DM

PROCSIM
An energy community simulator to develop and evaluate load balancing schemes

MASTER DISSERTATION

Nuno Alexandre Silva Velosa
MASTER IN INFORMATICS ENGINEERING

UNIVERSIDADE da MADEIRA
A Nossa Universidade
www.uma.pt

September | 2022

PROCSIM
An energy community simulator to develop and evaluate load balancing schemes

MASTER DISSERTATION

Nuno Alexandre Silva Velosa
MASTER IN INFORMATICS ENGINEERING

ORIENTATION
Filipe Magno Gouveia Quintal

CO-ORIENTATION
Amâncio Lucas de Sousa Pereira

Acknowledgments

I would start by dedicating this thesis to one of the most important people in my life: my dear deceased mother, who died last year, for her great role in my life and for all the numerous sacrifices she made for me. Although she is not present physically, I am sure that she is very proud of my effort and commitment. I would also like to dedicate it to my godmother for taking on this role and for always encouraging me to follow my goals and go as far as possible in life.

Secondly, I would like to present my sincere thankfulness to my sister, my father, all my family and all my friends for the support given and for always being by my side.

Last but not least, I owe my deepest gratitude to my advisor and co-advisor, Filipe Quintal and Lucas Pereira, for the mentoring given, especially Professor Lucas Pereira for being an indispensable help throughout my master's thesis and for providing opportunities for me to grow professionally. It would not have been possible without this assistance.

Abstract

Climate change is one of the biggest challenges of the present millennium. The energy sector is the biggest contributor to this problem with approximately 25% of the global emissions.

In order to mitigate this problem, one of the main solutions concerns to the use of energy from renewable sources. It is important to begin taking better advantage of the renewable resources more effectively and more often. In this sense, it is very important to develop mechanisms to balance the demand and supply, with the goal of minimizing, as much as possible, the use of energy from non-renewable sources. For this reason, Renewable Energy Communities (RECs) started to emerge. They allow the sharing of the resources, contributing to a better management of them.

However, these are not problem free. There are two main challenges that need to be solved: avoid a bad management of the renewable resources, hence avoiding the need to acquire energy from outside the community, and guarantee a fair distribution of the resources.

In this regard, many researchers are focusing their attentions in load shifting approaches (adapt the appliances running time to better balance the load). Nevertheless, most of them use implicit approaches through the use of incentives (such as tariffs and dynamic pricing), which can be considered unfair approaches since richer people tend to benefit (which is not supposed, because ideally all community members should benefit the same).

Based on this, in this work it is suggested an explicit load shifting approach based on the distribution of the timeslots, using the Multiple Knapsack combinatorial optimization problem. Although there are some literature which demonstrate the applicability of Knapsack in a variety of real world problems, the same does not happen in the energy field.

Furthermore, since a large quantity of data is required to test and evaluate multiple scenarios in this load balancing scheme, and taking in consideration that only two energy community datasets were found on the literature, in this thesis it is also proposed an energy community simulator that allows to create different Energy Community (EC) datasets and evaluate the impact of the optimization, considering only Photovoltaics (PV) production (other types of renewable sources as well as batteries are not considered).

Finally, in order to evaluate the impact of the developed load balancing strategy, the developed simulator was used in three different experiments: variation in bin size, variation in community size and variation in flexibility. The results were positive and showed that this strategy can provide a better management of the PV resources once it increased the PV use, decreased the PV waste and also decreased the use of energy from the grid.

Keywords: energy communities, multi knapsack, combinatorial optimization, simulator, python, renewable energy

Resumo

As alterações climáticas são um dos maiores desafios do presente milénio. O sector da energia é o que mais contribui para este problema com aproximadamente 25% das emissões globais.

A fim de mitigar este problema, uma grande solução está relacionada com a utilização de energia proveniente de recursos renováveis. É importante começar a tirar um melhor partido das fontes renováveis de forma mais eficaz e mais frequente. Neste sentido, é muito importante o desenvolvimento de mecanismos para equilibrar a procura e a oferta, com o objectivo de minimizar, tanto quanto possível, a utilização de energia proveniente de fontes não renováveis. Por esta razão, RECs começaram a surgir. Elas permitem a partilha dos recursos, contribuindo para uma melhor gestão dos mesmos.

No entanto, elas não estão isentas de problemas. Dois dos desafios mais importantes a resolver são: evitar uma má gestão dos recursos renováveis, evitando assim a necessidade de adquirir energia de fora da comunidade, e garantir uma distribuição justa dos recursos.

A este respeito, muitos investigadores estão a concentrar as atenções em abordagens load shifting (adaptar o tempo de funcionamento dos aparelhos para melhor equilibrar a carga). No entanto, a maioria deles utiliza abordagens implícitas através do uso de incentivos (tais como tarifas e preços dinâmicos), o que pode ser considerado injusto, uma vez que as pessoas mais ricas serão beneficiadas (o que não é suposto, pois idealmente todos os membros da comunidade devem beneficiar o mesmo).

Com base nisto, neste trabalho é sugerida uma abordagem explícita de load shifting baseada na distribuição de timeslots, utilizando o problema de otimização combinatorial Multiple Knapsack. Embora haja alguma literatura que demonstra a aplicabilidade do Knapsack numa variedade de problemas do mundo real, o mesmo não acontece no campo da energia.

Além disso, uma vez que é necessária uma grande quantidade de dados para testar e avaliar múltiplos cenários neste esquema de load balancing, e tendo em consideração que apenas dois datasets de comunidades energéticas foram encontrados na literatura, nesta tese é também proposto um simulador de comunidades energéticas que permite criar diferentes datasets de EC e avaliar o impacto da optimização, considerando apenas a produção fotovoltaica (não são considerados outros tipos de fontes renováveis, bem como baterias).

Finalmente, com o intuito de avaliar o impacto da estratégia de load balancing desenvolvida, o simulador desenvolvido foi utilizado em três experiências diferentes: variação no tamanho do bin, variação no tamanho da comunidade e variação na flexibilidade. Os resultados foram positivos e mostraram que esta estratégia pode proporcionar uma melhor gestão dos recursos do PV uma vez que aumenta a utilização do PV, diminui o desperdício do PV e também diminui a utilização de energia da rede.

Keywords: comunidades energéticas, multi knapsack, otimização combinatorial, simulador, python, energia renovável

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter, the main topics of the thesis will be presented. It is divided in three main sections: Motivation of the thesis, where will be described what is the problem that this thesis aims to solve, what is the purpose, what should it be used for, and, what should be the final result; Research Objectives, where it is explained why is it important for the energy research community (i.e. the main contributions of this thesis); and, finally, the Document Structure, describing the organization of the document.

1.1 Context

Mitigating climate change is one of the biggest challenges that confront mankind in the present millennium [6]. 25% of the global emissions corresponds to the Energy sector, making it the most polluting one [7]. It's important to know that this is also related to energy consumption by the domestic sector and not just by industry. The use of energy consuming appliances in domestic residences increases the consumption of energy. Likewise, the use of Heating,ventilation and air conditioning (HVAC) also contribute to the increase of energy consumption, leading to an increase in the carbon footprint [8].

The generation of electricity have an impact on the environment. To illustrate this, in the United States of America (USA), about 40% of the total energy consumed is used to generate electricity [9]. Besides this, in the European Union (EU), in 2018, 45.5% of the electricity generated came from fossil fuels [10]. Nevertheless, there has been an effort in the use of renewable energy. In Portugal, 51% of the energy used in 2019 was supplied from renewable sources [11, 12].

In this way, in every country, it is very important to produce and use electricity more efficiently, in order to reduce the amount of fuel needed to generate electricity and the amount of greenhouse gases and other air pollution emitted [13]. Using energy in a efficient way is using less energy to do the same job. For example, using energy-saving light bulbs instead of incandescent light bulbs, creates the same amount of light without the creation of wasted heat, and, most importantly, using less energy [14].

Although reducing the energy waste and using it efficiently is a good starting point, a effective solution concerns to the use of energy from renewable resources [15] [16] (such as solar, geothermal and wind) to provide all of society's energy needs, without fuels being burned, as shown in Figure 1.1. So, it is essential that countries begins to utilise renewable resources more effectively and more often [17, 18].

Nevertheless, increasing the amount of energy from Renewable Energy Source(s) (RES) in the electricity mix is easier said than done. Among the main challenges to the seamless integration of RES is the uncertain nature of generation from such sources, in particular when considering solar and wind-generation. As such, it is of crucial importance to be able to develop mechanisms to properly balance

[1] see https://www.gosunpro.com/news/renewable-vs-non-renewable-energy-sources/

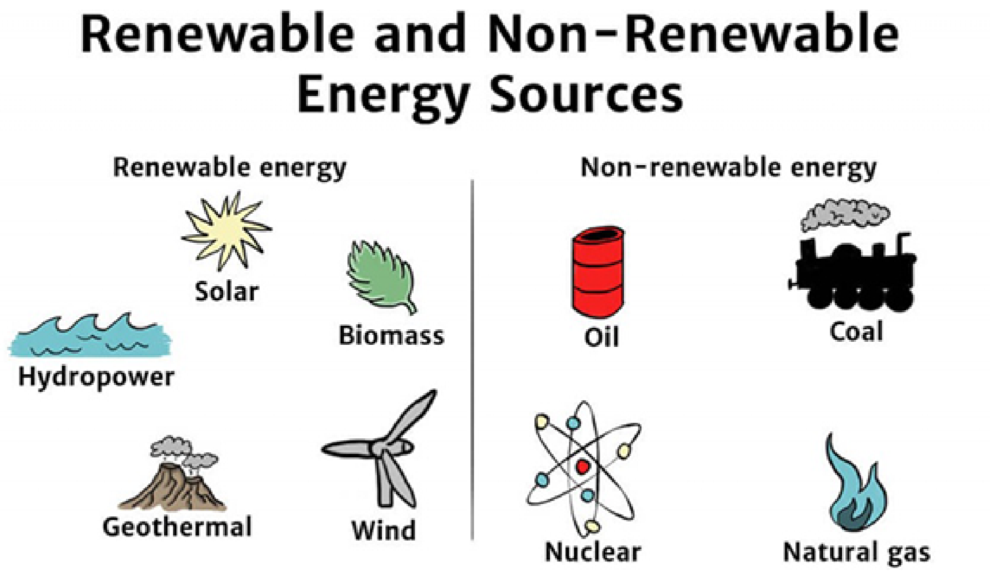

Figure 1.1: Renewable and Non-Renewable Energy Sources [1]

the demand and supply of energy, such that in periods where RES are not available it is possible to keep the demand and supply matched.

Taking this challenge into account, RECs started to emerge. Renewable energy communities show great potential and should be implemented in today's society, transforming the present centralized energy systems towards a more distributed and decentralized future [19]. In very simple terms, they produce or invest in the production of clean energy to meet their own consumption needs, trying to avoid, as much as possible, the use of energy from non-renewable sources [20]. An illustration of the principles of RECs is presented in Figure 2.1.

There are two main types of Energy Communities: the Citizen Energy Communities (CECs) and the RECs. In few words, Citizen energy communities participate across the electricity sector, while renewable energy communities focus only on renewable energy. Moreover, any size entity can participate in a citizen energy community, while renewable energy communities limit participation to micro, small and medium sized enterprises. Finally, CECs must be controlled by natural persons, local authorities or micro and small enterprises, while RECs must be effectively controlled by members that are located in proximity to the community's projects. In this thesis, the focus is on the Renewable Energy Communities since the goal is to use, at much as possible, energy from renewable sources.

1.2 Motivation

Despite their potential, RECs are not problem free. In this regard, two of the most crucial challenges that need to be addressed are: 1) the need to avoid a bad management of the renewable resources, hence avoiding the need to acquire energy from outside the community, which may or not be from RES, and 2) guarantee a fair attribution of the community resources, to avoid having community members that have higher benefits than others [21].

One of the possible solutions to this problem is to have the ability to schedule the usage of electric appliances within the Renewable Energy Community (REC) in such a way that the peak demand occurrences are lower, and the community members are satisfied with the schedule i.e., their electricity needs are met, and the allocation of the resources is considered fair by the community members.

In the current literature, many different works have been done based on load shifting approaches [22], [23] (it is, adapt the running time of appliances to better meet the supply of currently available energy). Most of the papers do it in a implicit way. In few words, the adaptation of the appliances running time is done through the use of incentives which lead people to make their decisions not completely consciously.

These implicit approaches uses tariffs and dynamic pricing. Lower prices are set to periods where an higher amount of energy production exists in order to encourage people to shift their consumption to these periods. On the other hand, when there is a lower amount of production, people should use it as less as possible, so, in order to achieve that, the prices are increased [24]. These approaches take in consideration that people will choose the periods where they pay less for the same use of the appliance.

Nevertheless, this is not considered a fair approach since that the richest people have a higher flexibility in the choice of the consumption periods while the impoverished ones does not have another option than going to the periods with lower price.

Based on this, it is suggested an explicit load shifting approach based on the distribution of timeslots instead of setting dynamic prices. A timeslot allows to use a specific appliance in a time interval with a determined duration. The appliances usage is done through timeslots where the community members need to have them to use the flexible appliances. Otherwise these appliances can not be used.

In Brooks et al. [21], it is proposed that the scheduling of the appliances is performed by so-called "social agents" by using time-slots to guarantee that the peak demand is not met.

In order to do that, this thesis studies the use of Multiple Knapsack to manage the community PV resources (solar panels), having the idea of scheduling the appliances use according to the appliance consumption pattern and also based on the production. The main goal of it is to make netload (total demand minus production) as close to zero as possible (i.e. netload = 0 means demand = production).

Multi Knapsack is a combinatorial optimization problem which generalizes the standard knapsack problem. In this problem, an agent must make irrevocable decisions about which items to pack into some knapsacks without knowing which items will arrive in the future [25]. There are some constraints that should be respected and an Objective Function (OF) that is used to find the optimal solution. Based on this OF (that can be a minimization or maximization function), and taking in consideration the constraints, the Knapsack decides which items should be packed and which of them should not, returning the optimal value of the OF.

Although there are some literature which demonstrate the utility and the applicability of Knapsack in a variety of real world problems, the same does not happen in the energy field.

For example, in [26] the Knapsack is used to achieve the optimal management of the appliances in peak hours taking in account the users preferences and without exceeding the available budget, while in [27] the authors propose a combinatorial optimization algorithm for load leveling and peak demand reduction which combines two combinatorial problems: Bin Packing (BP) and Subset Sum Problem (SSP).

Having this in mind, this thesis will explore in detail how can Multi Knapsack be used in the context of RECs, more precisely to balance the demand and supply, in order to mitigate the problems described previously.

With the aim of evaluating the impact of the optimization, a large quantity of data is required, namely, different energy community datasets. However, in the existent literature only two datasets papers were found: a norwegian real-one [28] and a synthetic one [29], both from 2022, which means, in fact, a lack of available EC datasets to develop and evaluate such techniques.

In this way, in this thesis it is also proposed an energy community simulator in order to make it possible to create different EC datasets to evaluate and analyse the effect of the optimization process in different community types: smaller communities, bigger ones, more flexible communities, communities with a lower flexibility, etc. Above all, this simulator allows to create different scenarios and assess the generalization of the proposed load balancing strategies.

1.3 Research Objectives

To sum up, this thesis makes two main research contributions. The first one is related to the development of a EC simulator that provides a strong contribution to the research community in RECs, which allows, not only to test and evaluate different load balancing strategies, but also to define and configure different communities to generate a consumption dataset composed by demand, production and netload of the community as well as specific demand data for each house and appliance. The second one refers to the development of a load balancing scheme through the use of Multi Knapsack in order to allow an efficient management of the resources in order to reduce, as most as possible, the use of energy from non-renewable sources. This community optimization is also useful to guarantee that the waste of renewable energy will be minimized.

1.4 Document Structure

This thesis is organized as follows: Chapter 2 where is presented some research and related work on Clean Energy Communities, Scheduling Home Appliances, Datasets and Bin Packing and Knapsack Application; Chapter 3, where the overall concept is explained as well as the simulator goals are described; Chapter 4, where the implementation approach is described and the modules are explained; Chapter 5 where all the optimization process is clarified (constraints, objective functions, etc), Chapter 6, where the simulator is evaluated and some experiments are done in order to test the potential of the simulator, and finally Chapter 7, where an overview about the thesis is provided as well as some limitations and future work.

Chapter 2

Background and State of the Art

In this chapter it is presented some existing literature in the research field of energy communities and renewable energy. Moreover, some related works will be explained in order to understand which type of approaches that researchers are currently using with the goal of balancing the demand and the production. Finally, some tools for the EC research will be described.

2.1 Renewable Energy Communities

A Renewable Energy Community, also known as REC, refers to a group of citizens, social entrepreneurs, public authorities and community organizations which participate directly in the energy transition by jointly investing in, producing, selling and distributing renewable energy [30]. It means that, the energy generated from renewable resources is distributed and used to fill community members' needs.

Although there are many single houses that have already integrated renewable energy sources, using, for example, solar panels, there are more benefits when they work together as a community.

In RECs, by reducing the greenhouse gas emissions through the use of renewable resources, there are several advantages, such as economic development, creation of new jobs, cheaper energy, self-sufficiency, community cohesion and energy security [30] [31].

In adittion to RECs, there are the Citizen Energy Communities (CECs). While the primary purpose of both are the same: provide environmental, economic or social community benefits for its stakeholders or members, rather than financial profits, there are some differences between them. Citizen energy communities participate across the electricity sector, while renewable energy communities focus only on renewable energy.

In RECs, the members and stakeholders can be: persons, local authorities (including municipalities) and micro, small or medium enterprises. In CECs, in addition to all these options, it can also be large enterprises.

On the other hand, while in Renewable Energy Communities the participation is only open to local members, it must be autonomous in the internal decision making and have to be controlled by members/stakeholders that are in "proximity" to the project owned by the community without any size constraint, in Citizen Energy Communities the participation is not restricted to local members (all potential members can participate), do not need to be autonomous, and can be controlled by members/stakeholders that are persons, local authorities (including municipalities) and micro/small enterprises (mixed organization or just one set). In contrast to RECs, medium and large companies are prohibited from exercising control over CECs.

A Renewable Energy Community is almost always a Citizen Energy Community, except if the effective control of the community is a medium-size or large-size company. In our approach, since the main objective is related to the management of the renewable resources, we only focus in the REC.

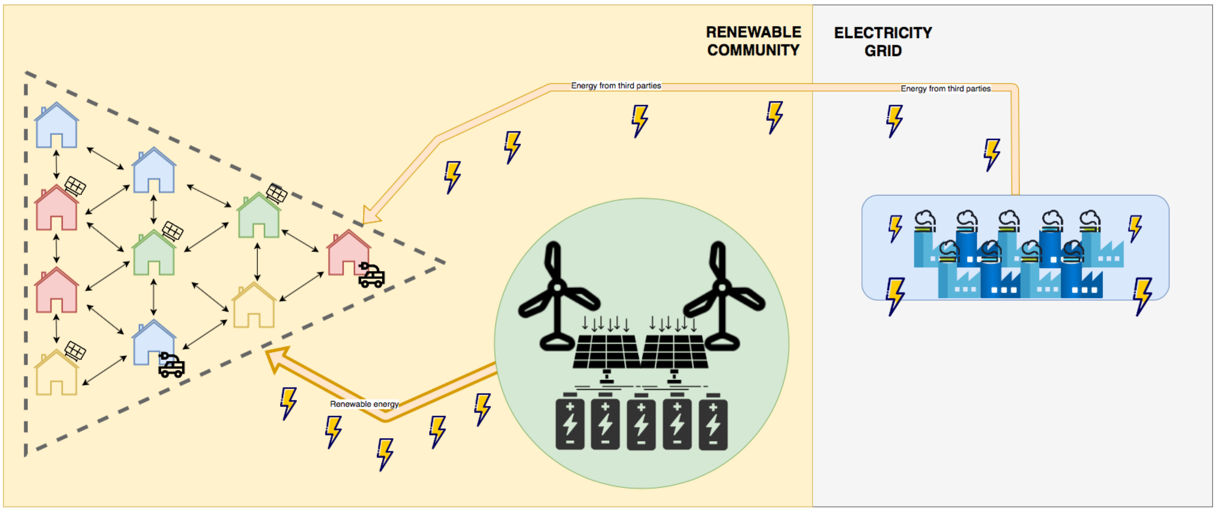

Figure 2.1: Illustration of a Renewable Energy Community

Figure 2.1 illustrates a Renewable Energy Community where all the community houses share their renewable energy resources. It is not isolated since it has also an electricity grid connection. The role of the electricity grid is to cover the community needs when the energy from renewable sources is not enough. Consequently, this energy can come from non-renewable sources, leading to an increase in the carbon footprint.

One thing that we should have in mind is that the Peak Energy Consumption should be reduced. It is essential for the effective use of renewable energy sources, in order to ensure that as much household demand as possible can be met by renewable sources [21].

Basically, to take better advantage of renewable energy, the members should not use the energy all at the same time, making periods of time which there are almost no energy consumed, and other periods which the energy consumed is too high (i.e. peak demand). If the energy consumed in a period of time is very high, the energy from renewable sources can not be sufficient, having the need to use energy from other sources.

As shown in Figure 2.2, the gray curve represents the real power curve and the blue one represents the curve after reducing the peak demand. In the gray curve, the power is very low between 0 and 5

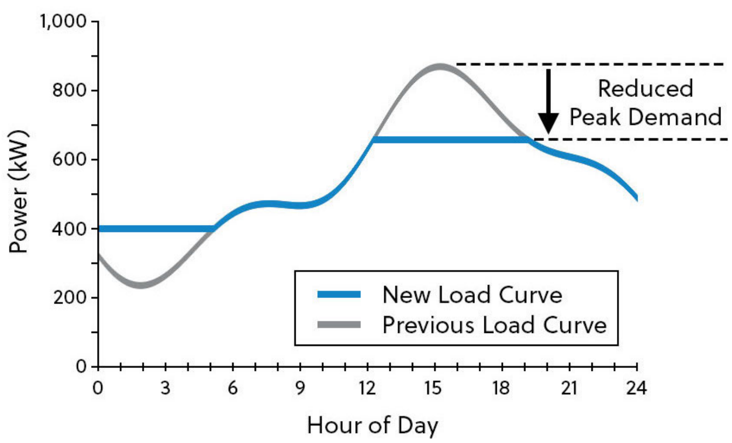

Figure 2.2: Reduction of the Peak Demand [1]

hours and it is very high between 12 and 19 hours. In order to reduce the peak demand (reducing the discrepancy), we draw the new curve by decreasing the power when it is very high.

Furthermore, other aspect that should be analysed is the way the energy will be distributed to the members, in order to make it fair for all. It is not fair having members spending more energy than others, without receiving any other reward (as shown in Figure 2.3). In this sense, some techniques/approaches have to be searched and developed, to ensure that everyone will benefit the same.

There are easier techniques to reduce peak demand that everyone can do, as follows:

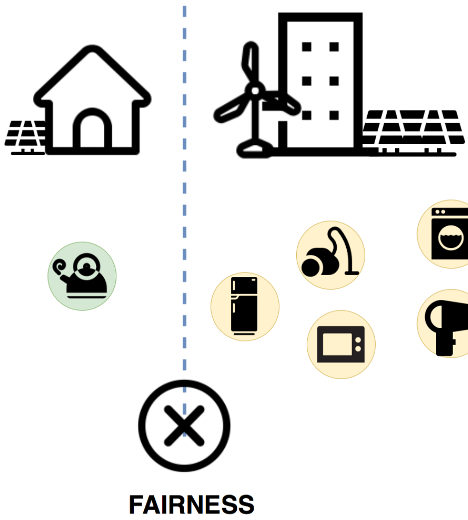

Figure 2.3: Unfair distribution of the time-slots should be avoided (Fairness should be guaranteed).

- Using some appliances in hours where there are less appliances being used, shifting from hours with less production to hours with higher production [32] - like using the washing machine during sleeping hours, where the used energy is less.
- Using appliances just when necessary - for instance, it's better using the washing machine just one time with more clothes than using it 2 times with less clothes.
- Using appliances that spend less energy - like using energy-saving light bulbs instead of incandescent light bulbs.

Unfortunately, although these techniques are easy, they are not put into practice because they require major changes. They require changes in routines, investments, and, at the same time, are probably harder to put into practise because of the living conditions. For instance, for someone that lives in an apartment, having the washing machine working during the night can make some undesirable noise for the neighbors, or, for a person with less economic wealth, changing the appliances to more "environment-friendly" appliances can be very hard. Furthermore, constantly changing devices will result in an increase of electronic waste (e-waste), which is another source of pollution.

Saying this, as referred in Introduction, one technique to avoid these issues is the introduction of the concept of timeslots, and scheduling home appliances.

2.2 Scheduling Home Appliances

In order to achieve a load balance among the supply and demand energy in a REC, many researchers are focusing in the scheduling of home appliances, as is the case of [32], which presents an efficient home energy management system for consumer appliance scheduling with the goal of reducing the energy consumption cost of the service provider. According to Lior [33], 30%-40% of the energy provided is used in residential houses, in home appliances. Thus, an important step is the reduction of the appliances consumption in a house.

However, in a house, not all the timeslots consume the same. There are appliances which require a bigger amount of energy to work than others, leading to a higher cost. This way, it's important to have an idea about which appliances consume the most electricity in a house.

Energy Intensive Appliances

According to *Payless Power*, the first cheap Texas electricity company, HVAC are the appliances which use more energy, corresponding to 46% of the home energy consumption (namely, 27% refers to heating, followed by cooling with 19%). After that, 14% of the total consumption refers to the water heater, 13% to washer and dryer machines, 12% to lights, 8% to refrigerator, 3% to electric oven, 2% to dishwasher, 1% to computer, and, finally, 1% refers to TV [34].

In this sense, it is clear that HVAC systems should be used more carefully, in order to avoid a big energy consumption [35]. However, although these appliances consume a lot of electricity, they are flexible, which are also important to reduce the peak demand.

Flexible Appliances

It should be noted that the flexibility is very important when scheduling home appliances, especially because not all appliances are flexible. In this sense, an appliance is considered flexible when its scheduling does not considerably affect a person's daily habits or activities [36].

Overall, there are two types of flexibility [37]:

- Flexibility in Time: when the appliance usage can be deferred in time, without disrupting activities.
- Flexibility in Power: when the level of power consumption of the appliance can be changed without affecting much the activity.

In order to better understand the difference between a flexible and unflexible appliance, an example will be provided. Undoubtedly, a coffee maker can be considered a not flexible appliance, in that if someone wants to make a coffee for the breakfast, that person will not use it some hours later (it has to be used in the exact moment). In the opposite case, a washing machine, for instance, can be considered a flexible appliance, because it does not need to be used in the exact moment, and can be used later (flexible in time) [38].

With this in mind, only appliances that are flexible in time, should be considered for the scheduling problem. Below some examples of flexible and non-flexible appliances (in time) are provided.

- Flexible appliances: washing machine, clothes dryer, dishwasher, vacuum cleaner, water heater.
- Non-flexible appliances: coffee maker, refrigerator, kettle, microwave.

Chavali et al. [22] created a distributed algorithm of appliance scheduling for a Home Energy Management System (HEMS) (which the objective of a HEMS is to monitor and control energy consumption [39]), using Demand Response (DR). According to Sianaki et al. [24], DR "refers to the consumers' ability to modify their consumption pattern in response to time-varying electricity prices". In this approach, the electricity prices for different time-periods are shown to the users, in order to encourage them to adapt their electricity consumption to it. According to this electricity price, each user will make his own energy consumption schedule.

However, these prices do not depend just on the time-period but also on the aggregated load of the users. Using a greedy iterative algorithm, the sub-optimal energy consumption of each user is calculated, where, in each iteration, all the users have to communicate their energy consumption schedule to the company. In this sense, according to the overall system load, the company will adjust the prices and send it to all users, so they can update their schedule based on the new price.

Aslam et al. [23] proposed a Demand Side Management (DSM) approach for residential consumers based on scheduling techniques, with the main purpose of reducing the electricity cost and obtain a desirable user waiting time, because 10-30% of electricity can be saved by scheduling of home appliances via DSM approach. However, due to this scheduling, the user waiting-time increases, creating, consequently, an optimization problem. This way, the authors suggest an efficient scheme in (HEMS) using (Genetic Algorithm (GA)) and Cuckoo search algorithm to solve this optimization problem. In this paper, the results demonstrated that the scheme achieves the affordable user waiting time, as well as, the GA and Cuckoo search decreases the electricity cost by 12.64% and 13.96%, respectively.

Further works in this topic can be found in the literature. For instance, in [40], the authors analyze available information concerning energy consumption in buildings, particularly related to HVAC systems, and concludes that 50% of the building consumption as well as 20% of the total consumption in the USA are related to the use of of HVAC systems. In [41], it is proposed a new approach to energy efficient control of systems by combining optimization and scheduling, which reduces the peak demand from small to large scales. On the other hand, in [24], it is demonstrated a fuzzy TOPSIS decision-making approach to quantify and evaluate the preferences of the consumers when using electrical devices according to a real-time price of demand response in order to do the best management of the appliances. Finally, since the user waiting time increases due to the scheduling of the home appliances, the authors of [42] propose an HEMS using meta-heuristic GA, Cuckoo Search Optimization Algorithm (CSOA) and Crow Search Algorithm (CSA) with the goal of electriciy cost reduction and peak load alleviation with minimum user waiting time.

2.3 Tools for EC Research

In a Renewable Energy Community, in order to schedule the appliances and distribute appliances time-slots for all the agents, so they can use them, an important aspect is to know which appliances the household has, the activities that the agent wants do do and also the appliances that it wants to use. With this in mind, some tools for household activities and consumption data must be available.

In this section, we aim to describe possible alternatives to gather the required consumption data. This includes publicly available electricity consumption datasets (real datasets and synthetic datasets) as well as household consumption simulators.

2.3.1 Electricity Consumption Datasets

A quick survey on the published literature revealed that there is a lack of real datasets specific for energy communities (EC). More precisely, only 2 datasets were found on real energy communities: a real-word one in Norway [28] and a synthetic one [29]. The dataset of a norwegian energy community [28] includes household consumption data collected from smart meters measurements filtered by season, weekday/weekend and time segment, and, divided into 5 groups: Household consumption, appliance consumption, electric vehicle charging, photovoltaic power generation and wholesale electricity price.

It aims to allow researchers to implement situations and experiments on realistic ECs. The synthetic one is presented in [29] where a dataset of a 49-household residential community with a public building (municipal library) is created by attributing sample consumption and photovoltaic production profiles and disaggregating the total power consumption of these households into the consumption of several individual appliances.

Taking into consideration this lack of EC's datasets, it becomes hard to create some case studies and experiments on this field. For this reason, as done by the authors in [29], the solution relies on the use of synthetic EC datasets. The electricity consumption datasets can be, mainly, from two different types: Real-World datasets or Synthetic datasets. The real-world datasets contains data obtained by direct measurements, using, for example, smart meters. In contrast, the synthetic ones uses data from simulators instead of direct measurements.

Real-World Datasets According to Pereira and Nunes [43], a real-world electricity consumption dataset is a collection of electrical energy measurements taken from real-world scenarios, without disrupting the everyday routines in the monitored space, that is, trying to keep the data as close to reality as possible. Such datasets contain measurements from the aggregate consumption and of the individual appliances. Examples of such datasets are REDD [44], UK-DALE [45], REFIT [46], and SustDataED [47].

All of these datasets have one thing in common: they contain individual and aggregate load. Nevertheless, there are real datasets which only contains aggregate load (e.g. PLAID). Usually, what varies between them are the location, number of houses, appliances and frequency.

Moreover, the datasets can be separated into residential and commercial. Residential datasets refers to a dataset of residential houses (EMBED, REDD, BLUED, PLAID, UK-Dale, etc) while commercial datasets refers to a dataset of commercial places (BLOND, I-BLEND, COMBED, etc) [48].

Synthetic Datasets In addition to real-world datasets, there are also synthetic datasets in which the data is not obtained by direct measurements. Since collecting real-world datasets require real measurements, which are time-consuming and expensive to get, besides the bureaucratic burdens, generating synthetic data is gaining popularity as a viable alternative that reduces measurement costs and saves valuable work hours [49]. Furthermore, in synthetic datasets, the energy consumption data is provided by simulators without limitations on measurement periods and without suffering from missing gaps, misaligned timestamps and corrupted data (as a result of sensor miscalculation or malfunction), as in real datasets [49].

SynD is a synthetic energy consumption dataset for Non-Intrusive Load Monitoring (NILM) that focus on residential buildings, having a household simulated with 21 household appliances and 180 days of synthetic power data. SynD provides aggregate power data as well as individual household appliances power data. [49].

Another example of a synthetic dataset is the SHED dataset [2], consisting of synthetic high-frequency electricity consumption measurements for commercial buildings, that can be used to evaluate NILM algorithms. SHED consists of 8 buildings, and includes the total current consumption, as well as the individual consumption corresponding to different categories. It can be downloaded at https://nilm.telecom-paristech.fr/shed/.

Finally, PowerGan [2] is an approach to generate random appliance power signatures using Generative Adversarial Networks (GAN), i.e. a GAN-based synthesizer for appliance power traces. Deep Neural Networks (DNN) are useful to solve problems such as classification, regression or segmentation, but, they are not able to generate synthetic data. In 2014, with the introduction of generative adversarial networks, instead of having one neural network trained to solve an optimization problem, there are two competing neural networks trained to find the equilibrium of a game. In a GAN game, there are 2 players:

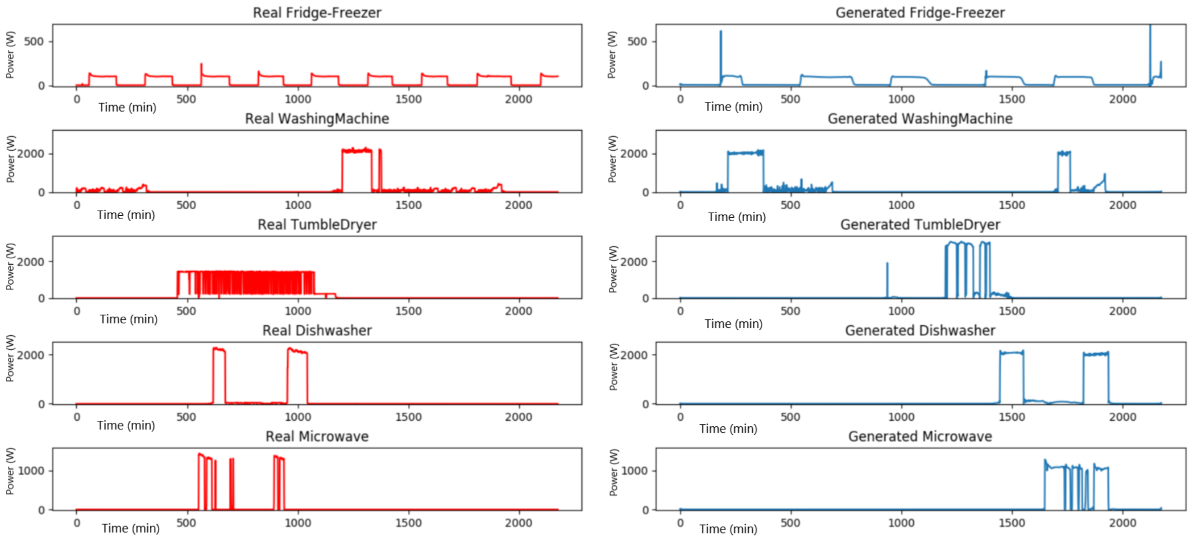

Figure 2.4: Examples of appliance power traces generated by PowerGAN [2]

the generator and the discriminator. The generator tries to generate realistic signals and the discriminator attempts to successfully distinguish these generated signals from real ones. The equilibrium of the GAN game is achieved when the generator can not distinguish the perfectly realistic generated signals from the real ones. This way, as shown in Figure 2.4, PowerGAN is able to produce diverse and realistic power appliances signatures, without the need to copy directly the training data [2].

2.3.2 Household Consumption Simulators

A big limitation related to the use of datasets is the fact that they are deterministic (i.e. each dataset return always the same output) and does not allow to test different scenarios. In opposite, simulators can provide different outputs every time they run, leading to the opportunity to explore different approaches and simulate different cases. Moreover, simulators have also a big advantage: they can be totally customized (it is possible to specify the users, the appliances, etc)

In Buneeva et al. [50], the authors proposed Automated Model Builder for Appliance Loads (AMBAL) which provides models from real device power consumption data collected by smart plugs, since many current models for synthetic trace generation just uses statistical information about household occupancy and only considers the energy consumption of the most frequently performed user activities (which is not enough to reflect the diversity of consumption signatures). Also, it does not need any manual interactions to establish the power consumption models of appliance loads and its models allows the quick generation of traces.

In order to investigate the quality of the traces synthetically generated from the AMBAL's models, the authors created the AMBAL-based NILM Trace generator (ANTgen) [51], a synthetic appliance traces generator, which synthesizes device models into aggregate traces (Figure 2.5) and uses them for the evaluation of disaggregation algorithms. In ANTgen, the operation of the appliances are scheduled according to the models of daily activities (which an activity defines a set of appliances and the sequence that they will run) (for instance, the meal should be cooked before eaten). On the other hand, user models determine which times the activities will take place, the frequency and the user presence [52]. Finally, it provides the opportunity to vary the number and types of appliances and outputs an aggregated trace for the whole household and a specific trace for each appliance. An example of a refrigerator trace is present in Figure 2.6.

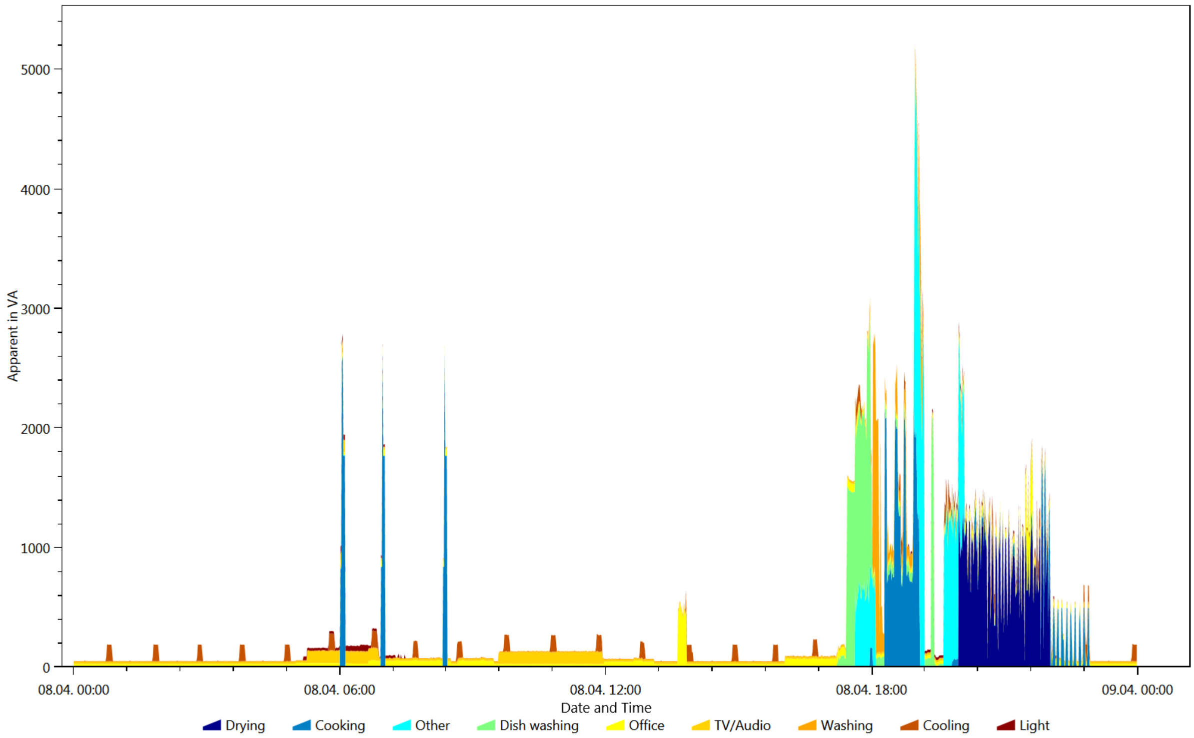

Figure 2.5: Aggregate load curve generated by LoadProfileGenerator [3]

On the other hand, SMACH [53, 54] is a meta-model and multi-agent simulator of human activity, which allows to model, simulate and study the household activities as well as their relation with electricity consumption. Considering this, in this simulator, according to the appliance use or pricing policies, it is possible to notice the evolution of the energy consumption. The output of SMACH is an activity diagram and a load curve for every electrical appliance that exists in the household [53].

Finally, the LoadProfileGenerator [3] is another household consumption simulator. This simulator enables the generation of load profiles (Figure 2.5) for electricity, gas, hot water and cold water, and returns individual load curves. It is possible to use one of 60 predefined households or define our own household. Also it is useful to simulate energy systems and completely customize full behaviour simulations.

After explaining the main limitations of datasets (reals and synthetics) and the main advantages of simulators, it is important to mention that this thesis will be focused on the use of simulators (namely the ANTGen simulator), instead of using specific datasets.

Since the consumption data source has already been chosen, it is now possible to start scheduling the home appliances using timeslots. For this distribution, combinatorial optimization will be used.

2.4 Combinatorial Optimization Application to Energy Communities

Combinatorial optimization is the process of searching for a maximum or minimum value of an objective function. [25] It determines the best solution from a finite set of possibilities. There are a lot of known combinatorial optimization problems. Some of the most common ones are: Subset Sum, Bin Packing and Knapsack.

Subset Sum problem determines if, given a set of non-negative integers and a value sum, there is a

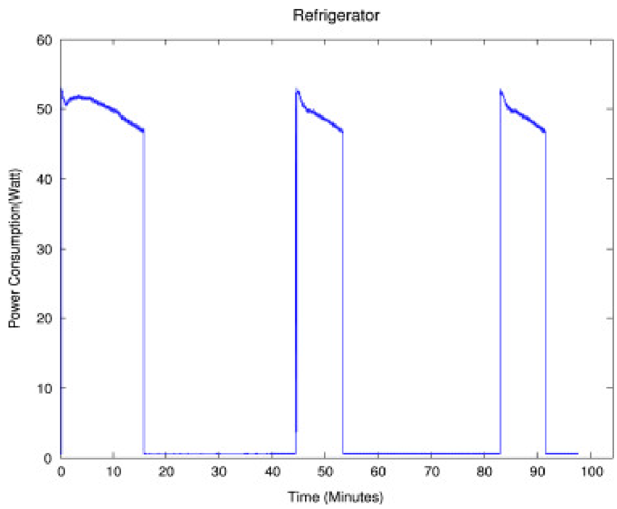

Figure 2.6: Refrigerator power consumption trace [4].

subset of the given set with a sum equal to the given sum. It can be used to retrieve all the combinations of a list which the sum is a specific value. For instance, considering a list of nums = [1, 2, 3, 4]. If the target is 7, there are two subsets that achieve this sum: 3, 4 and 1, 2, 4. If the target is 11, there are no solutions. [25]

On the other hand, Bin Packing is the problem of, given a set of items with a specific size, fit them into as few bins as possible, taking in consideration that it can not exceed the capacity of the bins. It may be assumed that all items have weights smaller than bin capacity. It can be used, for example, to calculate the minimum number of trucks that can hold all the boxes (there are some boxes to be transferred from a source to a destination and it is necessary to calculate the minimum number of trucks which can hold all the boxes taking in consideration its sizes). [25]

Unlike Bin Packing, in the traditional Knapsack there is only one bin (also called as knapsack). Each bin has it own capacity. Given weights and values of n items, the idea is to put the items in a knapsack of capacity W in order to maximize the total value in the knapsack, without exceed the knapsack/bin capacity. There are some variants of this problem, namely:

- 0/1 Knapsack (Classic one): It is not allowed to break items. We either take the whole item or do not take it. Only there are two options for each item: 0 if it is not placed and 1 if it is placed. [1]

- Fractional Knapsack: Unlike 0/1 Knapsack, the items can be divided and do not need to place the whole item. [2]

- Unbounded Knapsack: In this approach, an unlimited number of instances of an item can be used. [3]

- Multi Knapsack: The previous three variations of Knapsack only uses 1 bin to put all the items. In this variation, it is the same as 0/1 Knapsack but uses more than one bin and every bin has it capacity.

In the current days, although the application of bin packing and knapsack is not very common in the energy field, there are some literature review related to the use of it to some real life problems of different research areas.

[1]https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/
[2]https://www.geeksforgeeks.org/fractional-knapsack-problem/
[3]https://www.geeksforgeeks.org/unbounded-knapsack-repetition-items-allowed/

In [26], the authors use the Knapsack problem in order to achieve the optimal management of the appliances during peak periods with the objective of saving the budget and preferences by selecting the appliances which the use is most important according to the users preferences. The users do not want to change its consumption behavior, leading to the need to pay more. Nevertheless, since they do not want to exceed the available budget, some appliances have to be turned off or/and some consumption have to be shifted to off-peak hours. This way, the paper applies a simple Knapsack (with only one constraint and a simple objective function) to solve the following problem: "Which appliances should be turned off during peak hours without exceeding the budget, maximizing the user preference?" and uses LINGO software as a powerful optimization software to solve the problem.

In [27], the authors propose an algorithm which applies combinatorial optimization to Energy Storage System(s) (ESS) scheduling, in order to achieve load leveling and reduce the peak demand. They divide the problem in two steps, and, for each step, a different combinatorial optimization problem is used.

In the first step, where the aim is to define combinations of energy blocks which the sum of them equals the total ESS storage capacity, the authors use the SSP. Basically they divide the storage capacity in energy blocks with different capacities and retrieve all the possible combinations where the sum is the storage capacity. For instance, considering the example of Figure 2.7, using an ESS with a storage capacity of 20 kWh, it may be split into blocks of then 2 KWh units, or using four blocks of 3 kWh and two blocks of 4 kWh. There are multiple possible combinations. It is a fundamental aspect of SSP, which generates different viable options each time it runs.

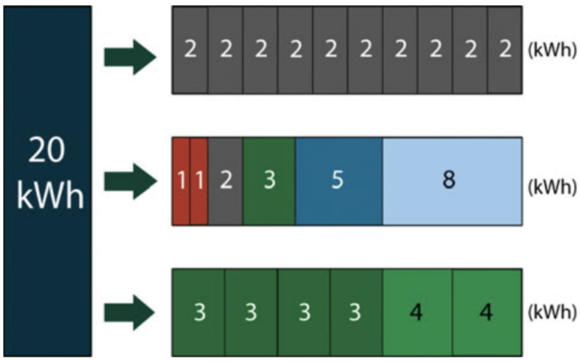

Figure 2.7: Possible combinations of blocks for a ESS with a storage capacity of 20 kWh. [5]

In the second step, using Knapsack, the idea is to use the set of energy blocks generated in the previous phase, and allocate it into finite consecutive time intervals of a given duration taking in consideration that the maximum amount of energy is proportional to the peak demand. Intervals of 1 hour results in 24 bins over 24 hours, 15-min intervals will result in 96 bins, and so on.

To achieve the peak demand shaving and load leveling, the ESS energy blocks from the first step are the items packed on the Knapsack using the worst-fit heuristic. Worst-fit heuristic places the items in the bin that leaves the most capacity left after a placement.

In [5], the authors propose the extension of the bin packing problem, called Bin Packing with Usage Cost Problem (BPUC), in order to manage the energy in data centres. The goal is to control the energy consumption in a data centre and guarantee that it is consistent with the application needs, economic constraints and service level agreements. Each server is represented by a bin with a capacity equal to the available amount of resources and a bin (server) is used when it contains at least one item. The objective is to assign each item to a bin, taking in consideration the capacity constraints, in order to minimize the sum of the costs of all bins.

Taking in consideration the characteristics of the five bins B = {(9,0,1), (3,0,2), (3,0,2), (3,0,2), (3,0,2)}, the ordered sizes of the five items S = {2,2,2,2,3,3,3} and f = 0 for every bin, two different scenarios will be presented in Figure 2.8. In the first scenario we consider that the packing (P1): {{2,2,2,2}, {3}, {3}, {3}, {}} is using the minimum number of bins (four bins) and has a cost of 26 (8*1 + 3*2 + 3*2 + 3*2) while the packing (P2): {{3,3,3}, {2}, {2}, {2}, {2}} is using one more bin (five bins) but has a cost of 25 (9 + 2*2 + 2*2 + 2*2 + 2*2).

As it is possible to see, (P2) is better than (P1) because (P2) has a lower cost although it uses more bins. It ca be concluded that, for this problem, using the minimum number of bins is not a good strategy. On the other hand, in the second scenario, changing the last unit cost to 3, we get B = {(9,0,1), (3,0,2), (3,0,2), (3,0,2), (3,0,3)}. The total cost of (P1) remains the same once it does not use the bin 5. However, the total cost of (P2) increases to 27, and thus, (P1) is better than (P2) in the 2nd scenario.

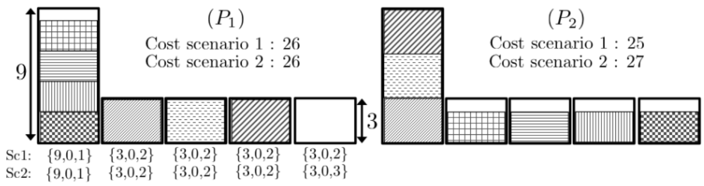

Figure 2.8: Example of two scenarios using BPUC [5]

2.5 Summary

In this section, a general overview of the SOA and an identification of the main gaps was done. it has been discussed how implementing Renewable Energy Communities (REC) can help mitigate environmental problems, such as climate change. Nevertheless, the reduction of the peak demand and the fair distribution of the resources should be solved in order to have fully functional REC.

In this sense, this thesis proposes the use of Combinatorial Optimization (namely using Multiple Knapsack problem) to schedule home appliances based on the use of timeslots, in order to balance the demand and supply of the REC. Although there are not enough research on the use of timeslots and Knapsack to solve this type of problems, it seems to be a promising approach.

However, with the goal of optimizing the management of the community resources, it is extremely important to have consumption data of all houses of the community. For this reason, the use of EC datasets or EC simulators is needed.

Neverthless, the full potential of these tools is still vastly under-explored and there is a lack of real energy communities datasets in the published literature: only 2 datasets were found. In this way, the development of a EC simulator is also proposed in this thesis, using ANTgen simulator to generate and provide household activities and consumption at the appliance-level.

For all these reasons, the main objective of this thesis is to investigate and explore how the demand and supply can be balanced inside a REC. More concretely, it provides two different contributions for the energy research field: the development of a EC simulator which allows to explore and evaluate different scenarios, and a load balancing scheme to manage the community RES taking the most advantage of them (avoid the waste of energy and avoid the use of energy from non-renewable sources).

Chapter 3

Proposed Solution

This chapter aims to describe the solution that will be created in this thesis, namely explain the overall concept of this simulator, highlight the goals of the three different components of the simulator and explain how these components can be combined in order to create the PROCSim simulator.

3.1 Overall Concept

As previously referred, this thesis gives two main contributions to the EC research community: the development of a EC simulator due to the lack of publicly available energy communities datasets and also due to the flexibility it provides to create different scenarios and evaluate multiple approaches/experiments; and, the development of a load balancing scheme which tries to maximize the use of the renewable energy resources and reduce, as most as possible, the use of energy from non-renewable sources.

The overall concept is represented in Figure 3.1, where it is possible to see the whole process, namely all the steps and the interaction between them. It demonstrates the flow when the EC simulator is used together with the load balancing scheme, although they can be used separately.

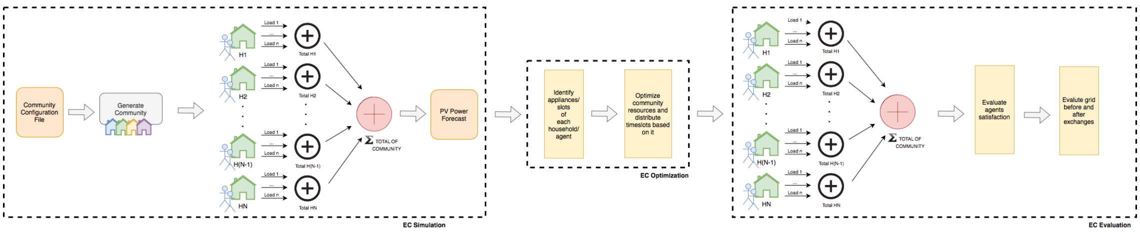

Figure 3.1: Overall Concept of PROCSim: Combination of three components (Simulator, Optimization and Evaluation)

It is divided in three main components: EC Simulator, EC Optimization and EC Evaluation. EC simulator is responsible for generating the community by generating and combining the consumption and production profiles, EC Optimization is responsible for optimizing the community (doing the management of the resources) in order to take the most advantage of the renewable resources and EC Evaluation is used to recalculate the consumption profiles after the optimization and evaluate the impact of the process in the community. Some more details will be provided in the next subsections.

3.2 Energy Community Simulator

The first component is the energy community simulator which allows to create and simulate energy communities based on the aggregation of the consumption profiles of electric appliances in individual households. The consumption profiles of the different households are generated automatically based on a set of properties. For example, the time of the day that appliances are most likely to be used, number of members in the house, contracted power, and flexibility levels of the house (i.e., an house with very strong routines would have a low flexibility).

In the present version of PROCSIM, two main data sources are considered: 1) the consumption demand, and 2) the generation from RES. The first data source is responsible for generating the consumption profiles of the different households, which are later aggregated to form the total demand of the energy community. The second data source is responsible for simulating the generation from RES, which in the present version of PROCSIM is from PV. After these data sources are generated they are combined to form the energy community netload (demand - generation), which corresponds to the amount of energy that must be acquired from the grid.

The whole process to calculate the netload is presented in Figure 3.2, starting with the sum of every consumption profile of the community and concluding with the subtraction of the production from that community demand.

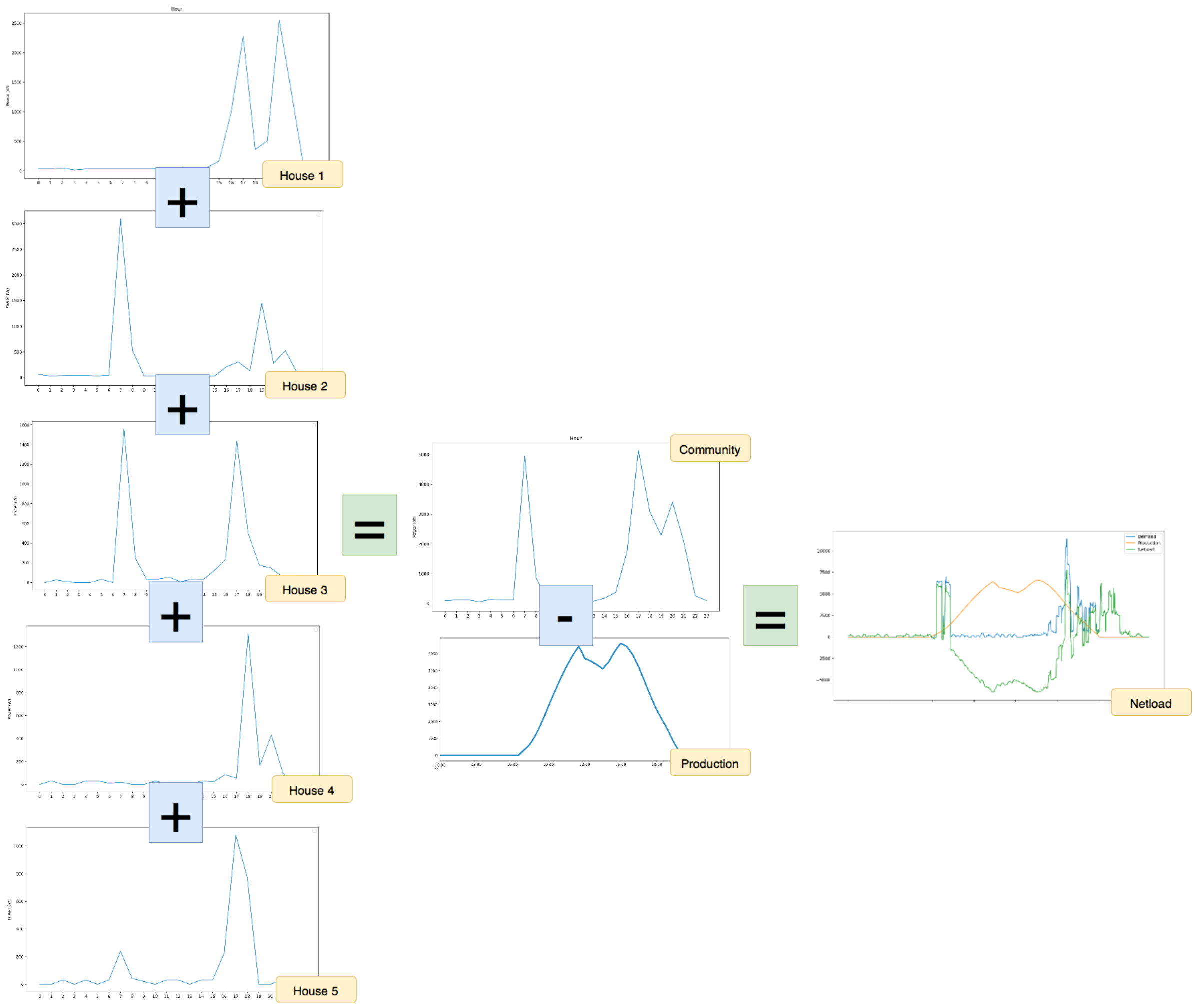

Figure 3.2: Netload calculation by subtracting the production from the community demand

3.3 Energy Community Optimization

The second component is the energy community optimization, which as seen in the state-of-the-art can be achieve in several different ways. In the current version of PROCSIM the optimization is performed using the concept of timeslots, which is a slot that allows to use a specific appliance in a period of the day.

More precisely, the optimization step aims to find the optimal way to manage the community resources in order to guarantee three equivalent goals: the PV renewables use will be maximized, the energy from grid will be minimized, and, mainly and ideally, the netload will tend to 0 (will not be too negative - meaning a waste of resources (once batteries or electric vehicles are not considered in the simulator) or too positive - meaning not enough renewable resources to cover the consumption needs).

In this concrete case, the optimization is performed in two steps. The first step aims at maximizing the self-consumption, by considering only PV production forecasts in the optimization. The second step aims at distributing the remaining timeslots taking into consideration that there is a maximum value of power that can be acquired from the grid, which in this work is referred as the Peak Power Contract (PPC).

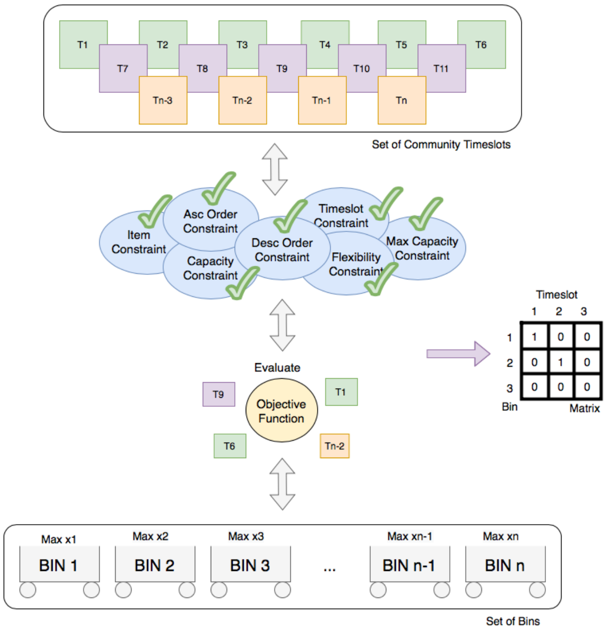

Figure 3.3: Overview of the optimization process.

Figure 3.3 gives a brief overview of the timeslots distribution process. There is a set of community timeslots, where each one is divided in a set of items depending the time interval of the bins, that have to be distributed over the day. Also there is a set of bins (that corresponds to a time interval which contains all the timeslots used in that time interval) with a fixed capacity. The timeslots have to be packed inside the bins ("box" that corresponds to a specific period of the day), but, in order to add a timeslot to a bin, some constraints have to be verified. The figure shows seven different constraints defined in order to solve this problem: Item Constraint, Asc Order Constraint, Capacity Constraint, Desc Order Constraint, Timeslot Constraint, Flexibility Constraint and Max Capacity Constraint. Each one has a different purpose and all together allows to solve this load balancing problem.

If all the constraints checked, the solver uses a matrix of 0's and 1's that stores a binary value (0 if item x not packed in bin y and 1 otherwise) and calculates the objective function for every combination in order to retrieve the best solution. The optimal solution is the one which maximizes or minimizes the objective function). This process is similar in the second step once only the objective function is different than the first step.

All these details of the day-ahead optimization process will be explained in Chapter 5.

3.4 Energy Community Evaluation

After generating and optimizing the community, in the EC Evaluation component, the community demand, production and netload are calculated again, and all the consumption profiles of every house of the community are then updated.

After these updates, we are able to compare the results before and after the optimization, and, using some defined metrics, evaluate the results.

This is the component which allows to understand if the optimization was useful and if it helped in solving the main problem: reduce the use of energy from non-renewable resources and increase the use of energy from renewable resources (i.e. balance the load).

As shown in Figure 3.4, it receives as input the location of the consumption profiles of all houses of the community before and after the optimization, and, based on that, the consumption profiles stored in csv files are read in dataframes.

After reading/obtaining the results, the metrics are calculated for three different phases: before the action of the community manager, after the first step of the optimization and after the second step. These metrics will be better explained and more details will be provided in Section 4.3.2. However, in summary, the most important ones are: quantity of energy used from the grid, quantity of energy used from the production, quantity of energy not used from the production, number of placed timeslots, number of unplaced timeslots and energy of unplaced timeslots.

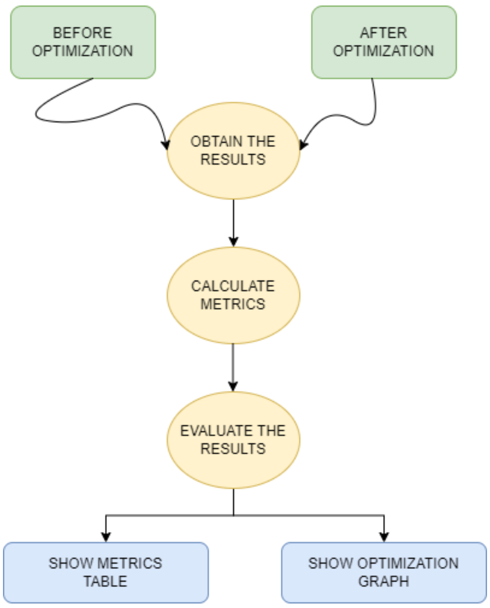

Figure 3.4: Overview of the EC Evaluation Component

Chapter 4

PROCSIM Simulator Development

In this chapter, all the details about the development of the PROCSIM will be explained. More than the functionality, information about the used tools and languages will be provided as well as relevant information about the decisions that were taken in account when developing the simulator.

4.1 Implementation Approach

The PROCSIM is a python platform that allows the simulation and management of energy communities, taking into consideration the available RES. The current simulator is divided into two main concepts: 1) the community simulation where the community is created and the individual and aggregated energy flows are generated, and 2) the load balancing evaluation where the different load balancing schemes are developed and simulated. The implementation approach is presented in Figure 4.1.

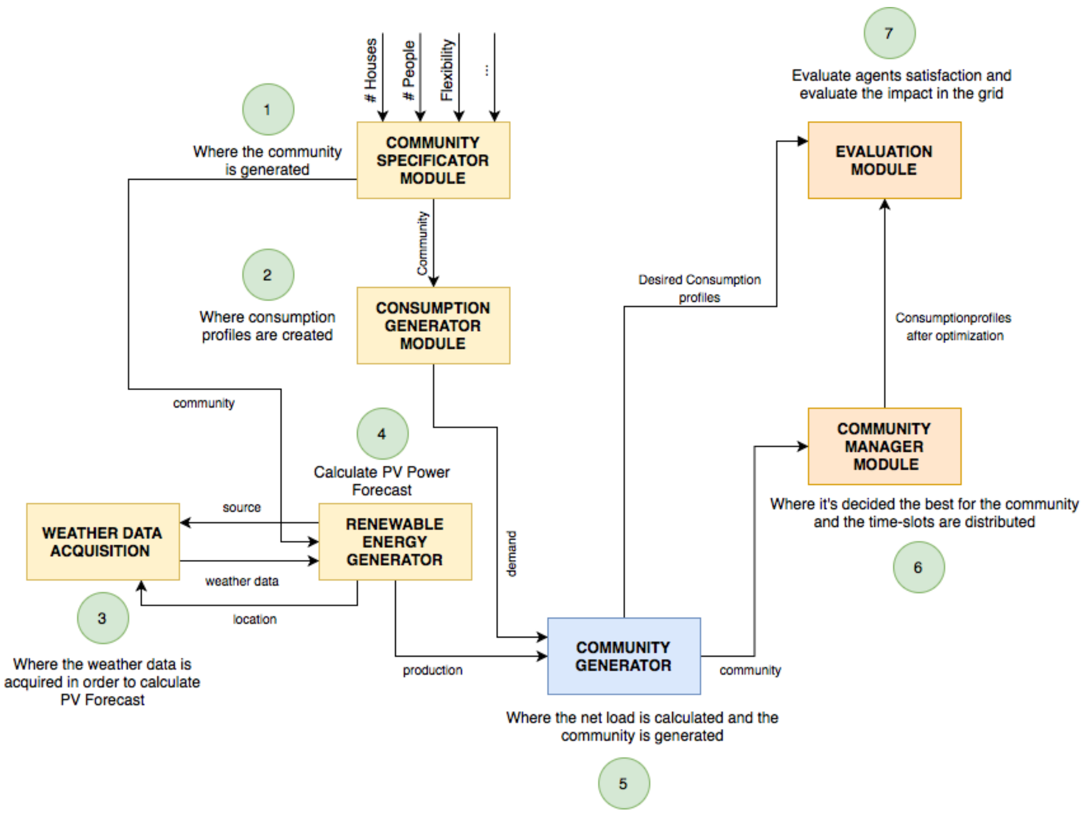

Figure 4.1: Implementation Approach in modules.

4.2 Community Simulation

In the present version, the community simulation is comprised of the following five modules: Community Specificator, Consumption Generator, Data Acquisition, Renewable Energy Generator and Community Generator.

4.2.1 Community Specificator

The *Community Specificator* module provides the tools and mechanisms to create an EC. The user should specify some important data, namely, the number of households, which appliances are available in each house, the number of householders, and the main activities performed by each one. In this module, the users also have the possibility to set its flexibility as well as the PPC.

Everyone who wants to create a community using this simulator should create a json file with all this information in order to allow it to create the consumption profiles based on the activities and schedules defined, and also based on the number of people of each house. For each house, for each user of the house and also for each activity, a consumption profile is created. Moreover, the PV power forecast will be done based on the house PPCs values because the maximum of PV that community can generate is 50% of the sum of the PPCs of all houses. Finally, the flexibility will allow to do a better optimization in the following way: Flexibility indicates how flexible is the house, between 0 and 1, where 1 means totally flexible (the appliance usage can be swifted to some hours before or some hours after).

Listing 4.1 represents the json representation of a house of a community. The house (Nuno House) has 1 person (Ann), a contracted power of 3.45kW and a flexibility of 0.8 (equivalent to 80%). Baseload is the one who is responsible for all not-flexible appliances (the appliances which consumption can not be changed - eg. fridge).

This house has four different appliances, which are: friddge, vaccumm cleaner, washing machine and drying machine. What concerns to the schedules, each user of the house (baseload is included) has to have the presence and activities arrays. The presence array means the intervals of the day where the user is at home for each day of the week, while the activities array refers to the activities the user do every week day.

In this case, since baseload is used for not-flexible appliances, it does not need presence (its always present) and the refrigerator runs from 00:00 to 24:00 (runs all day). On the other hand, Ann is not at home from 08:30 to 14:00, except sunday (because she is working during week days and saturday), and, for that reason, she watches TV twice a day on different schedules depending on the day of the week. Ann also uses the dishwasher 1.5 times a day, which means that she uses it once a day and every 2 days she does an extra run (50% of probability of running). The model present in each activity is the model that will be used to understand the appliance consumption pattern.

Listing 4.1: JSON Representation of a Community

```
{
    "house": "Nuno House",
    "num_people": 2,
    "contracted_power": 3.45,
    "flexibility": 0.8,
    "people": ["baseload", "Ann"],
    "appliances": "Fridge", "Vaccuum cleaner", "Washing machine", "Drying machine"],
    "schedules":
    [
        {
            "presence": [],
            "activities":
```

```
        [
            {
                "activity": "activity_fridge",
                "model": "KITCHEN/refrigerator.conf",
                "daily_runs": 25,
                "schedules":
                [
                    {
                        "day": monday",
                        "schedule": ["00:00-24:00"]
                    },
                    ...
                    {
                        "day": "sunday",
                        "schedule": ["00:00-24:00"]
                    }
                ]
            }
        ]
    },
    {
        "presence":
        [
            {
                "day": "monday",
                "schedule": ["00:00-08:30", "14:00-24:00"]
            },
            ...
            {
                "day": "sunday",
                "schedule": ["00:00-24:00"]
            }
        ],
        "activities": [
            {
                "activity": "activity_tv",
                "model": "LEISURE/watch_tv.conf",
                "daily_runs": 2,
                "schedule":
                [
                    {
                        "day": "monday",
                        "schedule": ["16:00-23:30"]
                    },
                    ...
                    {
                        "day": "sunday",
                        "schedule": ["09:00-23:30"]
                    }
                ]
            },
            {
                "activity": "activity_dishwashing",
                "model": "KITCHEN/dishwasher.conf",
                "daily_runs": 1.5,
                "schedule":
                [
                    {
                        "day": "monday",
```

```
                    "schedule": ["07:00-12:00","17:00-23:00"]
                },
                ...
                {
                    "day": "sunday",
                    "schedule": ["09:00-22:00"]
                }
            ]
        }
    ]
  }
 ]
}
```

4.2.2 Consumption Generator

The *Consumption Generator* is used to generate the consumption profiles of the community based on the information provided in the previous module. More precisely, for each household, the module generates the baseline load and the consumption of each appliance. These are then summed to form the households and community aggregated consumption profiles.

The current version of PROCSIM relies on the AMBAL appliance trace generator (ANTgen) [50], which generates appliance consumption traces at 1Hz.

Based on the specification json file, this module creates a configuration file for each house of the community, with the extension conf, which is also composed by a set of configuration files - one for each user of the house.

Listing 4.2: ANTgen - House Configuration file

```
{
  [GENERAL]
  name = House Configuration file

  [users]
  baseload = house1_baseload.conf
  Ann = house1_Ann.conf
  Billy = house1_Billy.conf
}
```

Then, each user configuration file contains the user presence schedule as well as a set of the activities done by the user. An example is shown in the Listing 4.3.

After the creation of these configuration files for all the houses of the community, the ANTgen uses it to create the consumption profiles. Since ANTgen has its own appliances models to understand the appliance consumption pattern, it knows the behavior of each appliance, so, based on the schedule chosen by the user, the consumption profiles are created. It gives as an output a set of CSV files containing power consumption values and their timestamps with a frequency of 1Hz (1 value per second) for every house, activity, appliance and user. Then, it is resampled to a frequency of 1/60Hz (1 power value per minute) because it is more appropriate and easier to work with.

Figure 4.2 demonstrates a house consumption profile generated by this AMBAL trace generator. In the image it is possible to visualize four different consumption graphs:

- Per-appliance consumption: consumption of each appliance of the house
- Per-activity consumption: consumption of each activity done

Listing 4.3: ANTgen - User Configuration file

```
{
    [GENERAL]
    name = Ann

    [presence]
    monday = 00:00-08:30, 14:00-24:00
    tuesday = 00:00-08:30, 14:00-24:00
    wednesday = 00:00-08:30, 14:00-24:00
    thursday = 00:00-08:30, 14:00-24:00
    friday = 00:00-08:30, 14:00-24:00
    saturday = 00:00-08:30, 14:00-24:00
    sunday = 00:00-24:00

    [activity_dishwashing]
    model = KITCHEN/dishwasher.conf
    daily_runs = 1
    monday = 18:00-22:00
    tuesday = 07:00-12:00, 17:00-22:00
    wednesday = 07:00-14:00
    thursday = 07:00-12:00, 17:00-22:00
    saturday = 09:00-22:00
    sunday = 09:00-20:00

}
```

- Per-user consumption: consumption done by each user of the house
- Aggregate consumption: total consumption of the house

Using these consumption profiles, the Consumption Generator module aggregates the consumption of all community and separate the flexible consumption from the not-flexible consumption, as showing in Figures 4.3 and 4.4.

Then, the flexible timeslots are extracted from these profiles according to the periods of the day where the appliances are being used. From each timeslot, it is extracted the start time (where consumption starts) and end time (where consumption ends), appliance (which appliance is being used), house (which house is using the appliance), duration (in minutes), power (average of its power consumption) and max power, as shown in Listing 4.4.

Listing 4.4: JSON Representation of a Timeslot.

```
{
    "Start": "27-03-2022 14:30",
    "End": "27-03-2022 14:45",
    "Duration": 15,
    "MaxPower": 6329.33,
    "Power": 4950.59
    "Appliance": "Dishwasher",
    "House": 2
}
```

As previously said, only the flexible appliances are taken in account in order to extract the timeslots. This way, an appliance is considered flexible if its consumption can be changed over the time; i.e., when it can be used in another period of the day. On the other hand, a not-flexible one has to be used in the exact time and it consumption can not be moved over the time.

For instance, a washing machine is flexible once we can use it at 9 am or 1 pm or another time and does not make much difference. The same does not happen with the tv or the pc. If someone wants to watch tv at 9 am, it will not start watching tv at 1 pm instead of 9 am - makes a big difference.

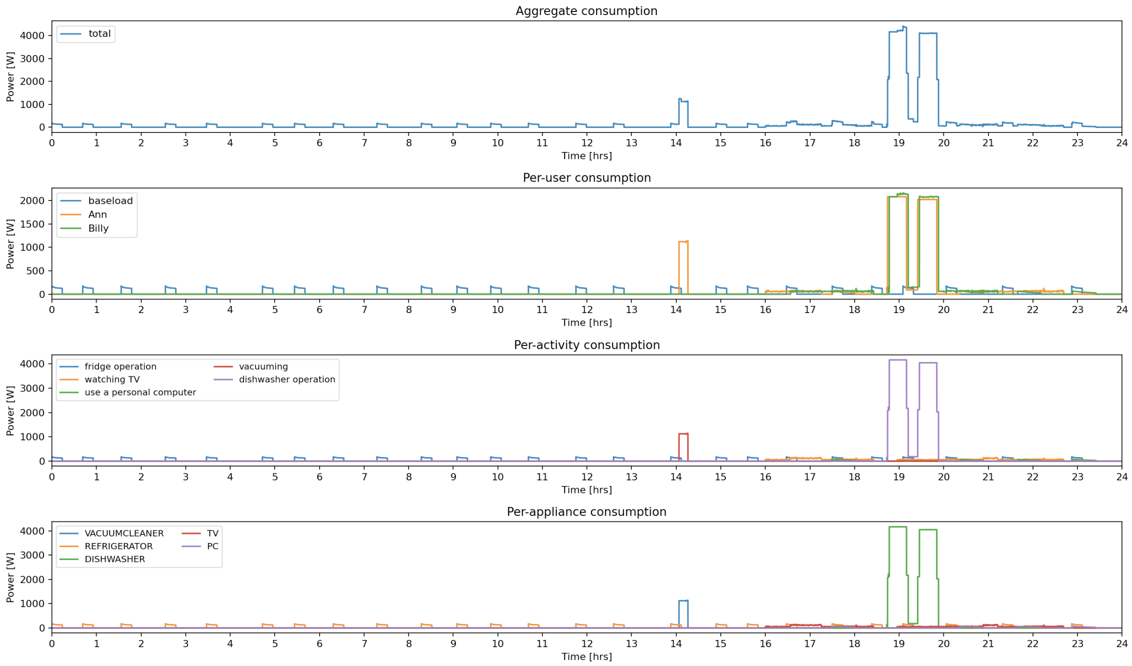

Figure 4.2: Consumption profile of house

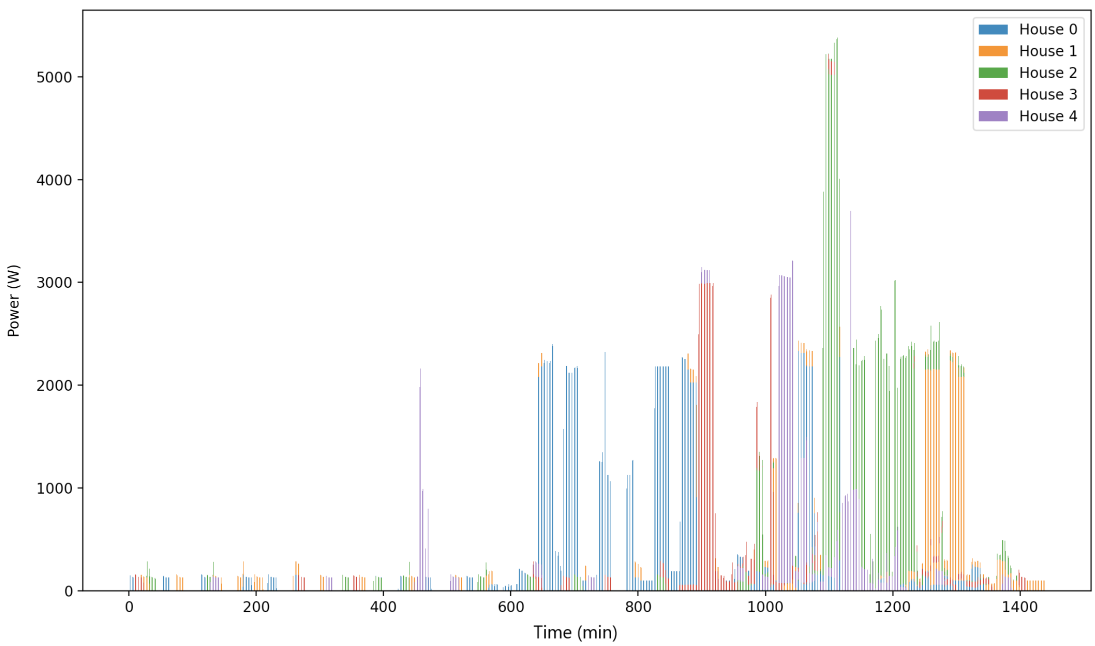

Figure 4.3: Aggregated consumption profile of the community

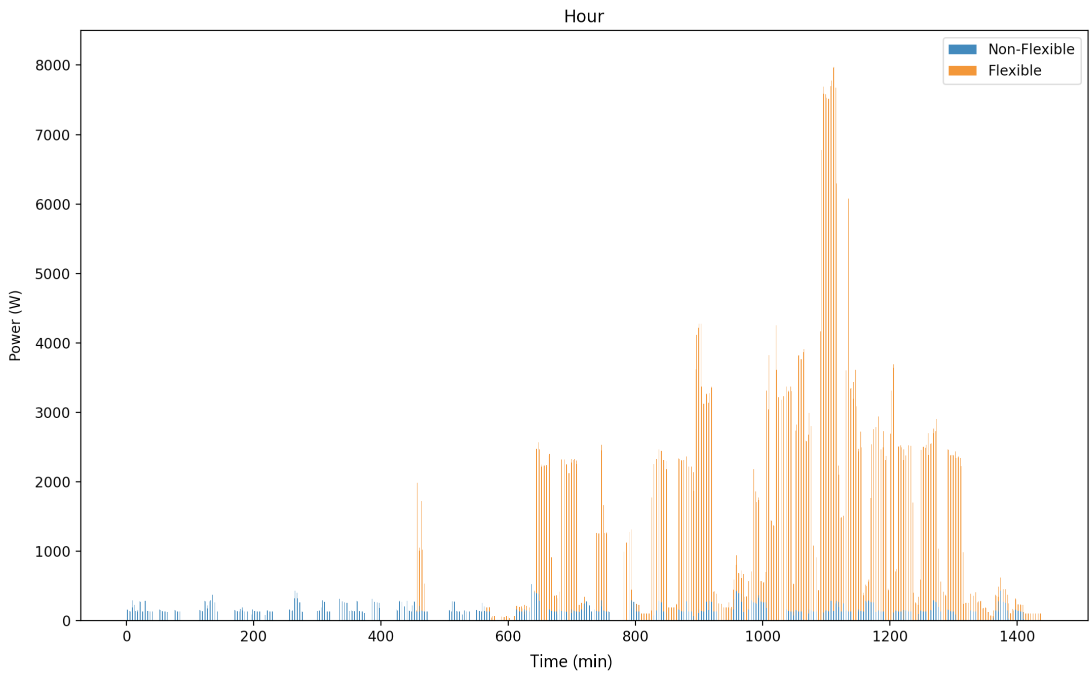

Figure 4.4: Flexible and not-flexible consumption of the community

Furthermore, in general, the flexible appliances are the ones which consume more energy, leading to a big increase of the total consumption, and, for that reason, are the ones which need more attention.

To conclude, another important aspect is that an appliance can be flexible in power or in time. Flexible in power means that we can adjust its power to consume more in less time or to consume less in more time (for example, an electric oven which consumes 2.3kWh in a hour, can be adjusted to consume 1.15kWh per hour in two hours) while flexible in time means that we can adjust it time to consume the same in other times of the day. For this simulator, only flexible in time appliances are considered.

Saying this, follows a list of the flexible and not-flexible appliances:

- Flexible Appliances: Dishwasher, Vacuum Cleaner, Washing Machine, Dryer, Iron, Cooking Stove
- Not-Flexible Appliances: Amplifier, Breadcutter, CD Player, Cofee Maker, Cooking Stove, Freeze, Kettle, PC, Printer, Refrigerator, Toaster, TV

4.2.3 Data Acquisition

Data Acquisiton is the module responsible for retrieving the weather data from a specific location (lat and long coordinates), namely the solar irradiance data (Global Horizontal Irradiance (GHI), Direct Normal Irradiance (DNI) and Diffuse Horizontal Irradiance (DHI)), for the same dates as the consumption profiles.

The DNI refers to the amount of solar radiation that comes perpendicular to a surface (ground or something parallel to the ground) and DHI represents the solar radiation that does not arrive on a direct path from the sun as DNI, but is scattered by clouds and particles in the atmosphere and comes equally from all directions. Both DNI and DHI are components of GHI, which is the total amount of radiation received by a surface which is horizontal parallel to the ground.

$$GHI = DNI * cos(\alpha) + DHI \text{ , where } \alpha \text{ is solar zenith angle} \tag{4.1}$$

These weather data are important once they are used to get the PV power forecast. This module was developed to support data from multiple sources: Solcast (real-time data), weather models and datasets (CSV files - offline data). In the future, other sources can be integrated.

Weather Models

What concerns to the Weather models, there are five (GFS, HRRR, RAP, NAM and NDFD) and all of them are supported by PVLib [1].

- Global Forecast System (GFS) [2]: GFS is the US model that provides forecasts for the entire globe. It is updated every six hours, runs at two resolutions (0.25 deg and 0.5 deg) and has a time resolution of 3 hours. It should be used for forecasts for 1-7 days anywhere on Earth.
- High Resolution Rapid Refresh (HRRR) [3]: HRRR is the most accurate model, however, it is only available for 15 hours. In addition, it runs at 3 km resolution and it is updated every hour. It should be used for forecasts for less than 24 hours and covers only the American continent.
- Rapid Refresh (RAP) [4]: RAP is the parent of HRRR. It is updated every hour and runs at 13, 20 and 40 km resolutions. It should be used for forecasts for less than 24 hours and only covers most of North America.
- North America Mesoscale (NAM) [5]: NAM just covers North America and is updated every six hours with a 20 km resolution.
- National Digital Forecast Database (NDFD) [6]: NDFD, in opposite to others mentioned, is not a model. It is a collection of forecasts made by National Weather Service updated every six hours. NDFD is available for United States and should be used for forecasts at all time horizons.

The summary of them are present in the following table.

Weather Model	Update Freq	Resolution	Forecasts	Cover
GFS	6 hours	0.25 deg and 0.5 deg	1-7 days	anywhere on Earth
HRRR	1 hour	3 km	< 24 hours	continental United States
RAP	1 hour	13, 20 and 40 km	< 24 hours	most of North America
NAM	6 hours	20 km	-	North America
NDFD	6 hours	-	all time horizons	United States

Table 4.1: Summary of weather models

The main problem in using these weather models to retrieve solar irradiance data is related to the fact that the minimum update frequency is 1 hour, which is not appropriate for the pv power forecast. When downsampling to a granularity of 1 minute, the values do not demonstrate a good representation of the real values.

Datasets (CSV)

Although datasets are supported in the simulator, the main limitation is that the datasets are previously generated (historical weather data) and are not as real as the real time data.

[1] see https://pvlib-python.readthedocs.io/en/v0.4.0/forecasts.html#weather-models
[2] see https://pvlib-python.readthedocs.io/en/v0.4.0/forecasts.html#gfs
[3] see https://pvlib-python.readthedocs.io/en/v0.4.0/forecasts.html#hrrr
[4] see https://pvlib-python.readthedocs.io/en/v0.4.0/forecasts.html#rap
[5] see https://pvlib-python.readthedocs.io/en/v0.4.0/forecasts.html#nam
[6] see https://pvlib-python.readthedocs.io/en/v0.4.0/forecasts.html#ndfd

The platform supports the use of datasets if they contain at least three specific weather columns: ghi, dni and dhi.

Solcast allows to download some weather datasets with different granularities: 5 minutes, or 30 minutes, for example. Compared to weather models, five minutes is a very good granularity but it is composed by pre-acquired data.

Solcast (Real Time)

Solcast also provides actual and forecast solar irradiance and power data, globally, using satellites and surface measurements. The datasets limitation is solved by using the solcast real-time data. However, it has a maximum limit of 10 data requests that can be done without a premium plan. About the granularity, it has a minimum granularity of 30 minutes.

Date	DNI	DHI	GHI
2021-12-31 17:00:00	77	42	55
2021-12-31 17:01:00	301	77	135
2021-12-31 17:02:00	443	100	223
2021-12-31 17:03:00	566	108	302
2021-12-31 17:04:00	645	115	378
2021-12-31 17:05:00	664	130	442
2021-12-31 17:06:00	701	134	495
2021-12-31 17:07:00	725	135	531

Table 4.2: Example of Solar irradiance data for 7 minutes

4.2.4 Renewable Energy Generator

The *Renewable Energy Generator* module provides the tools necessary to generate the profiles from RES. In the current version, there is no wind production. Only PV production is supported using PVLib-Python library[7] and weather data acquired from Solcast[8].

This enables the generation of solar PV production profiles with sampling intervals down to 5-minutes when using historical weather data and 30-minutes when using real-time measurements. This tool provides a set of functions to simulate the performance of photovoltaic energy systems.

PVLib is composed by a set of different PV modules and a set of different inverters. When running the model responsible to converting the weather data to PV forecast (ModelChain), a system (with a PV module as well as an inverter) and a location have to be chosen. In this case, the simulator uses the module Canadian Solar CS5P 220M 2009 and inverter ABB MICRO 0 25 I OUTD US 208 208V.

The number of houses in the community is also a factor which affects the PV power forecast because depending on the number of houses, more PV modules should be installed to cover the members needs, leading to an increase of the PV production.

In order to simplify, the PV is generated for a specific panel type and then it is normalized to 1 and multiplied by the installed power (as shown in Figure 4.5). This way, it is easier to configure the system instead of needing to identify the quantity of solar panels and inverters.

4.2.5 Community Generator

The last module of the Community Simulation is the Community Generator. It receives as input the community consumption (from the Consumption Generator module) as well as the PV production (from

[7] https://github.com/pvlib/pvlib-python
[8] https://solcast.com/

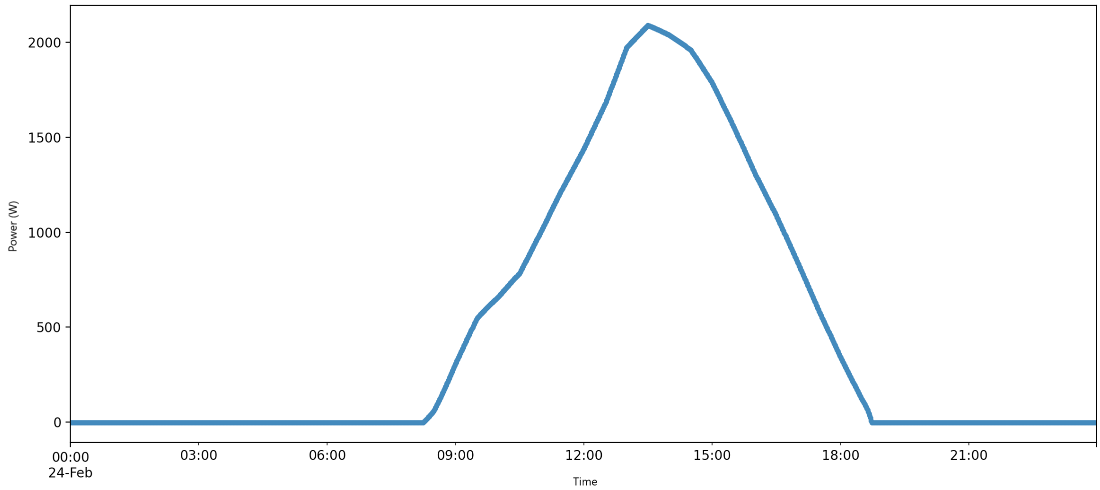

Figure 4.5: PV Power Production after the normalization and multiplied by 3.45 kW

the Renewable Energy Generator) with a time resolution of 1 minute (Figure 4.6). The main goal of this module is to return a community with the consumption, production and also the netload, which is calculated based on the other two columns.

The netload (Figure 4.7) is the total community demand minus the solar production, as represented in the following equation:

$$Netload = Nt_h = D_h - P_h \text{ , } \forall h \in [1, 24] \tag{4.2}$$

(where D_h is the Demand in hour h and P_h is the PV Production in hour h)

$$Nt_d = \begin{cases} = 0 & \text{, } D_h = P_h \text{ (production and demand perfectly balanced)} \\ < 0 & \text{, } D_h < P_h \text{ (waste of production)} \\ > 0 & \text{, } D_h > P_h \text{ (not enough production)} \end{cases} \text{ , } \forall h \in [1, 24] \tag{4.3}$$

One important aspect in this module is the fact that it was created in isolation in order to allow to be replaced by a community dataset (which has the three mentioned columns) or another approach.

As showed in Table 4.3, in the first 4 minutes, the netload is negative, which means that the production is higher than demand (there are a waste of some production because not all production is used). In the opposite, in the last 4 minutes there are more consumption than production (netload is positive), what means that there are not enough production to fullfill the consumption needs, leading to the need to acquire energy from the grid.

Date	Demand	Production	Netload
2022-02-25 18:23:00	388.21 W	605.29 W	-217.08 W
2022-02-25 18:24:00	386.56 W	581.52 W	-194.97 W
2022-02-25 18:25:00	385.17 W	557.88 W	-172.72 W
2022-02-25 18:26:00	413.46 W	534.35 W	-120.89 W
2022-02-25 18:27:00	537.63 W	510.92 W	26.72 W
2022-02-25 18:28:00	535.62 W	487.55 W	48.07 W
2022-02-25 18:29:00	526.22 W	464.22 W	62 W
2022-02-25 18:30:00	451.24 W	440.88 W	10.36 W

Table 4.3: Example of Netload calculation.

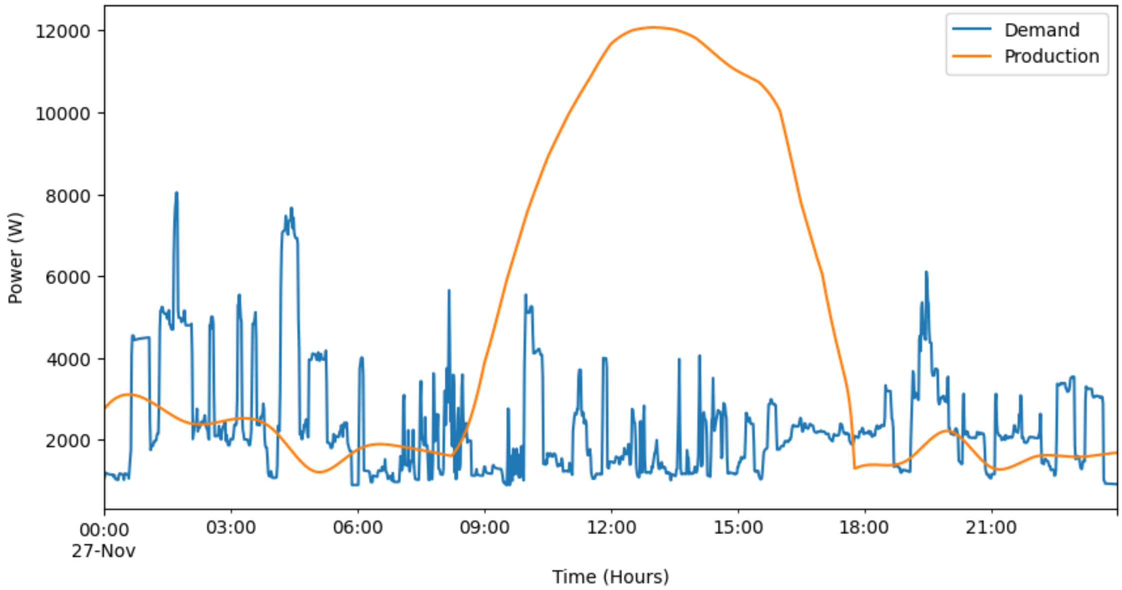

Figure 4.6: Demand and Production Graph

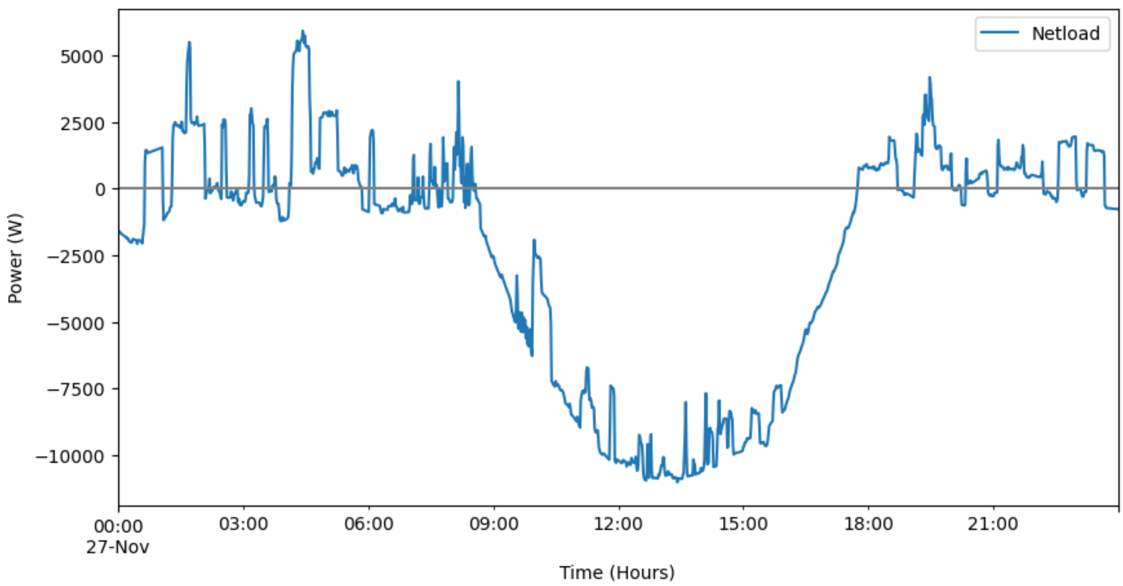

Figure 4.7: Netload Graph.

4.3 Load Balancing Evaluation

The second part of the simulator, named "load balancing evaluation", is composed of three modules: Community Manager, Exchanges Module and Evaluation. Similar to the first part of the simulator, this part can be used independently. It is responsible for using the Community Generator output to take some decisions and do the best management of the renewable resources of the community, i.e., make the best use of the resources.

4.3.1 Community Manager

This is the module where the optimization that will be explained in detail in Chapter 5 happens.

In few words, Community Manager (CM), which is the responsible for avoiding a bad use of the resources, have to analyse all the community consumption and production information and take decisions based on them, taking in consideration the existent constraints.

The schedule that the users want is not considered (which decreases the user satisfaction). In this optimization, although the user flexibility is considered (how flexible is the user to use the appliances some hours later or some hours before), the community is what is most taken in account, even if it not the best option for the user.

This module tries to give as many timeslots as possible without violating the defined constraints, having as priority the community management. As previously referred, the main goal is to avoid, as much as possible, a waste of renewable resources, and, at the same time, use, as less as possible, energy from grid (from non-renewable sources). This way, it changes the timeslots usage schedule according to the best for the community. For instance, if a user wants to use a timeslot at 7 pm but we have more production at 5pm instead of 7pm, then it moves the timeslot to start 2 hours earlier. It will allow a better management of the commuity.

It is important to highlight that only the start and end times of the timeslots can be changed. The other timeslots information (duration, house, appliance) can not be changed.

In Chapter 5 will be introduced the 2-step optimization. In addition, will be explained the sets and parameters of the optimization, as well as the decision variables. Furthermore, details will be provided about the different assumptions, constraints and objective functions used in order to have good results and also an efficient community.

Timeslots Division

Before the optimization starts, the CM has to divide each timeslot in a set of subitems according to the number of bins per hour. When there is a timeslot with a duration of 2 hour and 45 minutes, it will correspond to 3 subitems considering 1 bin per hour (bins of 1 hour), or 6 subitems considering 2 bins per hour (bins of 30 minutes) or 11 subitems when there are bins of 15 minutes (4 bins per hour). For each subitem of a timeslot, the energy for that time period (i.e. according to bin size) is calculated, and then used as input parameter in the optimization process.

In general, since bins of 1 minute are not used due to time limitations and complexity, it was decided that each subitem should be composed by Energy instead of Power. Although they are related, each one has it own purpose. Power is the rate at which work is done while the Energy is the capacity to do work. Basically, Energy is Power integrated over time.

Since each bin has a time interval (start and end), it makes more sense calculating the energy in that period of time and can also provide more realistic results. For instance, a vaccum cleaner with a power of 1000W being used for 2 hours consumes 2 kWh of Energy (2000 Wh).

Furthermore, in order to have a better idea of how these two units affects each other, see the following example: A Liquid Crystal Display (LCD) TV (low-power appliance) with a power of 100W has to be used during 10 hours to reach 1kWh, while a clothes dryer (high-power appliance) with a power of 2kW just needs 30 minutes to reach the same energy (1kWh). For this reason, this optimization only focus in flexible appliances (appliances which consumption can be shifted to another period of the day), which are also high-power appliances.

Consumption Profiles Update

After the distribution of the timeslots by the CM, the consumption profiles of every house as well as the community profile have to be updated. In order to do this, the bins where the timeslots are placed have to be converted to time periods, according to the number of bins per hour. For instance, when using bins of 30 minutes, the bin 1 should correspond to the period 00:00-00:29, the bin 2 should be 00:30-00:59, and so on. Table 4.4 demonstrates the conversion from bins to time periods when using bins of 15 minutes.

Bin (b)	Start Period	End Period
1	00:00	00:14
2	00:15	00:29
3	00:30	00:44
4	00:45	00:59
5	01:00	01:14
6	01:15	01:29
7	01:30	01:44
8	01:45	01:59

Table 4.4: Conversion from bins to time periods when using bins of 15 minutes

As it is possible to see, bin 6 corresponds to the period 01:15-01:29, but, one question arises: "How can it be converted from bin number to time period programatically?". It is simple using Equations (4.4) and (4.5), where fact is the number of bins per hour and s is the bin size.

$$h_{start} = floor((b-1)/(fact)) \tag{4.4}$$

$$m_{start} = (b-1)\%(fact) * s \tag{4.5}$$

In this case, bin 6 corresponds to b = 6, and fact = 4 because, since it is being used bins of 15 minutes, one hour contains 4 bins. In this way, in the start period, the h equals to floor(5/4) = floor(1.25) = 1 hour, and m equals to 5%4*15 = 1*15 = 15 minutes (start period = 01:15). To calculate the end period, just need to sum the bin size to the start period minus 1.

4.3.2 Evaluation

With the goal of evaluating different load balancing schemes, some performance metrics have been developed. The main idea behind these metrics are to facilitate the way that the simulator can be used to test and evaluate different situations. It calculates the metrics for the consumption data before the action of the CM, after the first step of the optimization and after the second step, thus allowing to compare the results. For example, with this module, it is possible to change or add a new constraint to the optimization, and check how it affects the way that the community is optimized, it is, the way the timeslots are distributed. Furthermore, the objective function can be adapted to support other goals and analyse the impact on the grid. Imagine someone which wants to prioritize the community members which have bigger houses. This way, having in mind that β represents the production, Nt represents the Netload and D represents the Demand, the developed metrics are:

P_g: the average power (W) used from grid in the day

$$P_g = \frac{\sum_{h=1}^{60*24} Nt_h}{60*24} \text{ , } Nt_h > 0 \tag{4.6}$$

P_{wpv}: the average power (W) not used (wasted) from PV in the day

$$P_g = \frac{\sum_{h=1}^{60*24} Nt_h}{60*24} \text{ , } Nt_h < 0 \tag{4.7}$$

P_{pv}: the average power (W) used from PV in the day

$$P_{pv} = \frac{\sum_{h1=1}^{60*24} D_{h1} + \sum_{h2=1}^{60*24} Nt_{h2}}{60*24*2} \text{ , } Nt_{h1} <= 0, Nt_{h2} > 0 \tag{4.8}$$

E_g: the amount of energy (Wh) used from grid in the day

$$E_g = \frac{\sum_{h=1}^{60*24} Nt_h}{60*24} * 24 \text{ , } Nt_h > 0 \tag{4.9}$$

E_{wpv}: the amount of energy (Wh) not used (wasted) from PV in the day

$$E_g = \frac{\sum_{h=1}^{60*24} Nt_h}{60*24} * 24 \text{ , } Nt_h <= 0 \tag{4.10}$$

E_{pv}: the amount of energy (Wh) used from PV in the day

$$E_{pv} = \frac{\sum_{h1=1}^{60*24} D_{h1} + \sum_{h2=1}^{60*24} Nt_{h2}}{60*24*2} * 24 \text{ , } Nt_{h1} <= 0, Nt_{h2} > 0 \tag{4.11}$$

$PK_{g,max}$: the maximum grid peak (W) in the day

$$PK_{g,max} = MAX(D_h) \text{ , } Nt_h > \beta \tag{4.12}$$

$PK_{g,min}$: the minimum grid peak (W) in the day

$$PK_{g,min} = MIN(D_h) \text{ , } Nt_h > \beta \tag{4.13}$$

$PK_{mg,max}$: the maximum magnitude peak (W) in the day

$$PK_{mg,max} = MAX(Nt_h) \text{ , } Nt_h > \beta \tag{4.14}$$

$PK_{mg,min}$: the minimum magnitude peak (W) in the day

$$PK_{mg,min} = MIN(Nt_h) \text{ , } Nt_h > \beta \tag{4.15}$$

PK_n: the number of peaks

$$PK_n = COUNT(Nt_h) \text{ , } Nt_h > \beta \tag{4.16}$$

T_p: the number of placed (flexible) timeslots
T_{np}: the number of (flexible) timeslots that were not placed
E_p: the amount of energy (Wh) of the placed (flexible) timeslots
E_{np}: the amount of energy (Wh) of the (flexible) timeslots that were not placed.

Using these metrics, it is possible to analyse the potential of some load balancing schemes. The smaller the amount of energy used from grid, the better the timeslots are distributed in the community, since it will give the possibility to use more energy from solar panels and will also avoid a waste of renewable production. On the other hand, the number of placed and not placed timeslots will allow to have an idea of the quantity of timeslots that were been distributed, taking in consideration that the best scheme is also the one which gives the highest number of timeslots, according to the problem constraints.

4.4 Dataset Structure

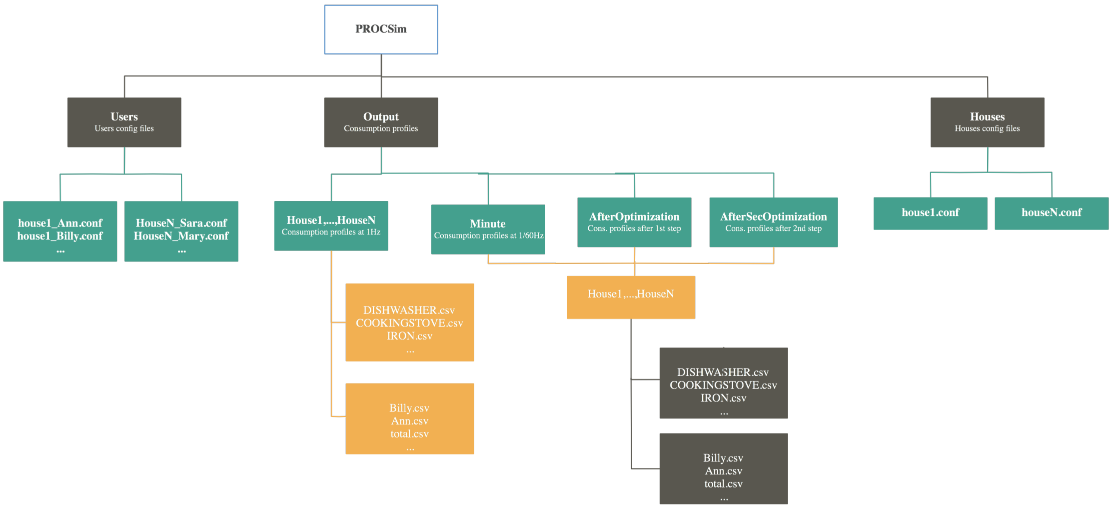

Figure 4.8: Dataset Structure

Once the simulations are concluded, the EC dataset is generated. The generated datasets are organized as represented in Figure 4.8.

The users folder stores all the user configuration files that are created by the simulator using the json input file. Each user file (.conf) contains the presence schedule as well as the schedule of each activity that will be done by the user. The name of it is composed by the number of the house followed by the user name (e.g. house1_Mary.conf).

On the other hand, the houses configuration files are stored in the houses folder. Basically it is composed by a set of files (one for each house) containing the list of user configuration files of the house.

Furthermore, after generating the 1 Hz houses consumption profiles based on the configuration files, they are stored in the output folder. It is the folder which contains all the consumption profiles for all different phases of the simulator process.

Inside of it, there are the following folders:

- "house1, ..., houseN" - consumption profiles generated by ANTGen with a frequency of 1Hz
- "minute" - consumption profiles resampled from 1 value per second (1Hz) to 1 power value per minute (1/60Hz).
- "afteroptimization" - consumption profiles after the 1st step of the optimization done by the community manager

- "aftersecoptimization" - consumption profiles after the 2nd step of the optimization done by the community manager

In each of these folders there are some dataframes: community (containing the total consumption of the community), community_baseload (total baseload of the community), community_not_baseload (total consumption of the community minus the total baseload), energy (community production) and, finally, netload (demand minus the production, which corresponds to the grid consumption). Besides this, there are folders for each house containing the total consumption of the house, consumption of each activity, consumption of each user and also the consumption for each appliance.

Chapter 5

Energy Community Optimization

In this chapter, the day-ahead 2-step approach will be explained. All the details about the complete optimization process will be provided, namely, the problem formulation (how the problem was formulated in order to use Knapsack to achieve the goals), Sets and Parameters (input sets that will be used in the process and input parameters that corresponds to the data that have to send to the optimization), Decision Variables (variables which store 0 or 1 depending if the item is in the bin or not and a matrix which will be used to store all these boolean variables), Assumptions (what we are assuming in order to solve this problem), Constraints (rules that allow to filter the problem combinations), Objective functions (defined function which allow to choose the best solution based on it) and OR Tools (the software used to simulate this optimization problem).

5.1 Introduction

As previously explained, Knapsack Problem is a combinatorial optimization problem used in linear programming for a user to learn how to formulate an equation that will optimally pack a knapsack with items of various weights. Each item usually has an associated value as well. The goal is to optimize the value of the bag while not exceeding the weight limit of the bag. This problem extends from the Basic Knapsack problem to Multiple Knapsack problem with multiple constraints and more complex objective functions.

5.2 Problem Formulation

In this problem, Multi Knapsack is used to distribute the timeslots over the day with the goal of balancing the demand and supply of the energy in a community in a way that the peak demand is also reduced, avoiding the need to acquire energy from non-renewable sources. The main challenge is to adapt the Multi Knapsack in order to use it to solve this problem.

There are B bins, where each bin corresponds to a specific period of the day, which can vary from some minutes to some hours. For instance, if 1-hour bins are used, there will be a total of 24 bins per day, while if bins of 15 minutes are used, the total will be 96 bins in a day (4 bins per hour).

Also there are N timeslots, and each timeslot is divided in a set of subitems depending on the bin size, and only the consumption in that bin period is considered. As the bin size decreases, the more subitems each timeslot will have. To clarify, consider the following example: There is a timeslot of 2 hours (from 9am to 11am). When using bins of 1 hour, it is divided in 2 subitems (09am-10am and 10am-11am), but, when using bins of 30 minutes, it corresponds to 4 subitems (09am-09.30am, 09.30am-10am, 10am-10.30am aand 10.30am-11am). The weight/consumption of each subitem will be calculated independently according to the consumption in that time period (it is, the weight of the subitem 09.30am-10am will consider just the production in the interval of 09.30am-10am).

The objective is that each bin will contain some subitems of the timeslots (where every subitem has a specific weight) taking in consideration that it can not exceed the bin maximum capacity. Only the timeslots that are in line with the defined constraints can be a part of the optimal solution. In addition, bins without attributed timeslots are also allowed.

This optimization problem was divided in two different steps, and although they have the same constraints, each step has its own objective function. While in the first step of the 2-step optimization the aim is to use only the available PV production without using any energy from grid, in the second step, energy from the grid can be used.

The main goal of the 2nd step is to pack all the timeslots that were not placed in the previous step (because there were not enough renewable energy to cover their needs), minimizing the energy from the grid, and taking in consideration that the amount of energy acquired from the grid can not exceed the established baseload value, which is the sum of the PPCs of every house.

The reason for being divided in two steps was essentially due to the fact that the second part will only be used if not all timeslots have been placed in the first part. First of all, it tries to cover the appliances consumption using only energy from renewable sources in the first step (without buying from grid), and, if not possible, the second part is used.

In this process, the optimal solution will be the one which packs the maximum number of timeslots, using, as less as possible, energy from the grid, and also, wasting, as less as possible, the available production. The Multiple Knapsack will return the best combination of items of each bin which gives the best value for the objective function.

With this goal, a set of parameters and linear constraints were defined, as well as an objective function, which will contribute to getting the best solution.

Energy (Wh)	w1	w2	w3	w4	w5	...
Date (h)	d1	d2	d3	d4	d5	...
Flexibility (h)	f1	f2	f3	f4	f5	...
Max (Wh)	m1	m2	m3	m4	m5	...
Number	n1	n2	n3	n4	n5	...

Table 5.1: Parameters for each item.

Capacity (Wh)	c1	c2	c3	c4	c5	...
Max (Wh)	b1	b2	b3	b4	b5	...

Table 5.2: Parameters for each bin

Factor (min)	t1
Nº Bins per hour	h1
Baseload (Wh)	l1

Table 5.3: Other parameters

5.3 Sets and Parameters

The Sets and Parameters are, in few words, the inputs of the problem. The problem is not solved if not all the following sets and parameters are defined. They are extremely necessary to solve the problem.

The sets are the data collections which contains the timeslots, the subitems and also the bins, while the parameters are the information that the solver knows about each bin and subitem (some examples are: subitem flexibility, subitem capacity, bin capacity, bin maximum and subitem maximum).

Tables 5.1 to 5.3 demonstrates the parameters used for each item, for each bin and for the whole problem.

5.3.1 Sets

Number of Items: $N = \{0, 1, 2, 3, 4, 5, 6, ...\}$
Number of SubItems: $K = \{\{0, 1, ...\}, \{0, 1, ...\}, \{0, 1, 2, 3, ...\}, ...\}$
Number of Bins: $M = \{1, 2, 3, 4, 5, 6, ...\}$

5.3.2 Parameters

$w_{i,p}$: the amount of energy used in Watt-hour (Wh) for subitem p of timeslot i, $\forall i \in N$
$d_{i,p}$: the hour wanted by the member for subitem p of timeslot i, $\forall i \in N$
$f_{i,p}$: the flexibility of subitem p of timeslot i, $\forall i \in N$
$m_{i,p}$: the maximum/peak power used (W) for subitem p of timeslot i, $\forall i \in N$
$n_{i,p}$: the timeslot number of the subitem p of timeslot i, $\forall i \in N$
c_j: the energy capacity (Kwh) for bin j, $\forall j \in M$
b_j: the power capacity maximum/peak (W) for bin j, $\forall j \in M$
t: the size of each bin in minutes (min)
h: the number of bins per hour, where h = $\frac{t}{60}$

5.4 Decision Variables

In this problem, there is a 3D matrix which stores a boolean variable in every position that indicates if the subitem of the timeslot is placed or not in the bin (as shown in Equation (5.1)). By default all the values in the matrix are 0. Every time that a subitem is placed in a bin, the value for that subitem and bin in the matrix is changed to 1. For instance, if the subitem 0 of the timeslot 3 is placed in the bin 2, then $x_{0,3,2}$ = 1.

Tables 5.4 and 5.5 show the 2D matrix for each timeslot, considering that it is not easy to represent a 3D matrix.

$$x_{i,p,j} = \begin{cases} 0 & \text{if subitem } p \text{ of timeslot } i \text{ is not placed in bin } j \\ 1 & \text{if subitem } p \text{ of timeslot } i \text{ is placed in bin } j \end{cases}, \forall i \in N, \forall p \in K, \forall j \in M \quad (5.1)$$

$x_{0,p,j}$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**...**
1	$x_{0,1,0}$	$x_{0,1,1}$	$x_{0,1,2}$	$x_{0,1,3}$	$x_{0,1,4}$	$x_{0,1,5}$	$x_{0,1,6}$	...
2	$x_{0,2,0}$	$x_{0,2,1}$	$x_{0,2,2}$	$x_{0,2,3}$	$x_{0,2,4}$	$x_{0,2,5}$	$x_{0,2,6}$	...
3	$x_{0,3,0}$	$x_{0,3,1}$	$x_{0,3,2}$	$x_{0,3,3}$	$x_{0,3,4}$	$x_{0,3,5}$	$x_{0,3,6}$	...
4	$x_{0,4,0}$	$x_{0,4,1}$	$x_{0,4,2}$	$x_{0,4,3}$	$x_{0,4,4}$	$x_{0,4,5}$	$x_{0,4,6}$	...
5	$x_{0,5,0}$	$x_{0,5,1}$	$x_{0,5,2}$	$x_{0,5,3}$	$x_{0,5,4}$	$x_{0,5,5}$	$x_{0,5,6}$	...
...	...	...	...	...	...	...	...	...

Table 5.4: Matrix of timeslot 0 - $x_{0,p,j}$

$x_{i,p,j}$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**...**
1	$x_{i,1,0}$	$x_{i,1,1}$	$x_{i,1,2}$	$x_{i,1,3}$	$x_{i,1,4}$	$x_{i,1,5}$	$x_{i,1,6}$	...
2	$x_{i,2,0}$	$x_{i,2,1}$	$x_{i,2,2}$	$x_{i,2,3}$	$x_{i,2,4}$	$x_{i,2,5}$	$x_{i,2,6}$	...
3	$x_{i,3,0}$	$x_{i,3,1}$	$x_{i,3,2}$	$x_{i,3,3}$	$x_{i,3,4}$	$x_{i,3,5}$	$x_{i,3,6}$	...
4	$x_{i,4,0}$	$x_{i,4,1}$	$x_{i,4,2}$	$x_{i,4,3}$	$x_{i,4,4}$	$x_{i,4,5}$	$x_{i,4,6}$	...
5	$x_{i,5,0}$	$x_{i,5,1}$	$x_{i,5,2}$	$x_{i,5,3}$	$x_{i,5,4}$	$x_{i,5,5}$	$x_{i,5,6}$	...
...	...	...	...	...	...	...	...	...

Table 5.5: Matrix of timeslot i - $x_{i,p,j}$

5.5 Assumptions

When setting up a Mixed Integer Linear Programming (MILP) problem, it is important to include the main assumptions that we are making. For this problem, the main assumptions are the following:

- A flexible appliance can be used some hours before or some hours after the expected (does not need to be used in the exact period).
- The appliance is not turned off from the begin until the end of the timeslot, i.e., the timeslot is used all the time during its duration.
- The community members will only request timeslots that really want and that will be used.

5.6 Constraints

In this problem, there is also constraints. Constraints refer to establishing some rules that affect the solution directly and can be used to filter the number of possibilities.

In this problem, seven constraints were defined for different purposes, and, all together, allow to obtain a reliable solution.

5.6.1 Item Constraint

The first constraint of this problem is represented in Equation (5.2) and refers that an item of a timeslot can be, at most, in one bin. This means that if a timeslot is divided in a set of subitems, a specific subitem can not be in two or more different bins, meaning that it can not be repeated multiple times over the day.

$$\sum_{j=1}^{M} x_{i,p,j} <= 1 \text{ , } x_{i,p,j} \in \{0,1\}, \forall i \in N, \forall p \in K_i \tag{5.2}$$

5.6.2 Capacity Constraint

What concerns to the capacity constraint (Equation (5.4)), in few words, the energy capacity of each bin can not be exceeded. In other words, this constraint is used to guarantee that the sum of the amount of energy used in every subitem of one bin is not higher than the capacity of that bin.

In the first step, the capacity of each bin is just the PV Power production. On the other hand, in the second step, instead of just having the production, also a baseload β is taking in account, as shown in the Equation (5.3). β is 85% of the sum of all Peak Power Contracts (PPC) of all houses of the community.

$$\beta = 0.85 * \sum_{i=1}^{Z} PPC_i \tag{5.3}$$

(where PPC_i is the Peak Power Contract of the house i and Z = number of houses)

$$\sum_{i=1}^{N} \sum_{p=1}^{K} w_{i,p} * x_{i,p,j} <= c_j \ , \ x_{i,p,j} \in \{0,1\}, \forall j \in M \tag{5.4}$$

5.6.3 Timeslot Constraint

As referred previously, a timeslot is composed by a set of subitems. This way, one important constraint, represented in Equation (5.5), is that an item of a timeslot can not be in the same bin as other item of the same timeslot (a bin can not contain more than one item of the same timeslot). So, it is possible to have more than one subitem in one bin but it has to be from different timeslots. For instance, in a timeslot (appliance) with two items, the first item can not happen at the same time as the second item.

$$\sum_{i=1}^{N} \sum_{p=1}^{K} x_{i,p,j} <= 1 \ , \ x_{i,p,j} \in \{0,1\}, \forall j \in M \tag{5.5}$$

5.6.4 Asc Order Constraint

In addition, the Equation (5.6) shows that the items of a timeslot should be placed in consecutive bins ordered according to its position. For instance, if the first subitem is placed in the bin x, the second bin has to be placed in the bin x+1 and the third bin has to be placed in the bin x+2. This constraint is not applied in the first bin because there are no previous bin.

It makes sense for the problem since if we have a washing machine running from 2 to 4 pm, we can't change the order that it will be running. For instance, in a timeslot with 2 items, the first part can not happen after the second part.

$$0 <= \sum_{j=1}^{M} j * x_{i,p,j} - \sum_{j=1}^{M} j * x_{i,p-1,j} <= 1 \ , \ x_{i,p,j} \in \{0,1\}, \forall i \in N, \forall p \in K_i/\{0\}, \tag{5.6}$$

5.6.5 Desc Order Constraint

Similar to the previous constraint, a desc order was also implemented in order to guarantee that the timeslots are consecutively placed, especially what concerns to the first bin. Since the asc order constraint is not applied to the first bin, this one will make sure that we can only put subitems in the first bin if the others are in the following bins (except if the timeslot only has one subitem). The idea is the following: If we have a subitem of a bin placed in the first bin, and the bin of the second subitem is the bin x, then the first subitem should be placed in the bin x-1. This constraint is not applied in the last bin because there are no subsequent bins.

Moreover, it is also important to refer that, with this constraint, the optimization does not allow to give to a member of the community just a part of the timeslot - the entire timeslot should be used (for instance, if there's a timeslot for washing machine from 2pm to 4pm, it can't be used from 3pm to 4pm (50%) or 2.15 to 2.45 (25%) of the timeslot – it has to be used from 2 pm to 4pm as asked by the member).

This equation is represented in Equation (5.7).

$$0 <= \sum_{j=1}^{M} j * x_{i,p+1,j} - \sum_{j=1}^{M} j * x_{i,p,j} <= 1 \ , x_{i,p,j} \in \{0,1\}, \forall i \in N, \forall p \in K_i / \{len(K_i) - 1\}, \tag{5.7}$$

5.6.6 Maximum Capacity Constraint

Another capacity constraint is related to the maximum capacity of each bin. In simple terms, since we are considering Energy (Wh) instead of Power (W), the Capacity Constraint defined previously is not enough to guarantee the reduction of the peak demand. In order to avoid having that consumption peaks, a new constraint was created. It is responsible to ensure that the sum of the timeslots maximum power in a bin will never be higher than the maximum PV production power, as demonstrated in Equation (5.8).

$$\sum_{i=1}^{N} \sum_{p=1}^{K} * m_{i,p} x_{i,p,j} <= b_j \ , x_{i,p,j} \in \{0,1\}, \forall j \in M \tag{5.8}$$

5.6.7 Flexibility Constraint

The last constraint of the problem is the Flexibility one (Equation (5.10)). It is used to fix the limit of hours that the bin can be moved from the real wanted time. In other words, the flexibility is considered the number of hours which we can subtract or add to the wanted hour. For example, if a member wants to use the dishwasher from 8am to 9am (1 hour of duration), and considering a flexibility of 2 hours, the dishwasher can be used between 6am to 11am (2 hours before the start and 2 hours after the end). This flexibility is a parameter which is calculated according to the flexibility of the appliance (defined in Table 5.6) and also based on the house flexibility, in the following way:

$$f_i = f_a * f_h \tag{5.9}$$

(where f_a is the appliance flexibility and f_h is the house flexibility)

$$-h * f_{i,p} <= \sum_{j=1}^{M} (j - d_{i,p}) * x_{i,p,j} <= h * f_{i,p} \ , x_{i,p,j} \in \{0,1\}, \forall i \in N, \forall p \in K_i \tag{5.10}$$

Appliance	Flexibility (in hours)
Dishwasher	12
Vacuum Cleaner	8
Washing Machine	10
Dryer Machine	5
Iron	5
Cooking Stove	1

Table 5.6: Appliances Flexibility

5.7 Objective Functions

For this 2-step optimization problem, two different objective functions were created: one for each step. The goal of the objective function of the original Knapsack is to maximize the value of each item packed in the bin, as shown in Equation (5.11), where v_i corresponds to the item value and $x_{i,j}$ the decision variable (0 if item i is not placed in the bin j and 1 otherwise).

In this case, that OF is not appropriate because the aim is not to give priority to the appliances which consume more energy. If we use that OF in our problem, the optimization will start by adding to the bins the most energy-consuming timeslots, and the ones which consume less will only be placed after all others if there are space for them. Obviously that this is not fair and also does not make any sense in our problem: what is expected is that all members have the same benefits.

$$maximize \sum_{i=1}^{N}\sum_{j=1}^{M} v_i * x_{i,j} \text{ , } x_{i,j} \in \{0,1\} \tag{5.11}$$

$$minimize \sum_{i=1}^{N}\sum_{p=1}^{K}\sum_{j=1}^{M}(\frac{c_j}{v_{i,j}} - v_{i,j} - min(0, c_j - v_{i,j})) * x_{i,p,,j} \text{ , } x_{i,p,j} \in \{0,1\} \tag{5.12}$$

$$maximize \sum_{i=1}^{N}\sum_{p=1}^{K}\sum_{j=1}^{M}(v_{i,j} + c_j - \beta_j) * x_{i,p,,j} \text{ , } x_{i,p,j} \in \{0,1\} \tag{5.13}$$

In the equation of the first step, once it is a minimization function, the lower values will be the chosen ones. It is, the best value for this function will be the lowest. In this sense, $\frac{c_j}{v_{i,j}}$ is the lowest value when $v_{i,j}$ is higher than c_j (when the load is higher than the bin capacity). But, since loads which exceeds the bin capacity can not be placed, a penalti have to be added (which is the minimum between 0 and the difference between the load and the capacity.

c_j - $v_{i,j}$ is positive (higher than 0) when bin capacity is higher than the timeslot load. On the other hand, if the capacity minus load is negative, then the timeslot does not exceed the bin capacity, so, the minimum between 0 and a negative number will be the negative number (which means a better solution). Otherwise, the minimum will be 0.

Furthermore, the idea of the objective function is also to start placing the most-consuming timeslots. For that reason, in the objective function the $v_{i,j}$ is subtracted (the timeslots with higher energy will minimize the function).

Finally, this function is multiplied by $x_{i,p,j} \in \{0,1\}$ because the function can only be calculated for the placed timeslots. If the timeslot i is placed, $x_{i,p,j}$ will return 1 and the function will multiply by 1. Otherwise, if the timeslot i is not placed, the function will multiply by 0, and will not be taken in consideration for the final result.

What concerns to the objective function of the second step, it is simpler because it is only used after some timeslots are placed. In contrast to the previous one, this is a maximization function and not a minimization function (the objective is to get the max value of the function).

All timeslots that can be used just with PV production and without any baseload were placed in the first step. In this step, there are just timeslots which the PV production is not enought to fullfil, leading to the need to get some energy from the grid (with a maximum value β). β refers to the baseload, and it is calculated in Equation (5.3). Subtracting β from c_j provides the bin capacity without considering the baseload (only the PV production). So, the goal is to add the timeslots to the bins which has more capacity, because, since there are no bins with sufficient production to cover the timeslot energy, then the timeslot have to be used in bins which have more production, allowing to reduce the quantity of energy used from the baseload.

Besides this, in the Equation (5.13) the $v_{i,j}$ (timeslot energy) is added because the goal is to start placing the timeslots which consume more energy (in the bins with more energy). For this reason, by adding the timeslot energy, it maximizes the function for the most-consuming timeslots.

Table 5.7 demonstrates the importance of choosing an objective function appropriate to the problem

	c_j	$v_{i,j}$	$c_j/v_{i,j}$	c_j-$v_{i,j}$	min(0, c_j-$v_{i,j}$)	minimize f(x)
c_j = 100 | $v_{i,j}$ = 10	100	10	10	90	0	0
c_j = 100 | $v_{i,j}$ = 20	100	20	5	80	0	-15
c_j = 100 | $v_{i,j}$ = 30	100	30	3.3333	70	0	-26.6667
c_j = 100 | $v_{i,j}$ = 40	100	40	2.5	60	0	-37.5
c_j = 100 | $v_{i,j}$ = 50	100	50	2	50	0	-48

Table 5.7: First Objective Function Example

of the objective function (OF) in this optimization problem. It is, since this OF is a minimization function, the best value for it will be the lowest one. Obviously that it is calculated by adding a set of values instead of just a value, taking in consideration that it can not exceed the bin capacity (100kWh in this case).

In this table, there are five timeslots (each one with a specific energy load) and just a bin (with a constant bin size). It is possible to see that the best value of the function (the lowest one) is the -48, which means starting by placing the timeslot of 50kWh instead of placing the others with lower capacities.

After that, since the bin capacity is 100kWh and there are 50kWh left, there are 3 solutions: Placing the timeslots $v_{i,j}$ = 10 and $v_{i,j}$ = 40 or the timeslots $v_{i,j}$ = 20 and $v_{i,j}$ = 30 or $v_{i,j}$ = 10 and $v_{i,j}$ = 30.

Adding the values of the objective functions of each timeslot, the result is: -37.5 for the bins of 10kWh and 40kWh, -41.6667 for the bins of 20kWh and 30kWh, and, finally, -26.6667 for the bins of 10kWh and 30kWh. The highest/worst value is the -26.6667 and makes sense since placing these two timeslots will allow a waste of 10kWh. On the other hand, the lowest (best) value is the -41.6667 (which corresponds to the bins of 20kWh and 30kWh).

Saying this, it is possible to conclude that, in this case, the best timeslots to be placed in the bin of 100kWh are the timeslots of 50kWh, 30kWh and 20kWhs, which avoids the need to get energy from outside of the community and also avoids the waste of energy.

	c_j	$v_{i,j}$	$c_j/v_{i,j}$	c_j-$v_{i,j}$	min(0, c_j-$v_{i,j}$)	minimize f(x)
c_j = 30 | $v_{i,j}$ = 40	30	40	0.75	-10	-10	-29.25
c_j = 40 | $v_{i,j}$ = 40	40	40	1	0	0	-39
c_j = 50 | $v_{i,j}$ = 40	50	40	1.25	10	0	-38.75
c_j = 60 | $v_{i,j}$ = 40	60	40	1.5	20	0	-38.5

Table 5.8: First Objective Function Example 2

Table 5.8 is similar to the previous table example, but, instead of varying the number of the timeslots using the same bin, there are just a timeslot and the goal is to use the objective function of the first step to decide in which bin the timeslot should be placed.

Taking this in account, according to the information present in the table, the lower value for the OF is the -39, which corresponds to placing the timeslot in the bin of 40kWh (which avoids energy waste as well as energy from grid). The second option is the 50kWh (which wastes 10kWh of energy). The third option is the 60kWh (leading to a waste of 20kWh of energy), and, finally, the worst option (which is not supported due to the capacity constraint) is placing in the bin of 30kWh (leading to the need to buy 10kWh of energy from non-renewable sources).

What concerns to the second step of the optimization process, where a baseload is added when there are not enough energy to cover timeslots, the results are present in Table 5.9.

In simple terms, there exists five bins and just a timeslot, and the main goal is to decide which bin should be chosen to place it. Since it is the second step, the timeslot has to be placed in the bin with the highest capacity, once the PV production will be higher and less energy have to be acquired from the grid (taking in account that the baseload is constant for all bins).

It is possible to see in column c_j - β_j that all the values are lower than the timeslot energy (12kWh).

	c_j	$v_{i,j}$	β_j	c_j - β_j	maximize g(x)	energy from grid
c_j = 5 | $v_{i,j}$ = 12	5	12	4	1	13	11
c_j = 10 | $v_{i,j}$ = 12	10	12	4	6	18	6
c_j = 12 | $v_{i,j}$ = 12	12	12	4	8	20	4
c_j = 13 | $v_{i,j}$ = 12	13	12	4	9	21	3
c_j = 15 | $v_{i,j}$ = 12	15	12	4	11	23	1

Table 5.9: Second Objective Function Example

That was the reason that it was not placed in the first step. However, adding a β = 4 allows to cover the timeslot.

Analysing the table, the conclusion is that the timeslot should be placed in the bin with a capacity of 15kWh because only 1kWh needs to be acquired from the grid. In the other cases, more energy has to be acquired from non-renewable sources.

5.8 OR Tools

Google OR Tools is an open-source software which allows to simulate different optimization problems in C++, Python, C# and Java. In few words, it is used for combinatorial optimization, which finds the best solution of a problem from a very large set of possible solutions.

It includes solvers for Constraint Programming (finds feasible solutions to a problem expressed as constraints), Linear and Mixed-Integer Programming (finds the optimal value of a linear objective function given a set of linear inequalities), Vehicle Routing (identifies best vehicle routes give constraints), Graph Algorithms (find shortest path in graphs), Scheduling (finds the optimal schedule for a complex set of tasks, some of which need to be performed before others), Bin Packing (packs as many objects of various sizes into a fixed number of bins with maximum capacities) [1], etc.

About Packing, there are many types of packing problems. Three main packing problems that are supported by OR Tools are the following:

- Knapsack: The goal is to pack a set of items with given values and sizes into a container with a maximum capacity in order to maximize the total value packed in the bag.
- Multiple Knapsacks: There are multiple knapsacks and the goal is to maximize the total value of the packed items in all knapsacks.
- Bin Packing: There are multiple bins with the same capacity. Unlike the knapsack, the number of bins is not fixed. The goal is to find the smallest number of bins that can hold all the items.

An example that shows the difference between the Knapsack and the Bin Packing is the following: In Multiple Knapsack, there are five trucks and a set of items have to be loaded into the trucks. In Bin packing, there are 20 trucks and the fewest trucks that hold all of them should be used.

[1] see https://developers.google.com/optimization/introduction/overview for more details

Chapter 6

Evaluation

In order to evaluate the 2-step optimization, some experiments will be done using the simulator, and the results will be evaluated according to some metrics. First of all, twenty different houses will be created (using different PPC values, different members and different appliances) to be used later in the experiments. After that, three scenarios will be experimented using three different strategies for each scenario. The first one will be varying the bins size and understand if the results are similar as expected, the second one will be varying the community sizes and analyze the scalability of the simulator, and the last one will be related to varying the houses flexibility and understand how can it affect the way that the timeslots are distributed.

6.1 Houses definition

Table 6.1 shows 20 different houses that will be used in the next experiments. The contracted power, total number of appliances, number of flexible appliances and number of people are described in the table, as well as the list of flexible appliances of each house. The flexible appliances presented in the table are: Washing Machine (WM), Vaccum Cleaner (VC), Dryer Machine (DM), Water Heater (WH) and DishWasher (DW).

In the next experiments, communities will be created based on combinations of these houses. The consumption profiles of each house is present in Appendix A.

6.2 Experiment 1: Variation in bins size

Varying the bins size, which is equivalent to varying the number of bins per hour, can create an impact in the optimization process. As the number of the bins per hour increases, more subitems each timeslot will have, leading to an increase of the possible combinations.

In this way, with the increase of possible combinations, the constraints and the objective function have to be evaluated for more cases (for every subitem).

Furthermore, since the consumption profiles have a granularity of 1 minute, depending on the bin size, this consumption has to be resampled because each subitem of a timeslot must have its own energy consumption according to the bin period.

For instance, if there are being used bins of 1 hour, a timeslot of 2 hours is divided in two subitems and needs to be placed in two bins while if the bin size is 30 minutes, the same timeslot has to be placed in four bins (since it will be divided in four subitems instead of two). In this sense, when using bins of 30

	Contracted Power	Num Flex Appl	Num Appl	Num People	Flexible Appliances
House 1	3.45 kVA	2	13	1	WM, VC
House 2	3.45 kVA	2	11	1	WM, VC
House 3	5.75 kVA	3	13	2	WM, VC, DM
House 4	10.35 kVA	5	16	4	WM, VC, DM, WH, DW
House 5	6.9 kVA	3	16	3	VC, WH, DW
House 6	3.45 kVA	2	13	2	DM, WH
House 7	5.75 kVA	2	13	3	WM, VC
House 8	4.6 kVA	3	14	4	WM, VC, WH
House 9	3.45 kVA	1	12	1	VC
House 10	3.45 kVA	2	13	2	VC, WH
House 11	5.75 kVA	3	14	4	WM, VC, DW
House 12	6.9 kVA	4	15	4	VC, DM, WH, DW
House 13	6.9 kVA	3	14	2	WM, VC, DM
House 14	3.45 kVA	1	10	1	WM
House 15	13.8 kVA	5	16	7	WM, VC, DM, WH, DW
House 16	10.35 kVA	5	16	6	WM, VC, DM, WH, DW
House 17	6.9 kVA	3	14	1	WM, VC, DM
House 18	3.45 kVA	2	13	1	WM, VC
House 19	3.45 kVA	1	12	1	DM
House 20	6.9 kVA	3	14	3	DM, WH, DW

Table 6.1: Houses definition

minutes it is just needed to resample 30 minutes while in bins of 1 hour it is needed to resample 1 hour of consumption data (which provides less accuracy in the results).

Taking this into consideration, it is possible to conclude that the subitem energy is more realistic when using a higher number of bins (when the bin size is lower). The most accurate results should be provided when using bins of 1 minute but it has a big disadvantage: creates many more combinations, as shown in table Table 6.2.

Bin size	Nº Bins per hour	Nº bins in a day
1h	1	24
30min	2	48
15min	4	96
10min	6	144
5min	12	288
1min	60	1440

Table 6.2: Calculation of number of bins in a day based on bin size

6.2.1 Experiment Setup

Consider the scenario composed by a community with five houses: house 3, house 4, house 5, house 12 and house 17, each one with a flexibility of 100% (completely flexible). The consumption profiles of houses 3, 4 and 5 are presented in Figures 6.1 to 6.3.

Summing all the consumption profiles of the five houses, the community consumption profile is presented in Figure 6.4, where the flexible and not-flexible load are separated as well as the baseline.

After that, the PV Power is forecasted for the same day as the consumption profiles and they are combined (demand and production) in order to generate the complete community profile (as shown in Figure 6.5). Finally, in order to calculate the netload, the production is subtracted from the demand, and the result is available in Figure 6.6.

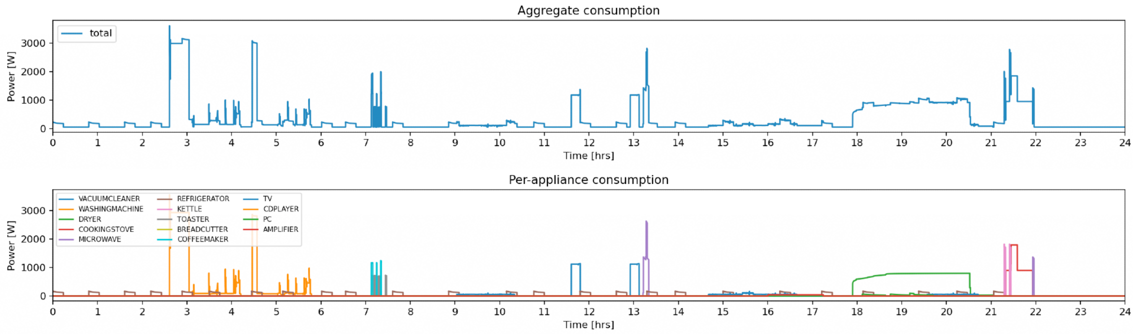

Figure 6.1: Consumption profile of house 3

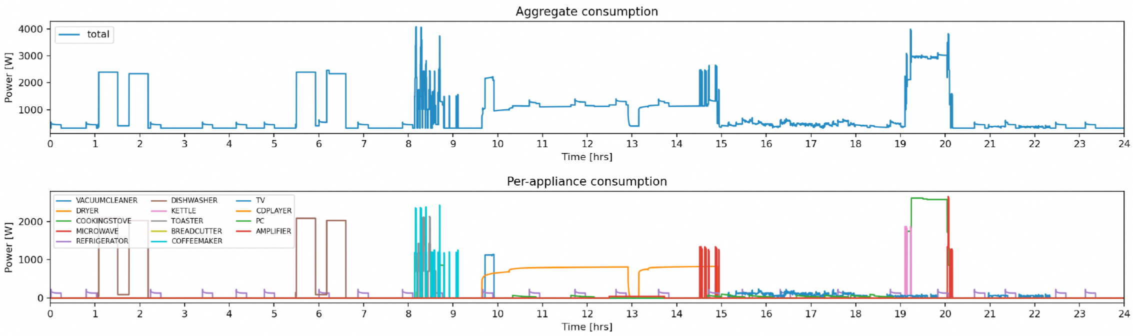

Figure 6.2: Consumption profile of house 4

For this experiment, three different cases were considered: a case with a bin size of 1 hour (which is equivalent to 1 bin per hour), another with a bin size of 30 minutes (which is 2 bins per hour) and, finally, a case using a bin size of 15 minutes (i.e, 4 bins per hour).

The main goal of this experiment is to evaluate the different results when increasing or decreasing the number of bins, and, above all, understand if the results are very similar and provide good results for the community management.

6.2.2 Results

Bins of 1 hour

Starting with the case that uses bins of 1 hour, as already explained, there are 24 bins in a day. As the number of bins decrease, the number of subitems also decrease, which leads to the decrease of possible combinations. In this sense, since there is a lower number of bins in comparison to bins of 30 minutes or bins of 15 minutes, the set of subitems is also lower.

Bin Number	Bin Capacities	Bin Maximum
6	137.04 Wh	651.94 W
7	1802.19 Wh	2755.15 W
8	3403.98 Wh	4066.45 W
9	5260.86 Wh	6629.8 W
10	8038.57 Wh	9336.41 W
11	8726.79 Wh	9443.22 W

Table 6.3: Portion of input sent to optimization

Taking this into account, Tables 6.3 and 6.4 show a portion of the input sent in the optimization. In Table 6.3 it is possible to see the capacity and the maximum peak of six bins while in Table 6.4 it is

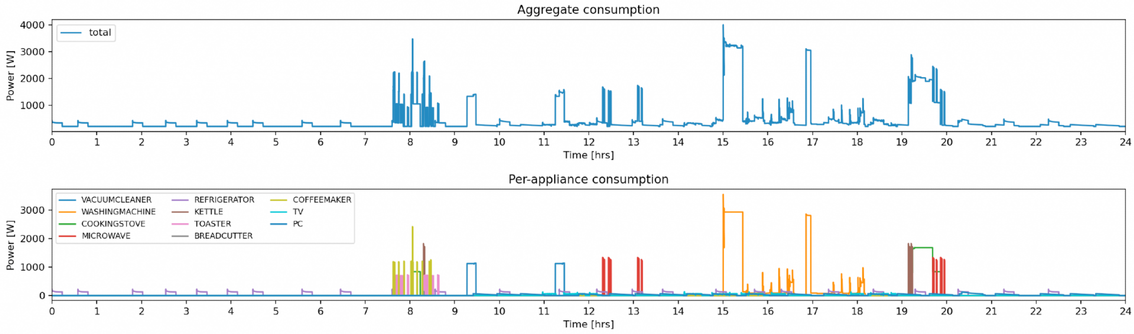

Figure 6.3: Consumption profile of house 5

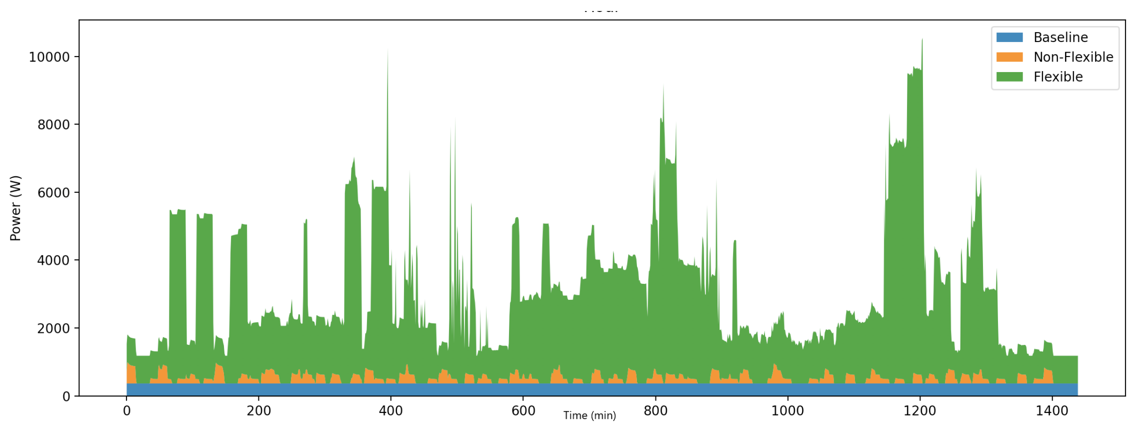

Figure 6.4: Consumption profile of the community

possible to see the subitems of three timeslots and its parameters: subitem value (which is the subitem energy consumption), subitem maximum (which is the energy consumption peak of the subitem), the flexibility (that corresponds to the number of bins where the appliance can be shifted) and the desired hour (which is the preffered hour of the user to use the timeslot subitem).

After solving this experiment, Figure 6.7 illustrates the optimal solution for the resources management when using bins of one hour; i.e., it shows the best way that the resources should be distributed in order to balance the demand and supply using this bin size.

It is divided in 4 graphs with different purposes. Graph A shows the demand desired by the community members without any action from the community manager and figure B shows only the non-flexible consumption (that consumption that can not be moved or optimized and has to be maintained - using non-flexible appliances).

On the other hand, graphs C and D show the demand after the intervention of the community manager (after optimizing the community load) for the first and second steps of the optimization respectively.

In the first graph, where there is no action from the community manager, in comparison to the graphs C and D, three main aspects can be observed:

- There are many more peaks
- A big quantity of production is not used (it is wasted)
- There are many more energy consumption that is not covered by the PV production (Renewable Energy)

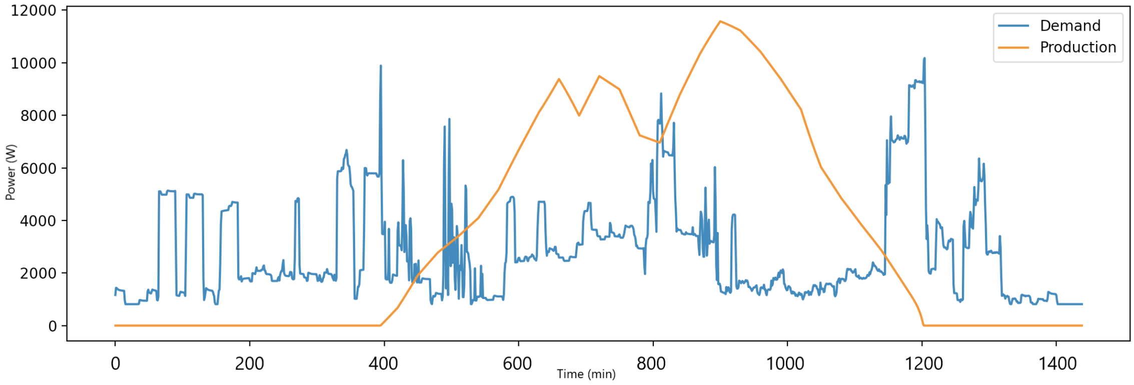

Figure 6.5: Demand and Production of Community

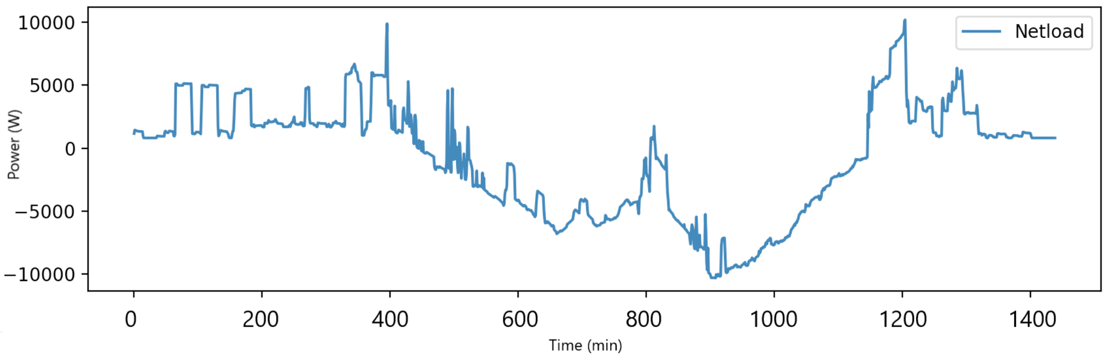

Figure 6.6: Netload of Community

Bins of 30 minutes

The same process was done using bins of 30 minutes (using 48 bins) and the Figure 6.8 visually shows the optimal solution for distributing the timeslots when having bins of this size.

In this case, a timeslot is divided in more subitems compared to the previous case because the bins are smaller. The timeslot number 4 that had 4 subitems in the first run (Table 6.4), doubled the number (i.e. eight).

Furthermore, it is important to understand that, even if using the same timeslots and the same consumption profiles, the optimization output is not the same when varying the bins size, and different results can be obtained.

Bins of 15 minutes

Finally, in order to conclude this experiment, bins of 15 minutes were used with the goal of better understanding what is the impact of increasing and decreasing the bin size. The timeslots distribution is presented in the Figure 6.9.

As it is possible to observe, the consumption profile after the first and second optimizations are a bit different compared to the previous two cases, but, visually it is not easy to compare and analyse the three approaches.

In order to analyse the results obtained in this experiment and better understand the approaches, some promising metrics have to be calculated for each case.

Timeslot Number	SubItem Number	SubItem Value	SubItem Maximum	Flexibility	Desired Bin/Hour
1	1	225.36 Wh	1138.05 W	8	12
2	1	87.97 Wh	1123 W	8	13
2	2	137.39 Wh	1137.77 W	8	14
3	1	1132.98 Wh	2935.09 W	10	3
3	2	248.93 Wh	2935 W	10	4
3	3	394.13 Wh	2839.19 W	10	5
3	4	118.67 Wh	600.51 W	10	6

Table 6.4: Sample of the input of the optimization

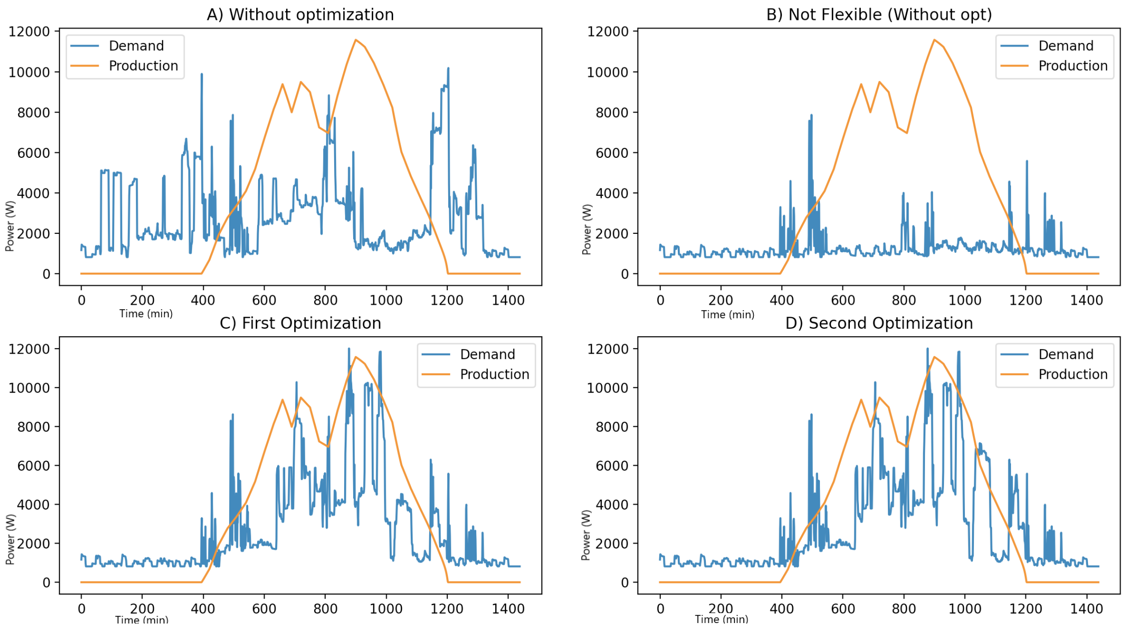

Figure 6.7: Bins of 1 hour

6.2.3 Analysis and Conclusion

Based on these consumption profiles, the results obtained using some metrics for the three cases are presented in Table 6.5, divided in two steps: after the first optimization (where only the timeslots which renewable energy is enough are distributed) and after second optimization (where not only the renewable energy can be used but also a PPC value which indicates the quantity of energy that can be acquired from grid for the remaining of the timeslots that have not been distributed in the first phase).

First of all, the quantity of energy from the grid is one of the most important metrics that can be used to evaluate the result of the optimization process for each case and analyse which provides the best results. Ideally the energy that comes from the grid should be the least possible, and the best result would be not needing to acquire any energy from grid and just use the energy from renewable sources to cover the consumption needs, although it is not possible in most of the situations.

Furthermore, energy used from the PVs and energy not used from the PVs are also two important metrics as they will allow to check if most of the production is being used and not wasted. In the best cases, the energy used from the PVs should be as higher as possible (and, of course, the unused energy should not be too high).

In addition, the number of peaks is also important because it allows to understand how many times (which is equivalent to the non-consecutive quantity of minutes) that renewable energy is not enough and energy from the grid have to be acquired in order to fulfill the needs. Once again, in the ideal case it should be as less as possible.

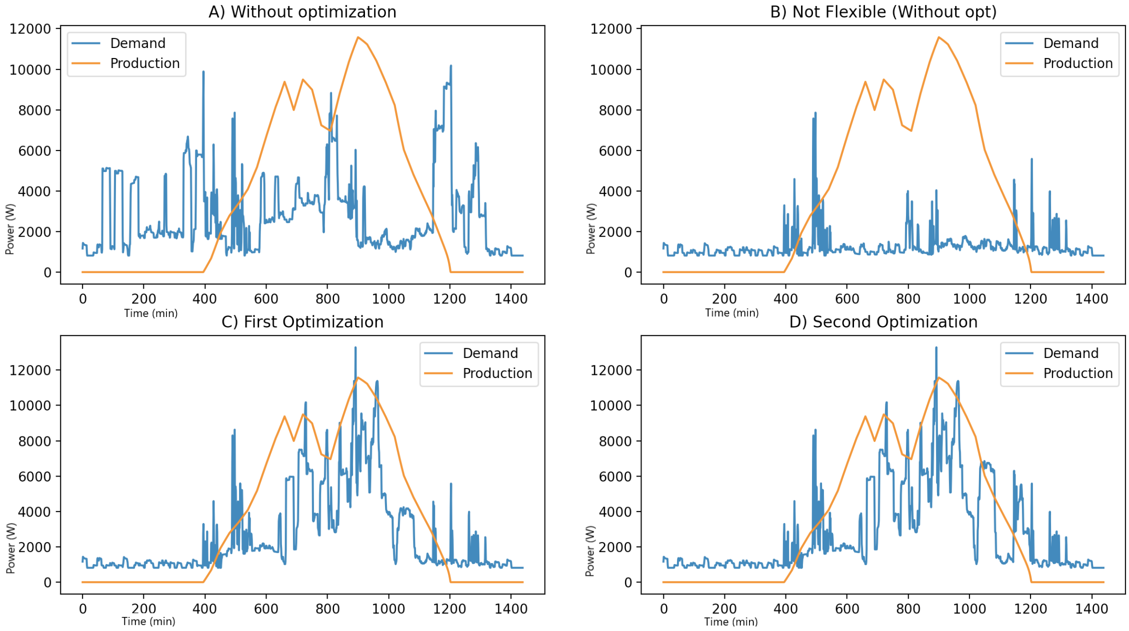

Figure 6.8: Bins of 30 minutes

Finally, the last two metrics that will be used to evaluate this experience are related to the number of timeslots. It is important to understand the quantity of timeslots that have been placed (and can be used by the community members) and the ones which members are not able to use. This optimization process tries to give as many timeslots as possible, however, in many cases it is not possible because the community manager (which is the module responsible for optimizing the resources) prioritize the community instead of the members satisfaction. It preferes to give less timeslots but have a better optimized community.

Saying this, in Table 6.5, it is possible to verify that before the optimization there were 35.11 kWh that came from non-renewable sources (grid). When optimizing using bins of one hour, in the second optimization (after placing all the timeslots), this value is reduced to 15.65 kWh (more than half). In bins of 30 minutes, the energy used from the grid is 15.54 kWh and it is 15 kWh in bins of 15 minutes. In this way, bins of 15 minutes allow to get less energy from the grid in comparison with the other bin sizes.

		Bins 1 h		**Bins 30 min**		**Bins 15 min**	
	Before Opt	**1st Opt**	**2nd Opt**	**1st Opt**	**2nd Opt**	**1st Opt**	**2nd Opt**
E_g (kWh)	35.11	14.27	15.65	13.17	15.54	13.25	15
E_{wpv} (kWh)	53.97	36.15	34.51	36.55	34.4	34.28	33.85
E_{pv} (kWh)	32.18	50	51.64	49.6	51.75	51.87	52.3
T_p	27	25	2 (27)	24	3 (27)	25	2 (27)
T_{np}	0	2	0	3	0	2	0

Table 6.5: Experiment Results

However, as previously referred, it is also necessary to understand which one takes the most advantage of the PV resources. In the three cases presented in the table, there is an evident increase of the PV usage compared to the values before the action of the community manager, but the one which uses more power from renewable energy is the third (uses 3137.93 kW).

Finally, in order to conclude the analysis of this experiment, the Table 6.5 also shows that before the optimization there was 3238.21 kW of PV that was not used while in the other three cases this value is more than 1000 kW lower.

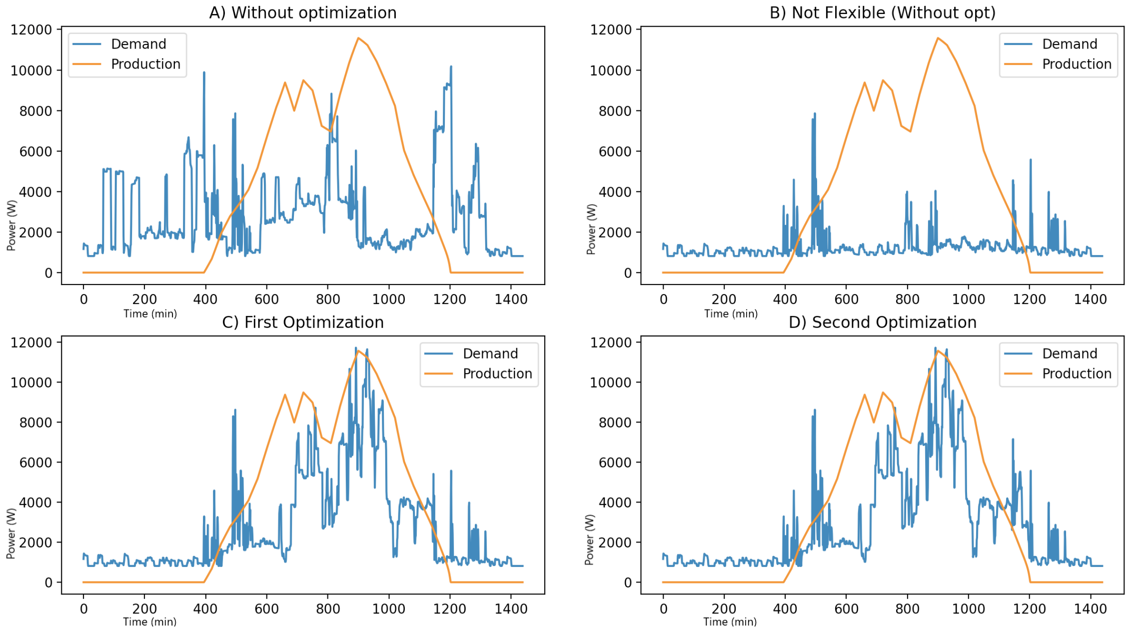

Figure 6.9: Bins of 15 minutes

Based on these results, two distinct metrics can be calculated: Self-Consumption and Self-Sufficiency. The Self-Consumption (SC) is the amount of electricity generated and consumed with respect to the total generation, while the Self-Sufficiency (SS) measures the consumption amount supplied by generation with respect to the total consumption (independence from the grid) [55]. They are calculated according to the following formulas.

$$SC = \frac{E_{pv}}{E_{pv} + E_{wpv}} \tag{6.1}$$

$$SS = \frac{E_{pv}}{E_{pv} + E_{g}} \tag{6.2}$$

The values of SC e SS for every case after the second step of the optimization are shown in Table 6.6.

With respect to these calculated metrics, not significant conclusions can be drawn from this experience. As expected, the results provided by the bins of 1 hour are almost the same as the bins of the other sizes, meaning that the load balancing strategy works using different bin sizes. Bins of 15 minutes provides better results in comparison to bins of 1 hour and bins of 30 minutes relatively to the power acquired from the grid. What concerns to the one which maximizes the power used from PV and minimizes the power wasted from PV, bins of 15 minutes also takes the advantage. However, these differences are not significant enough, meaning that it is not a pattern and can vary depending on the experiment. Based on this, the same experience was run for five other communities of the same size (five houses). The results are summarized on the Tables 6.7 and 6.8.

After interpreting the results presented in the Table 6.7, it is possible to conclude that the size of the bin is not a variable that creates a big impact in the optimization. While in the previous experiment, the bins of 15 minutes provided the best results, on the other hand, in the community A they provide the worst ones (uses more power from grid, uses less from the PV and wastes more PV power). Furthermore, there are no big differences when comparing the three approaches because the process is always the same. The unique difference is that, as the bin size decreases, the more realistic the solution becomes.

	Before Opt	Bins 1 h	Bins 30 min	Bins 15 min
SC	37.4%	59.9%	60%	60.7%
SS	47.8%	76.7%	76.9%	77.7%

Table 6.6: SC e SS calculation for every case

		Bins 1 h		**Bins 30 min**		**Bins 15 min**	
	Before Opt	**1st Opt**	**2nd Opt**	**1st Opt**	**2nd Opt**	**1st Opt**	**2nd Opt**
Community A							
E_g (kWh)	27.86	14.68	16.25	13.72	16.07	13.9	16.25
E_{wpv} (kWh)	51.13	39.52	39.52	39.36	39.34	39.54	39.52
E_{pv} (kWh)	26.09	37.71	37.71	37.86	37.88	37.68	37.7
T_p	24	20	4 (24)	19	5 (24)	19	5 (24)
T_{np}	0	4	0	5	0	5	0
Community B							
E_g (kWh)	34.08	16.1	20.92	16	20.82	16.21	21.1
E_{wpv} (kWh)	65.05	52.94	51.89	52.84	51.79	52	51.02
E_{pv} (kWh)	27.02	39.12	40.17	39.23	40.28	39.02	39.99
T_p	26	24	2 (26)	24	2 (26)	24	2 (26)
T_{np}	0	2	0	2	0	2	0
Community C							
E_g (kWh)	31.4	16.53	19.58	16.53	19.58	16.54	19.58
E_{wpv} (kWh)	55.9	44.98	44.09	44.98	44.1	44.99	44.1
E_{pv} (kWh)	33.19	44.12	45	44.12	45	44.11	45
T_p	22	20	2 (22)	20	2 (22)	20	2 (22)
T_{np}	0	2	0	2	0	2	0
Community D							
E_g (kWh)	30.07	13.11	19.06	13.22	19.16	13.1	19.04
E_{wpv} (kWh)	51.65	40.65	40.63	40.76	40.74	40.63	40.61
E_{pv} (kWh)	25.9	36.9	36.92	36.79	36.81	36.91	36.93
T_p	29	24	5 (29)	24	5 (29)	24	5 (29)
T_{np}	0	5	0	5	0	5	0
Community E							
E_g (kWh)	20.18	12.85	15.34	12.76	15.24	12.1	15.2
E_{wpv} (kWh)	48.55	43.71	43.71	43.61	43.61	43.76	43.57
E_{pv} (kWh)	25.77	30.61	30.61	30.71	30.71	30.55	30.75
PK_n	823	743	743	735	735	707	707
T_p	20	15	5 (20)	15	5 (20)	13	7 (20)
T_{np}	0	5	0	5	0	7	0

Table 6.7: Experiment Results of five different communities

6.3 Experiment 2: Variation in community size

6.3.1 Experiment Setup

The second experiment of this chapter is related to the variation of the community size. It is very important to understand how the community size affects the optimization and what is the behavior as the community increases.

Like the first experiment, the flexibility is 100% for every house and three different cases were considered: a community with five houses, other having ten houses and, finally, a big community with twenty houses.

It is also important to mention that, in this experience, in order to the get real results and provide real conclusions, the same day was considered (i.e., the same weather data).

	Before Opt	Bins 1 h	Bins 30 min	Bins 15 min
Community A				
SC	33.8%	48.8%	49.1%	48.8%
SS	48.4%	69.9%	70.2%	69.9%
Community B				
SC	29.3%	43.6%	43.7%	43.9%
SS	44.2%	65.8%	65.9%	65.5%
Community C				
SC	37.3%	50.5%	50.5%	50.5%
SS	51.4%	69.7%	69.7%	69.7%
Community D				
SC	33.4%	47.6%	47.5%	47.6%
SS	46.3%	66%	65.8%	66%
Community E				
SC	34.7%	41.2%	41.3%	41.4%
SS	56.1%	66.6%	66.8%	66.9%

Table 6.8: SC e SS calculation of five different communities

6.3.2 Results

Community of 5 houses

For the first test using a community with five different houses, the chosen houses were: house 2, house 6, house 10, house 14 and house 18.

Before the optimization, the SS was 49.1% (it is, 49.1% of the used energy comes from renewable energy - less than an half) and the SC was 27.3% (from the total PV production, only 27.3% of it was really used - almost one quarter of the production).

After the 1st step of the optimization, these values increased as expected. SS increased to 69.2% (20% more of the energy used is from PV production) and SC increased to 35.4%.

Finally, in the second step, placing the timeslots that were not placed in the first step (all timeslots placed), the SS was reduced to 66% and the SC was the same (because the remaining timeslots were placed using energy from grid).

At the first stage there can be a surprise since there is 40.89 kW of PV production that is not used in the first step, and it continues not being used in the second step. One question that may arise is "if there is enough production that is not being used, why can not it be used to place the remaining of the timeslots, avoiding the need to acquire it from the grid?". The answer is very simple: as previously said, one of the main constraints of this process is related to the flexibility, which is a multiplication of two different flexibilities: appliance flexibility and house flexibility. In this case, the two placed timeslots in the second step refers to the Cooking Stove, and its flexibility is just 1 hour. Taking in consideration that the houses flexibility are 1 (between 0-1, equivalent to 100%), this appliance only can be shifted one hour (before or after) and there are no production in that time interval.

These results are summarized in Tables 6.9 and 6.10.

Community of 10 houses

Considering a community with ten houses: houses 2, 4, 6, 8, 10, 12, 14, 16, 18 and 20, the process that was done in the previous section has been repeated, but, instead of having a 5-houses community, using a 10-houses one.

It will allow to understand if there are better results when having bigger communities, or, in the other hand, the smaller communities provides the best results. Also, it will allow to understand how scalable the simulator is.

Using a community of five houses, before the optimization, the SS was 46.1% while the SC was 29.8%. After the first step, SS increased to 73.1% and SC to 44.1%, and after the second step, the SC decreased to 70.7% and SC increased to 45.8%.

All these results are also available in tables Tables 6.9 and 6.10.

	5 houses		**10 houses**		**20 houses**	
	1st Opt	**2nd Opt**	**1st Opt**	**2nd Opt**	**1st Opt**	**2nd Opt**
E_g (kWh)	9.96	11.55	29.7	34.61	60.48	68.26
E_{wpv} (kWh)	40.89	40.89	102.18	99.04	223.12	223.12
E_{pv} (kWh)	22.37	22.37	80.58	83.71	149.42	149.42
T_p	14	2 (16)	48	4 (52)	88	8 (96)
T_{np}	2	0	4	0	8	0

Table 6.9: Experiment Results using different community sizes

	5 houses	10 houses	20 houses
SC	35.4%	45.8%	40.1%
SS	66%	70.7%	68.6%

Table 6.10: SC e SS calculation for different community sizes

Community of 20 houses

Finally, the last test of this experiment was using a community of twenty houses (from house 1 to house 20).

Before starting the optimization process, the SS obtained was 48.4% and the SC was 28.3%. Then, when starting the process, the SS was increased to 71.2% (more production came from renewable sources instead of non-renewable ones) and the SS increased to 40.1% (less PV production was wasted).

In order to conclude, in the last step, the SS decreased to 68.6% because the remaining of the timeslots that were not placed in the first step were all placed using the grid (for the reasons aforementioned). The SC was the same taking in account that the PV used was the same.

6.3.3 Analysis and Conclusion

From the previous section, can be concluded that the community of 10 houses provided the best results (i.e. takes more advantage of the renewable resources). However, the community of 20 houses has better results in comparison with the 5-houses one.

It is not possible to know beforehand which community size provides the best results because the community size is not supposed to affect the experiment directly. What really affects the experiment and can provide different results are the PPC of each house, the appliances each house uses and the flexibility of the houses. It can be completely different having a community with 10 houses with a PPC of 3.45 or having other 10 houses with other PPC values.

In this case, the 10-houses community has the best results. However, changing one of the houses of this community (for instance, increasing the number of people and used appliances) can be enough to reverse the results.

The main goal of this experiment was not to compare and analyse directly the differences between the three cases, but essentially to check if the optimization approach is scalable (i.e. if SC and SS values have increased in comparison with the results obtained before running the optimization, for every case). In this regard, based on the results of the previous section, it is possible to conclude that the results are positive when considering communities with a low number of houses, but also when considering bigger ones.

6.4 Experiment: Variation in flexibility

6.4.1 Experiment Setup

The last experiment of this chapter is related to the variation of the flexibility given as input.

As previously said, there is a flexibility constraint which considers the appliance and house flexibilities. This constraint is very important because it affects the way that the timeslots are distributed.

Imagine that there is a timeslot with a flexibility of 2 hours. Then, this timeslot can only be shifted to a time interval between the two previous hours and the two next hours according to the requested time. It is completely different when considering another timeslot with a flexibility of 10 hours. As the name indicates, there is more flexibility to choose the adequate time period where the appliance will be used.

As the timeslot flexibility increases, many more options to put the timeslot exists, leading to the increase of the possible combinations, which can also increase the runtime.

For this experiment, using a community of 10 houses (houses 1, 3, 5, 7, 9, 11, 13, 15, 17 and 19) and a bin size of 30 minutes, four different scenarios were considered. Without changing the appliance flexibilities (which are considered in Table 5.6), only the houses flexibility were changed in order to take some conclusions. In this way, the scenarios are: using a flexibility of 25% for all houses of the community, using 50%, using 75%, and, finally, using 100% (completely flexible, as similar to the previous experiments).

6.4.2 Results

Flexibility of 25%

In this first scenario, considering a flexibility of 25% for every house (a quarter of the timeslot flexibility), the flexibilities are: (1) Dishwasher - 3 hours, (2) Vaccum Cleaner - 2 hours, (3) Washing Machine - 2 hours, (4) Dryer Machine - 1 hour, (5) Iron - 1 hour, and, (6) Cooking Stove - 25 minutes (0 hours).

Since we are using bins of 30 minutes, it corresponds to: (1) Dishwasher - 6 bins, (2) Vaccum Cleaner - 4 bins, (3) Washing Machine - 4 bins, (4) Dryer Machine - 2 bins, (5) Iron - 2 bins, and, (6) Cooking Stove - 0 bins.

In this sense, after running the experiment, for this scenario, the results were the following: The energy used from the grid (non-renewable resources) was 53.97 kWh after the optimization, in comparison with the 63.55 kWh obtained before it. Moreover, the energy not used from the PV decreased, although it was not a significant decrease (135 kWh to 125.21 kWh). Finally, what concerns to the energy used from the PV (renewable sources), it increased from 54.78 kWh to 64.35 kWh (approximately 10 kWh).

Figure 6.10 shows the graphs for the different steps of the process and can be used to better understand and compare the results.

Flexibility of 50%

In the second scenario, using a house flexibility of 50% (half of the timeslot machine), the flexibilities are: (1) - Dishwasher - 6 hours, (2) Vaccum Cleaner - 4 hours, (3) Washing Machine - 4 hours, (4) Dryer Machine - 2 hours, (5) Iron - 2 hours, and, (6) Cooking stove - 30 minutes (0 hours).

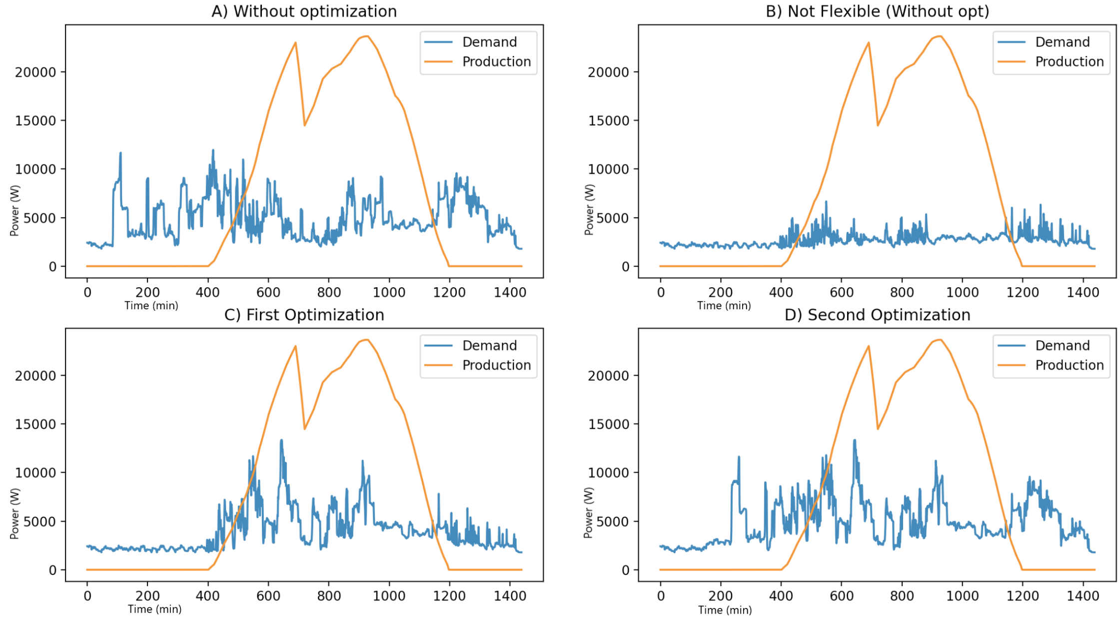

Figure 6.10: Results obtained using a flexibility of 25%

Taking in consideration the bin size of 30 minutes, the equivalent in bins are: (1) Dishwasher - 12 bins, (2) Vaccum Cleaner - 8 bins, (3) Washing Machine - 8 bins, (4) Dryer Machine - 4 bins, (5) Iron - 4 bins, and, (6) Cooking Stove - 1 bin.

In this case, the energy used from the grid in the second optimization was 46.78 kWh (less compared to the previous scenario), the energy not used from the PV was 120.24 kWh (less compared to the previous scenario) and the energy used from the PV was 71.54 kWh (more in comparison with the flexibility of 25%). This is presented graphically in Figure 6.11.

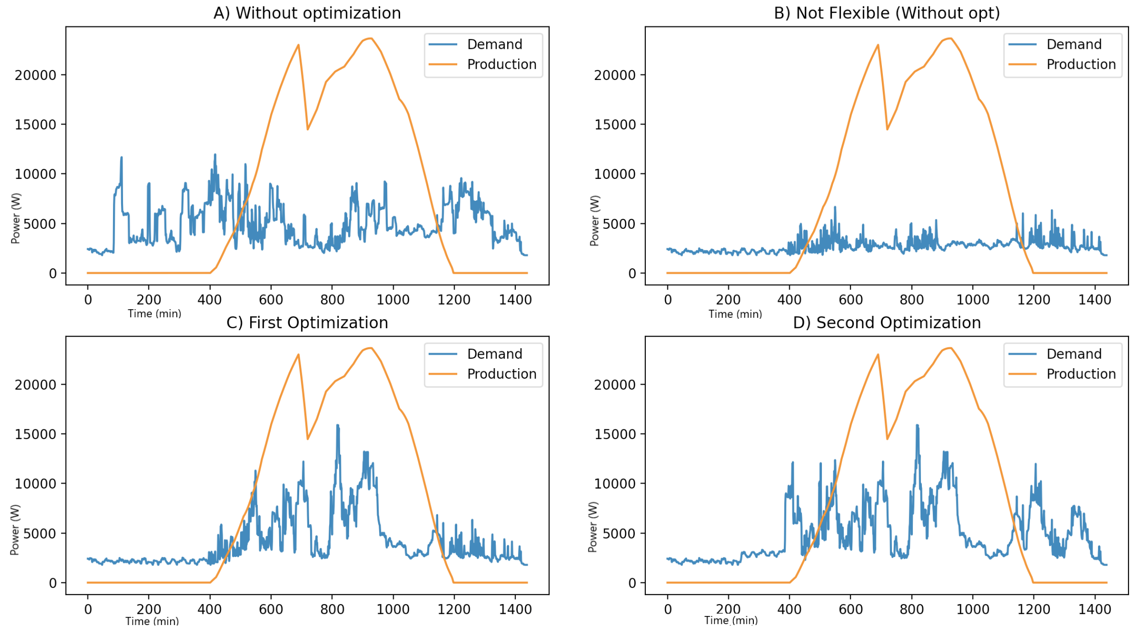

Figure 6.11: Results obtained using a flexibility of 50%

Flexibility of 75%

The third scenario consists of assigning a flexibility of 75% for each community house and running the experiment for the same ten houses using a bin size of 30 minutes.

In this way, the flexibilities are: (1) - Dishwasher - 9 hours (18 bins), (2) Vaccum Cleaner - 6 hours (12 bins), (3) Washing Machine - 6 hours (12 bins), (4) Dryer Machine - 3 hours (6 bins), (5) Iron - 3 hours (6 bins), and, (6) Cooking stove - 45 minutes (2 bins).

In this case, since there are more possible combinations due to the increase of the flexibility, the results are expected to be better, or, at least, equal. It is not possible to have worst results taking in consideration that it includes the combinations of the previous cases as well as new combinations.

In Figure 6.12 and Table 6.11, it is possible to see that, in the second step of this test the power obtained from the grid was 42.07 kWh, the energy wasted from the PV was 113.53 kWh and the energy used from the PV was 76.26 kWh.

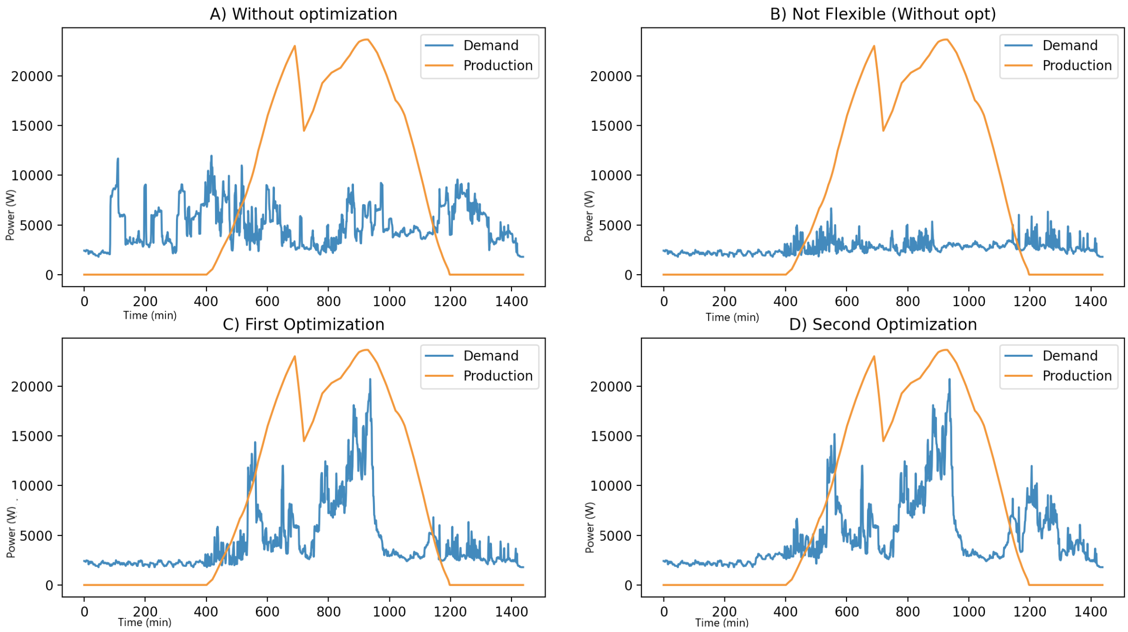

Figure 6.12: Results obtained using a flexibility of 75%

Flexibility of 100%

At the end, the last scenario consists of using a flexibility of 100% (completely flexible). This is the case which has more possible combinations because the flexibility is the highest possible.

The flexibilities used are the same as the ones referred in Table 5.6 because the house flexibility (which is 1) multiplies by the appliance flexibility. They are: (1) - Dishwasher - 12 hours (24 bins), (2) Vaccum Cleaner - 8 hours (16 bins), (3) Washing Machine - 8 hours (16 bins), (4) Dryer Machine - 4 hours (8 bins), (5) Iron - 4 hours (8 bins), and, (6) Cooking stove - 1 hour (2 bins).

The result is displayed in Figure 6.13. It allowed to decrease the energy from the grid (to 38.24 kWh) as well as the energy wasted from the PV (to 109.7 kWh), while, on the other hand, allowed to increase the energy from the PV (to 80.08 kWh).

The analysis of these results will be done in the next section.

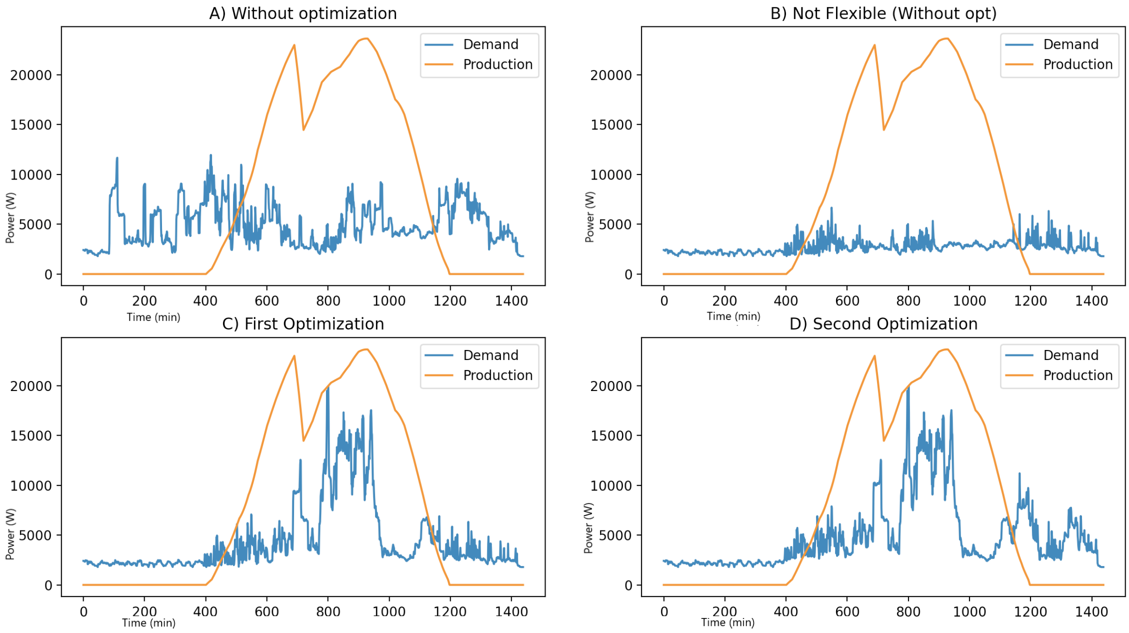

Figure 6.13: Results obtained using a flexibility of 100%

6.5 Analysis and Conclusion

Before analysing the four different scenarios and taking some conclusions, it is important to calculate two other metrics: SS and SC. Both metrics will allow to understand the percentage of the PV which is being used in comparison with all the power used, and the percentage of the PV which is being used in comparison with all the production. The values are presented in Table 6.12.

First of all, after analysing the results of the Table 6.11, it is interesting to see that, as the flexibility increases, more timeslots are placed in the first step of the optimization. For instance, using a flexibility of 25%, only 37 were placed using just renewable energy, while, when using a flexibility of 100%, 9 more timeslots have been placed.

	Before	**Flexibility 25%**		**Flexibility 50%**		**Flexibility 75%**		**Flexibility 100%**	
		1st	**2nd**	**1st**	**2nd**	**1st**	**2nd**	**1st**	**2nd**
E_g (kWh)	63.55	30.7	53.97	28.91	46.78	29.4	42.07	28.95	38.24
E_{wpv} (kWh)	135	126.27	125.43	120.24	118.24	115.05	113.53	112.78	109.7
E_{pv} (kWh)	54.78	63.51	64.35	69.55	71.54	74.74	76.26	77	80.08
T_p	52	37	15 (52)	42	10 (52)	45	7 (52)	46	6 (52)
T_{np}	0	15	0	10	0	7	0	6	0

Table 6.11: Experiment Results using different houses flexibilities

	Before	Flexibility 25%	Flexibility 50%	Flexibility 75%	Flexibility 100%
SC	28.9%	33.9%	37.7%	40.2%	42.2%
SS	46.3%	54.4%	60.5%	64.4%	67.7%

Table 6.12: SC e SS calculation for different houses flexibilities

Furthermore, although all cases provide significant improvements in comparison to before the optimization, it is possible to understand that, the case which gets more power from the grid is the first one (flexibility of 25%) while the case which acquires less power from the grid is the last one (flexibility of 100%). The same happens to the power not used from the PV. What concerns to the amount of power from the PV, the one which takes the most advantage is, once again, the last one.

Table 6.12 summarizes these results. Before the optimization only 28.9% of the total PV was used and only 46.3% of the total power used came from the renewable sources. As the flexibility increases, these values also increase. At the end, using a flexibility of 100%, 42.2% of the total production was used and almost 70% of the total consumption was acquired from renewable resources (more 20% compared to before the action of the community manager).

It is important to highlight that using a flexibility of 100% allows to use approximately 25000 Wh more of production (avoiding the waste of approximately 25k Wh) and also allows to use approximately 25 kWh less of power from non-renewable sources due to the better load balancing.

Timeslots Placement

In addition, it would be also interesting to explore two random timeslots and understand what is the effect in their placements when changing the flexibility of the appliances. In order to do that, the bins positions of the timeslots 2 and 38 (each one composed by 7 subitems) were analysed, and the results are the following:

- Flexibility of 25%: Timeslot 2 has been placed from 4am to 7am in the 2nd optimization with the following capacities (Wh): 52785.0, 52785.0, 52785.0, 52785.0, 52785.0, 52956.9, 53962.11, and the Timeslot 38 has been placed from 6am to 9am in the 2nd optimization with the capacities (Wh) 52785.0, 52785.0, 52956.9, 53962.11, 54750.58, 57237.8.

- Flexibility of 50%: Timeslot 2 has been placed from 6.30am to 9.30m in the 2nd optimization with the capacities (Wh) 52956.9, 54060.91, 55726.3, 57733.31, 59293.93, 61777.74, 66187.62 and the Timeslot 38 has been placed from 8.30am to 11.30am in the 1st optimization with the capacities (Wh) 7659.52, 10493.85, 14065.1, 17246.55, 19834.84, 22018.75, 19191.23.

- Flexibility of 75%: Timeslot 2 has been placed from 9am to 12pm in the 1st optimization with the following capacities (Wh): 10493.85, 14065.1, 17246.55, 19834.84, 22018.75, 19191.23, 15439.81, and Timeslot 38 has been placed from bins 11am to 2pm in the 1st optimization with the capacities (Wh) 22018.75, 19191.23, 15439.81, 17820.58, 19761.48, 20545.55 and 21417.27.

- Flexibility of 100%: Timeslot 2 has been placed from 11.30am to 2.30pm in the 1st optimization with the following capacities (Wh): 19191.23, 15439.81, 17820.58, 19761.48, 20545.55, 21417.27, 22743.05, and the Timeslot 38 has been placed from 1.30am to 4.30am in the 1st optimization with the capacities (Wh) 20545.55, 21417.27, 22743.05, 23565.94, 23014.37, 21246.33 and 18857.78.

By analysing the timeslot hours in the previous 4 cases, it is possible to verify that the timeslot 2 is placed in the 2nd step of the optimization when using flexibilities of 25% and 50% because there is no enough renewable energy to cover it needs with a flow flexibility (has to acquire from the grid), but, when increasing to 50% or 100%, it is placed in the first step and it can be covered only using power from renewable sources. However, since the house wanted to use the timeslot from 1.30am to 4.30am, it is also possible to see that, as expected, as the flexibility increases, even if the results are better, the timeslot is further away (the difference between the desired hour and the given hour is increasing).

Finally, another interesting approach would be understanding how the timeslots' placement are changed according to flexibility changes. In this sense, three random timeslots (timeslot 2 - which corresponds to the washing machine of the house 1, timeslot 25 - which is the dishwasher of house 6, and timeslot 33 - that is the washing machine of house 7) were selected and its positions throughout the day were analysed. Figure 6.14 shows the position of the timeslots before the action of the community manager, which corresponds to the desired positions (what was expected by the community houses). on the other hand, Figure 6.15 shows the positions for the same bins after the optimization process.

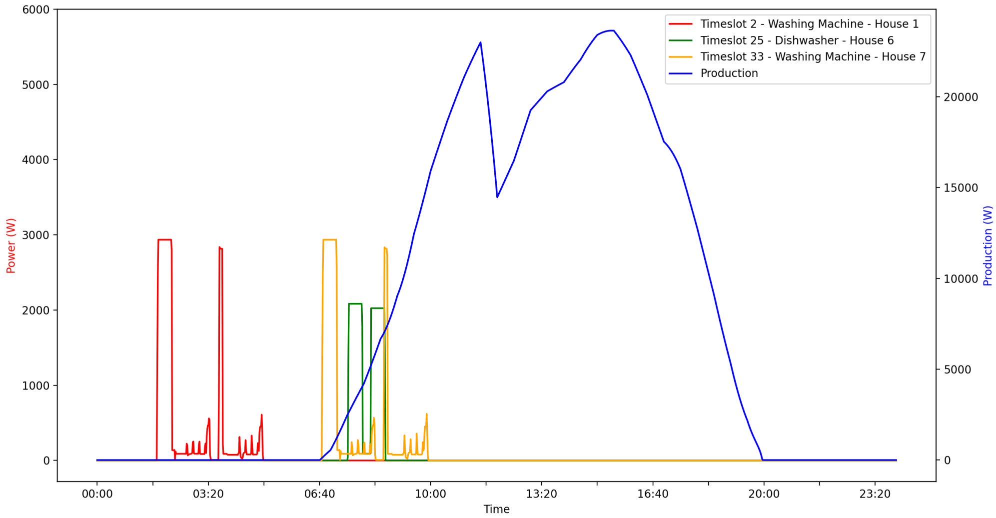

Figure 6.14: Three timeslots positions desired by the members without running the simulation

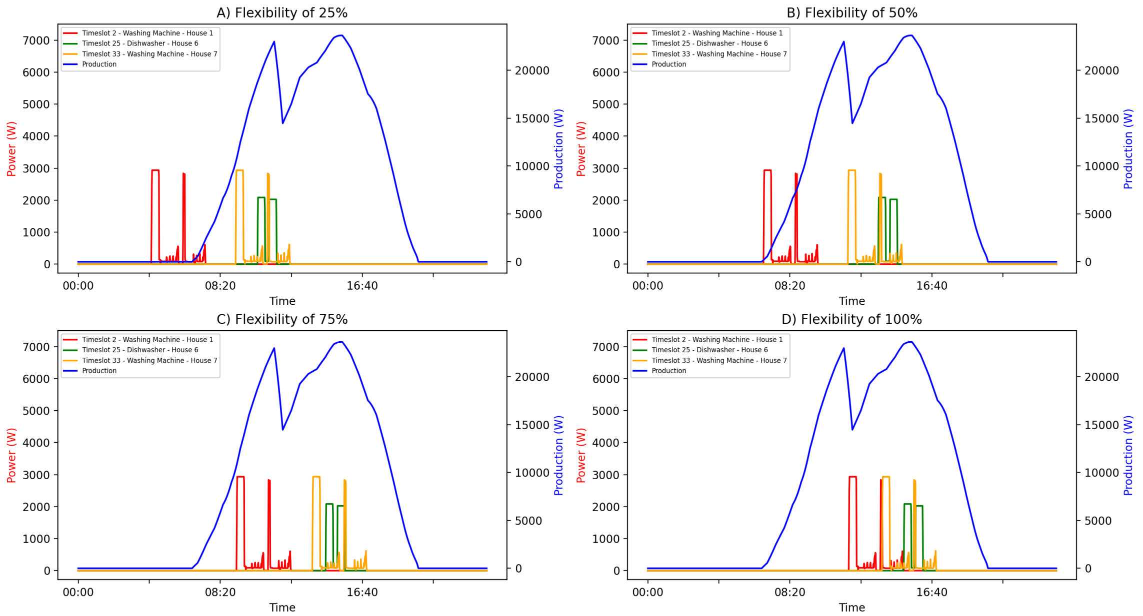

Figure 6.15: Three timeslots positions using different flexibilities (25%, 50%, 75%, 100%)

In the subplot A, using a flexibility of 25%, it is possible to see that the timeslot number 2 is running when there is no production (which means that power from the grid is being acquired). When comparing the other two timeslots in the different subplots, it is also possible to know that they are not placed in the best positions even if only production is being used.

As the flexibility increases, all the timeslots shift to the right, until the suplot D which has the best solution that only uses renewable energy. However, it is also important to mention that, although better results are being obtained when shifting the timeslots to the right (in this specific case), the members satisfaction is decreasing (taking in consideration that the distance between the placed position and the desired position are increasing).

Chapter 7

Conclusion

In this section, a summary will be provided based on the objectives of this thesis and the results obtained. In this way, the potential of the simulator as well as the load balancing strategy will be commented according to the optimization results.

Furthermore, the thesis implications will be referred, namely, the potential of this thesis in the research academy, and how can it be useful for researchers or other people.

At the end, the limitations as well as the future work will be described. In few words, it will be mentioned what are the limitations of the simulator and how can they be improved in the future in order to build a better simulator.

7.1 Summary

Balancing the demand and supply is, undoubtedly, an important aspect to take in consideration in the Energy Communities, especially when the goal is to take the most advantage of the renewable resources. Many researchers are focusing their work in finding possible solutions to help people avoid, as much as possible, the acquisition of energy from non-renewable sources. In this way, in order to achieve these goals, multiple works have been done in the scheduling of the home appliances using DR approaches (such as dynamic pricing and tariffs).

In this thesis, a different approach was suggested using the concept of timeslots (explicit approach) instead of different prices depending on the period of the day (implicit approach). In this sense, the main contributions of this thesis, as previously explained, are the development of an EC simulator which aims to generate different communities, test multiple scenarios, do different experiments and evaluate some strategies, and the development of a load balancing scheme which optimizes the community resources and distributes the timeslots based on the members' needs.

Although initially there were some difficulties related to the definition of the constraints and the objective functions using linear equations in order to adapt the Multi Knapsack to be used in our approach, and also related to the update of the consumption profiles (converting bins to time periods), it is possible to conclude that both goals were accomplished. The simulator is fully implemented because it creates consumption profiles for every house, calculates the community demand, community PV production and community netload, allows to test different schemes and evaluates the results based on some defined metrics.

What concerns to the optimization part, it is also possible to understand, based on the results obtained in the experiments, that the load balancing scheme proposed is working as expected and the Multi Knapsack can be effectively used to optimize the community. In every experiment, when analyzing the results before the action of the CM and after its action, it is possible to conclude that the CM has an important role because the results were much more positive after the optimization (i.e. the PV use increased and the PV waste decreased as well as the energy used from the grid).

In the first experiment, which consists in the variation of the bin sizes, the SS and SC increased a lot (Table 6.6). Using bins of 15 minutes, the optimization allowed to increase the SC in 23.3% and the SS in 30.1%, meaning that, more 23.3% of the PV was used (avoiding a waste of 23.3%) and an additional 30.1% of the energy came from renewable sources (instead of non-renewable ones).

In the third experiment, which the goal was to vary the community flexibility and analyze the results, also the values of SS and SC have increased (Table 6.12). For instance, using a flexibility of 100%, the SC was 42.2% instead of 28.9% (increased 13.3%) and the SS was 67.7% instead of 46.3% (increased 21.4%).

7.2 Implications

With PROCSIM, it was possible to reduce a gap existent in the literature, which is the lack of datasets for ECs. As aforementioned, a large quantity of data is necessary to evaluate different perspectives and steps of an algorithm. The use of simulators instead of real datasets allow to simulate different scenarios and obtain different results, otherwise the results will always be the same for the same dataset. Also, using this simulator, people can test and evaluate their strategies and algorithms in a personalized community by choosing the appliances, the house members, the schedules and activities of each member of the house for every appliance, the flexibility and the contracted power.

In general, this simulator can be helpful for every researcher in this field which would like to create ECs datasets for different purposes. This can be used, for example, to analyze and evaluate the impact of introducing electric vehicles in a community.

Moreover, in this thesis, a different strategy is given. In this strategy, instead of creating mechanisms to encourage people to change their consumption patterns based on financial incentives, timeslots are distributed in a way that improves the use of community resources, taking in consideration two factors: the reduction of the peak demand (which is related to maximization of the use of renewable resources) and the fair distribution of the timeslots for all community members. This scheme can be used to improve the management of the available resources.

7.3 Limitations and Future Work

Although both objectives have been reached, there are some limitations and improvements that can be done in the future. For instance, once there is no wind production implemented in the PROCSIM, there are no renewable energy at night, and all the appliances used at night only can use energy from the grid. A future step would be implementing wind generation for use in conjunction with solar production, and implementing batteries or/and electric vehicles, in a way that the energy which is not used in some periods can be stored and used later. In this regard, instead of needing to use the energy in the exact moment, it gives us flexibility to store the excess of production and use it when there is lower production.

Future developments of this platform can also include updates to the different community simulation modules in order to generate ECs that are as realistic as possible, and also a graphical interface should be created to simplify the way that the houses of the community can be configured. In the current version, these configurations are done in a json file, which makes it hard to create communities with many houses.

What concerns to the optimization, it is important to mention that one thing that could be improved is adding one more step to the process which gives the remaining of the timeslots that have not been placed in the first two steps. Currently, when a timeslot is not placed in the first or second steps due to the defined constraints, the member can not use the appliance. This third step would be used to allow the members to use the appliances when there is no other solution.

Furthermore, better results can be obtained in the optimization by changing the objective function, changing some of the constraints, or even restructuring the way that the problem is formulated. For example, other optimization techniques, like priorities, could be implemented, where not all timeslots have the same priority and the ones with higher priorities have to be placed first. As it is visible, although the results were very positive, there is a margin for improvements to increase the SS and SC values.

Finally, the last improvement could be integrating this simulator with an agent-based platform, like a market, having as the main goal the trade of the timeslots distributed by the CM. Imagine that a community member receives a timeslot in an unwanted period, then he could have the opportunity to trade it with other members which have a timeslot he wants.

Bibliography

[1] O. Energy, “Peak Demand Reduction with Energy Storage,” 2016.

[2] A. Harell, R. Jones, S. Makonin, and I. V. Bajic, “PowerGAN: Synthesizing Appliance Power Signatures Using Generative Adversarial Networks,” *arXiv:2007.13645 [cs, eess]*, July 2020. arXiv: 2007.13645.

[3] N. Pflugradt, “LoadProfileGenerator,” 2015.

[4] N. Roy, N. Pathak, and A. Misra, “Fine-grained appliance usage and energy monitoring through mobile and power-line sensing,” *Pervasive and Mobile Computing*, vol. 30, pp. 132–150, Aug. 2016.

[5] H. Cambazard, D. Mehta, B. O’Sullivan, and H. Simonis, “Bin Packing with Linear Usage Costs - An Application to Energy Management in Data Centres,” vol. 8124, Sept. 2013.

[6] U. F. Akpan and G. E. Akpan, “The Contribution of Energy Consumption to Climate Change: A Feasible Policy Direction,” *International Journal of Energy Economics and Policy*, vol. 2, pp. 21–33, Dec. 2011. Number: 1.

[7] F. Dews, “Charts of the Week: Tackling climate change,” Dec. 2019.

[8] S. Perry, J. Klemeš, and I. Bulatov, “Integrating waste and renewable energy to reduce the carbon footprint of locally integrated energy sectors,” *Energy*, vol. 33, pp. 1489–1497, Oct. 2008.

[9] O. US EPA, “Learn about Energy and its Impact on the Environment,” Aug. 2015.

[10] Eurostat, “Electricity production, consumption and market overview - Statistics Explained,” 2020.

[11] IEEFA, “Renewable energy provided 51% of Portugal’s electricity needs in 2019,” Feb. 2020.

[12] N. Santos, “Portugal is Leading the Way in Renewable Energy – Here’s Why,” 2018.

[13] O. US EPA, “Reduce the Environmental Impact of Your Energy Use,” Aug. 2015.

[14] O. Energy, “What is energy efficiency? | OVO Energy,” 2009.

[15] G. Powells, H. Bulkeley, S. Bell, and E. Judson, “Peak electricity demand and the flexibility of everyday life,” *Geoforum*, vol. 55, pp. 43–52, Aug. 2014.

[16] H. Lund, “Renewable energy strategies for sustainable development,” *Energy*, vol. 32, pp. 912–919, June 2007.

[17] S. R. Bull, “Renewable energy today and tomorrow,” *Proceedings of the IEEE*, vol. 89, pp. 1216–1226, Aug. 2001. Conference Name: Proceedings of the IEEE.

[18] N. Duić and M. da Graça Carvalho, “Increasing renewable energy sources in island energy supply: case study Porto Santo,” *Renewable and Sustainable Energy Reviews*, vol. 8, pp. 383–399, Aug. 2004.

[19] E. M. Gui and I. MacGill, “Typology of future clean energy communities: An exploratory structure, opportunities, and challenges,” *Energy Research & Social Science*, vol. 35, pp. 94–107, Jan. 2018.

[20] G. Dóci, E. Vasileiadou, and A. C. Petersen, “Exploring the transition potential of renewable energy communities,” *Futures*, vol. 66, pp. 85–95, Feb. 2015.

[21] N. A. Brooks, S. T. Powers, and J. M. Borg, “A mechanism to promote social behaviour in household load balancing,” *arXiv:2006.14526 [cs]*, June 2020. arXiv: 2006.14526.

[22] P. Chavali, P. Yang, and A. Nehorai, “A Distributed Algorithm of Appliance Scheduling for Home Energy Management System,” *IEEE Transactions on Smart Grid*, vol. 5, pp. 282–290, Jan. 2014. Conference Name: IEEE Transactions on Smart Grid.

[23] S. Aslam, R. Bukhsh, A. Khalid, N. Javaid, I. Ullah, I. Fatima, and Q. Hasan, “An Efficient Home Energy Management Scheme Using Cuckoo Search,” Aug. 2017.

[24] O. Sianaki and M. Masoum, “A fuzzy TOPSIS approach for home energy management in smart grid with considering householders’ preferences,” pp. 1–6, Feb. 2013.

[25] A. Polak, L. Rohwedder, and K. Węgrzycki, “Knapsack and Subset Sum with Small Items,” May 2021. arXiv:2105.04035 [cs].

[26] O. A. Sianaki, O. Hussain, and A. R. Tabesh, “A Knapsack problem approach for achieving efficient energy consumption in smart grid for endusers’ life style,” in *2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply*, pp. 159–164, Sept. 2010.

[27] S. U. Agamah and L. Ekonomou, “A Heuristic Combinatorial Optimization Algorithm for Load-Leveling and Peak Demand Reduction using Energy Storage Systems,” *Electric Power Components and Systems*, vol. 45, pp. 2093–2103, Nov. 2017. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/15325008.2017.1406017.

[28] K. Berg and M. Löschenbrand, “A data set of a Norwegian energy community,” *Data in Brief*, vol. 40, p. 107683, Feb. 2022.

[29] C. Goncalves, R. Barreto, P. Faria, L. Gomes, and Z. Vale, “Energy Community Consumption and Generation Dataset with Appliance Allocation,” *IFAC-PapersOnLine*, vol. 55, pp. 285–290, Jan. 2022.

[30] I. Europe, “2018-08-30_policy_brief_renewable_energy_communities_pb_to4_final.pdf,” 2018.

[31] Andreas Uihlein and Aura Caramizaru, “Energy communities: an overview of energy and social innovation.,” Tech. Rep. EUR 30083 EN, Publications Office of the European Union, Luxembourg, Feb. 2020. doi: 10.2760/180576.

[32] M. Iqbal, M. I. A. Sajjad, S. Amin, Haroon, R. Liaqat, Khan, M. Waseem, and Shah, “Optimal Scheduling of Residential Home Appliances by Considering Energy Storage and Stochastically Modelled Photovoltaics in a Grid Exchange Environment Using Hybrid Grey Wolf Genetic Algorithm Optimizer,” *Applied Sciences*, vol. 9, p. 5226, Dec. 2019.

[33] N. Lior, “Sustainable energy development: The present (2011) situation and possible paths to the future,” *Energy*, vol. 43, pp. 174–191, July 2012.

[34] P. Power, “What Uses The Most Electricity In Your Home? | Payless Power,” 2018.

[35] D. Energy, “What Uses the Most Electricity in My Home? – Biggest Electricity Hogs | Direct Energy,” 2020.

[36] A. Lucas, L. Jansen, N. Andreadou, E. Kotsakis, and M. Masera, “Load Flexibility Forecast for DR Using Non-Intrusive Load Monitoring in the Residential Sector,” *Energies*, vol. 12, p. 2725, Jan. 2019. Number: 14 Publisher: Multidisciplinary Digital Publishing Institute.

[37] M. Degefa, “How to estimate flexibility potential of household appliances? (To reduce peak load) - #SINTEFblog,” 2019.

[38] B. Neupane, T. B. Pedersen, and B. Thiesson, “Towards Flexibility Detection in Device-Level Energy Consumption,” in *Data Analytics for Renewable Energy Integration* (W. L. Woon, Z. Aung, and S. Madnick, eds.), no. 8817 in Lecture Notes in Computer Science, pp. 1–16, Springer International Publishing, Sept. 2014.

[39] H. Lee, W. Park, and I. Lee, “A Home Energy Management System for Energy-Efficient Smart Homes,” in *2014 International Conference on Computational Science and Computational Intelligence*, vol. 2, pp. 142–145, Mar. 2014.

[40] L. Pérez-Lombard, J. Ortiz, and C. Pout, “A review on buildings energy consumption information,” *Energy and Buildings*, vol. 40, pp. 394–398, Jan. 2008.

[41] T. X. Nghiem, M. Behl, R. Mangharam, and G. J. Pappas, “Green scheduling of control systems for peak demand reduction,” in *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pp. 5131–5136, Dec. 2011. ISSN: 0743-1546.

[42] S. Aslam, Q. Z. Iqbal, N. Javaid, Z. Khan, K. Khursheed, and S. Haider, “Towards efficient energy management of smart buildings exploiting heuristic optimization with real time and critical peak pricing schemes,” 12 2017.

[43] L. Pereira and N. Nunes, “Performance evaluation in non-intrusive load monitoring: Datasets, metrics, and tools—A review,” *WIREs Data Mining and Knowledge Discovery*, vol. 8, no. 6, p. e1265, 2018. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1265.

[44] J. Kolter and M. Johnson, “REDD: A Public Data Set for Energy Disaggregation Research,” *Artif. Intell.*, vol. 25, Jan. 2011.

[45] J. Kelly and W. Knottenbelt, “The UK-DALE dataset, domestic appliance-level electricity demand and whole-house demand from five UK homes,” *Scientific Data*, vol. 2, Mar. 2015.

[46] D. Murray, L. Stankovic, and V. Stankovic, “An electrical load measurements dataset of United Kingdom households from a two-year longitudinal study,” *Scientific Data*, vol. 4, p. 160122, Jan. 2017. Number: 1 Publisher: Nature Publishing Group.

[47] L. Pereira, F. Quintal, R. Gonçalves, and N. J. Nunes, “SustData: A Public Dataset for ICT4S Electric Energy Research,” pp. 359–368, Atlantis Press, Aug. 2014. ISSN: 2352-538X.

[48] H. Kazmi, I. Munne-Collado, F. Mehmood, T. A. Syed, and J. Driesen, “Towards data-driven energy communities: A review of open-source datasets, models and tools,” *Renewable and Sustainable Energy Reviews*, vol. 148, p. 111290, Sept. 2021.

[49] C. Klemenjak, C. Kovatsch, M. Herold, and W. Elmenreich, “A synthetic energy dataset for non-intrusive load monitoring in households,” *Scientific Data*, vol. 7, p. 108, Apr. 2020. Number: 1 Publisher: Nature Publishing Group.

[50] N. Buneeva and A. Reinhardt, “AMBAL: Realistic load signature generation for load disaggregation performance evaluation,” in *2017 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 443–448, Oct. 2017.

[51] A. Reinhardt, “areinhardt/antgen,” Nov. 2020. original-date: 2020-05-14T08:42:49Z.

[52] A. Reinhardt and C. Klemenjak, “How does Load Disaggregation Performance Depend on Data Characteristics? Insights from a Benchmarking Study,” in *Proceedings of the Eleventh ACM International Conference on Future Energy Systems*, e-Energy ’20, (New York, NY, USA), pp. 167–177, Association for Computing Machinery, June 2020.

[53] E. Amouroux, T. Huraux, F. Sempe, N. Sabouret, and Y. Haradji, “SMACH: Agent-Based Simulation Investigation on Human Activities and Household Electrical Consumption,” in *Agents and Artificial Intelligence* (J. Filipe and A. Fred, eds.), Communications in Computer and Information Science, (Berlin, Heidelberg), pp. 194–210, Springer, 2014.

[54] J. Albouys, N. Sabouret, Y. Haradji, M. Schumann, and C. Inard, “SMACH: Multi-agent Simulation of Human Activity in the Household,” in *Advances in Practical Applications of Survivable Agents and Multi-Agent Systems: The PAAMS Collection* (Y. Demazeau, E. Matson, J. M. Corchado, and F. De la Prieta, eds.), Lecture Notes in Computer Science, (Cham), pp. 227–231, Springer International Publishing, 2019.

[55] A. Ciocia, A. Amato, P. Di Leo, S. Fichera, G. Malgaroli, F. Spertino, and S. Tzanova, “Self-Consumption and Self-Sufficiency in Photovoltaic Systems: Effect of Grid Limitation and Storage Installation,” *Energies*, vol. 14, p. 1591, Jan. 2021. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

Appendix A

Appendix chapter

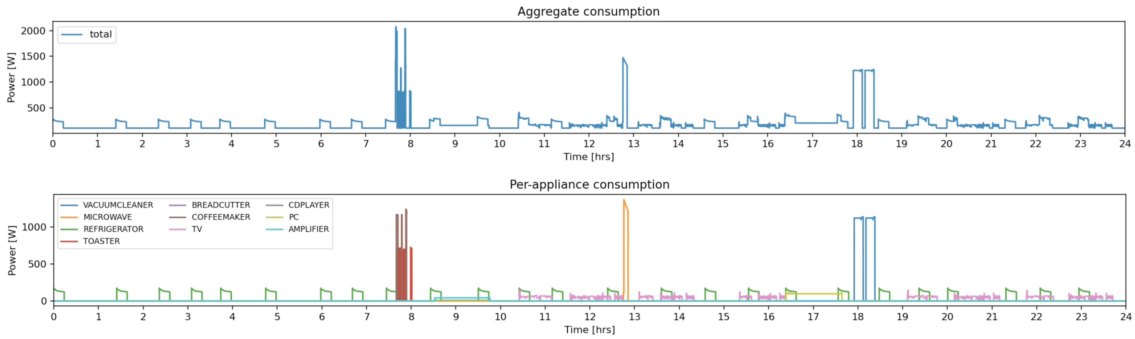

Figure A.1: Consumption profile of house 1

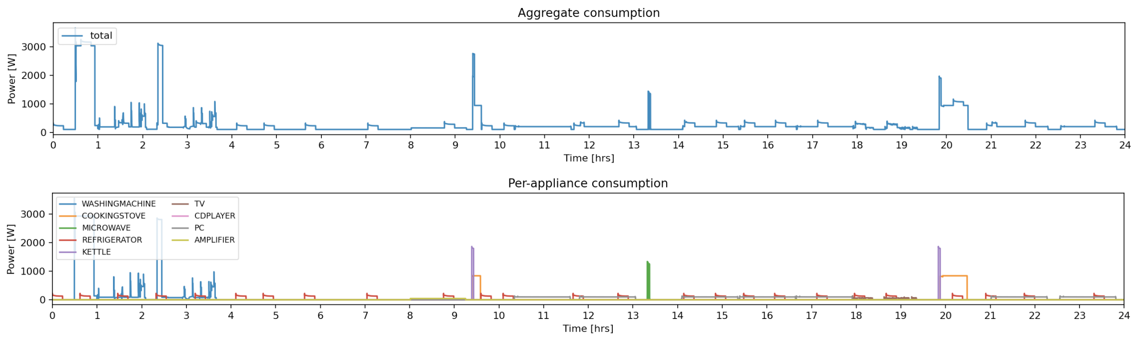

Figure A.2: Consumption profile of house 2

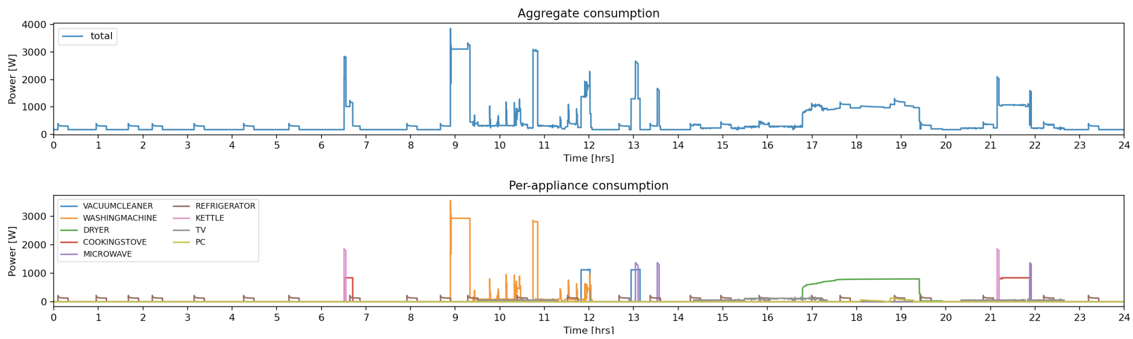

Figure A.3: Consumption profile of house 3

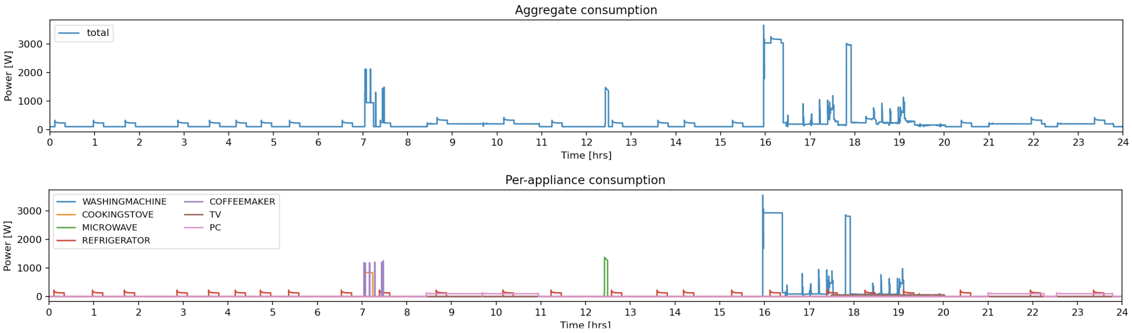

Figure A.4: Consumption profile of house 4

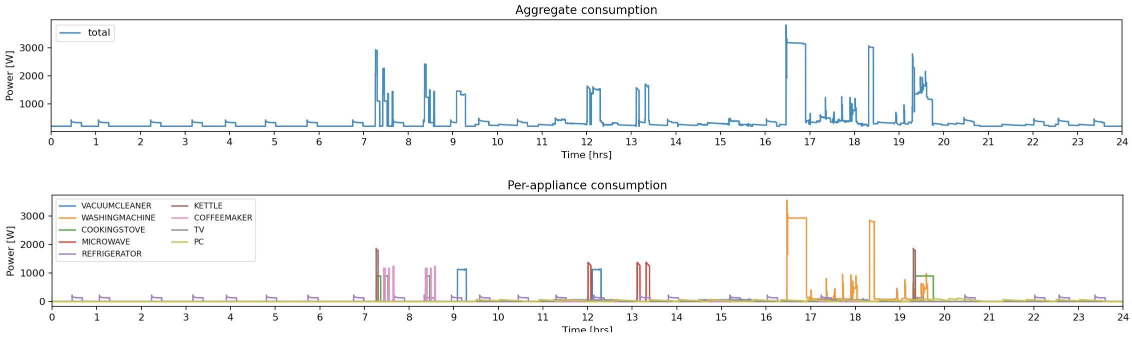

Figure A.5: Consumption profile of house 5

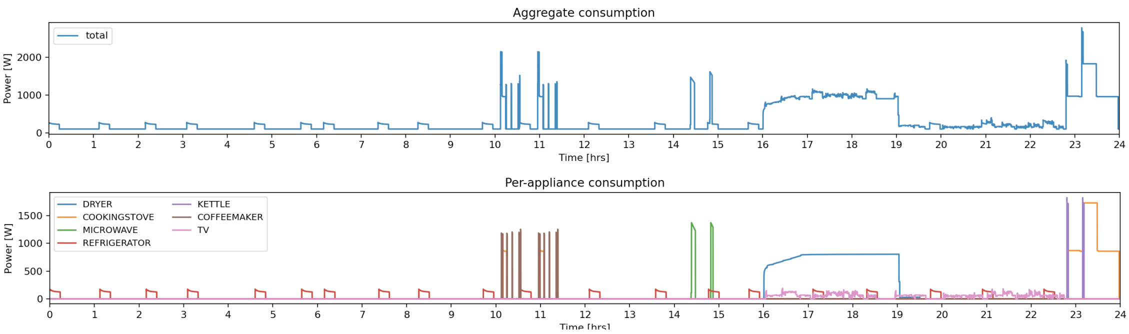

Figure A.6: Consumption profile of house 6

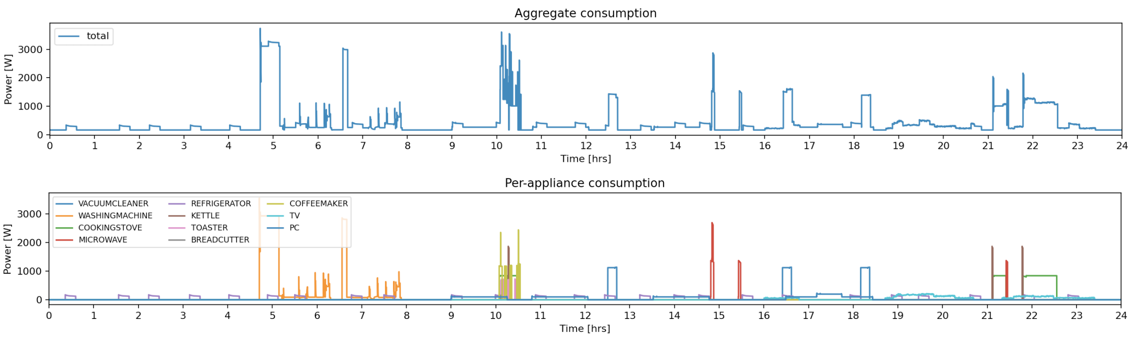

Figure A.7: Consumption profile of house 7

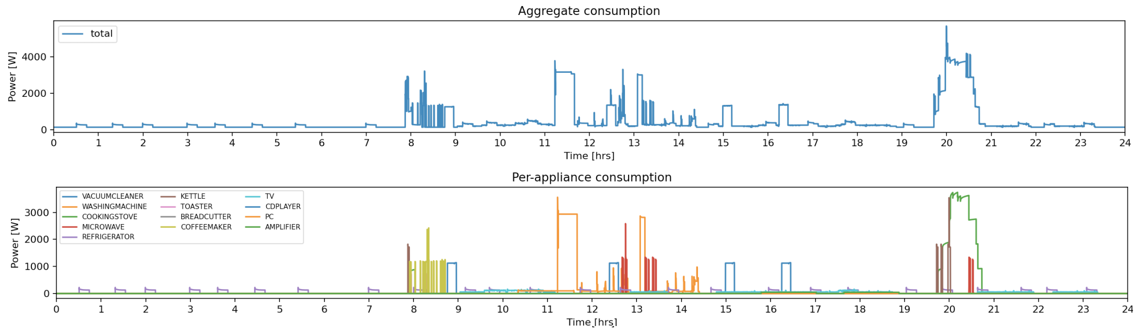

Figure A.8: Consumption profile of house 8

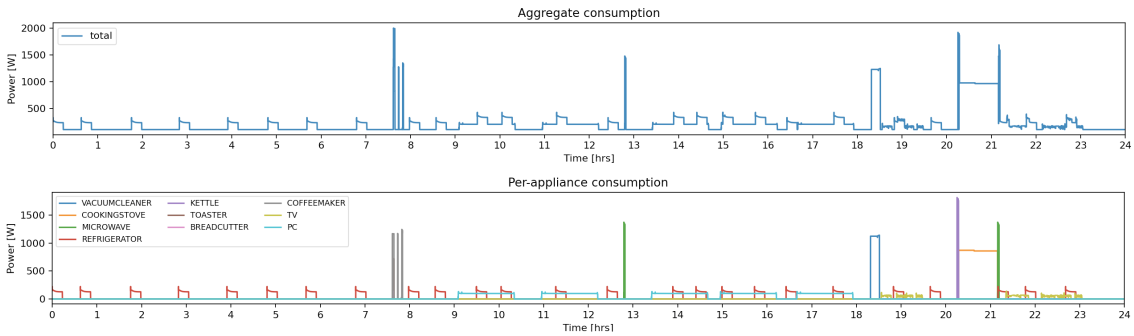

Figure A.9: Consumption profile of house 9

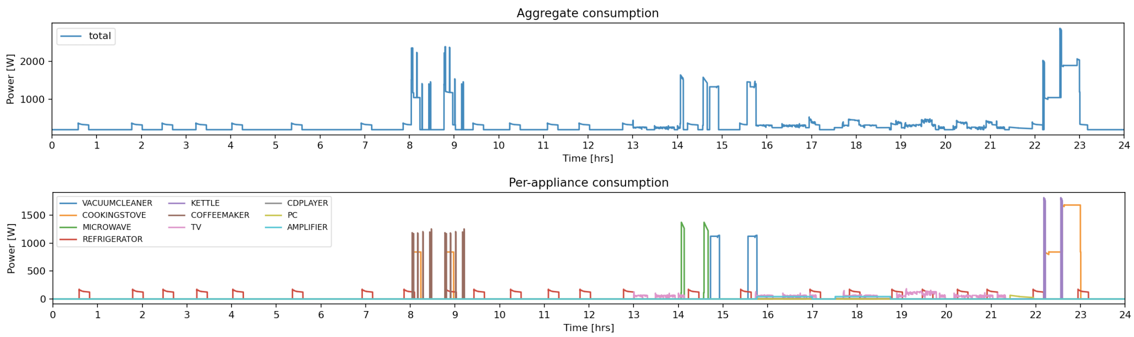

Figure A.10: Consumption profile of house 10

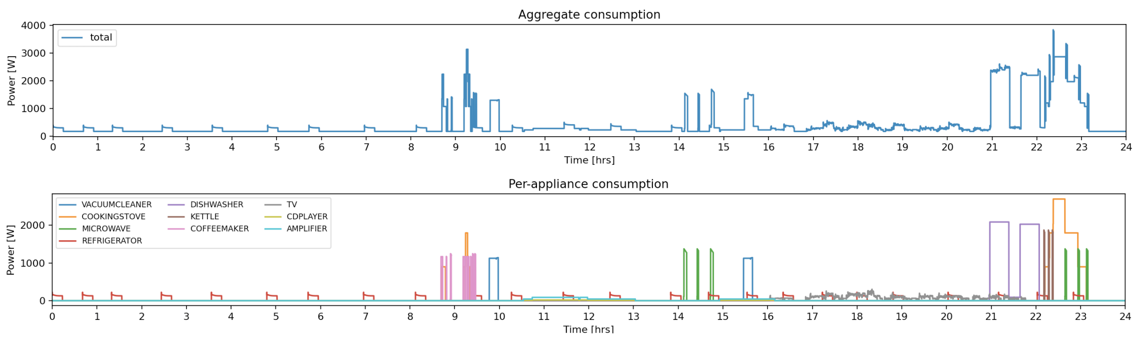

Figure A.11: Consumption profile of house 11

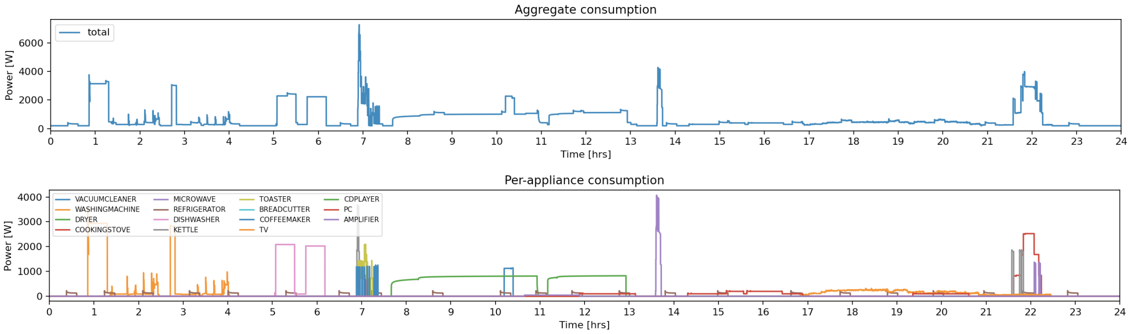

Figure A.12: Consumption profile of house 12

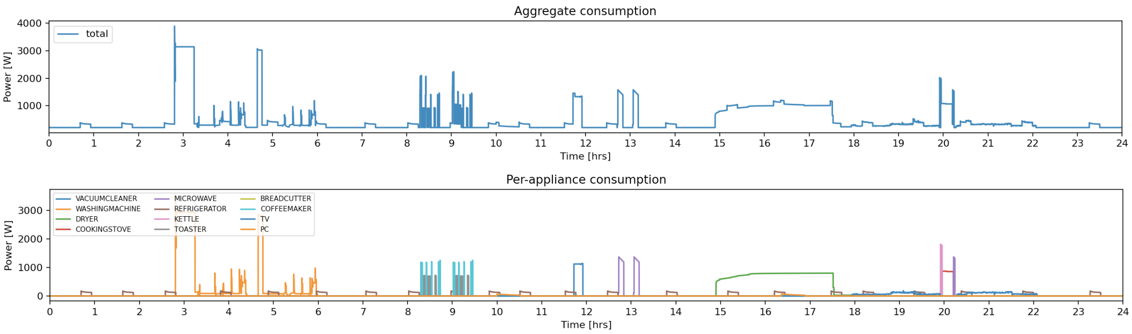

Figure A.13: Consumption profile of house 13

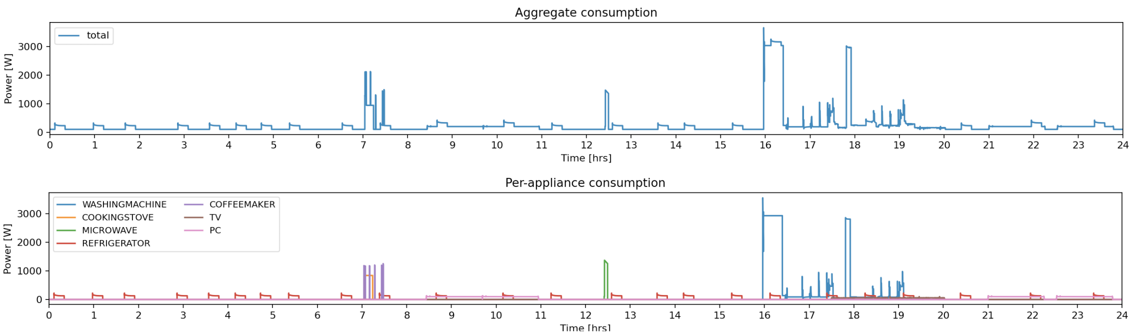

Figure A.14: Consumption profile of house 14

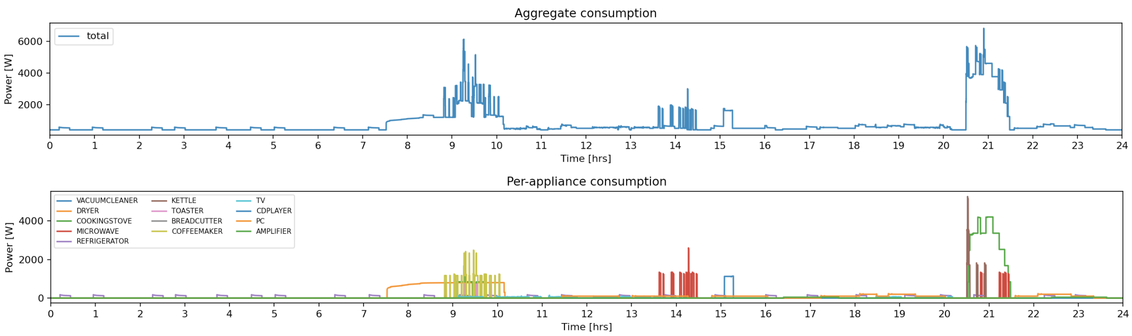

Figure A.15: Consumption profile of house 15

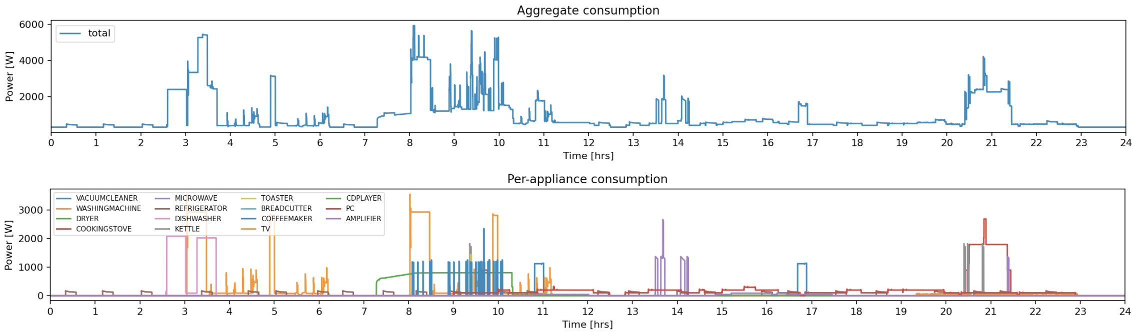

Figure A.16: Consumption profile of house 16

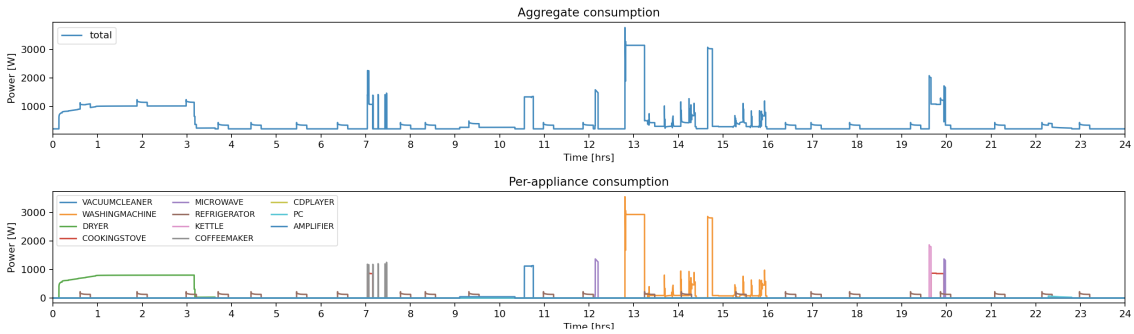

Figure A.17: Consumption profile of house 17

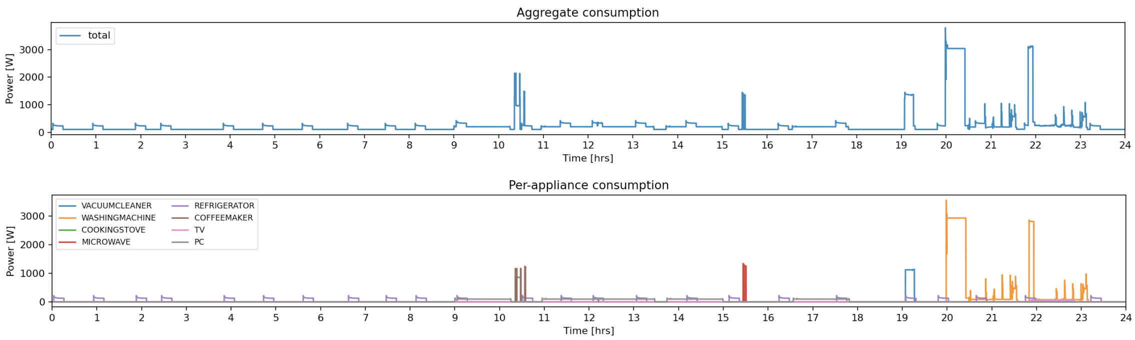

Figure A.18: Consumption profile of house 18

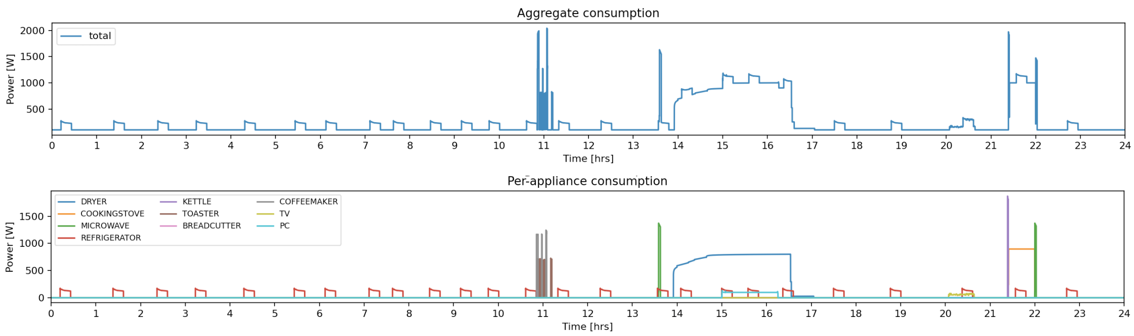

Figure A.19: Consumption profile of house 19

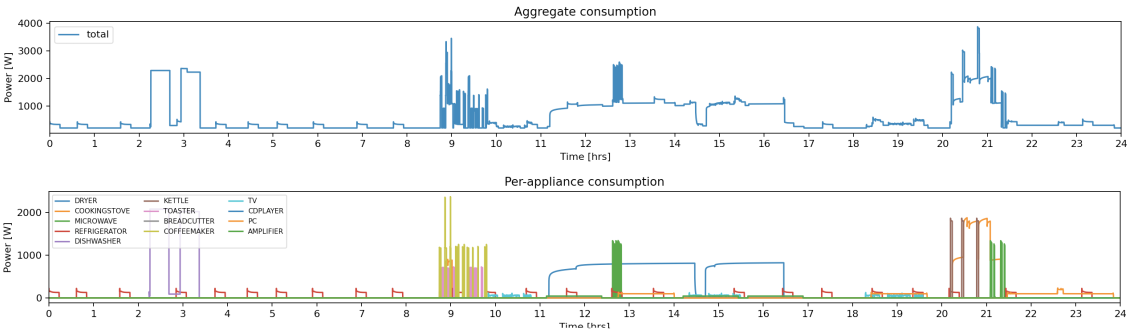

Figure A.20: Consumption profile of house 20

Acronyms

AMBAL Automated Model Builder for Appliance Loads. 12, 24

ANTgen AMBAL-based NILM Trace generator. 12, 16, 24, 25

BP Bin Packing. 4

BPUC Bin Packing with Usage Cost Problem. 15

CECs Citizen Energy Communities. 3, 6

CM Community Manager. 32, 33, 65, 66

CSA Crow Search Algorithm. 10

CSOA Cuckoo Search Optimization Algorithm. 10

DHI Diffuse Horizontal Irradiance. 27

DM Dryer Machine. 46

DNI Direct Normal Irradiance. 27

DNN Deep Neural Networks. 11

DR Demand Response. 9, 64

DSM Demand Side Management. 10

DW DishWasher. 46

EC Energy Community. iii, v, 4, 5, 16, 17, 20, 22, 35, 64, 65

ESS Energy Storage System(s). 15

EU European Union. 2

GA Genetic Algorithm. 10

GAN Generative Adversarial Networks. 11, 12

GFS Global Forecast System. 28

GHI Global Horizontal Irradiance. 27

HEMS Home Energy Management System. 9, 10

HRRR High Resolution Rapid Refresh. 28

HVAC Heating,ventilation and air conditioning. 2, 9, 10

LCD Liquid Crystal Display. 33

MILP Mixed Integer Linear Programming. 40

NAM North America Mesoscale. 28

NDFD National Digital Forecast Database. 28

NILM Non-Intrusive Load Monitoring. 11

OF Objective Function. 4, 43, 44

PPC Peak Power Contract. 19, 22, 38, 41, 46, 51, 56

PV Photovoltaics. iii, v, 4, 18, 19, 29, 34, 38, 40, 42, 43, 44, 47, 49, 51, 52, 53, 55, 56, 57, 58, 59, 60, 61, 64, 65

RAP Rapid Refresh. 28

REC Renewable Energy Community. 4, 6, 7, 8, 16

RECs Renewable Energy Communities. iii, v, 3, 4, 5, 6

RES Renewable Energy Source(s). 2, 3, 16, 18, 21

SC Self-Consumption. ix, 53, 55, 56, 60, 65, 66

SS Self-Sufficiency. ix, 53, 55, 56, 60, 65, 66

SSP Subset Sum Problem. 4, 15

USA United States of America. 2, 10

VC Vaccum Cleaner. 46

WH Water Heater. 46

WM Washing Machine. 46