the performance of centralized control while requiring much less information sharing.

The load game algorithm presented in this project thus appears as an interesting and resilient tool for energy community management.

To further improve the work done during this project, a first step would be to pursue its theoretical analysis. Although the load game algorithm converged with the right learning rate in all the experiments it was tested on, there has been no theoretical proofs of this behavior, motivating the introduction of the backup projected gradient descent algorithm. By providing theoretical convergence guarantees this secondary algorithm could be eliminated, improving the quality of the results.

Another point of improvement would be to diversify the energy communities the algorithm was applied to, to get some more holistic results. For now the algorithm was tested on energy communities with a central, shared PV installation and members who were only consumers. This setup could represent for example a residential building with several apartments and a PV installation on the roof. A significant share of energy communities are however composed of individual households each having their own PV panels. Testing the algorithm with prosumers would thus also be useful in determining its applicability. Along those lines, diversifying the initial loads of the agents could also be valuable, as there were only different because of some noise for now.

Finally, integrating the load game approach in more complex energy systems would also be of interest. For now the energy systems considered are only composed of batteries, PV panels and loads. In reality energy systems are more complex especially if the electrical system is connected to the heating system. Some of the other devices also require integer variables to be modelled (heatpump which are on or off), which could potentially modify the properties of the game solution or the way to find it. This additional work could thus make the load game approach beneficial for a wider range of energy community and could even be broadened to energy systems of other scales like districts or microgrids.

References

[1] Carlo Vezzoli, Fabrizio Ceschin, Lilac Osanjo, Mugendi K. M'Rithaa, Richie Moalosi, Venny Nakazibwe, and Jan Carel Diehl. Distributed/decentralised renewable energy systems. In Carlo Vezzoli, Fabrizio Ceschin, Lilac Osanjo, Mugendi K. M'Rithaa, Richie Moalosi, Venny Nakazibwe, and Jan Carel Diehl, editors, *Designing Sustainable Energy for All: Sustainable Product-Service System Design Applied to Distributed Renewable Energy*, Green Energy and Technology, pages 23–39. Springer International Publishing, 2018.

[2] IEA. Unlocking the potential of distributed energy resources – analysis. Technical report, International Energy Agency, 2022.

[3] Anna-Riikka Kojonsaari and Jenny Palm. Distributed energy systems and energy communities under negotiation. *Technology and Economics of Smart Grids and Sustainable Energy*, 6(1):17, 2021.

[4] Batchu Rajasekhar, Naran Pindoriya, Wayes Tushar, and Chau Yuen. Collaborative energy management for a residential community: A non-cooperative and evolutionary approach. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(3):177–192, 2019.

[5] Hesam Farzaneh, Hamed Kebriaei, and Farokh Aminifar. Deterministic Mean Field Game for Energy Management in a Utility with Many Users. In *2018 Smart Grid Conference (SGC)*, pages 1–6, Sanandaj, Iran, November 2018. IEEE.

[6] Hesam Farzaneh, Mohammad Shokri, Hamed Kebriaei, and Farrokh Aminifar. Robust Energy Management of Residential Nanogrids via Decentralized Mean Field Control. *IEEE Transactions on Sustainable Energy*, 11(3):1995–2002, July 2020.

[7] Maksym Koltunov, Simon Pezzutto, Adriano Bisello, Georg Lettner, Albert Hiesl, Wilfried van Sark, Atse Louwen, and Eric Wilczynski. Mapping of energy communities in europe: Status quo and review of existing classifications. *Sustainability*, 15(10):8201, 2023. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute.

[8] Council of the European Union European Parliament. Directive (eu) 2018/2001 of the european parliament and of the council on the promotion of the use of energy from renewable sources. *Off. J. Eur. Union*, pages 82–209, 2018.

[9] IEM. Directive (eu) 2019/944 on common rules for the internal market for electricity and amending directive 2012/27/eu. *Off. J. Eur. Union*, pages 125–199, 2019.

[10] SuisseEnergie. Guide pratique de la consommation propre. Technical report, SuisseEnergie, 2023.

[11] Sébastien Faivre. Rcp - regroupement dans le cadre de la consommation propre (rcp). Technical report, CECB, 2022.

[12] Climkit. Comptage électrique et regroupements, 2023.

[13] NeoVac. Rendre le rcp plus équitable, 2023.

[14] Dario Raffaele, Vincent Bolwerk, Lisette ten Boske, and Ivar Tjallingii. Peer to peer energy trading, 2023.

[15] Entrnce. Local energy trading for energy communities, 2023.

[16] Chao Long, Jianzhong Wu, Chenghua Zhang, Lee Thomas, Meng Cheng, and Nick Jenkins. Peer-to-peer energy trading in a community microgrid. 2017.

[17] Valeria Casalicchio, Giampaolo Manzolini, Matteo Giacomo Prina, and David Moser. Optimal Allocation Method for a Fair Distribution of the Benefits in an Energy Community. *Solar RRL*, 6(5):2100473, 2022. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/solr.202100473.

[18] Miguel Manuel de Villena, Samy Aittahar, Sebastien Mathieu, Ioannis Boukas, Eric Vermeulen, and Damien Ernst. Allocation of locally generated electricity in renewable energy communities, January 2022. arXiv:2009.05411 [cs, eess].

[19] Nian Liu, Xinghuo Yu, Cheng Wang, Chaojie Li, Li Ma, and Jinyong Lei. Energy-Sharing Model With Price-Based Demand Response for Microgrids of Peer-to-Peer Prosumers. *IEEE Transactions on Power Systems*, 32(5):3569–3583, September 2017. Conference Name: IEEE Transactions on Power Systems.

[20] Adewale Zakariyahu Obaro, Josiah Lange Munda, and Adedayo Adedamola Yusuff. Modelling and energy management of an off-grid distributed energy system: A typical community scenario in south africa. *Energies*, 16(2):693, 2023. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[21] Chao Long, Jianzhong Wu, Yue Zhou, and Nick Jenkins. Peer-to-peer energy sharing through a two-stage aggregated battery control in a community Microgrid. *Applied Energy*, 226:261–276, September 2018.

[22] Abdulrahman A. Alrobaian and Ali Sulaiman Alsagri. Multi-agent-based energy management for a fully electrified residential consumption. *Energy*, 282:128852, November 2023.

[23] Cunbin Li, Xuefeng Jia, Ying Zhou, and Xiaopeng Li. A microgrids energy management model based on multi-agent system using adaptive weight and chaotic search particle swarm optimization considering demand response. *Journal of Cleaner Production*, 262:121247, 2020.

[24] Mahmoud Elkazaz, Mark Sumner, and David Thomas. A hierarchical and decentralized energy management system for peer-to-peer energy trading. *Applied Energy*, 291:116766, June 2021.

[25] Suyang Zhou, Zijian Hu, Wei Gu, Meng Jiang, and Xiao-Ping Zhang. Artificial intelligence based smart energy community management: A reinforcement learning approach. *CSEE Journal of Power and Energy Systems*, 5(1):1–10, March 2019. Conference Name: CSEE Journal of Power and Energy Systems.

[26] Xinghua Liu, Shenghan Xie, Chen Geng, Jianning Yin, Gaoxi Xiao, and Hui Cao. Optimal Evolutionary Dispatch for Integrated Community Energy Systems Considering Uncertainties of Renewable Energy Sources and Internal Loads. *Energies*, 14(12):3644, January 2021. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute.

[27] Heiko Hotz. A short introduction to game theory, 2006.

[28] Mohadese Movahednia, Hamid Karimi, and Shahram Jadid. A cooperative game approach for energy management of interconnected microgrids. *Electric Power Systems Research*, 213:108772, 2022.

[29] Changsen Feng, Fushuan Wen, Shi You, Zhiyi Li, Farhad Shahnia, and Mohammad Shahidehpour. Coalitional game-based transactive energy management in local energy communities. *IEEE Transactions on Power Systems*, 35(3):1729–1740, 2020. Conference Name: IEEE Transactions on Power Systems.

[30] Liyang Han, Thomas Morstyn, and Malcolm McCulloch. Incentivizing prosumer coalitions with energy management using cooperative game theory. *IEEE Transactions on Power Systems*, 34(1):303–313, 2019. Conference Name: IEEE Transactions on Power Systems.

[31] Omaji Samuel, Nadeem Javaid, Adia Khalid, Wazir Zada Khan, Mohammed Y. Aalsalem, Muhammad Khalil Afzal, and Byung-Seo Kim. Towards real-time energy management of multi-microgrid using a deep convolution neural network and cooperative game approach. *IEEE Access*, 8:161377–161395, 2020. Conference Name: IEEE Access.

[32] Perukrishnen Vytelingum, Thomas D. Voice, Sarvapali D. Ramchurn, Alex Rogers, and Nicholas R. Jennings. Agent-Based Micro-Storage Management for the Smart Grid. pages 39–46, 2010.

[33] Amrit Paudel, Kalpesh Chaudhari, Chao Long, and Hoay Beng Gooi. Peer-to-Peer Energy Trading in a Prosumer-Based Community Microgrid: A Game-Theoretic Model. *IEEE Transactions on Industrial Electronics*, 66(8):6087–6097, August 2019.

[34] Mohamad Aziz, Hanane Dagdougui, and Issmail Elhallaoui. A decentralized game theoretic approach for virtual storage system aggregation in a residential community. *IEEE Access*, 10:34846–34857, 2022. Conference Name: IEEE Access.

[35] Dario Paccagnan, Maryam Kamgarpour, and John Lygeros. On aggregative and mean field games with applications to electricity markets. ISBN: 9781509025916 Meeting Name: 2016 European Control Conference (ECC) Place: Aalborg, Denmark Publisher: IEEE.

[36] Zhongjing Ma, Duncan S. Callaway, and Ian A. Hiskens. Decentralized charging control of large populations of plug-in electric vehicles. *IEEE Transactions on Control Systems Technology*, 21(1):67–78, 2013.

[37] Edstan Fernandez, M. J. Hossain, Khizir Mahmud, Mohammad Sohrab Hasan Nizami, and Muhammad Kashif. A Bi-level optimization-based community energy management system for optimal energy sharing and trading among peers. *Journal of Cleaner Production*, 279:123254, January 2021.

[38] Wayes Tushar, Bo Chai, Chau Yuen, David B. Smith, and H. Vincent Poor. Energy management for a user interactive smart community: A stackelberg game approach. In *2014 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, pages 709–714, 2014. ISSN: 2378-8542.

[39] Xin Dong, Xianshan Li, and Shan Cheng. Energy management optimization of microgrid cluster based on multi-agent-system and hierarchical stackelberg game theory. *IEEE Access*, 8:206183–206197, 2020. Conference Name: IEEE Access.

[40] Francisco Facchinei and Jong-Shi Pang, editors. *Finite-Dimensional Variational Inequalities and Complementarity Problems*. Springer Series in Operations Research and Financial Engineering. Springer, 2004.

[41] Francisco Facchinei and Christian Kanzow. Generalized nash equilibrium problems. 5(3):173–210, 2007.

A Literature review summary

Paper	Type of community	Central Coordinating Agent	Chronology	Energy storage	Uncertainty	Demand response	Centralised	Approach
[16]	Prosumers	Yes	Ex-post	No	No	No	Yes	Bill sharing Mixed market rate
[17]	Prosumers	Yes	Ex-post	Yes	No	No	Yes	Cost allocation based on contribution to local energy
[18]	Prosumers	Yes	Ex-post	No	No	No	Yes	Repartition keys
[19]	Prosumers	Yes	Day Ahead	No	Yes	Yes	Yes	SDR Based bi-level optimization
[20]	Off-grid distributed energy system	Yes	Day-ahead	Yes	No	No	Yes	MINLP
[21]	Prosumers and consumers	Yes	Real-time	Yes	Yes	Yes	Yes	CNLP optimization with a rule based layer
[22]	Group of microgrids	Yes	Day Ahead	Yes	No	Yes	Yes	GRASP
[23]	Group of microgrids	Yes	Real-time	Yes	NO	Yes	Yes	PSO
[24]	Prosumers	Yes	Real-time	Yes	Yes	Yes	No	Individual and pair-wise optimization
[25]	Prosumers and consumers	No	Real time	Yes	No	Yes	Yes	Fuzzy Q learning
[26]	Prosumers	Yes	Real time	No	No	Yes	Yes	PSO with several forecast scenarios
[33]	Prosumers	No	Real-time	No	No	Yes	Yes	Separate buyers and sellers games
[5]	Prosumers	Yes	Day ahead	Yes	No	Yes	No	Mean-field game
[6]	Prosumers	Yes	Day ahead	Yes	Yes	Yes	No	Mean-field game
[34]	Prosumers	Yes	Day ahead	Yes	No	Yes	No	Mean-field game
[4]	Prosumers	Yes	Day ahead	Yes	No	Yes	No	Stackelberg game
[37]	Prosumers	Yes	Day ahead	Yes	No	Yes	No	Stackelberg game
[38]	Prosumers	Yes	Day ahead	No	No	Yes	No	Stackelberg game
[39]	Clusters of microgrids	Yes	Day ahead	Yes	No	Yes	No	Stackelberg game

Table 1: Literature review summary

B Proof of Lipschitz continuity and strong monotony of F_Γ

To prove strong monotony of F_Γ, the jacobian JF_Γ can be used in a similar fashion to section 3.4.2, where it was used to prove strict monotony. Proposition 2.3.2c of [40] indeed states that F_Γ is strongly monotomous if there exists a constant $c \in \mathbb{R}_{>0}$ such that:

$$z^\top JF_\Gamma z \geq c||z||_2^2\,, \quad \forall z \in \mathbb{R}^{2NT}, \tag{52}$$

for all $y \in \mathcal{Y}$. Recall that JF_Γ is:

$$JF_\Gamma = 2 * \begin{pmatrix} \mathcal{A} & O_{NT\times NT} \\ O_{NT\times NT} & \mathcal{B}_N \end{pmatrix},$$

with $\mathcal{A}$ and $\mathcal{B}_N$ defined in equations (23) and (24). So $z^\top JF_\Gamma z = 2(z_1^\top \mathcal{A} z_1 + z_2^\top \mathcal{B}_N z_2)$, with $z = (z_1, z_2)$. For the term in $\mathcal{A}$:

$$z_1^\top \mathcal{A} z_1 = r * z_1^\top z_1 = r||z_1||_2^2, \tag{53}$$

with $r \in \mathbb{R}_{>0}$ the diagonal value of the scaling matrix R. For $\mathcal{B}_N$, it was demonstrated in equation (26) that:

$$z_2^\top \mathcal{B}_N z_2 = \sum_{i=1}^{N} z_{2,i}^\top D z_{2,i} + ||\sum_{i=1}^{N} k_i||_2^2.$$

Looking at the first term, the matrix D is the same for all players and its diagonal values are identical at each timesteps. So $\sum_{i=1}^{N} z_{2,i}^\top D z_{2,i} = d z_2^\top z_2 = d||z_2||_2^2$. Therefore:

$$z_2^\top \mathcal{B}_N z_2 = d||z_2||_2^2 + ||\sum_{i=1}^{N} k_i||_2^2 \geq d||z_2||_2^2. \tag{54}$$

Putting back equations (53) and (54) together:

$$z^\top JF_\Gamma z = 2(z_1^\top \mathcal{A} z_1 + z_2^\top \mathcal{B}_N z_2) \geq 2r||z_1||_2^2 + 2d||z_2||_2^2. \tag{55}$$

As D is designed such that its diagonal elements can be arbitrarily small, it can be assumed that $d \leq r$. Let $\delta_{d-r} = r - d$:

$$z^\top JF_\Gamma z \geq 2r||z_1||_2^2 + 2d||z_2||_2^2 = 2d(||z_1||_2^2 + ||z_2||_2^2) + 2\delta_{d-r}||z_1||_2^2 \geq 2d(||z_1||_2^2 + ||z_2||_2^2) = 2d||z||_2^2.$$

So, for all $z \in \mathbb{R}^{NT}$, $z^\top JF_\Gamma z \geq 2d||z||_2^2$. The map F_Γ is therefore strongly monotonous. ■

To prove that F_Γ is Lipschitz continuous, JF_Γ can also be used. Indeed, if JF_Γ is bounded for any $y \in \mathcal{Y}$, then F_Γ is Lipschitz continuous. This means that there exists a $L \in \mathbb{R}$ such that:

$$||JF_\Gamma(y)||_2 \leq L\,, \quad \forall y \in \mathcal{Y}.$$

If it exists, $L = \sup_y ||JF_\Gamma(y)||_2$. By construction, JF_Γ is a constant matrix, which does not depend on y. Its norm is also constant and the jacobian therefore satisfies the boundedness condition. F_Γ is thus Lipschitz continuous, with Lipschitz constant:

$$L = ||JF_\Gamma||_2.$$

■