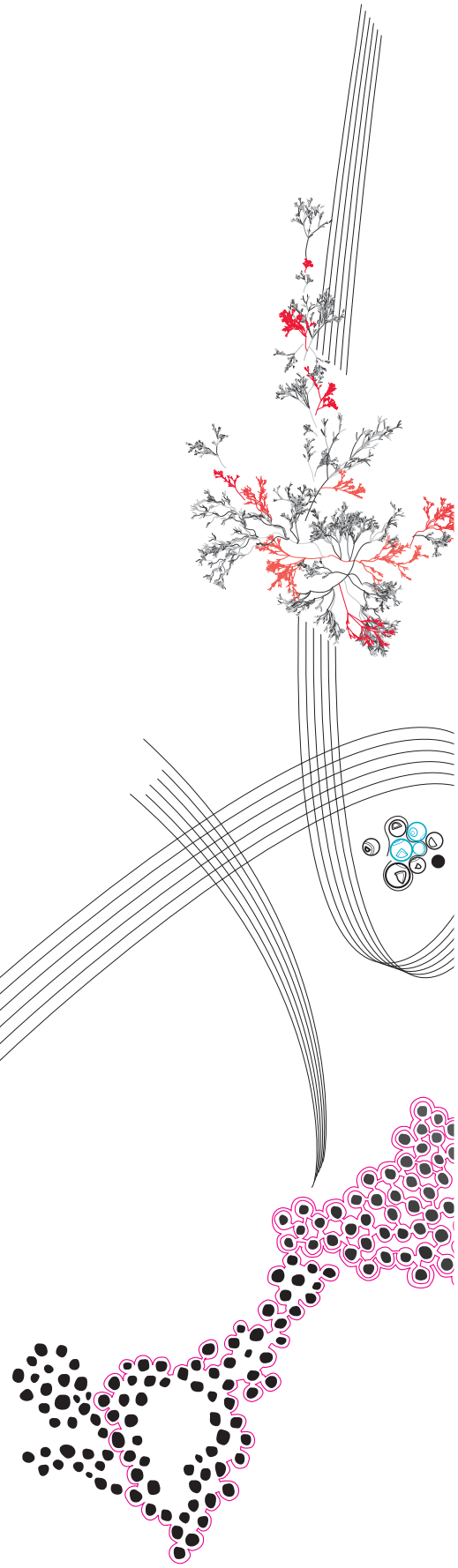


MSc Embedded Systems
Final Project

Investigating Holarchies to Perform Capacity Allocation in Smart Grids

D.H. Matena

Chair: dr. ir. Marco Gerards

Supervisors: dr. ir. Gerwin Hoogsteen
dr. Juan López Amézquita
ir. Flin Verdaasdonk

External: dr. ir. Alex Chiumento

September 10, 2024

Department of Computer Architecture for Embedded Systems (CAES)
University of Twente

UNIVERSITY OF TWENTE.

Abstract

The increase of prosumers in the electricity grid (grid users who both consume and produce electrical energy) results in higher grid load and even leads to congestion in some cases. However, congestion does not occur over the course of the entire day: there are moments where the peak load exceeds the limits of the substation. This shows that grid congestion can be managed by allowing grid users a more dynamic connection to the grid instead of a firm connection. If grid users have a non-firm connection to the grid, they are also able to use flexibility sources in order to alleviate even more stress from the grid. One problem with these flexibility sources is that they have their limitations. For example, a battery has the physical boundaries of drained and charged. Another problem is that the grid users need to coordinate with each other in order to effectively deploy these flexibility sources and not cause bigger problems for the electricity grid. Therefore, an algorithm needs to be designed to mitigate grid congestion while it prevents violations regarding the flexibility sources, by allowing for coordination between the grid users on a decentralized level.

This thesis presents a holarchy, combined with two developed algorithms. These algorithms aim to mitigate grid congestion and stay within the bounds of the flexibility sources. The first algorithm is called the Curtailment and Reallocation algorithm. It takes the preference profiles into account of the grid users, as well as preference bounds (bounds in which the grid user desires to stay in between). With these preference profiles, it calculates a profile in which these bounds are met as much as possible, while preventing grid congestion. It does not take the limits of the flexibility sources into consideration, which can result in violations in that area. The second algorithm deals with that by trading capacity with other holons in a fully distributed way, such that they are able to fully mitigate these violations regarding flexibility without a third party being involved.

In the end, the developed algorithms are compared to a state of the art congestion management algorithm by introducing multiple scenarios. These scenarios comprise a small-scale scenario and a real-life use case of an industrial district. During the comparison, the developed algorithms perform better when it comes to equality: every holon is treated equally. The results show this by having the metrics for each holon closer grouped together compared to Profile Steering. For the real-life use case, the developed algorithms adhere to the preference profile way more compared to Profile Steering, where the mean percentage of preference met is 83% for the developed algorithms, versus 20.8% for Profile Steering. The downside to the developed algorithms is the execution time, which drastically increases if the holons are only able to trade small quantities of capacity at a time.

Acknowledgment

Six years. That is how long it has taken me to complete my bachelor's and master's degrees at the University of Twente. In those six years, I have become a vastly different person and gained so much knowledge. I have also met so many people and friendships have been created that will last beyond the university. I am thankful for every person I have met, but writing about each and every one of them is going to take longer than writing this entire thesis, so I will keep it brief.

First of all, I would like to thank my daily supervisors Gerwin Hoogsteen, Juan Lopéz Amézquita and Flin Verdaasdonk. Gerwin, you have been my supervisor for three projects, including my bachelor assignment. The growth I have experienced in my work, a lot of that is thanks to you. We have had a lot of interesting discussions regarding the thesis, as well as more informal topics. Juan, thank you for your guidance and your enthusiasm regarding the Ecofactorij and Flin, thanks for every other bit of feedback I have received from you, as well as our interesting discussions and providing help when I needed it. Lastly, I would like to thank Marco Gerards, for providing interesting insights during the weekly meetings and Alex Chiumento, for taking his time to be my external committee member.

Furthermore, I would like to thank everyone in the research group of CAES, but mainly the ones who shared the same faith as me in writing their dissertations. Special thanks go out to you Frank, Remco, Larissa and Mustafa. Even though we all wanted to work on our assignments, you have ensured that we also had a lot of fun doing just that (and sometimes also lost track of time...).

Lastly, I would like to thank my friends and family for their support during these last couple of months. Het waren soms moeilijke tijden, maar samen zijn we er doorheen gekomen en is er licht aan het einde van de tunnel (en dan niet alleen met het afstuderen)! Mijn naam staat misschien voorop deze scriptie, maar eigenlijk hebben we hier met zijn allen aan gewerkt, en daarvoor wil ik jullie stuk voor stuk heel erg bedanken!

Contents

1 Introduction **1**
- 1.1 Energy Transition . . . 1
- 1.2 Dutch Grid . . . 1
  - 1.2.1 Non-Firm Capacity . . . 3
  - 1.2.2 Flexibility and Resource Allocation . . . 4
- 1.3 Communication and Control Topologies . . . 4
  - 1.3.1 Holarchies . . . 6
- 1.4 Problem and Research Questions . . . 6
- 1.5 Approach . . . 7

2 Related Work **8**
- 2.1 Congestion Management in the Netherlands . . . 8
- 2.2 Existing Congestion Management Approaches . . . 8
  - 2.2.1 Demand Side Management . . . 9
  - 2.2.2 Model Predictive Control . . . 9
- 2.3 Holarchies . . . 10
- 2.4 Resource Allocation in Other Fields . . . 12
- 2.5 Conclusion . . . 13

3 Problem Analysis **14**
- 3.1 Congestion . . . 14
- 3.2 Flexibility . . . 15
- 3.3 Preference Profiles . . . 17

4 Methodology **19**
- 4.1 Holarchy . . . 19
- 4.2 Resource Allocation . . . 19
  - 4.2.1 Curtailment and Reallocation . . . 20
  - 4.2.2 Capacity Trading . . . 27

5 Evaluation **37**
- 5.1 Simulations . . . 37
  - 5.1.1 Environment . . . 37
  - 5.1.2 Setup . . . 37
  - 5.1.3 Metrics . . . 41
- 5.2 Evaluation Method . . . 43
- 5.3 Results and Discussions . . . 43
  - 5.3.1 Small-Scale Scenario . . . 43
  - 5.3.2 Ecofactorij . . . 58

5.4 Conclusion 64

6 Conclusions and Future Work **66**
6.1 Research Questions 66
6.2 Future Work 68

Nomenclature

T	number of time steps in a day
t	current time step
H	number of holons
h	current holon
$P_{p,h}(t)$	preferred power usage of holon h at time step t
$P_{a,h}(t)$	allocated power usage of holon h at time step t
$P_s(t)$	power limit of the substation at time step t
$P_{ub,h}(t)$	upper power bound of holon h at time t
$P_{lb,h}(t)$	lower power bound of holon h at time t
$P_v(t)$	total violation amount at time t
α_h	priority of holon h
Δ_h	subtraction amount for holon h
f	factor for proportional curtailment
ζ_{max}	maximum amount of flexibility a source is able to provide
$SoC_h(t)$	available amount of flexibility at time step t
t_{begin}	earliest interval of trade window
t_{end}	latest interval of trade window
ϵ	infraction amount regarding flexibility

Chapter 1

Introduction

Climate change is becoming a bigger problem for humanity every day. Global temperatures rise annually and countries are confronted with extreme droughts and floods [1]. In order to decelerate climate change, multiple countries have signed the Paris Climate Agreement [2]. Additionally, the Netherlands implemented its own Climate Act, with the goal to reduce the amount of greenhouse gas by 49% in 2030 compared to 1990 and a 95% reduction by 2050 [3]. Since the Dutch Climate Act has been implemented, the government of the Netherlands is taking steps to accomplish the goals stated in that act.

1.1 Energy Transition

In order to accomplish the 2030 goals stated in the Dutch Climate Act, the government of the Netherlands encourages all sectors of society to reduce the amount of pollution and switch to more sustainable energy sources. For example, by promoting the use of heat pumps instead of central heating for households, driving electric vehicles (EVs), installation of PV panels and electrification of industrial processes. This movement towards using more electrical energy is called the energy transition [4].

A consequence of the energy transition is already noticeable due to increased grid utilization [5]. For example, some parts of the Dutch electricity grid are becoming congested, i.e., the grid infrastructure is unable to cope with the supply and demand of the electricity grid [6].

1.2 Dutch Grid

To understand electricity grid congestion in the Netherlands, it is important to understand the design of the Dutch electricity grid itself and the parties involved. There are three parties involved in the Dutch electricity grid. The first party comprises the grid operators, the Distribution System Operators (DSOs) and the Transmission System Operator (TSO). They ensure that the grid is operated safely, reliably and in a cost-effective manner [7]. The second party comprises the energy providers, who generate (via power plants, solar/wind farms etc.) the electrical energy that is transported through the electricity grid to the third party: the grid users. Grid users are the customers of the electricity grid, because they receive the energy from the energy providers. Examples of grid users are residential areas and industrial complexes.

Transportation of electrical energy is not as straight forward as creating a direct connection between the energy provider and grid users. A simplified overview is shown in Figure 1.1. This figure shows a centralized approach, where electricity is generated at a power plant. This output power is transported over the high voltage (HV) networks towards the substations. In the Netherlands, HV comprises voltages over 50kV. These substations reduce the voltage of the output from HV to medium voltage (MV). MV in the Netherlands is between 3kV and 25kV [8]. The electricity is then transferred towards local substations. These local substations are situated in areas with the grid users, such as residential areas and industrial complexes. At these substations, the voltage is reduced to low voltage (LV). In the Netherlands, LV is equal to 230V (phase to neutral). From these substations, the electricity is then distributed to the grid users of the grid [9].

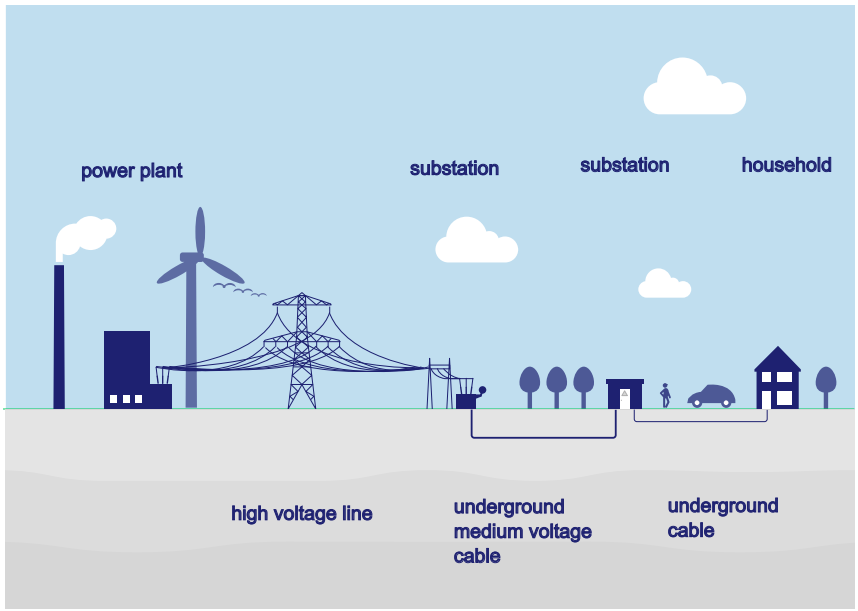
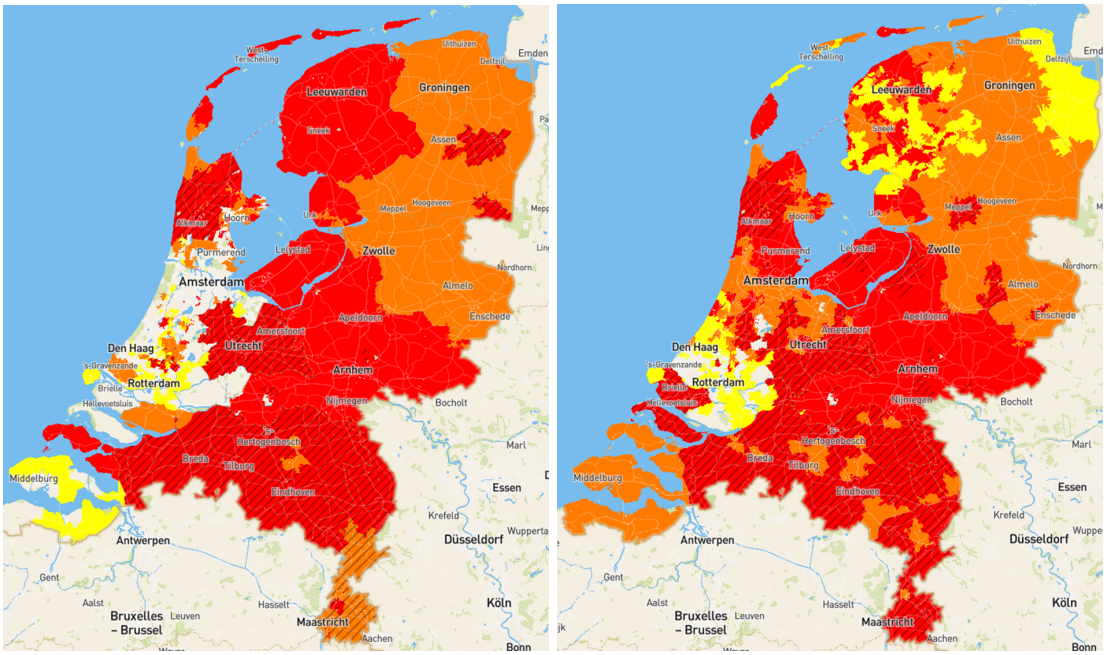


Figure 1.1: Overview of the Dutch electrical grid (adapted from [8]).

Traditionally, these grid users were always classified as consumers. The power plants were the only producers in the grid, whose output energy was transported to these grid users. However, with the introduction of photovoltaic (PV) panels, wind turbines and distributed energy resources (DERs), the grid users have slowly shifted into the role of a prosumer: they can both produce and consume energy. This new prosumer role, coupled with an increase in electricity usage results in higher grid utilization and the power generation becoming decentralized.

Whereas the grid was previously perceived as being capable of transporting unlimited amounts of power, it can no longer be considered as such. The reason behind this is that the cables and the transformers at the substations in the electricity grid have a certain upper bound to transport electrical energy, also called capacity. This capacity determines the maximum amount of electrical energy that can be transported at any time. However, with the increase in both production and consumption of electrical energy, this capacity is becoming a bottleneck. When the demand or supply exceeds the physical capacity of the cable, the grid is not able to cope with that extra demand and grid congestion occurs [10].

In the Netherlands, grid congestion is becoming an large problem, as shown in Figure 1.2. This figure shows grid congestion in the Netherlands both for supply and for feed-in of electricity. Here, orange indicates that there currently is no transport capacity available (pending congestion management research). Red indicates that there is no transport capacity available and congestion management cannot be applied [11].



(a) Map of grid congestion in supply Dutch Grid [11].

(b) Map of grid congestion in feed-in Dutch Grid [11].

Figure 1.2: Maps of grid congestion in the Netherlands.

1.2.1 Non-Firm Capacity

According to Figure 1.2, it appears that a large part of the Netherlands is congested. However, this is only the case on paper. In reality, grid congestion is a dynamic problem, i.e, it does not occur during the entire day, but only during peak hours. The grid appears to be fully utilized in Figure 1.2. In practice however, there still is capacity in the grid during off-peak moments to provide for households, industries etc. An example of the load at a substation over the course of a day is shown in Figure 1.3 [12]. The residual capacity shown in that figure indicates the amount of unutilized energy by the connected parties.

Previously, the only grid connection contracts grid users can have are firm contracts. With firm contracts, the grid user is guaranteed a certain predetermined amount of capacity during the entire day. For example, a factory can use 100 kW at all times. However, firm contracts do not recognize the time-dependent nature of grid congestion. Grid operators are starting to use non-firm capacity (NFC) contracts to deal with this time-dependent aspect. NFC contracts work by providing varying access to the grid depending on the amount of available capacity. It may be the case that they are not granted any access to the grid during peak hours. However, NFC contracts are generally cheaper than firm contracts, providing an incentive for customers with varying energy demands and flexibility [13].

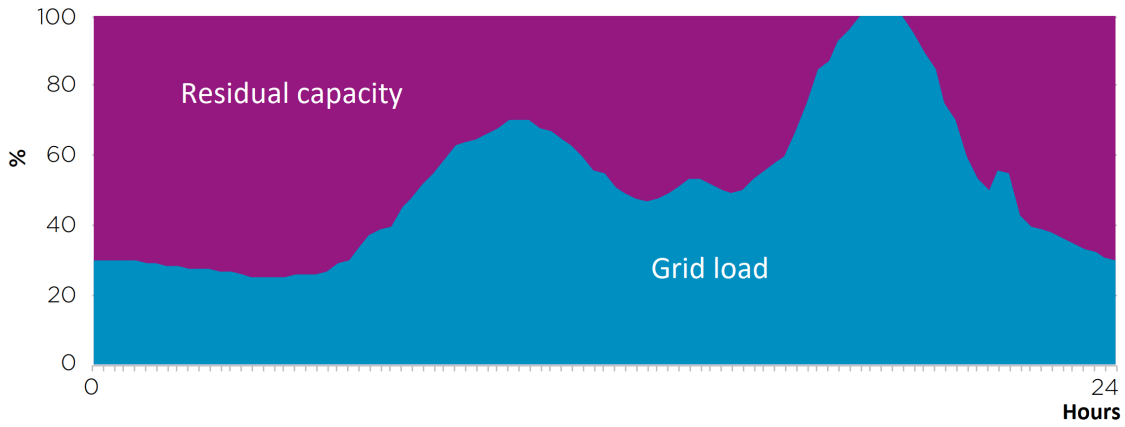


Figure 1.3: Load profile for one day (adapted from [12]).

There are three types of NFC: flexible transmission rights, time-based transmission rights and a combination between flexible, time-based and firm transmission rights [6]. Flexible transmission rights allow grid users to adjust the transmission capacity based on the grid conditions and market needs, allowing for a greater adaptability to fluctuations. Time-based transmission rights allow for the allocation of transmission capacity for specific time periods. The final type is a hybrid between the aforementioned types of NFC and firm contracts, where the grid user has a certain amount of firm capacity with an addition of some flexible capacity. This capacity can be allocated with the use of either flexible or time-based transmission rights.

1.2.2 Flexibility and Resource Allocation

Grid users with non-firm contracts need to become more flexible in their energy usage. This flexibility can come in several ways. Examples of flexibility include energy production management, load balancing and storage management.

With energy production management, grid users can opt for selectively activating their local energy production sources (such as PV panels) to adjust for the available capacity in the electricity grid. Alternatively, load balancing concerns the management of processes to comply with the limits of the grid. This can be done by for example delaying tasks with a low priority at moments of grid congestion. If the aforementioned methods still do not prevent grid congestion, energy storage can be used at the moments of grid congestion. By strategically deploying these storage elements during grid congestion and recharging them at moments where possible, it is still possible to operate without grid congestion occurring.

Deciding when to utilize the management of these flexibility resources can also be modelled as a resource allocation problem. It involves balancing the supply and demand to maintain the stability in the grid and prevent grid congestion. Resources can be allocated anywhere from minutes up to days in advance.

1.3 Communication and Control Topologies

An issue comes to light when multiple grid users with a NFC contract have a common connection to the grid. If these grid users allocate their resources to prevent grid congestion,

Table 1.1: Performance of the communication topologies.

	Control	**Scalability**	**Resilience**	**Complexity**
Centralized	Single centralized controller	Limited by capacity of controller	Single point of failure	Simple to implement
Decentralized	Multiple independent nodes	Dependent on number of control nodes	More robust to single node failures	More complex than centralized
Distributed	Equal participation by all nodes	Highly scalable	Highest resilience	Most complex

it is imperative to have some sort of coordination amongst each other. This coordination requires a form of communication network in order to function. In a generalized sense, there are three types of communication network topologies: centralized, decentralized and distributed.

Centralized communication networks contain one central controller. This central controller has a direct communication link with every node in the network and is responsible for the decision making in the network.

In decentralized communication networks, there is not one central controller. Instead, there are multiple controllers throughout the network. These controllers are able to communicate with each other, as well as having a direct communication link to (some of the) nodes in the network.

Distributed communication networks go one step further, where the control aspect is divided across every node. Furthermore, each node is connected to either some or all of the other nodes, depending on the goal of the topology.

An overview of the different communication topologies is also shown in Figure 1.4 with the performance of each topology shown in Table 1.1.

Figure 1.4: Overview of the three communication topologies, the yellow dots are the controllers and the blue dots are the clients [14].

This shift towards the prosumer role within the electricity grid results in regulating of the electricity grid in a centralized manner to become more difficult. In order to allow for

a flexible network to solve for grid congestion, both decentralized and distributed networks are viable options. One communication network topology that has shown to work in a smart grid scenario is the holarchy topology [15].

1.3.1 Holarchies

A holarchy is firstly introduced by Koestler [16] and described as a combination between a decentralized and distributed communication topology consisting of multiple holons (the nodes in a holarchy) that are able to communicate with each other, as well as have decentralized control. An overview of a holarchy is shown in Figure 1.5, in which the holons are able to operate in an autonomous and self-reliant way, but they also communicate and coordinate with other holons in order to achieve a common goal.

When looking at an industrial district, every company has its own processes to run, its own power generation and its own energy buffers. These resources are utilized within the complex itself. However, how much power can be drawn from the electricity grid while preventing grid congestion depends on the amount of power drawn by the other companies in the industrial district, thus making communication between each other beneficial. This can be modelled as a holarchy by considering every company in the industrial district as a holon. If that is the case, they are able to still fully utilize their own resources, but also communicate and coordinate with each other to accomplish that common goal of preventing grid congestion in the industrial district.

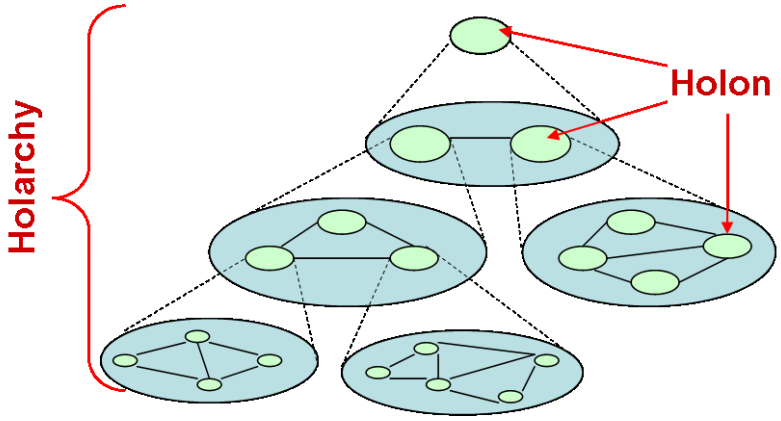


Figure 1.5: Schematic overview of a holarchy [15].

1.4 Problem and Research Questions

When grid users (consumers and prosumers) with NFC contracts and flexibility sources have a common connection to the grid, resource allocation can become computationally complex. The main reasons for this computational complexity are that both grid congestion needs to be prevented and the limits of the flexibility sources need to be taken into account. These limits exist because of the physical boundaries of the flexibility sources. This situation raises the following research question:

How can decentralized resource allocation be implemented in a holarchical smart grid, in which flexibility can be exploited to guarantee a feasible power

profile?

Along with this main research question, the following sub-questions have been defined:

- *Which algorithms can be used for the decentralized allocation of resources in a holarchy and how do they perform?*
- *What are the benefits and drawbacks of implementing a holarchy in a smart grid compared to the state of the art?*
- *How much flexibility is needed in order for the holarchy paradigm to have a beneficial effect?*

1.5 Approach

In order to answer the research question, several steps need to be taken. The first step is researching the existing state of the art regarding holarchies and the allocation of resources. The second step is to set up a holarchy and defining the holons. The third step is to allocate the resources using the holarchy. The allocation of resources aims to utilize the flexibility sources of each holon in order to reduce the peak load and hence prevent grid congestion.

The remainder of the thesis starts by looking at the existing congestion management approach used in the Netherlands, existing resource allocation algorithms in Chapter 2 and implementations of the holarchy. Chapter 3 analyzes the problem at hand in detail, after which Chapter 4 elaborates on the aforementioned approach used to tackle the research question. Chapter 5 describes a simulation setup and results of the developed algorithms in Chapter 4 with a use case consisting of a Dutch industrial district, after which the thesis concludes with Chapter 6.

Chapter 2

Related Work

This chapter presents previous research as a background. At first, the current situation in the Netherlands regarding congestion management is presented. The second part of this chapter concerns the research regarding existing (decentralized) congestion management approaches in smart grids. Thirdly, holarchies and its appearances in smart grids is discussed. Lastly, we focus on resource allocation algorithms in other fields that can be translated into the field of electricity grids. This chapter then wraps up with a summary.

2.1 Congestion Management in the Netherlands

As highlighted in the introduction, grid congestion is a big problem in the Netherlands. Currently, congestion management is largely responsive and on a voluntary basis [17]. When the electricity demand is higher than the capacity of the electricity grid, the TSO, TenneT, declares congestion in the affected area. Once congestion has been declared, TenneT assesses the situation along with the grid users in that area and evaluate possibilities regarding the redistribution of capacity. During the moments where the demand becomes too high in that area, TenneT requests the grid users to reduce their use of the grid momentarily. If the grid users respond and reduce their electricity usage, they are compensated for the discomfort via bids. Conversely, large consumers can offer to temporarily use more electricity when the demand for discharge is too large. If that is the case, TenneT requests other grid users to temporarily increase their electricity usage.

TenneT also states that congestion management is a temporary measure in the Netherlands [17]. TenneT is simultaneously working on reinforcing the electrical grid. When the grid has been reinforced, there is sufficient capacity for every grid user without resorting to congestion management.

2.2 Existing Congestion Management Approaches

Huang et al. [18] surveyed different types of congestion management approaches in smart grids. Implementations of these algorithms already exist. Some of the state of the art congestion management approaches in both theory and in practice are described in this section.

2.2.1 Demand Side Management

A main strategy to combat peak loads is Demand Side Management (DSM) [19]. DSM is a method that provides incentives for grid users to change their energy usage to combat grid congestion. This incentive can be done in several ways, such as price steering and profile steering.

Price Steering

With price steering, the grid users are incentivized to change their usage based on energy prices. These energy prices are predetermined and change depending on the estimated grid load (when the grid load is estimated to be high, the price is also high). These varying energy prices can be implemented in one of two ways: with uniform pricing and differentiated dynamic pricing [19].

Uniform pricing provides the same price signal to each grid user. This at first seems to solve for the peak load, since the price is the highest during for grid users during peak load. However, research shows that this might only shift the peak load toward the moments where the energy price is the lowest, thus not a reduction in the peak load [20].

With differentiated dynamic pricing, the price signal sent to the grid users is not the same. As a result, each grid user experiences the lowest prices at different moments. This encourages the grid user to shift their peak loads to these cheaper moments, resulting in the total load on the grid being more evenly distributed. The load being more distributed is also known as peak shaving.

Profile Steering

Van der Klauw [21] provides a different method for peak shaving. Where price steering results in (local) peak shifting, profile steering attempts to use flexible sources as a way of shifting the load. Instead of using energy prices as an incentive, the load is shifted by sending a preference power profile from the grid user to the controller. This preference profile consists of a day-ahead power profile that contains the amount of power the grid user desires to utilize. After this step, each individual grid user is requested to send a candidate schedule to optimize the objective function given the aggregate power profile (the combined power profiles for each grid user). From these schedules, the candidate schedule with the most improvement is chosen and the grid user who submitted that candidate schedule is asked to update to that candidate schedule. After which, this step of creating new candidate schedules is repeated until no more significant improvements are made according to the controller.

2.2.2 Model Predictive Control

Research has also been performed regarding predictive control. Predictive control methods are employed in order to prevent grid congestion instead of reacting when grid congestion occurs. In [22], two types of predictive control are evaluated and compared to each other by using price steering approaches that are developed from price steering using shadow prices in [23]. These types of predictive control are centralized model predictive control (CMPC) and distributed model predictive control (DMPC).

CMPC entails a system that revolves around a centralized communication topology, in which the central controller has knowledge of the entire system and also solves the optimization problem of the grid by using the aggregated power profile of the connected parties.

With DMPC, the system is distributed and thus comprises multiple controllers. These controllers each receive a shadow price from the DSO for the optimization algorithm. A shadow price refers to the cost of performing certain actions. Each controller minimizes costs based on these shadow prices by allocating their resources. After executing this process, the resulting profiles are sent to the DSO. The aggregated profile is then evaluated for congestion. If congestion still occurs, the shadow price is updated and the process restarts until either there are no significant power violations or if the maximum amount of iterations has been reached.

In the end, both methods prevented grid congestion, where DMPC utilizes these shadow prices and CMPC utilizes the global knowledge. Because of this global knowledge, CMPC was faster in terms of computation time compared to DMPC. DMPC is computationally more complex, but it provides almost identical results compared to CMPC. Especially with the grid becoming more decentralized, a centralized approach is becoming more difficult to maintain and DMPC is starting to become the more viable option.

2.3 Holarchies

This section describes multiple different implementations of the holarchy architecture in smart grids, focusing on the functionalities, benefits and challenges of the various holonic architectures.

Egert et al. [24] describe an implementation of a holarchy, in which a section of an electrical grid is divided into multiple holons connected hierarchically. In this holarchy, each holon comprises multiple prosumers containing flexibility sources. Utilizing these flexibility sources are able to counterbalance grid imbalances between supply and demand.

Delving deeper into a holarchy itself, understanding the functions of a single holon becomes crucial. According to [25], these functions include:

- Monitoring function: detects and analyzes the current situation in both the electrical grid and the energy prices in order to make decisions.
- Control function: balances the load and production within the holon.
- Scheduling and handling function: Allocates the load and production based on the information gathered by the monitoring function
- Shedding function: identifies and disconnects loads when needed according to the scheduling and handling function.
- Forecast function: predicts the energy usage and production for that holon.
- Prioritization function: provides every load with a priority when some loads may need to be disconnected according to the shedding function.

A practical example of a holarchy regarding power distribution is discussed in [26]. In this system, a power distribution system is divided into a three-layer holarchy. An overview

of this system is shown in Figure 2.1. The holons at the neighborhood level have a group in which they communicate with each other, as well as with a holon at the feeder level.

In order to allocate the available power in the grid according to [26] and Figure 2.1, three steps are taken. The first step is for the holon at the feeder level to aggregate the prosumptions (productions and consumptions) of the underlying group of holons. The second step is to calculate a three-phase optimal power flow at the substation level by using the information gathered at the feeder level. The holons at the substation level then transmit these results back to the holons at the feeder level, after which the holons at the feeder level calculate a one-phase optimal power flow for each of the holons in the neighborhood level. The nodes (producers and consumers) within the holons at the feeder level then transmit these values to the corresponding nodes in the holons at the neighborhood level. The third step is feedback. Each holon in the neighborhood level transmits its aggregated prosumption to the corresponding node at the feeder level. The holon encapsulating this node evaluates if there is a power flow mismatch. Should that be the case, then it readjusts the values calculated by the one-phase optimal power flow algorithm, after which that result is sent back to the holons at the neighborhood level. This step is also performed between the holons in the feeder and substation level.

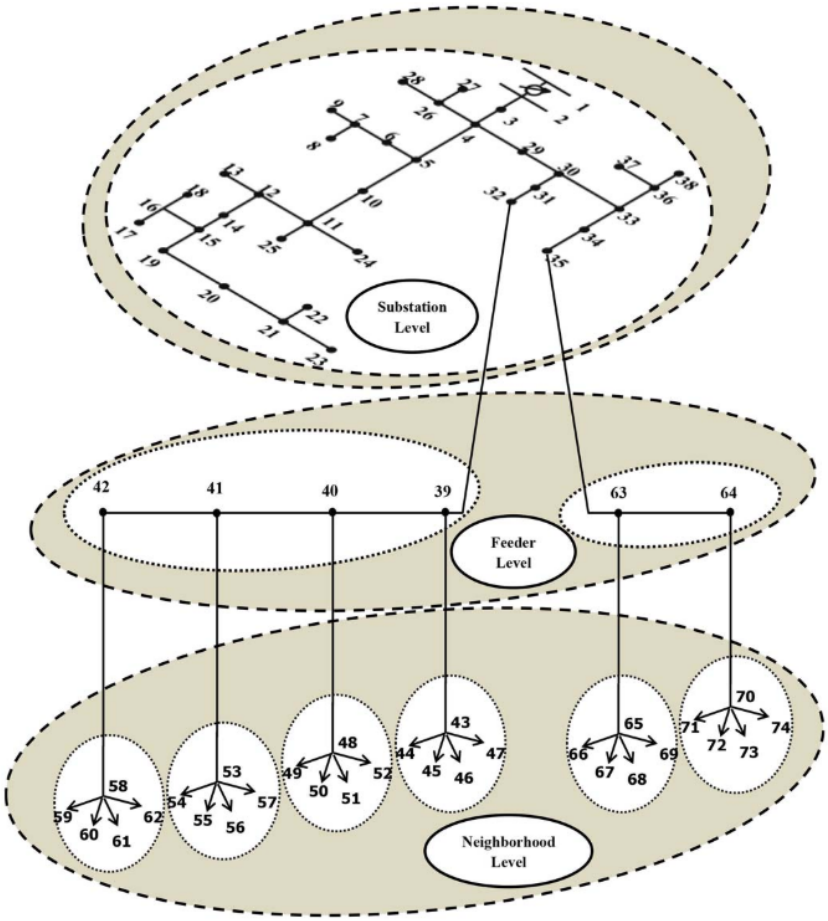


Figure 2.1: Overview of a power distribution system used in [26].

In another study, [27] explored the integration of a holarchy into an existing multi-agent

system (MAS) for the electrical grid described in [28]. This new holonic system is able to have fully decentralized communication and no holon has the knowledge of the global system. This removes the need for a centralized controller and hence results in a more scalable system.

Ashrafi and Shahrtash [29] describe their version of a holarchy to prevent voltage violations in the grid. Their holarchy is set up in a way that the every node within the holons is connected to a holon at the higher level based on a certain priority. This priority is determined by the amount of voltage violation occurring at that node (also known as the need of that node). These nodes are connected to the nodes in the upper level by determining the layout in which the voltage violation is minimized. However, deciding on the layout of the nodes in the different holons is still done by a centralized controller that receives all data from the system.

To improve upon that research in [29], Ashrafi and Shahrtash investigated the ability of negotiation in a holarchy [30]. They investigate the possibility of coordination among holons in a holarchy without the need for a controller. In their model, the nodes in the holons are able to read their own voltages and are able to communicate with other nodes to perform soft-negotiation. This round of negotiation allows for the absence of a centralized controller that makes the decisions in the network and the communication can be done locally, resulting in a more scalable system compared to centralized approaches.

Holarchies are also able to aid in case of blackouts because of their flexible nature [31]. In this research, the electrical grid is divided into multiple holons that each contain producers, consumers and junctions (the connections between the producers and consumers). The holons are set up in such a way that they are self-reliant, thus able to operate fully autonomously in case of a blackout somewhere else in the electricity grid.

Besides blackouts, it may occur that links between nodes in a holon suffer from a failure. In the case of failed links, Abdel-Fattah et al. [32] introduce a method of allowing holons in a holarchy to dynamically reconfigure. This method starts at the moment a link failure is detected, after which the holon that detected the failure broadcasts a message to its neighbors (holons that have a direct connection to the broadcasting holon). These neighbors also send a message to their neighbors. During this proximity-based information exchange, the holons establish new links with each other. Implementing this self-healing aspect in holarchies allow for a more robust and resilient communication network.

2.4 Resource Allocation in Other Fields

When investigating resource allocation in electricity grids, valuable lessons can be learned by looking at implemented strategies from other domains that are able to be implemented in electricity grids as well. One of these domains is in real-time systems, in which job scheduling is a relevant topic. Examples of job scheduling algorithms are First-Come, First-Serve (FCFS), Shortest Job First (SJF) and Round Robin (RR).

FCFS [33] schedules its jobs in according to the arrival time. The job that has arrived at the processor first, gets planned first. If there are not enough resources to plan a job at its arrival time, it gets shifted until there are enough resources to plan the job. SJF [34]

takes the job from the queue that has the shortest execution time and plans that job next. RR [35] executes the jobs by allocating a fixed time-slot to each job in the queue. If the job has not been executed at the end of the time-slot, it is inserted back into the queue and the next job starts. The differences between these algorithms is that FCFS focuses on prioritizing jobs that arrived first. SJF focuses on prioritizing the shorter jobs, thus giving a low priority to jobs that take longer and reducing their chance of finishing before the deadline. RR prioritizes fairness, since the jobs are allocated the same amount of time every cycle.

These algorithms can be translated to electrical grids, where the jobs are the energy profiles and the processor time is the capacity in the electrical grid, which also has an upper limit.

Another field in which resource allocation is key to optimize the systems is supply chain management. There are a lot of decisions to be made simultaneously in supply chain management regarding the planning of different jobs [36],[37]. In [36], both tardy and non-tardy jobs are taken into account. In order to finish the jobs on time as much as possible, their objective is to minimize the sum of the number of tardy jobs and delivery cost. To accomplish that, a branch and bound algorithm is used [38]. Branch and bound algorithms are used to split bigger problems into sub-problems (branching), that are used to create a tree structure. The optimal solution can be found by traversing this tree recursively and pruning the non-optimal solutions (bounding).

2.5 Conclusion

This chapter has provided an overview of relevant research. Initially, it explored the current congestion management approach in the Netherlands, highlighting the reactive and voluntary nature. Subsequently, it reviewed existing approaches regarding congestion management with a focus on decentralized methods. The concept of holarchies in smart electrical grids has also been examined. Finally, resource allocation methods from other fields are highlighted that can be utilized in the smart grid domain. The holarchy paradigm can be used in the smart grids scenario in combination with resource allocation methods can be used to combat grid congestion at the level of a substation.

However, a couple things need to be kept in mind when designing an algorithm to prevent grid congestion. The first problem is that a more decentralized grid results in worse scalability regarding centralized congestion management and resource allocation. Furthermore, using price signals when trying to steer the power usage of customers can result in peak shifting instead of peak shaving, which does not solve grid congestion.

Chapter 3

Problem Analysis

3.1 Congestion

Looking back at Figure 1.1, congestion can occur everywhere. Looking at an industrial district with H companies that have a common connection to the electrical grid, congestion can occur when the substation is not able to cope with either the aggregated power demand or supply of the companies. In order to determine if congestion occurs, the total power consumption at substation level is needed. This is calculated by taking the sum of the power usage of every company h:

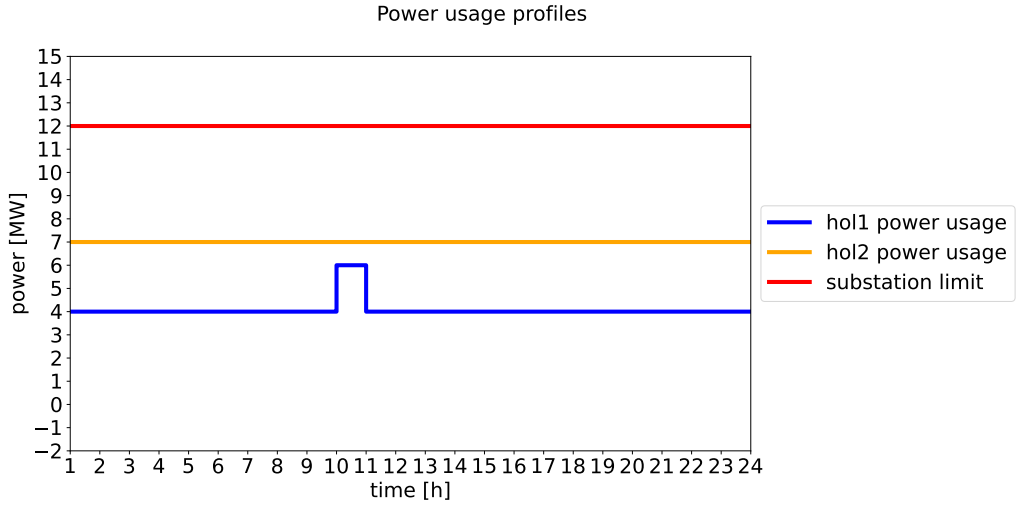
$$P_{tot}(t) = \sum_{h=1}^{H} P_{a,h}(t) \tag{3.1}$$

In Equation 3.1, $P_{tot}(t)$ is the aggregated amount of power of every company combined at time t. $P_{a,h}(t)$ is the amount of power used by company h at time t. Grid congestion occurs when P_{tot} exceeds the amount of available capacity to the industrial district at time t. I.e.:

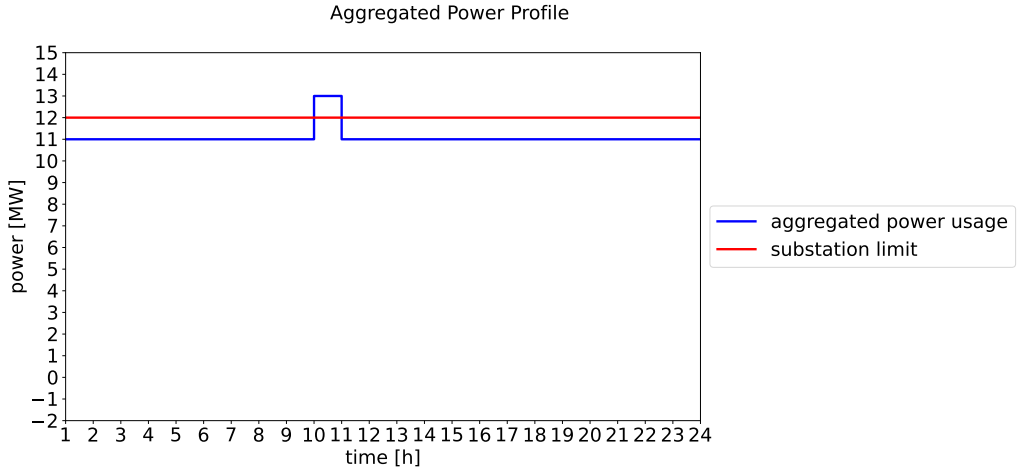
$$\exists t \in T \text{ where } |P_{tot}(t)| > P_s(t) \tag{3.2}$$

In Equation 3.2, $P_s(t)$ is the amount of available capacity at the substation, which is the total amount of capacity available to the industrial district at time t. Grid congestion occurs if Equation 3.2 holds.

An example of congestion is shown in Figure 3.1. Figure 3.1a shows the preferred power usage of two companies (com1 and com2), as well as the amount of available capacity at the substation over the course of one day. When taking the aggregate power using Equation 3.1, Figure 3.1b is created. It comes to light in Figure 3.1b that the aggregated power exceeds the amount of available power, resulting in grid congestion at $t = 11$ if these power profiles are realized the following day. This issue can be solved by utilizing flexibility from the industrial district.



(a) Preference profile of two companies and amount of available power.



(b) Aggregated power profile and amount of available power.

Figure 3.1: Example of grid congestion.

3.2 Flexibility

Each of the companies has different energy needs and characteristics. For example, some companies are logistical companies with charging stations for electric trucks that need to be charged at night, whereas other companies concern energy-intensive industrial processes. These companies have completely different energy needs and characteristics, but they are both able to provide flexibility in some ways. For the logistics companies for example, the electric trucks need to be fully charged in the morning; it does not matter when the truck is charged at night. Furthermore, the energy-expensive industrial processes might be able to turn down its production speed resulting in a lower power profile.

Besides these different characteristics that are specific to the type of company, there are also other types of flexibility sources. Flexibility sources are buffers, in which energy can be stored to drain at a later moment. From a higher point of view, this looks the same as load shifting, since they still need to be recharged at a later (or earlier) point in

time but the company is still able to operate like normal. Examples of such buffers include compressed air, batteries and thermal storage.

These flexibility sources can be utilized to shift the power profile of companies to prevent grid congestion, while keeping the same total energy usage over the course of a day.

However, one thing that needs to be taken into account when utilizing these flexibility sources is that they are not infinite: The state of charge (SoC) of the flexibility sources are always bound by a lower and upper limit. Not only do companies have to prevent exceeding the limit of the substation, but also the state of charge should always be in between (and including) 0 and the capacity (ζ_{max}) of that flexibility source:

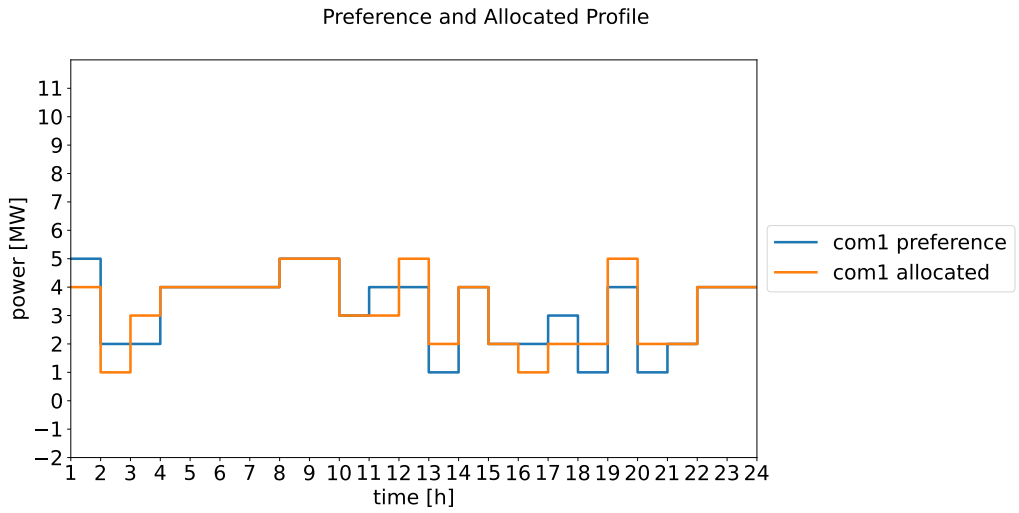
$$0 \leq SoC(t) \leq \zeta_{max} \quad \forall t \in T \tag{3.3}$$

An example is presented in Figure 3.2. Figure 3.2a shows both the preference and allocated profiles for company com1. The preference profile is the amount of capacity com1 would like to receive during the day. However, these amounts cannot be met if grid congestion needs to be prevented. To still allow for the same amount of capacity throughout the day, com1 is allocated either more or less power. This difference results in the usage of its battery: if com1 is allocated less capacity compared to the preferred amount, the battery can be discharged to bridge that gap and vice versa. In theory, the amount of allocated energy is the same as the preferred amount, but Figure 3.2b indicates that the SoC becomes smaller than 0 multiple times. This indicates a deficit in the energy usage that the flexibility sources of that company cannot compensate for.

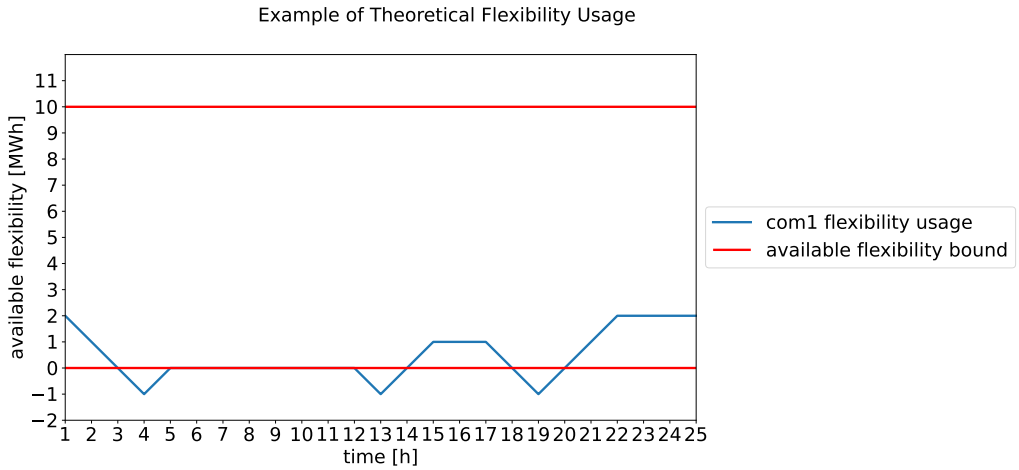
Cases may occur where grid congestion is solved by utilizing these flexibility sources, but the limits regarding the state of charge are exceeded Equation 3.3. In that case, companies should be able to coordinate with each other to trade capacity. Capacity trading entails two companies that are able to solve for the violation in Figure 3.2b by exchanging capacity before the violation, after which they exchange the capacity back without causing a new violation. This trading is not done directly (by directly transferring power between the two companies), but rather by changing their power usage, resulting in no change in the overall grid utilization.

However, implementing flexibility sources in the electricity grid can be detrimental for the electricity grid if there is no proper coordination between the companies. Should coordination not occur, it becomes very difficult to prevent grid congestion and results in the following undesired consequences:

The first consequence is the same consequence as when using uniform pricing: multiple companies ask for energy at the same time resulting in a simultaneous high demand, thus not preventing grid congestion. Without coordination, it may occur that even without grid congestion, the available resources are still inefficiently used. At one moment, the supply of renewable energy may be higher than the demand, whereas the roles are reversed at the next moment. These reasons make it critical to have some form of coordination between the companies if preventing grid congestion is the main goal.



(a) Preference profile and allocated capacity for company com1.



(b) Resulting desired flexibility profile for company com1.

Figure 3.2: Example of flexibility usage for company com1.

3.3 Preference Profiles

In order to coordinate the usage of these resources, day-ahead planning can be used. Day-ahead planning can help predict power usage and help companies to adapt their energy usage depending on the forecasts. In order to create a day-ahead planning, a preference profile is needed. This preference profile is a power profile that contains the amount of desired power of that company on an hourly basis. The companies are able to indicate how much they allow to be deviated from their preference profile in case of congestion. These extra bounds are called the preference bounds and comprise an upper bound ($P_{ub,h}(t)$) and a lower bound ($P_{lb,h}(t)$). An example of a preference profile is shown in Figure 3.3, where $P_{p,h}(t)$ is the preference profile of holon h.

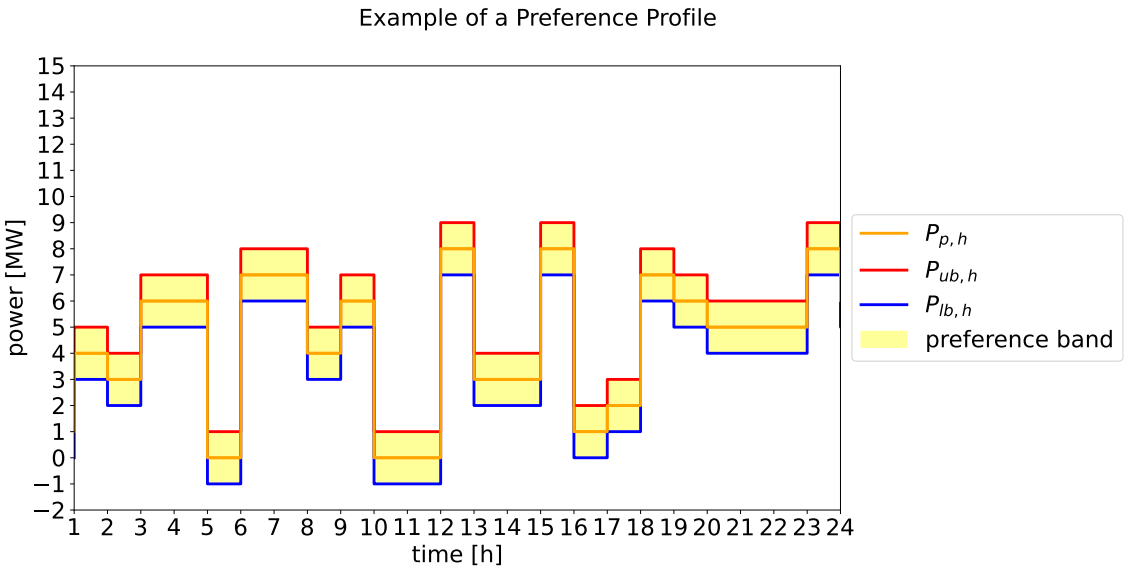


Figure 3.3: Example of a preference profile with a preference band. $P_{p,h}(t)$ is the preference profile, $P_{ub,h}(t)$ is the upper bound and $P_{lb,h}(t)$ is the lower bound.

Chapter 4

Methodology

This chapter starts by describing in detail how the holarchy concept is implemented and the distribution of the holons within the holarchy. Then, the resource allocation algorithm is described in detail, which consists of two different algorithms. The first algorithm is called the *Curtailment and Reallocation* algorithm. The goal of that algorithm is to prevent grid congestion. The second algorithm is the *Capacity Trading* algorithm, in which the flexibility sources trade capacity without introducing extra strain on the electrical grid to stay within the bounds of the flexibility sources.

4.1 Holarchy

In order to model the electrical grid as a holarchy, the holons need to be defined. In this research, every prosumer is a modelled as a holon. These holons are connected to a holon at a higher level in the holarchy (in the same manner as shown in Figure 2.1). This super-holon corresponds to the substation, since the substation is the link between the underlying connections and the rest of the electrical grid. The contrast with the setup in Figure 2.1 is the lack of holons at the feeder level. The holons at the lower level are able to communicate with each other in a distributed manner.

4.2 Resource Allocation

As mentioned in Chapter 3, the main goal is to prevent grid congestion by taking the flexibility sources into account. The implementation splits this goal into two separate goals: preventing grid congestion and staying within the limits of the flexibility sources. These goals are tackled separately. The process of staying within the limits of the electrical grid is described in Subsection 4.2.1, whereas Subsection 4.2.2 explains the process of staying within the limits of the flexibility sources for each holon.

A global overview of the system in its entirety is shown in Figure 4.1. The input for each part is shown in between the steps, whereas the output is shown at the end.

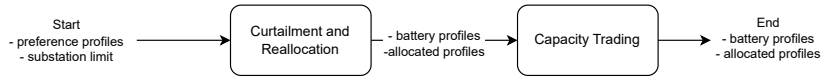


Figure 4.1: Global overview of the resource allocation algorithms.

4.2.1 Curtailment and Reallocation

To accomplish the goal of preventing grid congestion and providing as much capacity as possible for the connected holons, this part of the algorithm consists of two stages: curtailment and reallocation. A global overview of this algorithm is shown in Figure 4.2. Note that the following steps of this algorithm are executed over the course of T time steps.

Curtailment

The first stage aims to curtail the holons at moments of congestion. Congestion can occur if there is too much supply or too much feed-in. Both of these cases need to be taken into account when preventing congestion. Repeating the equations from the problem analysis, grid congestion is expressed using the following equations:

$$P_{tot}(t) = \sum_{h=1}^{H} P_{a,h}(t) \tag{4.1}$$

$$\exists t \in T \text{ where } |P_{tot}(t)| > P_s(t) \tag{4.2}$$

In (4.1), $P_{tot}(t)$ is the aggregated amount of power of every holon h combined at time t. $P_{a,h}$ is the amount of power used by holon h at time t. In (4.2), $P_s(t)$ is the amount of available power to the industrial district at time t. Grid congestion occurs if (4.2) holds. If congestion occurs, the next step is to determine the power violation (the amount by which the limit of the grid is exceeded). This violation is calculated using the following equation:

$$P_v(t) = \begin{cases} sgn(P_{tot}(t)) \cdot (|P_s(t)| - |P_{tot}(t)|), & \text{if } |P_{tot}(t)| > |P_s(t)| \\ 0, & \text{otherwise} \end{cases} \tag{4.3}$$

In (4.3), $P_v(t)$ is the power violation amount in the grid. The sgn function is the sign function, which determines the sign of $P_v(t)$ depending on the sign of $P_{tot}(t)$. This determines if the violation occurs in feed-in or supply. $|P_s(t)| - |P_{tot}(t)|$ determines the magnitude of the violation.

In case of congestion, the connected holons have to be curtailed. This curtailment is done in three steps:

1. The holons are sorted in ascending order using a priority (which is presented in step **Curtailment 1**);
2. The holons are curtailed based on the aforementioned order until their lower bound;
3. The holons are curtailed proportionally with respect to priority if grid congestion still occurs.

Curtailment 1: Priority Calculation and Sorting Priorities are assigned to the holons in order to determine which holons to curtail and are based on the amount that a holon has already been curtailed. In order to give holons who have been curtailed a higher priority compared to holons who have been curtailed less, the euclidean norm is used:

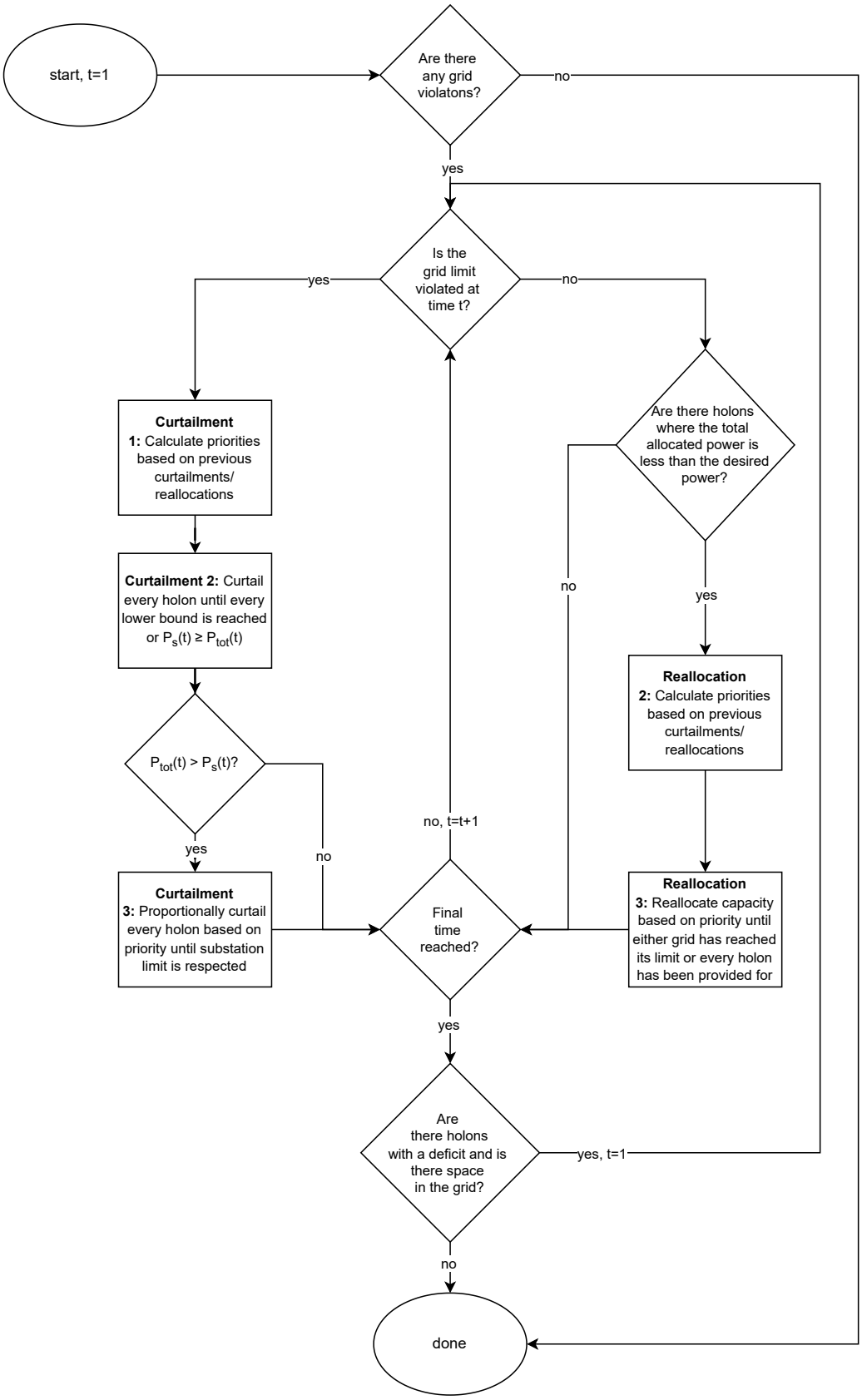


Figure 4.2: Flow diagram of the Curtailment and Reallocation algorithm.

$$\alpha_h = \sqrt{\sum_{t=1}^{T} \left(P_{a,h}(t) - P_{p,h}(t)\right)^2} \tag{4.4}$$

In (4.4), α_h is the priority of holon h, $P_{p,h}(t)$ is the preferred amount of power for node h at time t. This priority is taken over the course of the entire day divided into T time steps. The priority increases along with the difference between the allocated and preferred amount of power. A sorted list is created based on this priority, starting with holons with the lowest priority (since they have been curtailed the least). This list determines which holons are curtailed first. If each holon has a priority of 0, the order is randomly determined.

Curtailment 2: Curtailment Until Bounds The goal of this step is to solve grid congestion without the holons being curtailed beyond their preference bounds (depending on P_v). If grid congestion occurs at time t, the holons are curtailed based on the priorities calculated in (4.4). The following equation is used to determine the amount that needs to be curtailed:

$$\Delta_h(t) = \begin{cases} \max\left(P_v(t), P_{lb,h}(t) - P_{a,h}(t)\right), & \text{if } P_v(t) < 0 \\ \min\left(P_v(t), P_{ub,h}(t) - P_{a,h}(t)\right), & \text{otherwise} \end{cases} \tag{4.5}$$

(4.5) is split into two cases, since different bounds need to be used depending on the violation. If the violation is negative (holons need to reduce their energy usage), the lower bound ($P_{lb,h}(t)$) of the preference bound needs to be taken into account. In case of a positive violation (holons need to increase their energy usage), the upper bound ($P_{ub,h}(t)$) is used.

Next, we use $\Delta_h(t)$ to calculate the amount of allocated power of that holon:

$$P_{a,h}(t) \leftarrow P_{a,h}(t) + \Delta_h(t) \tag{4.6}$$

Not only the allocated power in (4.6) should be updated, the priority should also be updated according to (4.4). Furthermore, a deficit variable (σ_h^k) is introduced. This deficit variable keeps track how much a holon has been curtailed, to ensure that the curtailment occurs as distributed as possible. The deficit is updated with every change in the power profile:

$$\sigma_h^{k+1} = \sigma_h^k + \Delta_h(t) \tag{4.7}$$

Since there is no deficit before the algorithm is executed, $\sigma_h^{k=0} = 0$.

Curtailment 3: Proportional Curtailment If every holon has reached its lower bound regarding allocated capacity and grid congestion still occurs, extra measures need to be taken. The next curtailment step is done proportionally with respect to priority to ensure less curtailment for holons who have been curtailed the most: the holons with a higher priority are proportionally curtailed less than holons with a lower priority. To calculate how much each holon needs to be curtailed, the factor ($f_h(t)$) needs to be calculated. In order to calculate this factor, the total priority (α_{tot}) needs to be known, as well

as the highest priority value (α_{max}). This total priority is the sum of all priorities of the connected holons (H):

$$\alpha_{tot} = \sum_{h=1}^{H} \alpha_h \tag{4.8}$$

$$\alpha_{max} = \max(\alpha_h) \tag{4.9}$$

With α_{tot} calculated in (4.8) and α_{max} calculated in (4.9), the factor is calculated using the following equation:

$$f_h(t) = \begin{cases} \frac{1}{H}, & \text{if } \alpha_{tot} = 0 \\ \frac{\alpha_{max} - \alpha_h}{\alpha_{tot}}, & \text{otherwise} \end{cases} \tag{4.10}$$

Using the value of the factor calculated in (4.10), the new curtailment amount is calculated:

$$\Delta_h(t) = f_h(t) \cdot P_v(t) \quad \forall h \in H \tag{4.11}$$

(4.11) calculates the amount of allocated power every holon needs to be updated by. This update is again performed using (4.6) and (4.7).

Reallocation

The second stage consists of reallocating capacity to the holons. This stage counteracts the imbalance between the preference and allocated profile caused by the curtailment stage. Reallocation can only occur when there is available capacity in the grid, which is the opposite of the congestion evaluation equation in (4.2):

$$\exists t \in T \text{ where } |P_{tot}(t)| < |P_s(t)| \tag{4.12}$$

When the amount of available capacity is the same as the total amount of allocated power at time t, the grid is fully utilized from the point of view from the substation: No curtailment is necessary, but reallocation is not possible either.

When (4.12) holds, reallocation can be performed if there are holons with a deficit. I.e.: $\exists h \in H$ where $\sigma_h^k \neq 0$. When a holon has a deficit, the reallocation is divided in two steps:

1. The holons are sorted in descending order using a priority.
2. The holons have their capacity reallocated.

Reallocation 1: Priority Calculation The goal of this step is to determine the order in which the nodes receive reallocated capacity. The priorities are again calculated using (4.4). In contrast to the curtailment order, the reallocation order is in reverse. Since the holons with the highest priority have been curtailed the most, they are to be accounted for first.

Reallocation 2: Capacity Reallocation With the order of allocation determined, the holons can have some capacity reallocated. This step determines the amount that is reallocated. In order to determine the amount of capacity that needs to be reallocated, two things need to be evaluated: the upper bound of the preference band and the amount of remaining capacity in the grid:

$$\Delta_h(t) = \begin{cases} \max\left(P_v(t), P_{lb,h}(t) - P_{a,h}(t), -\sigma_h^k\right), & \text{if } \sigma_h^k > 0 \\ \min\left(P_v(t), P_{ub,h}(t) - P_{a,h}(t), -\sigma_h^k\right), & \text{otherwise} \end{cases} \tag{4.13}$$

$$P_{a,h}(t) = P_{a,h}(t) + \Delta_h(t) \tag{4.14}$$

The resulting amount at (4.13) is used to update holon h according to (4.14). This step is repeated until either every holon has fulfilled its deficit, every holon has reached its respective $P_{ub,h}(t)$ or $P_s(t) = 0$.

This process of curtailment and reallocation is performed sequentially for every time step. When the end of an iteration is reached (i.e. when $t = T$), the resulting profiles are evaluated for deficits. This is necessary in order to determine if the Curtailment and Reallocation algorithm is finished. If at least one holon ($h \in H$ has a deficit $\sigma_h^k \neq 0$), the profiles with a deficit need to be evaluated whether that deficit can be reduced/mitigated. In order to evaluate that possibility, there must be some time t where the following two conditions must hold:

- There must be available capacity in the grid, i.e., $|P_{tot}(t)| < |P_s(t)|$.
- For a holon with a deficit, the corresponding bound has not been reached yet. This entails that $P_{a,h}(t) < P_{ub,h}(t)$ for $\sigma > 0$ and $P_{a,h}(t) > P_{lb,h}(t)$ for $\sigma < 0$.

If the aforementioned condition holds (i.e. there still are possibilities to minimize deficits), t is set to 1 and the Curtailment and Reallocation algorithm starts over to allow for this extra reallocation. To avoid possible deadlocks, the algorithm has been set to run a maximum of n times. If those iterations have occurred and there still is no feasible profile with the submitted preference profiles, the holons can either opt to create new preference profiles or accept this new power profile with a deficit.

Pseudocode regarding the curtailment is presented in Algorithm 1, reallocation is presented in Algorithm 2 and the overview of the entire algorithm is shown in Algorithm 3.

Algorithm 1 Pseudocode for Curtailment.

Function Curtailment
$\alpha_h = \sqrt{\sum_{t=1}^{T} \left(P_{a,h}(t) - P_{p,h}(t)\right)^2} \quad \forall h \in H$
sort α_h, lowest first
for h in priority list **do**
    if $P_v(t) < 0$ **then** ▷ Calculate amount to curtail
        $\Delta_h(t) = \max(P_v(t), P_{lb,h}(t) - P_{a,h}(t))$
    else
        $\Delta_h(t) = \min(P_v(t), P_{ub,h}(t) - P_{a,h}(t))$
    end if
    update(h,$\Delta_h(t)$) ▷ Update holon h
end for
if $P_v(t) \neq 0$ **then**
    for $h = 1 : H$ **do** ▷ Proportionally curtail every holon h
        $\alpha_{tot} = \sum_{h=1}^{H} \alpha_h$
        $f_h(t) = 1 - \frac{\alpha_h}{\alpha_{tot}}$
        $\Delta_h(t) = f_h(t) \cdot P_v(t)$
        update(h,$\Delta_h(t)$)
    end for
end if

Algorithm 2 Pseudocode for Reallocation.

Function Reallocation
$\alpha_h = \sqrt{\sum_{t=1}^{T} \left(P_{a,h}(t) - P_{p,h}(t)\right)^2} \quad \forall h \in H$ ▷ Calculate priorities
sort α_h, highest first
for h in priority list **do**
    if $\sigma_h^k > 0$ **then** ▷ Calculate amount to reallocate
        $\Delta_h(t) = \max(P_v(t), P_{lb,h}(t) - P_{a,h}(t), -\sigma_h^k)$
    else
        $\Delta_h(t) = \min(P_v(t), P_{ub,h}(t) - P_{a,h}(t), -\sigma_h^k)$
    end if
    update(h,Δ_h)
end for

Algorithm 3 Pseudocode for the Curtailment and Reallocation algorithm.

$leaveCounter = 0$
$P_{tot}(t) = \sum_{h=1}^{H} P_{a,h}(t)$
if $P_{tot} < P_s(t) \quad \forall t \in T$ **then**
    done ▷ No curtailment and reallocation necessary
end if
for $t = 1 : T$ **do**
    $P_v(t) = \frac{P_{tot}(t)}{|P_{tot}(t)|} \cdot (P_s(t) - |P_{tot}(t)|)$
    if $|P_{tot}(t)| > P_s(t)$ **then** ▷ Curtailment necessary
        Curtailment(t)
    else if $|P_{tot}| < P_s(t)$ **then** ▷ Reallocation possible
        Reallocation(t)
    end if
    if $t = T$ **then**
        if $\exists t \in T$ where $|P_{tot}(t)| < P_s(t)$ **then**:
            if Possible to reallocate **then**
                $t = 1$
                $leaveCounter = leaveCounter + 1$
                if $leaveCounter > n$ **then** ▷ exit when an arbitrary amount of iterations, n, has been reached
                    return ▷ Not all deficits can be accounted for
                end if
            end if
        end if
    end if
end for

4.2.2 Capacity Trading

The previous section ensures that the the limit of the substation is always respected and the holons have been curtailed/reallocated within their preference bounds as much as possible. When deviated from the preference profile however, the flexibility sources need to be utilized. These sources have their bounds as mentioned in Chapter 3. The issue is that the Curtailment and Reallocation algorithm does not take the limits of these flexibility sources into account. A violation in these flexibility bounds requires the trading of capacity with other holons. Figure 4.3 provides a flow diagram of this negotiation process between two holons.

In order to evaluate if trading is needed, the state of charge of the flexibility sources needs to be known. This is calculated by using the $P_{a,h}(t)$ provided by the previous section. Since the state of charge is the result of the power usage of the previous interval, the equation becomes:

$$SoC_h(t+1) = SoC_h(t) + P_{a,h}(t) - P_{p,h}(t) \tag{4.15}$$

In (4.15), the assumption is made that holons only utilize the flexibility sources when the allocated profile deviates from the preference profile. To evaluate a possible violation regarding $SoC_h(t)$, the constraints have to be known. As mentioned in Section 3.2, $SoC_h(t)$ has to stay in between 0 and ζ_{max} at all times:

$$0 \leq SoC_h(t) \leq \zeta_{max} \quad \forall t \in T \tag{4.16}$$

If an infraction occurs according to (4.16), capacity needs to be traded with other holons. When a holon experiences an infraction, it first determines the time and magnitude of the infraction. This is necessary for the next step: creating a trade window.

Step 1: Creating Trade Window

At first, the trade window is determined by holon 1. In this window, holon 1 should be able to receive more or less capacity to resolve for that infraction without resulting in other infractions. The time of the infraction is set to the latest interval (t_{end}) of the window, since the holon should receive or provide the capacity before that moment. To determine the earliest interval (t_{begin}) of the window, it needs to be determined on which side of the bounds the infraction occurs. The earliest interval is either set to 1 or:

- The last occurrence of $SoC_{h1}(t) = \zeta_{max}$ on the interval $[1, t_{end})$, if $SoC_{h1}(t_{end}) < 0$
- The last occurrence of $SoC_{h1}(t) = 0$ on the interval $[1, t_{end})$, if $SoC_{h1}(t_{end}) > \zeta_{max}$

An overview of the state of charge, $SoC_{h1}(t)$, as well as the flexibility bounds and the interval boundaries (t_{begin} and t_{end}) are shown in Figure 4.4. The infraction amount, ϵ is set to $SoC_{h1}(t = t_{end})$. The trade window boundaries, t_{begin} and t_{end}, along with the ϵ, are communicated to the other holon (holon 2) for the next steps in capacity trading. This way, $hol1$ does not have to send privacy sensitive information.

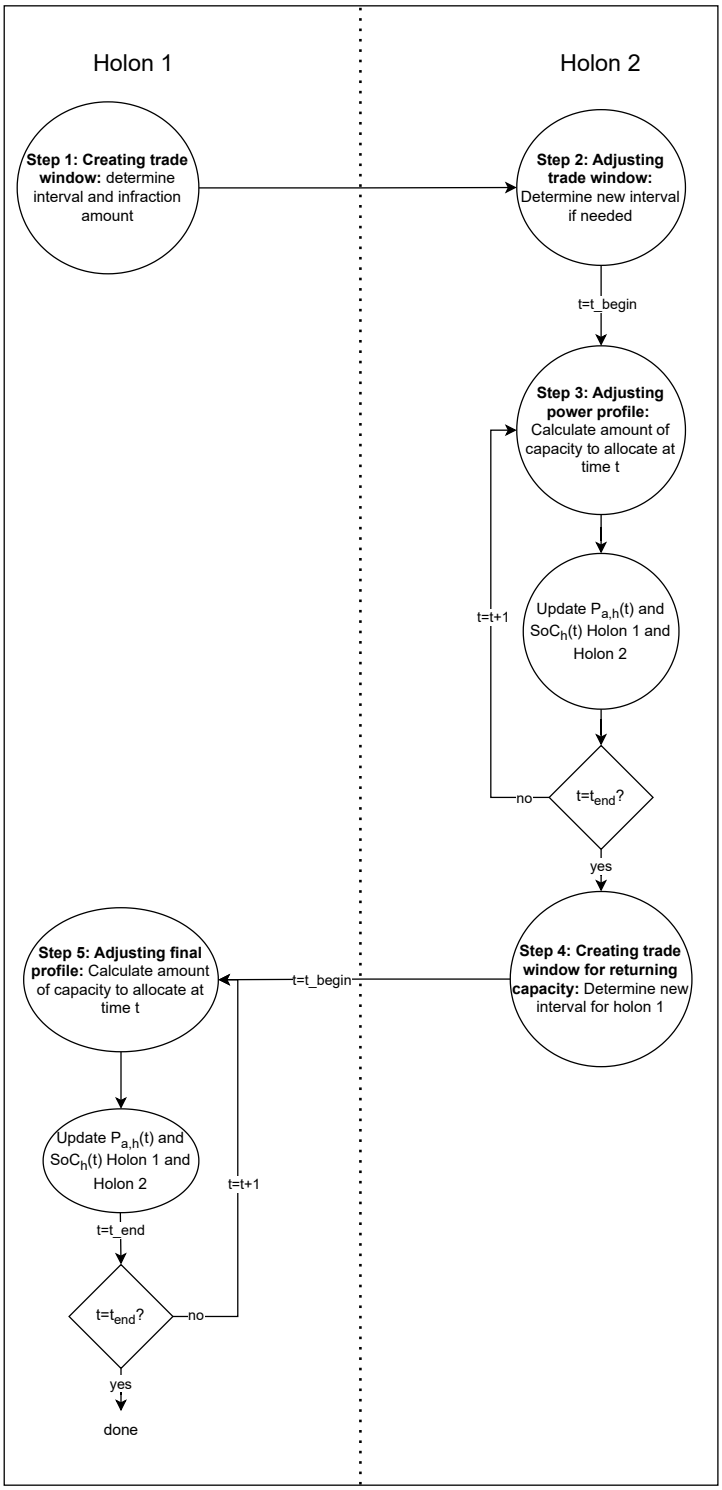


Figure 4.3: Flow diagram of the Capacity Trading algorithm between holons.

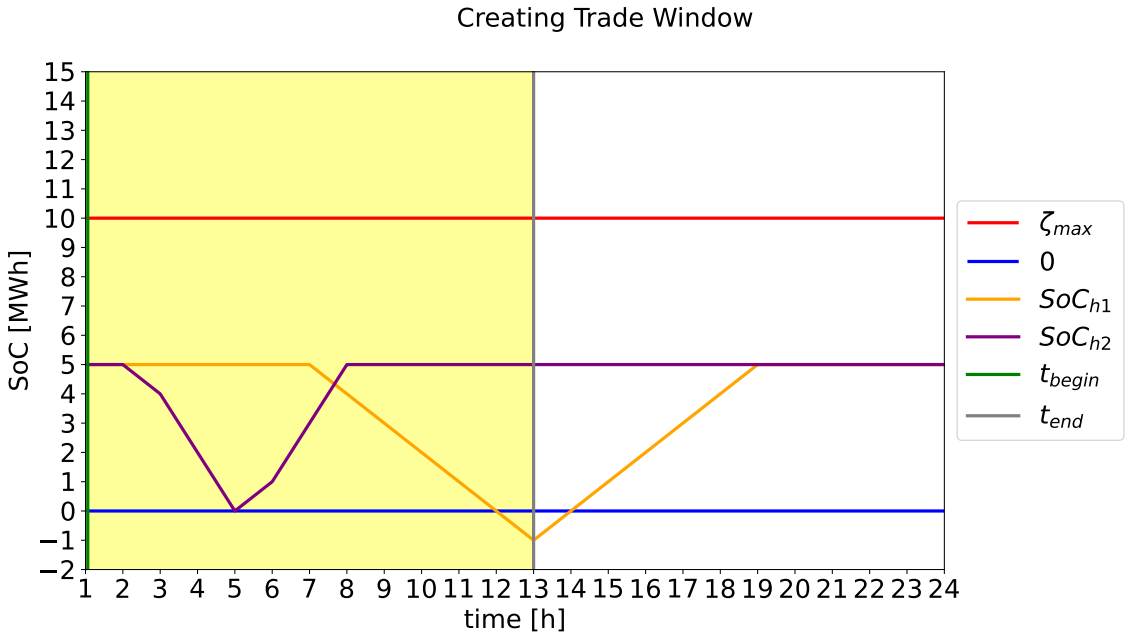


Figure 4.4: Creating Trade Window, where $t_{begin} = 1$, t_{end} is set to the moment the infraction occurs and $\epsilon = SoC_{h1}(t = t_{end})$.

Step 2: Adjusting Trade Window

When $hol2$ receives the trade window, $hol2$ might need to adjust the received trade window. The trading immediately fails between the two holons if $hol2$ does not have a sufficient $SoC_{h2}(t)$ at $t = t_{end}$ to compensate for the infraction (for example when $SoC_{h1}(t_{end}) + SoC_{h2}(t_{end}) < 0$). If the infraction can be compensated for, the trade window might be adjusted. In this adjusted window, t_{end} is kept the same, but t_{begin} is changed according to the availability of $hol2$:

- The last occurrence of $SoC_{h2}(t) \leq 0$ on the interval $[t_{begin}, t_{end})$ if $\epsilon < 0$.
- The last occurrence of $SoC_{h2}(t) \geq \zeta_{max}$ on the interval $[t_{begin}, t_{end})$ if $\epsilon > 0$.
- t_{begin} if there are none of these occurrences.

An overview of the adjusted trade window is shown in Figure 4.5. This new interval is where the trading occurs by adjusting the power profiles.

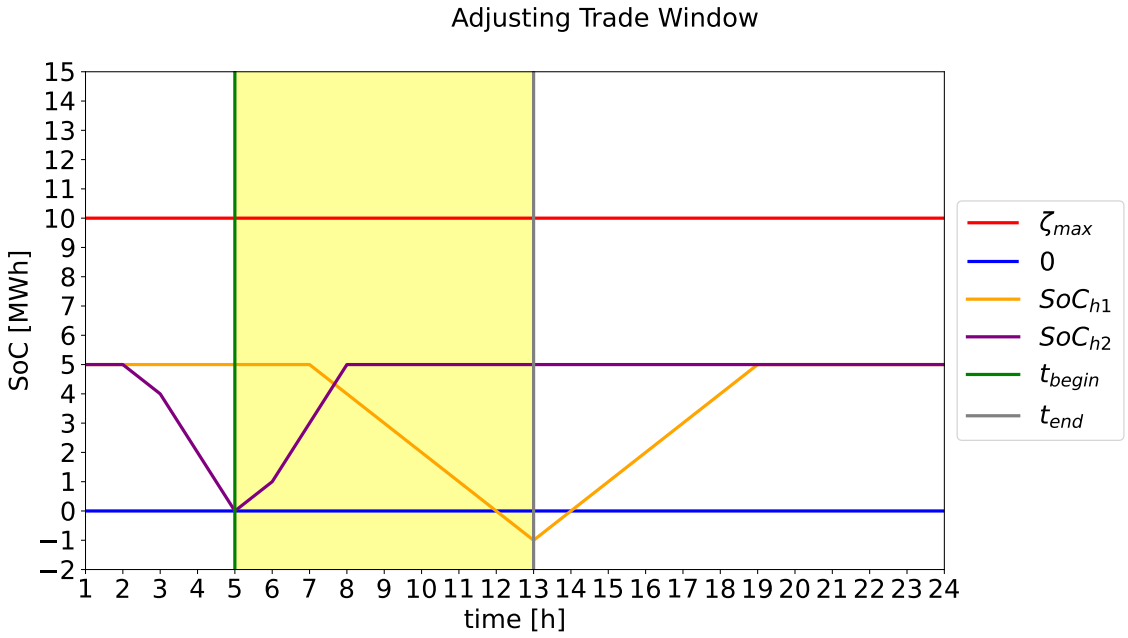


Figure 4.5: Adjusting Trade Window, where t_{begin} is now set to the last moment where $SoC_{h2} \leq 0$.

Step 3: Adjusting Power Profile

Now that the new window has been determined, the infraction can be mitigated from the point of view of $hol2$. This happens on a sequential basis: it starts with $t = t_{begin}$ and ends at $t = t_{end}$. At every time step, the amount to trade is calculated, depending on whether $\epsilon > 0$ ($hol2$ needs to receive capacity) or $\epsilon < 0$ ($hol2$ needs to provide capacity):

$$\Delta_h(t) = \begin{cases} \max\left(\epsilon, P_{lb,hol2}(t) - P_{a,hol2}(t),\right. \\ \quad \left. 0 - \min\left(SoC[t+1, t_{end}+1]\right)\right), & \text{if } \epsilon < 0 \\ \min\left(\epsilon, P_{ub,hol2}(t) - P_{a,hol2}(t),\right. \\ \quad \left. \max\left(0, \zeta_{max} - \max\left(SoC[t+1, t_{end}+1]\right)\right)\right), & \text{if } \epsilon > 0 \end{cases} \tag{4.17}$$

$$\epsilon \leftarrow \epsilon - \Delta_h(t) \tag{4.18}$$

After (4.17), a change window is updated, that records the time and capacity to allocate and ϵ is updated according to (4.18). This step is repeated until either $t = t_{end}$ or $\epsilon = 0$. If $\epsilon \neq 0$ at $t = t_{end}$, the holons are not able to fully mitigate that infraction and the trade fails, allowing for $hol1$ to trade with other holons. If $\epsilon = 0$, the problems of the first holon have been solved and the process reverses, where $hol1$ needs to trade back the same amount of capacity to $hol2$.

An overview of this step is shown in Figure 4.6, where 1 MW of capacity is provided by $hol2$ at interval $t = 5$, such that the $SoC_{h1}(t)$ and $SoC_{h2}(t)$ change between $t = 5$ and $t = 6$. With this step, $\epsilon = 0$, but the capacity provided by $hol2$ needs to be traded back by $hol1$.

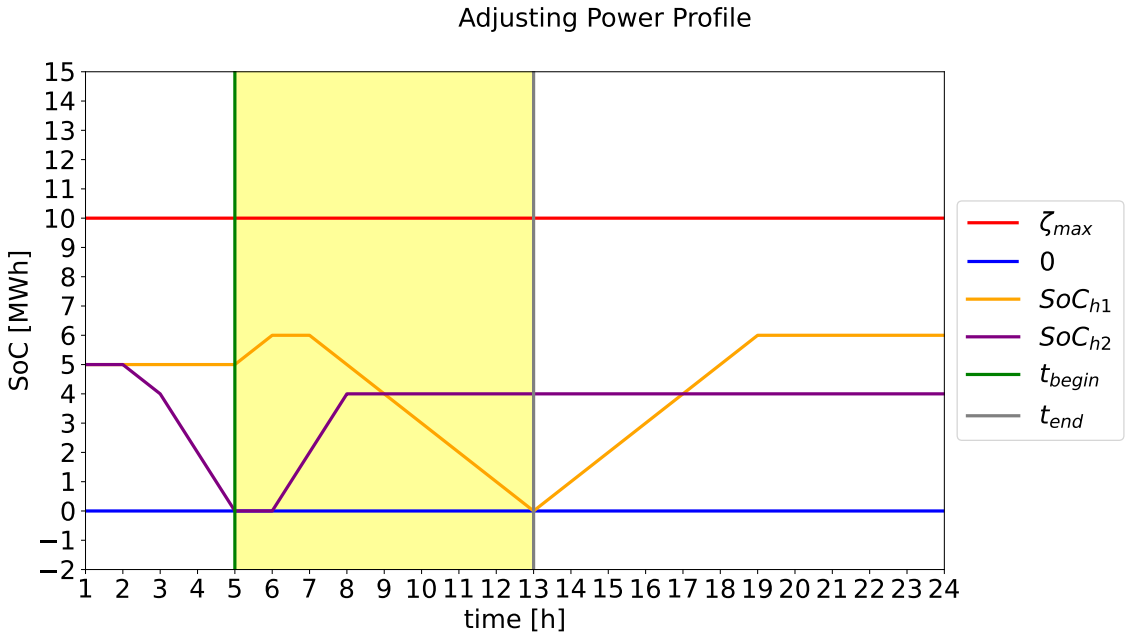


Figure 4.6: Adjusting Power Profile, where $hol1$ and $hol2$ have traded 1 MWh of capacity between $t = 5$ and $t = 6$.

Step 4: Creating Trade Window for Returning Capacity

Since the trading thus far has resulted in $hol2$ providing capacity for $hol1$, there is an imbalance for both holons that needs to be accounted for ($hol1$ has more capacity and $hol2$ has less capacity than it started with). Mitigating that imbalance is the goal of the next two steps. At first, $hol2$ creates a new trade window. This trade window allows for the second part of trading to commence. This new trade window starts at the moment the infraction occurred, since $hol1$ is only able to trade back the capacity after the infraction. Hence, the new t_{begin} is set to the the latest interval (t_{end}) of the previous trade window. The latest interval of the new trade window, t_{end}, is set to one of the following:

- The last occurrence of $SoC_{h2}(t) \geq \zeta_{max}$ on the interval $[t_{begin}, T]$ if infraction < 0.
- The last occurrence of $SoC_{h2}(t) \leq 0$ on the interval $[t_{begin}, T]$ if $\epsilon > \zeta_{max}$.
- T if there are none of the aforementioned occurrences.

These new values for t_{begin} and t_{end} are communicated back to $hol1$.

The new trading window created for returning capacity towards $hol2$ is shown in Figure 4.7. This new trade window is then being communicated back to $hol1$, along with the changes in power usage from the first step of trading, which is then used to update both $hol1$ and $hol2$ according to (4.6).

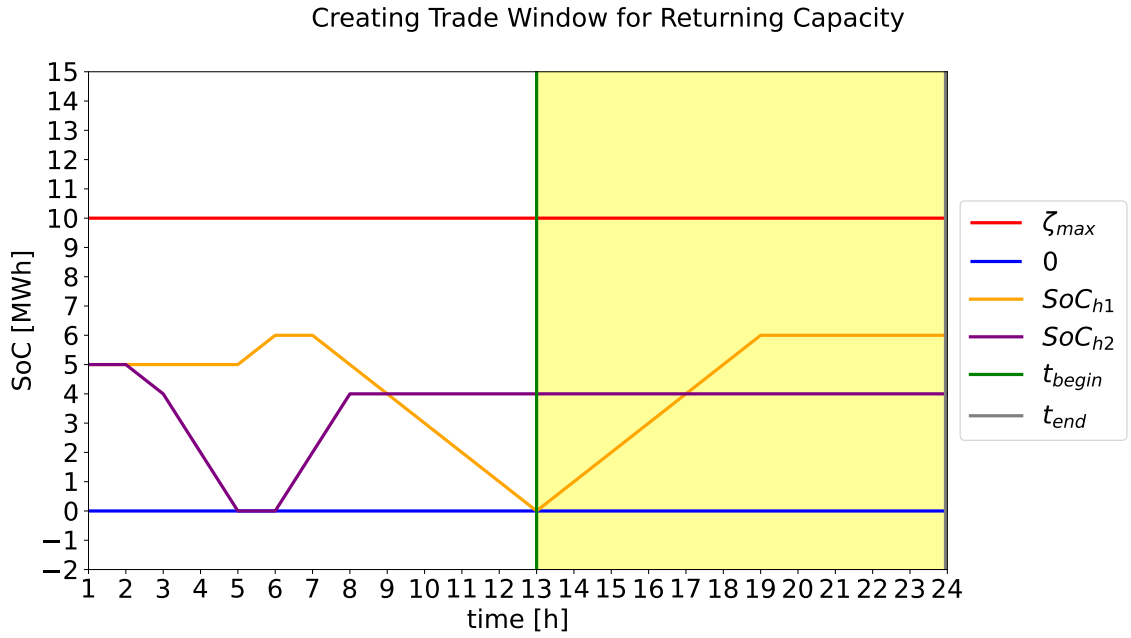


Figure 4.7: Creating Trade Window for Returning Capacity, where $t_{begin} = 13$ and $t_{end} = 24$.

Step 5: Adjusting Final Power Profile

When $hol1$ has received this new trade window from $hol2$, $hol1$ executes the same steps as described in the **Adjusting Trade Window** and **Adjusting Power Profile** steps: it first adjusts the trade window if needed, after which the power profile is adjusted to mitigate the infraction. If the resulting profile is energetically feasible, and does not exceed the bounds of its own flexibility sources, the final power profile is communicated to $hol2$, after which $hol1$ and $hol2$ are updated with these new power profiles according to (4.6). Once the holons are updated, the process of trading capacity is done for that specific violation.

An overview of the final SoC graphs for both holons is shown in Figure 4.8. With this step, the trading has come to an end. If the profile is not energetically feasible and the holons are not able to trade with each other, the trade fails and the first node needs to trade with a different holon. In the case that no holon is able to trade enough capacity to fully mitigate the violation, the holon with the violation can opt to trade with multiple holons with a reduced ϵ. This can still mitigate the infraction, but it does require multiple trades. The infraction to trade with is reduced if trading was unsuccessful with every holon. If that is the case, the reduction is stepwise. Should a new trade with this amount be successful, the new value for ϵ is calculated, which then results in a new trading cycle. This can result in a larger number of iterations when trading, but it does take data privacy into account. When the amount to be traded reaches 0 because of this stepwise reduction, the trade permanently fails.

Pseudocode regarding the Capacity Trading algorithm is shown in Algorithms 4-8.

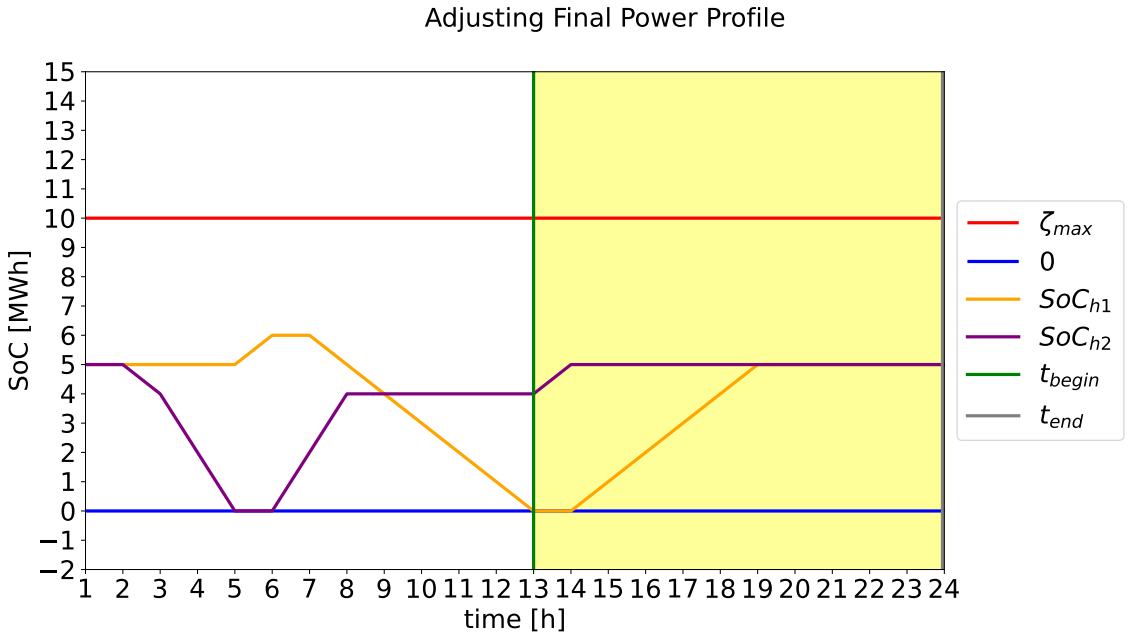


Figure 4.8: Adjusting Final Power Profile, where $hol1$ has traded back 1 MWh to $hol2$ between $t = 13$ and $t = 14$.

Algorithm 4 Pseudocode for step 1: Creating Trade Window function.

Function Creating_Trade_Window()
t_{end} = first occurrence of $SoC_h(t) < 0$ or $SoC_h(t) > \zeta_{max}$
if $SoC(t_{end}) < 0$ **then**
    if $\exists t \in 1 : t_{end}$ where $SoC_h(t) = \zeta_{max}$ **then**
        t_{begin} = last occurrence of $t \in [1, t_{end}]$ where $SoC_h(t) = \zeta_{max}$
    else
        $t_{begin} = 1$
    end if
    $\epsilon = SoC(t_{end})$
else
    if $\exists t \in 1 : t_{end}$ where $SoC_h(t) = 0$ **then**
        t_{begin} = last occurrence of $t \in [1, t_{end}]$ where $SoC_h(t) = 0$
    else
        $t_{begin} = 1$
    end if
    $\epsilon = SoC(t_{end}) - \zeta_{max}$ ▷ Determine ϵ
end if
return t_{begin}, t_{end}, ϵ

Algorithm 5 Pseudocode for step 2: Adjusting Trade Window function.

Function Adjust_Trade_Window(t_{begin}, t_{end}, $infraction$)
if $\epsilon < 0$ **then**
    if $\exists t \in t_{begin} : t_{end}$ where $SoC_h(t) \leq 0$ **then**
        t_{begin} = last occurrence of t where $SoC_h(t) \leq 0$
    end if
else ▷ $\epsilon > 0$
    if $\exists t \in t_{begin} : t_{end}$ where $SoC_h(t) \geq \zeta_{max}$ **then**
        t_{begin} = last occurrence of t where $SoC_h(t) \geq \zeta_{max}$
    end if
end if
return t_{begin}, t_{end}

Algorithm 6 Pseudocode for step 3: Adjusting Power Profile function.

Function Adjust_Power_Profile(t_{begin}, t_{end}, $infraction$)
if $\epsilon < 0$ **then**
    for $t = t_{begin} : t_{end}$ **do**
        $\Delta_h(t) = \max(\epsilon, P_{lb,h}(t) - P_{a,h}(t), -\min(SoC[t+1, t_{end}+1]))$
        update$(h, \Delta_h(t))$
        $\epsilon \leftarrow \epsilon - \Delta_h(t)$
        $change\_profile(t) = \Delta_h(t)$
    end for
else ▷ $\epsilon > 0$
    for $t = t_{begin} : t_{end}$ **do**
        $\Delta_h(t) = \min(\epsilon, P_{ub,h}(t) - P_{a,h}(t), \max(0, \zeta_{max} - \max(SoC[t+1, t_{end}+1]))$
        update$(h, \Delta_h(t))$
        $\epsilon \leftarrow \epsilon - \Delta_h(t)$
        $change\_profile(t) = \Delta_h(t)$
    end for
end if
if $\epsilon \neq 0$ **then** ▷ If trade fails
    return False
end if
return $change\_profile$

Algorithm 7 Pseudocode for step 4: Creating Trade Window for Returning Capacity function.

Function Creating_Return_Window(t_{begin}, t_{end})
if $\epsilon < 0$ **then**
    $t_{begin} = t_{end}$
    $t_{end} = T$
    if $\exists t \in t_{begin} : t_{end}$ where $SoC_h(t) < 0$ **then**
        t_{end} = first occurrence where$SoC_h(t) < 0$
    end if
    if $\exists t \in t_{begin} : t_{end}$ where $SoC_h(t) \geq \zeta_{max}$ **then**
        t_{begin} = last occurrence where$SoC_h(t) \geq \zeta_{max}$
    end if
else ▷ $\epsilon > 0$
    $t_{begin} = t_{end}$
    $t_{end} = T$
    if $\exists t \in t_{begin} : t_{end}$ where $SoC_h(t) > \zeta_{max}$ **then**
        t_{end} = first occurrence where$SoC_h(t) > \zeta_{max}$
    end if
    if $\exists t \in t_{begin} : t_{end}$ where $SoC_h(t) \leq 0$ **then**
        t_{begin} = last occurrence where$SoC_h(t) \leq 0$
    end if
end if

Algorithm 8 Pseudocode of Capacity Trading Algorithm.

for $h = 1 : H$ **do**
  for $t = 1 : T$ **do** ▷ Calculate SoC_h at every t for each h
    $SoC_h(t+1) = SoC_h(t) + P_{a,h}(t) - P_{p,h}(t)$
  end for
end for
for $hol1 = 1 : H$ **do**
  if $\neg(0 \leq SoC_{h1}(t) \leq \zeta_{max} \quad \forall t \in T)$ **then**
    t_{begin},t_{end},ϵ = $hol1$.Creating_Trade_Window()
    $r = 1$
    do
      for $hol2 \in H$ **do**
        Send t_{begin}, t_{end} and ϵ from $hol1$ to $hol2$
        t_{begin},t_{end} = $hol2$.Adjusting_Trade_Window(t_{begin}, t_{end}, ϵ)
        $change\_profile$ = $hol2$.Adjusting_Power_Profile(t_{begin},t_{end}, ϵ)
        t_{begin},t_{end} = $hol2$.Creating_Return_Window(t_{begin},t_{end})
        Send t_{begin}, t_{end} and $change\_profile$ from $hol2$ to $hol1$
        t_{begin},t_{end} = $hol1$.Adjusting_Trade_Window(t_{begin},t_{end}, ϵ)
        $change\_profile$ = $hol1$.Adjusting_Power_Profile(t_{begin},t_{end}, ϵ)
        if trade is successful **then**
          Send $change\_profile$ from $hol1$ to $hol2$
        end if
        t_{begin},t_{end},ϵ = $hol1$.Creating_Trade_Window()
      end for
      if no trade successful **then**
        $r = r - 0.05$
        if $r = 0$ **then**
          trade permanently fails
        end if
        $\epsilon\epsilon\epsilon \cdot r$
      else
        $r = 1$
      end if
    while $\epsilon \neq 0$
  end if
end for

Chapter 5

Evaluation

This chapter details the evaluation process of the Curtailment and Reallocation shown in Section 4.2.1 and the Capacity Trading algorithms shown in Section 4.2.2. We begin with a description of the environment, followed by the setup, which comprises two different scenarios: a small-scale scenario and an existing industrial complex in the Netherlands. After the setup has been described, the metrics used to evaluate the algorithms are presented, followed by the results of the scenarios. These results are then compared to an implementation of the profile steering algorithm [19] to compare it against the state of the art.

5.1 Simulations

5.1.1 Environment

The algorithms are implemented and evaluated in a self-developed Python environment. In this simulation environment, two design choices are made to simplify the system, but still result in viable results. The first design choice is that each holon has a direct communication link with all other holons. This does not have to be the case in general, but the algorithm does not change the functionality of the algorithm. The second point is an addition of an error margin. This error margin is used to deal with rounding errors during computations [39].

The algorithm is executed on hardware with the following specifications:

- CPU: Intel Core i7-8750H
- Memory: 16 GB DDR4

5.1.2 Setup

This section details the characteristics of each scenario. It starts with a small-scale scenario, which is used to evaluate the designed algorithms in a controlled environment that can be utilized to search for edge cases. The second scenario comprises an existing industrial complex to evaluate the functioning of the algorithm in realistic scenarios. The end result for every scenario is also compared to the profile steering algorithm [19], to allow for a direct comparison with a state-of-the-art alternative for congestion management.

Profile Steering

In order to provide a proper comparison between the developed algorithms and profile steering, one modification needs to be made to the profile steering algorithm from [19]. This change is when the algorithm is finished. Whereas the original goal of the profile steering algorithm is to finish when insufficient progress can be made per iteration, the new goal is whenever the limit of the substation are not exceeded anymore.

Small-Scale Scenario

The small scale scenario represent a setup comprising three holons that are interconnected and also have a direct communication link to the substation. These holons have a different preference profiles that are shown in Figure 5.1.

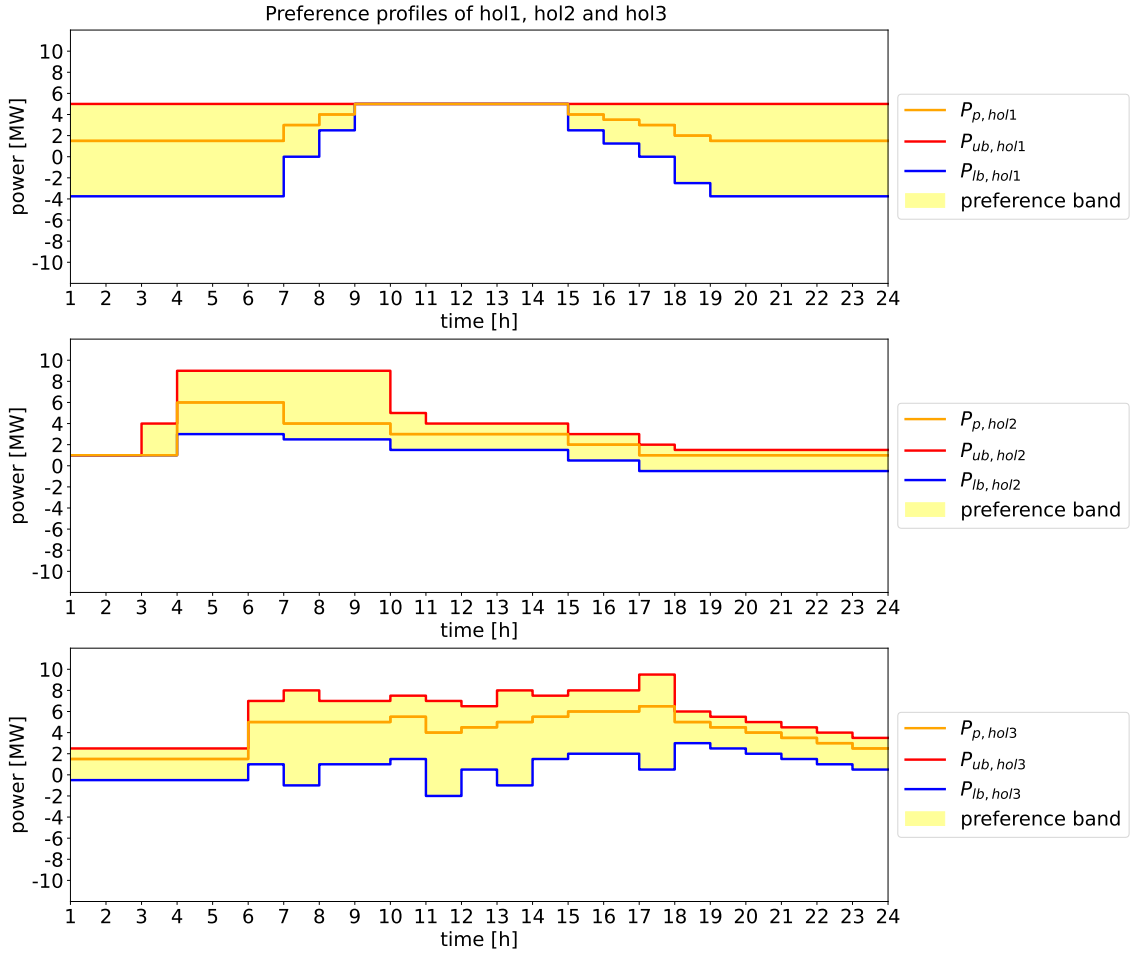


Figure 5.1: Preference profiles of three different holons, named *hol*1, *hol*2 and *hol*3.

In Figure 5.1, *hol*1 represents an office. This office has a maximum amount of power of 5 MW available at any time, resulting in a static upper bound. During the morning, employees arrive and hence the power increases, to decrease again at the end of the day. *hol*2 represents a logistics company, who has electric trucks arriving in the morning, to be dispatched at the end of the morning. This results in a higher peak during that time. These trucks need to be charged and are usually charged at maximum power. However,

they can also be charged with a reduced amount of power and still be fully charged when they have to leave. This results in some flexibility. When the trucks have left, the company can utilize a battery to offer flexibility towards the rest of the grid. *hol*3 is a manufacturing plant with variable production processes. This holon has an erratic power profile with a varying amount of flexibility. Figure 5.2 shows the aggregated preference profile of the three holons. It also shows that grid congestion occurs if these preference profiles are realized the next day.

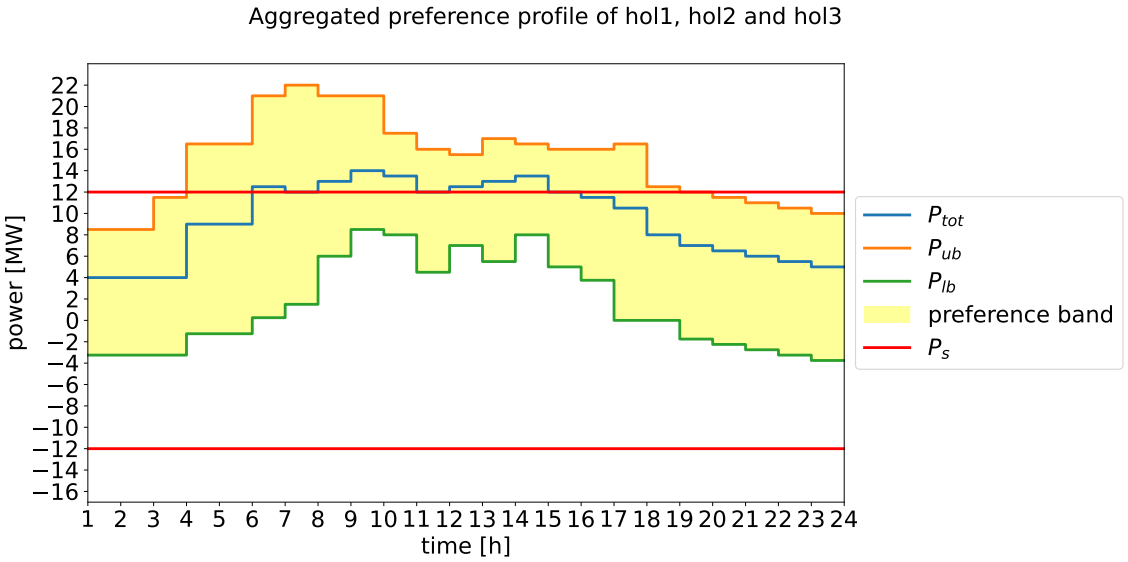


Figure 5.2: Aggregated preference profile and the substation limits.

Ecofactorij

The Ecofactorij [40] is an industrial complex in Apeldoorn, the Netherlands. They received an exemption from the Authority for Consumers & Markets (ACM), which allows them to regulate their own electricity grid without having a DSO involved [41]. The Ecofactorij is connected to the national grid at a central location, after which the electricity gets distributed across the connected parties. Since the grid (electricity and gas) of the Ecofactorij is private property, it is also considered as a closed distribution system (CDS) [42].

Normally, the DSO needs to guarantee the uptime of the grid. If the grid would be public, the DSO would aggregate the maximum capacity of each company to determine the connection size of the whole area. Since the Ecofactorij is a CDS, this is not needed and the Ecofactorij is able to determine the connection size to the national electricity grid, provided the DSO can supply that amount. Since a lower connection size is cheaper, it also provides an incentive towards the connected parties to actively participate in avoiding peak load in the electricity grid.

Furthermore, the Ecofactorij considers self-sustainability as a high priority. There are multiple energy sources and buffers connected, both in central locations and at the companies themselves. The energy sources include PV panels (both at the companies and external solar parks) and heat pumps [40], [43]. These sources can be regulated by the Ecofactorij to relieve strain on the grid in case of congestion, by scaling the production up

or down on fifteen minute intervals. The buffers comprise heat accumulators and batteries, which are able to be utilized at moments of grid congestion to shift the grid load [40], [44].

A simplified overview of the connections in the Ecofactorij is shown in Figure 5.3.

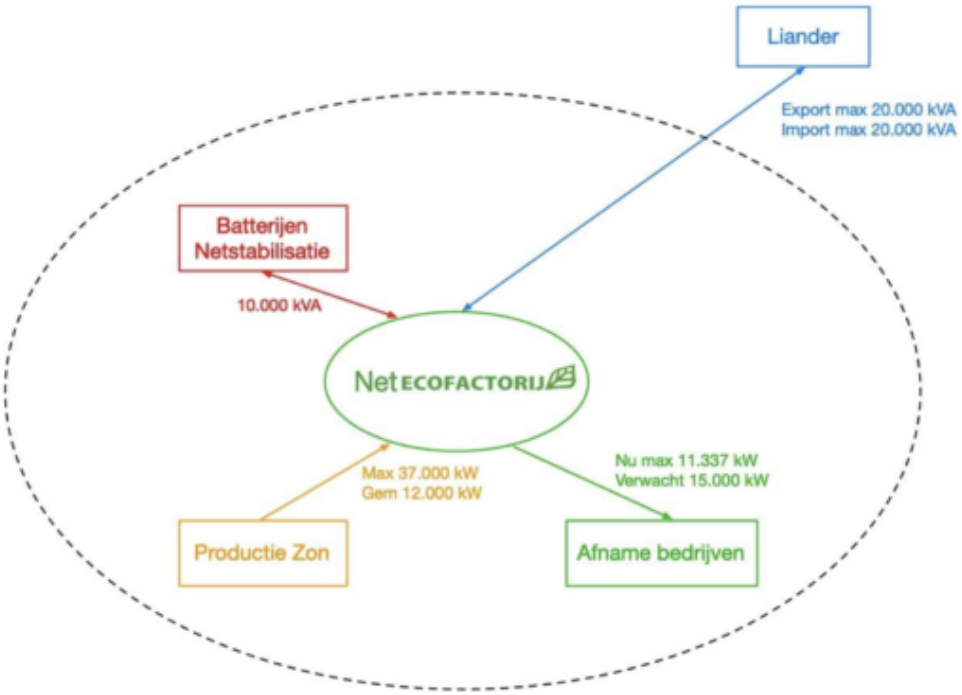


Figure 5.3: Overview of the Ecofactorij [40].

For a realistic scenario, the Ecofactorij is a good representation. The setup of the Ecofactorij consists of 24 holons that are connected to the grid. These holons vary in energy demand: some holons are bigger companies with their energy usage in megawatts, whereas other holons comprise smaller companies and solar parks. The availability of historical data allows for realistic scenarios. For the simulation, one extreme day during the winter (a lot of power usage and no solar production) is used to evaluate the algorithms. The aggregated preference profile of that day is shown in Figure 5.4.

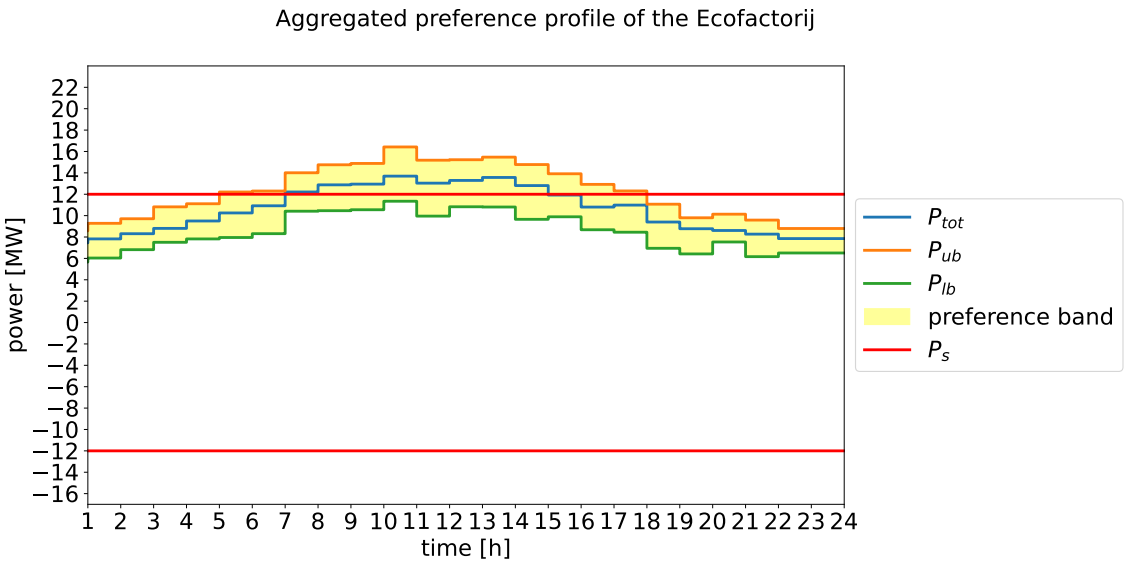


Figure 5.4: Preference profile of the Ecofactorij.

Historical data is used to create a preference profile for the Ecofactorij. The preference profile is determined by creating a data sample every two seconds over the course of the day. With these samples, the average is used for the preference profile itself and the bounds are based on the standard deviations of those measurements.

5.1.3 Metrics

In order to get a better understanding in the performance of the algorithms, metrics are defined. The metrics used are: complexity, user satisfaction and renewable energy usage.

Complexity

Complexity is useful to determine the mechanical characteristics of the algorithms. This evaluation can be split into two different types of complexity: time and resource complexity. Time complexity is determined by the operations required in order to fully execute the algorithm. This is determined as a function of the input (e.g. how much extra time it takes when the input size increases). Resource complexity is used to describe the amount of memory and computational power is needed to execute the algorithm as a function of the input (e.g. how the amount of extra resources scales when the input size increases). The complexity is also represented by the execution time of the algorithm in Section 5.3.

Curtailment and Reallocation For the curtailment and reallocation algorithm, the bottleneck when it comes to time complexity is the sorting of the priorities. The time complexity of sorting algorithm varies depending on which sorting algorithm used. For this algorithm, the sorted function from Python is used, which is a combination of merge sort and insertion sort. The time complexity of this algorithm is $O(Hlog(H))$, with H being the amount of holons.

As for resource complexity, the storage of variables scales linearly with H, hence the resource complexity of the curtailment and reallocation algorithm is $O(H)$.

Capacity Trading The trading always occurs between two holons, however the amount of trades performed can vary widely depending on the element of reduction in trade amount when it fails to trade with every holon until a trade is successful or the amount to trade has reached 0. Because of this, the time complexity of the Capacity Trading algorithm is $O(H^2)$. The resource complexity for this algorithm scales linearly with the amount of holons, H. Hence, the resource complexity is $O(H)$.

User Satisfaction

For user satisfaction, the focus is on the individual holon. The goal is that each holon is treated equally. In order to determine if they are treated equally, multiple aspects have to be evaluated regarding user satisfaction. These aspects are:

% of preference met determines to what extent the submitted preference profile, $P_{p,h}(t)$ is adhered to. This is calculated using the following equation:

$$perc\_preference = \left(1 - \frac{\sum_{t=1}^{T} |P_{a,h}(t) - P_{p,h}(t)|}{\sum_{t=1}^{T} |P_{p,h}(t)|}\right) \cdot 100\% \tag{5.1}$$

% of allocated power out of bounds goes one step further than **% of preference met**, since it evaluates how much the allocated power, $P_{a,h}(t)$ stays within the preference bounds. In order to determine this percentage, we first need to calculate how much from the preference bound is deviated at time t. Depending on if the allocated power ($P_{a,h}(t)$) is greater than the upper bound ($P_{ub,h}(t)$) or the allocated power is lower than the lower bound ($P_{lb,h}(t)$), the deviation, $d(t)$, can be calculated as follows:

$$d(t) = \begin{cases} P_{a,h}(t) - P_{ub,h}(t), & if P_{a,h}(t) > P_{ub,h}(t) \\ P_{lb,h}(t) - P_{a,h}(t), & if P_{a,h}(t) < P_{lb,h}(t) \\ 0, & otherwise \end{cases} \tag{5.2}$$

With the deviation calculated, the metric is calculated using the following equation:

$$perc\_out\_of\_bounds = \frac{\sum_{t=1}^{T} |d(t)|}{\sum_{t=1}^{T} |P_{p,h}(t)|} \cdot 100\% \tag{5.3}$$

% deficit determines how much of the preferred amount of power has been converted into a deficit. This deficit is calculated using the following equation:

$$deficit_h = \sum_{t=1}^{T} P_{p,h}(t) - P_{a,h}(t) \tag{5.4}$$

$$perc\_deficit = \frac{100 \cdot deficit_h}{\sum_{t=1}^{T} P_{p,h}(t)} \tag{5.5}$$

Flexibility violation describes how much of the change in the preference profile of a holon cannot be accounted for by the flexibility sources. These flexibility sources also consist of the sources from other holons. This metric calculates the biggest violation. A

violation is represented with a negative value, whereas a positive value indicates how much capacity the holon has to spare. when (3.3) does not hold. The violation is calculated as follows:

$$flexibility\_violation_h = \begin{cases} \zeta_{max} - \max(SoC_h(t)), & \text{if } \max(SoC_h(t)) > \zeta_{max} \\ \min(SoC_h(t)), & \text{otherwise} \end{cases} \tag{5.6}$$

5.2 Evaluation Method

Evaluating the metrics from Section 5.1.3 requires several steps. Since the defined metrics are defined for individual holons, these individual metrics are used to establish the metrics that are used for the entire system. This section details where and how these metrics are calculated.

The values for % preference met, % out of bounds and % deficit are calculated after the Curtailment and Reallocation algorithm, as well as after the Capacity Trading algorithm. However, the flexibility violation requires a slight modification. Since this depends on the characteristics of the flexibility sources of each holon, we perform parameter sweeps. During these sweeps, we vary the initial state of charge ($SoC_h(t = 1)$) of two flexibility sources. For each value of the $SoC_h(t = 1)$, the flexibility violation is calculated for each holon in the model, after which the minimum value is used to create a heat map.

Since the Capacity Trading algorithm has an effect on the power profiles of the holons, it also changes the % preference met metric. This can be combined with the aforementioned parameter sweep to show the effect of different values of $SoC_h(t = 1)$ on the % preference met. In order to to create a heat map for that metric as well, we use the mean. The minimum value is not chosen for the heat map, because it does not change once the holon with the lowest % preference met is unable or does not need to trade.

Lastly, to measure the complexity of the algorithms, we examine the convergence of the flexibility violation. If the flexibility violation for each holon is non-negative, the algorithms have converged and created an energetically feasible planning. Therefore, the flexibility violation for each holon is calculated after every iteration and shown in a graph.

5.3 Results and Discussions

This section describes and explains the results of the algorithm using the two aforementioned scenarios. For each scenario, it shows and elaborates the results of the Curtailment and Reallocation algorithm first, followed by the results and elaboration of the Capacity Trading Algorithm.

5.3.1 Small-Scale Scenario

The small-scale scenario is used to verify the functioning of the algorithms and to also benchmark the algorithms separately. Every metric is being evaluated for after both algorithms, however whereas the Capacity Trading algorithm only results in a change in the

% preference met, the **% out of bounds** and the **flexibility violation** metrics, since the total amount of allocated power does not change.

Curtailment and Reallocation Results

The results of the Curtailment and Reallocation algorithm are shown in Figure 5.5 with the aggregated power profile, P_{tot} shown in Figure 5.6.

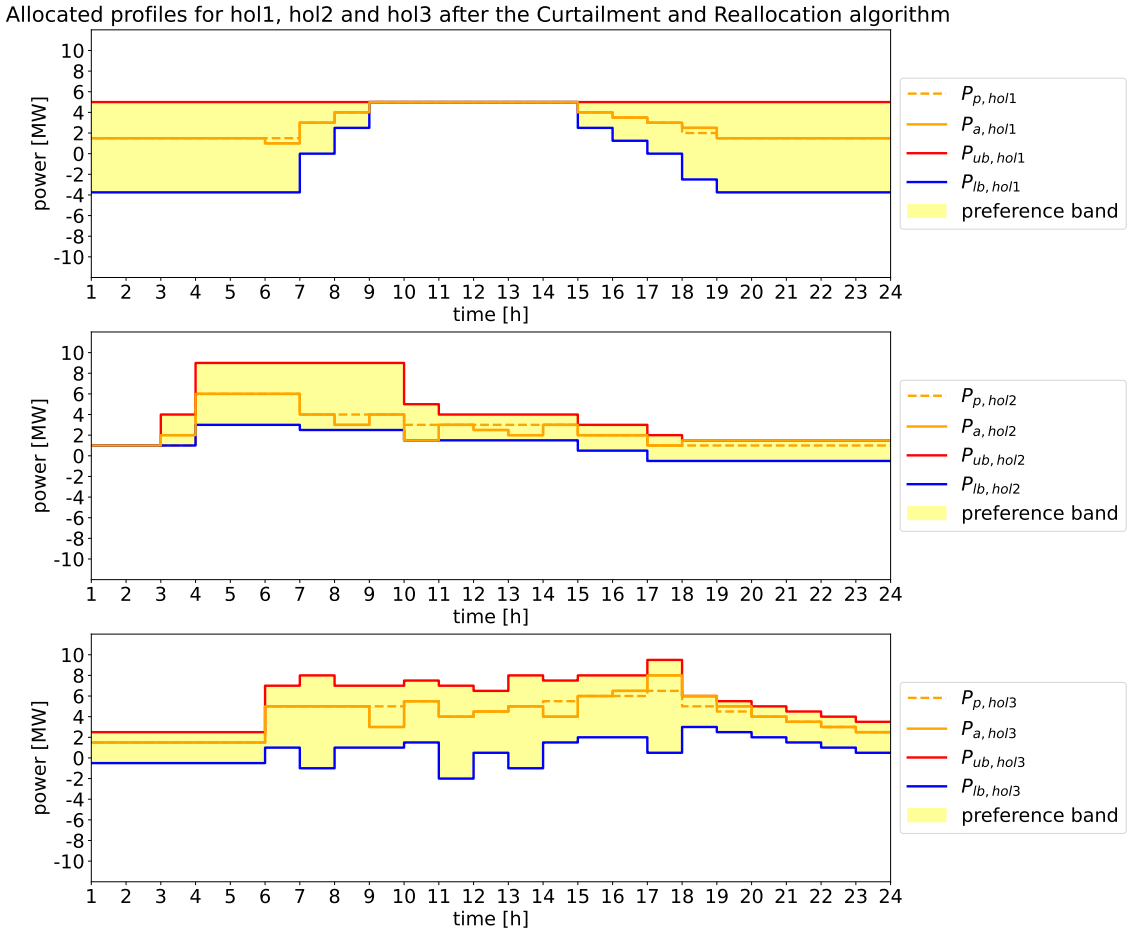


Figure 5.5: Allocated profiles of the three different holons, named *hol*1, *hol*2, *hol*3.

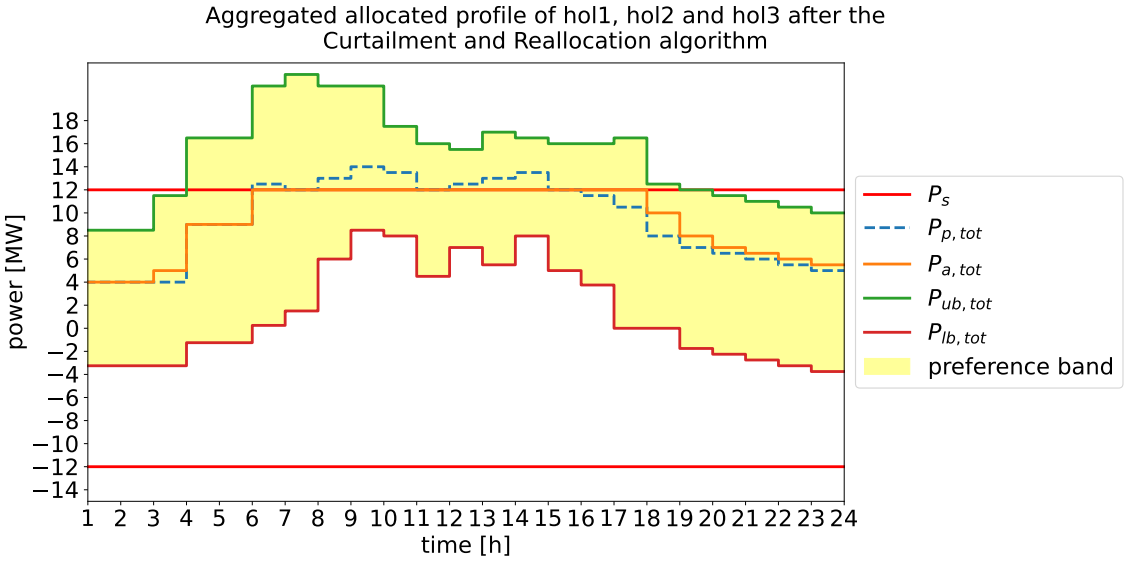


Figure 5.6: Aggregated allocated profile with the preference profile and the substation limits.

Figure 5.6 indicates that the curtailment and reallocation algorithm successfully achieves its main goal of preventing grid congestion at the substation given the preference profiles from Figure 5.1. The amount curtailed from each holon is also reallocated during moments where the aggregated power complies with the limits at the substation. Furthermore, Figure 5.5 shows that the allocated power, $P_{a,h}(t)$, stays within the boundaries of the preference band of each holon. The results are also summarized in Table 5.1.

Table 5.1: Results of the Curtailment and Reallocation algorithm.

	% preference met	**% out of bounds**	**% deficit**
hol1	98.5	0.0	0.0
hol2	86.7	0.0	0.0
hol3	92.6	0.0	0.0

Table 5.1 indeed shows that the profiles never exceed their preference bounds, as well as the holons receiving the same amount of capacity throughout the day as submitted in their preference profiles. Hence, both of these metrics are 0%.

For the evaluation of the final metric, flexibility violation, flexibility sources have been introduced into the system. These flexibility sources comprise three batteries, each of them connected to a single holon. As mentioned before in Section 5.2, this metric depends on the initial state of charge, $SoC_h(t = 1)$, of these batteries. Hence, a parameter sweep has been performed. For this parameter sweep, the capacity of the batteries (ζ_{max}) has been set to 5 MWh for these simulations. During the parameter sweep, two holons are swept, while not varying the $SoC_h(t = 1)$ of the third holon. This results in three different sweeps, in which the holon with the constant initial state of charge varies with each sweep. For the constant holon, $SoC_h(t = 1) = 2$ MWh (which corresponds to a battery that is 40% charged).

The parameter sweeps for the small-scale scenario after the Curtailment and Reallocation algorithm are shown in Figure 5.7.

	SoC$_{hol2}$(t=1) [MWh]											
SoC$_{hol1}$(t=1) [MWh]	hol1\hol2	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
	0	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	0.5	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	1	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	1.5	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	2	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	2.5	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	3	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	3.5	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	4	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	4.5	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	5	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500	-1500

(a) Heat map of the flexibility violation with varying $SoC_h(t=1)$ for $hol1$ and $hol2$.

	SoC$_{hol3}$(t=1) [MWh]											
SoC$_{hol1}$(t=1) [MWh]	hol1\hol3	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
	0	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	0.5	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	1	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	1.5	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	2	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	2.5	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	3	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	3.5	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	4	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	4.5	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	5	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000

(b) Heat map of the flexibility violation with varying $SoC_h(t=1)$ for $hol1$ and $hol3$.

	SoC$_{hol3}$(t=1) [MWh]											
SoC$_{hol2}$(t=1) [MWh]	hol2\hol3	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
	0	-3500	-3000	-3000	-3000	-3000	-3000	-3000	-3000	-3000	-3000	-3000
	0.5	-3500	-3000	-2500	-2500	-2500	-2500	-2500	-2500	-2500	-2500	-2500
	1	-3500	-3000	-2500	-2000	-2000	-2000	-2000	-2000	-2000	-2000	-2000
	1.5	-3500	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1500	-1500
	2	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-1000	-1000
	2.5	-3500	-3000	-2500	-2000	-1500	-1000	-500	-500	-500	-500	-500
	3	-3500	-3000	-2500	-2000	-1500	-1000	-500	0	0	0	0
	3.5	-3500	-3000	-2500	-2000	-1500	-1000	-500	0	500	500	500
	4	-3500	-3000	-2500	-2000	-1500	-1000	-500	0	500	1000	1000
	4.5	-3500	-3000	-2500	-2000	-1500	-1000	-500	0	500	1000	1500
	5	-3500	-3000	-2500	-2000	-1500	-1000	-500	0	500	1000	1500

(c) Heat map of the flexibility violation.

Figure 5.7: Heat map of the flexibility violation after the Curtailment and Reallocation algorithm.

Figure 5.7 shows the moments where the resulting power profiles are energetically feasible. A profile is energetically feasible if the resulting value in the heat map is non-negative. For Figure 5.7, this is only seen in the parameter sweep of both hol2 and hol3. The only viable profiles occur when $SoC_{hol2}(t=1) \geq 3\,\text{MWh}$ and $SoC_{hol3}(t=1) \geq 3.5\,\text{MWh}$, given that $SoC_{hol1}(t=1) = 2\,\text{MWh}$. With those values, the minimum flexibility violation is non-negative. However, more valid combinations can be created when executing the Capacity Trading algorithm.

Capacity Trading Results

For the capacity trading, the same parameter sweeps have been performed in order to evaluate if the flexibility violations can be solved for in more cases (two holons vary their initial state of charge ($SoC_h(t = 1)$), whereas the other holon has a static $SoC_h(t = 1)$ =2 MWh). These results are shown in Figures 5.8, 5.9 and 5.10. Subfigures 5.8a, 5.9a and 5.10a represent the flexibility violation results of the parameter sweeps. Subfigures 5.8b, 5.9b and 5.10b show the new preference met results after the Capacity Trading algorithm with the parameter sweep. This preference met is determined by taking the mean value of the three holons. Since the allocated profiles change, this metric also changes.

	SoChol2(t=1) [MWh]										
SoChol1(t=1) [MWh]	hol1\hol2	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
	0	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1500	-1050	-525	0
	0.5	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1050	-525	0	0
	1	-2550	-2000	-1500	-1500	-1500	-1500	-1050	-521.19	0	0	0
	1.5	-2025	-1500	-1500	-1500	-1500	-1050	-525	0	0	0	0
	2	-1500	-1500	-1500	-1500	-1050	-525	0	0	0	0	0
	2.5	-1500	-1500	-1500	-1050	-525	0	0	0	0	0	0
	3	-1500	-1500	-1050	-525	0	0	0	0	0	0	0
	3.5	-1500	-1050	-525	0	0	0	0	0	0	0	0
	4	-1050	-525	0	0	0	0	0	0	0	0	0
	4.5	-525	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0

(a) Heat map of the flexibility violation.

	SoChol2(t=1) [MWh]										
SoChol1(t=1) [MWh]	hol1\hol2	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
	0	92.59	92.59	92.59	92.59	92.59	92.59	92.59	92.59	92.59	92.06	91.48
	0.5	92.59	92.59	92.59	92.59	92.59	92.59	92.59	92.59	92.59	91.19	90.34
	1	92.15	92.1	92.1	92.1	92.1	92.1	92.15	92.12	92.1	91.19	90.34
	1.5	91.63	91.6	91.6	91.6	91.6	91.65	91.63	91.6	91.63	91.19	90.34
	2	91.11	91.11	91.11	91.11	91.16	91.14	91.11	91.11	91.11	91.11	91.11
	2.5	90.62	90.62	90.62	90.67	90.64	90.62	91.11	91.11	91.11	91.11	91.11
	3	90.12	90.12	90.17	90.15	90.12	90.62	91.11	91.11	91.11	91.11	91.11
	3.5	89.63	89.68	89.65	89.63	90.12	90.62	91.11	91.11	91.11	91.11	91.11
	4	89.19	89.16	89.14	89.63	90.12	90.62	91.11	91.11	91.11	91.11	91.11
	4.5	88.67	88.64	89.14	89.63	90.12	90.62	91.11	91.11	91.11	91.11	90.99
	5	88.15	88.64	89.14	89.63	90.12	90.62	91.11	91.11	91.11	91.05	90.99

(b) Heat map of the preference met.

Figure 5.8: Heat maps for parameter sweep of $SoC_h(t = 1)$ for $hol1$ and $hol2$, where $SoC_{hol3}(t = 1)$ =2 MWh.

	SoC$_{hol3}$(t=1) [MWh]											
SoC$_{hol1}$(t=1) [MWh]	hol1\hol3	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
	0	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-1000	-522.5	0
	0.5	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-500	0	0
	1	-3500	-3000	-2500	-2000	-1500	-1000	-500	-500	0	0	0
	1.5	-3500	-3000	-2500	-2000	-1500	-1000	-500	0	0	0	0
	2	-3000	-2550	-2000	-1500	-1050	-500	0	0	0	0	0
	2.5	-2500	-2025	-1500	-1050	-525	0	0	0	0	0	0
	3	-2000	-1500	-1050	-525	0	0	0	0	0	0	0
	3.5	-1500	-1050	-525	0	0	0	0	0	0	0	0
	4	-1050	-525	0	0	0	0	0	0	0	0	0
	4.5	-525	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0

(a) Heat map of the flexibility violation.

	SoC$_{hol3}$(t=1) [MWh]											
SoC$_{hol1}$(t=1) [MWh]	hol1\hol3	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
	0	92.59	92.59	92.59	92.59	92.59	92.59	92.59	92.59	92.24	91.9	91.53
	0.5	92.59	92.59	92.59	92.59	92.59	92.59	92.59	92.59	92.24	91.89	91.89
	1	92.1	92.1	92.1	92.1	92.1	92.1	92.1	92.1	91.75	91.89	91.89
	1.5	91.6	91.6	91.6	91.6	91.6	91.6	91.6	91.6	91.6	91.6	91.6
	2	91.11	91.16	91.11	91.11	91.16	91.11	91.11	91.6	91.6	91.6	91.6
	2.5	90.62	90.64	90.62	90.67	90.64	90.62	91.11	91.6	91.6	91.6	91.6
	3	90.12	90.12	90.17	90.15	90.12	90.62	91.11	91.6	91.6	91.6	91.6
	3.5	89.63	89.68	89.65	89.63	90.12	90.62	91.11	91.6	91.6	91.6	91.6
	4	89.19	89.16	89.14	89.63	90.12	90.62	91.11	91.6	91.6	91.6	91.6
	4.5	88.67	88.64	89.14	89.63	90.12	90.62	91.11	91.6	91.6	91.6	91.6
	5	88.15	88.64	89.14	89.63	90.12	90.62	91.11	91.6	91.6	91.6	91.6

(b) Heat map of the preference met.

Figure 5.9: Heat maps for parameter sweep of $SoC_h(t = 1)$ for $hol1$ and $hol3$, where $SoC_{hol2}(t = 1)$ =2 MWh.

SoC$_{hol2}$(t=1) [MWh] \ SoC$_{hol3}$(t=1) [MWh]	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
0	-3500	-3000	-2500	-2000	-1500	-1500	-1500	-1500	-1000	-500	0
0.5	-3500	-3000	-2500	-2000	-1500	-1000	-1000	-1000	-500	0	0
1	-3500	-3000	-2500	-2000	-1500	-1000	-500	-500	0	0	0
1.5	-3500	-3000	-2500	-2000	-1500	-1000	-500	0	0	0	0
2	-3000	-2550	-2000	-1500	-1050	-500	0	0	0	0	0
2.5	-2500	-2025	-1500	-1050	-525	0	0	0	0	0	0
3	-2000	-1500	-1050	-525	0	0	0	0	0	0	0
3.5	-1500	-1050	-521.19	0	0	0	0	0	500	500	500
4	-1050	-525	0	0	0	0	0	0	500	1000	1000
4.5	-525	0	0	0	0	0	0	0	500	1000	1500
5	0	0	0	0	0	0	0	0	500	1000	1500

(a) Heat map of the flexibility violation.

SoC$_{hol2}$(t=1) [MWh] \ SoC$_{hol3}$(t=1) [MWh]	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
0	91.11	91.11	91.11	91.11	91.11	91.11	91.11	91.11	90.76	90.41	90.05
0.5	91.11	91.11	91.11	91.11	91.11	91.11	91.11	91.11	90.76	90.41	90.41
1	91.11	91.11	91.11	91.11	91.11	91.11	91.11	91.11	90.76	90.9	90.9
1.5	91.11	91.11	91.11	91.11	91.11	91.11	91.11	91.11	91.11	91.11	91.11
2	91.11	91.16	91.11	91.11	91.16	91.11	91.11	91.6	91.6	91.6	91.6
2.5	91.11	91.14	91.11	91.16	91.14	91.11	91.6	92.1	92.1	92.1	92.1
3	91.11	91.11	91.16	91.14	91.11	91.6	92.1	92.59	92.59	92.59	92.59
3.5	91.11	91.11	91.13	91.11	91.11	91.6	92.1	92.59	92.59	92.59	92.59
4	91.11	91.11	91.11	91.14	91.11	91.6	92.1	92.59	92.59	92.59	92.59
4.5	90.58	90.56	91.05	91.54	91.11	91.6	92.1	92.1	92.1	92.1	92.1
5	90	90.56	91.05	90.7	91.11	91.6	91.6	91.6	91.6	91.6	91.6

(b) Heat map of the preference met.

Figure 5.10: Heat maps for parameter sweep of $SoC_h(t = 1)$ for $hol2$ and $hol3$, where $SoC_{hol1}(t = 1)$ =2 MWh.

Figures 5.8, 5.9 and 5.10 show that more combinations have become energetically feasible power profiles after the Capacity Trading algorithm. For example, a feasible profile can already be accomplished when one battery is charged at the beginning of the day. Should that be the case and should another battery contain at least 2 MWh, the initial state of charge of the third battery does not matter to create a feasible power profile. Combining this with the preference met, it illustrates an interesting first observation: a feasible profile does not necessarily indicate that it performs well in other areas. The aforementioned example might result in a feasible power profile, but the preference met is lower compared to a more balanced approach (instead of one holon having a charged battery).

Another interesting observation from the preference met heat maps is the higher percentage of preference met in the top left regions of Subfigures 5.8b and 5.9b. These regions with a higher preference met exist because trading only occurs when possible. Trading is not possible in those regions, hence no trades commence and the allocated power profiles remain unchanged. This results in the same preference met compared to the preference met after the Curtailment and Reallocation algorithm. Additionally, there are also cases where the preference met is lower, but the resulting power profile still have a flexibility violation (i.e., the profiles are energetically infeasible). This occurs when trading is partially possible: some parts of the flexibility violation are accounted for, but not all.

In order to evaluate what exactly happens during the Capacity Trading algorithm, we isolate three different scenarios from these heat maps. These combinations of $SoC_h(t = 1)$ are shown in Table 5.2.

Table 5.2: Parameters of the batteries for each of the holons.

	$SoC_h(t=1)$ [MWh]		
Scenario	*1*	*2*	*3*
$hol1$	2.0	2.0	1.5
$hol2$	3.0	3.0	3.5
$hol3$	3.5	2.5	2.0

With the parameters from Table 5.2, the profile of the state of charge, $SoC_h(t)$, of each of the holons is shown in Figure 5.11. This figure shows that there are no violations even before trading, resulting in no difference in the metrics before and after the Capacity Trading algorithm. Figure 5.11 also includes the resulting $SoC_h(t)$ profiles of the profile steering algorithm. In a comparison, it becomes clear that profile steering does not utilize every battery, whereas the developed algorithms do utilize every battery when needed.

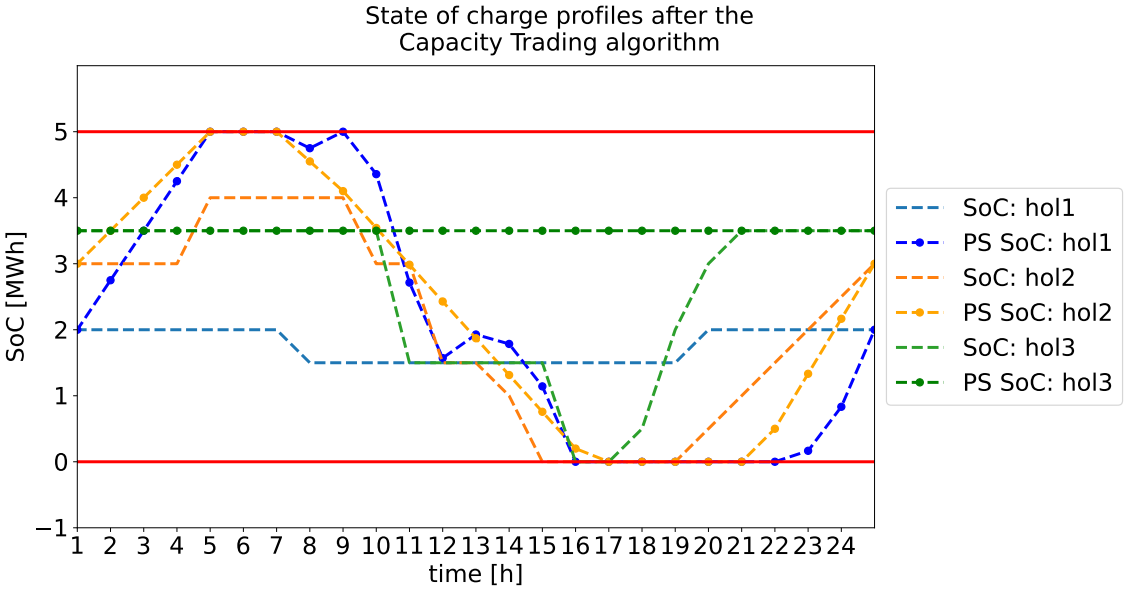


Figure 5.11: Profiles of the $SoC_h(t)$ for scenario 1 after the Capacity Trading algorithm (in filled lines) versus the $SoC_h(t)$ of the profile steering algorithm (in dashed lines).

The final allocated profiles for each holon, as well as the resulting profile from the profile steering algorithm are shown in Figure 5.12 with the aggregated power profile shown in Figure 5.13. As can be seen in that figure, the allocated profiles of the developed algorithms stick more to the preference bounds compared to the profiles from the profile steering algorithm. The profile steering algorithm also exceed the preference bounds, since it only takes $P_{p,h}(t)$ into account when creating a planning. This is also reflected in the metrics, which are summarized in Table 5.3. The table also shows the difference in preference met between the two algorithms: even though $hol3$ has a lower preference met compared to profile steering, the other holons have a higher preference met, resulting in a higher average customer satisfaction.

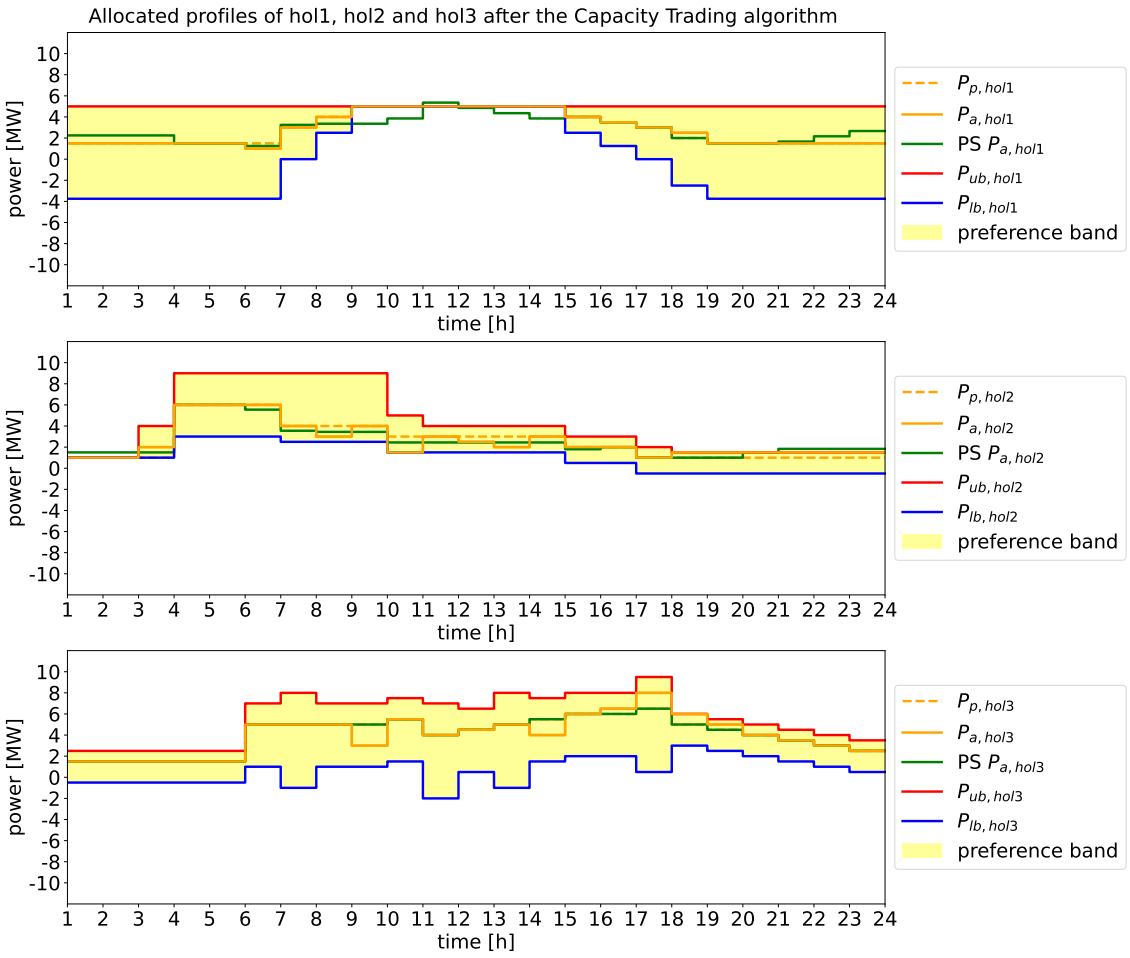


Figure 5.12: Allocated profiles after the Capacity Trading algorithm for scenario 1, along with the profile steering algorithm results.

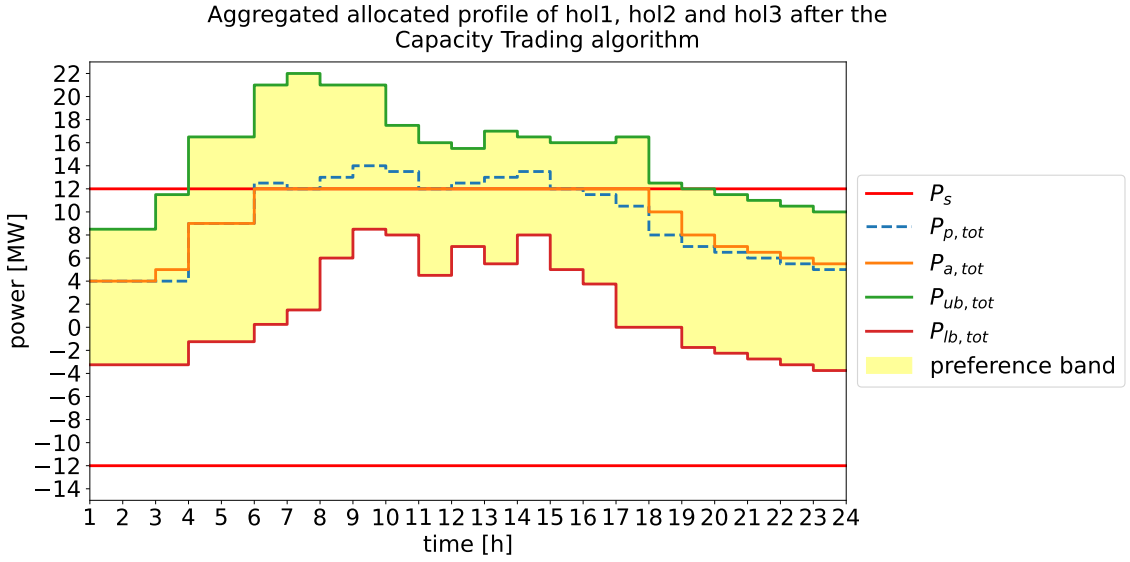


Figure 5.13: Aggregated power profile after the Capacity Trading algorithm for scenario 1.

Table 5.3: Results of the Developed algorithms for scenario 1 compared to profile steering.

		% preference met	% out of bounds	% deficit
Developed Algorithms	**hol1**	98.5	0.0	0.0
	hol2	86.7	0.0	0.0
	hol3	92.6	0.0	0.0
Profile Steering	**hol1**	83.3	0.5	0.0
	hol2	83.3	4.2	0.0
	hol3	100	0.0	0.0

When looking at the amount of iterations the algorithm takes to converge, we need to evaluate when the algorithm actually has converged. This is the case when the flexibility violations caused by the Curtailment and Reallocation algorithm are mitigated if possible. This results in Figure 5.14, which shows that the algorithm fully converges in two iterations. In total, the average time it took for the algorithm to fully executed is 0.67 seconds.

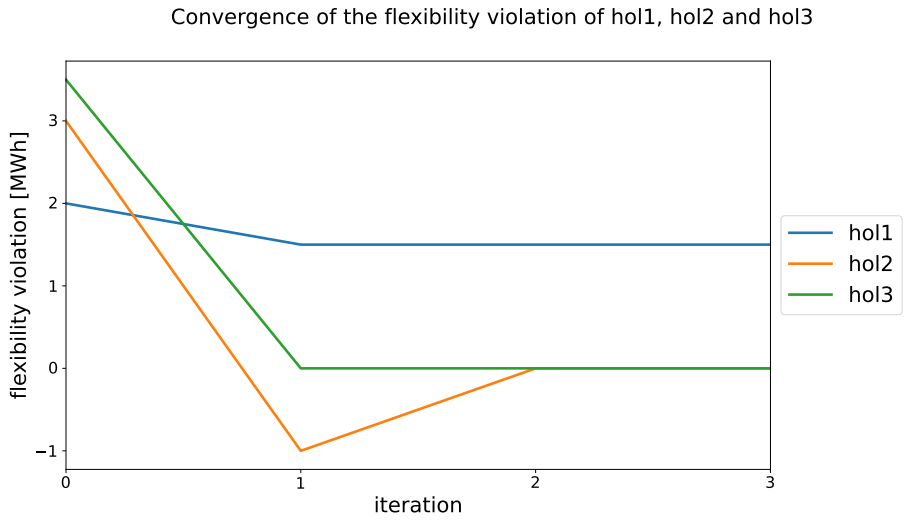


Figure 5.14: Convergence graph of scenario 1, where the system has converged after two iterations.

For the second scenario (see Table 5.2), we notice from Figure 5.7 that the Capacity Trading algorithm needs to be executed in order to provide a feasible schedule. This is also represented in Figure 5.15, where $hol3$ has a deficit that the battery cannot account for. Hence, $hol3$ attempts to trade with $hol1$. Since $hol1$ is able to provide for that capacity and $hol3$ is able to trade back that amount later, the violation can be mitigated. The $SoC_h(t)$ of the batteries after the Capacity Trading algorithm is shown in Figure 5.16.

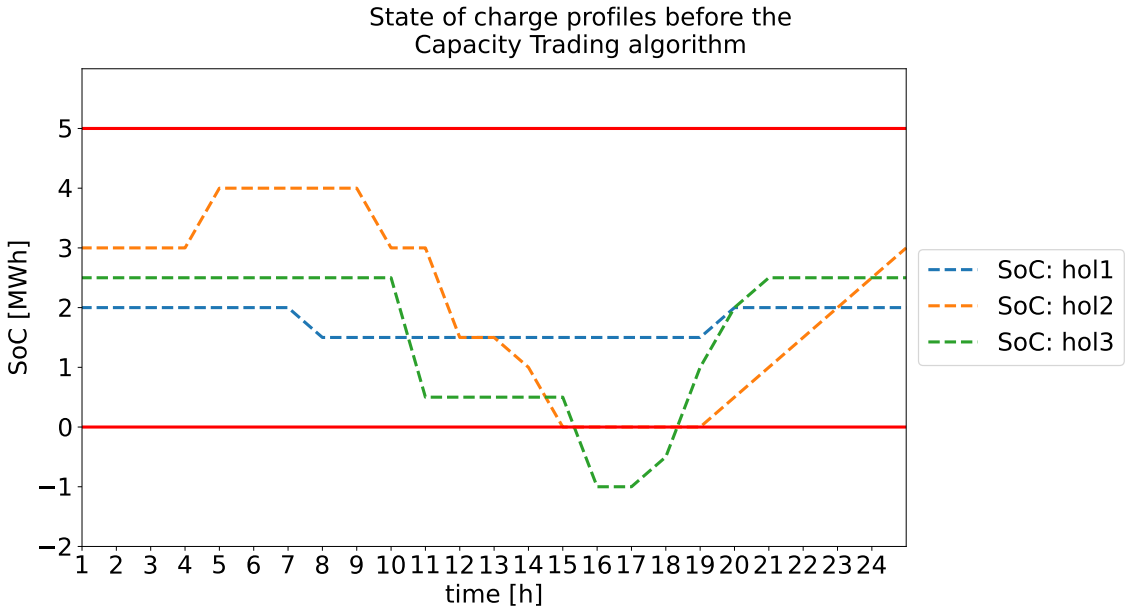


Figure 5.15: Profiles of the $SoC_h(t)$ for scenario 2 before the Capacity Trading algorithm.

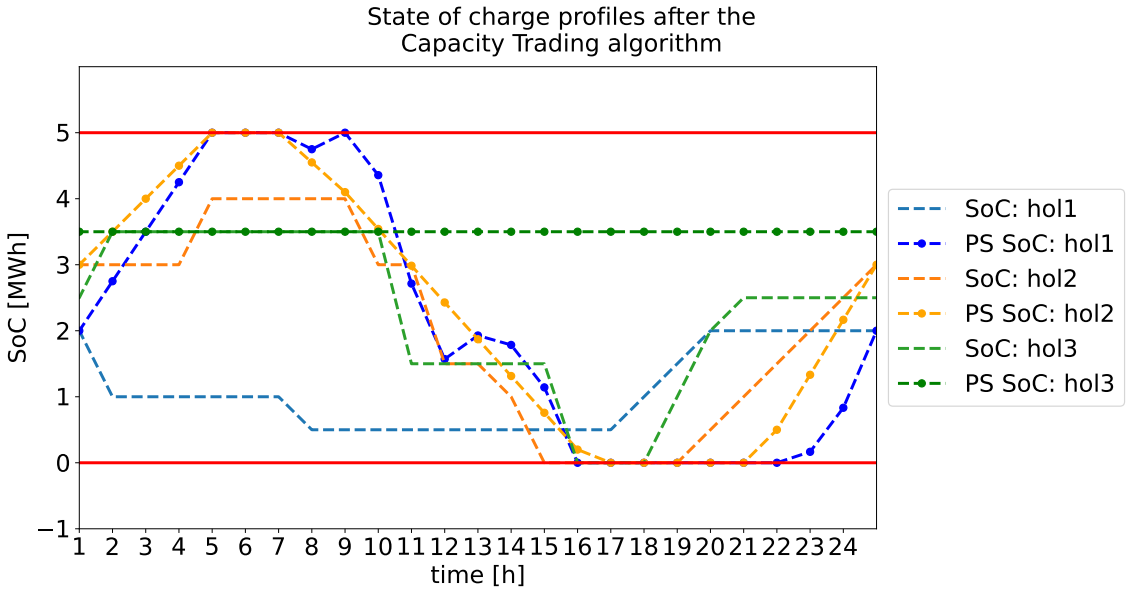


Figure 5.16: Profiles of the $SoC_h(t)$ for scenario 2 after the Capacity Trading algorithm (in filled lines) versus the $SoC_h(t)$ of the profile steering algorithm (in dashed lines).

The final allocated profiles for each holon, combined with the final allocated profiles from the profile steering algorithm are shown in Figure 5.17. It shows that the developed algorithms still provide a feasible solution where the bounds are not exceeded. This is also represented in Table 5.4, where the results for the % preference met metric are closer to each other compared to the result from profile steering The % preference met for $hol1$ is a little lower compared to scenario 1, which is due to the trade that occurred.

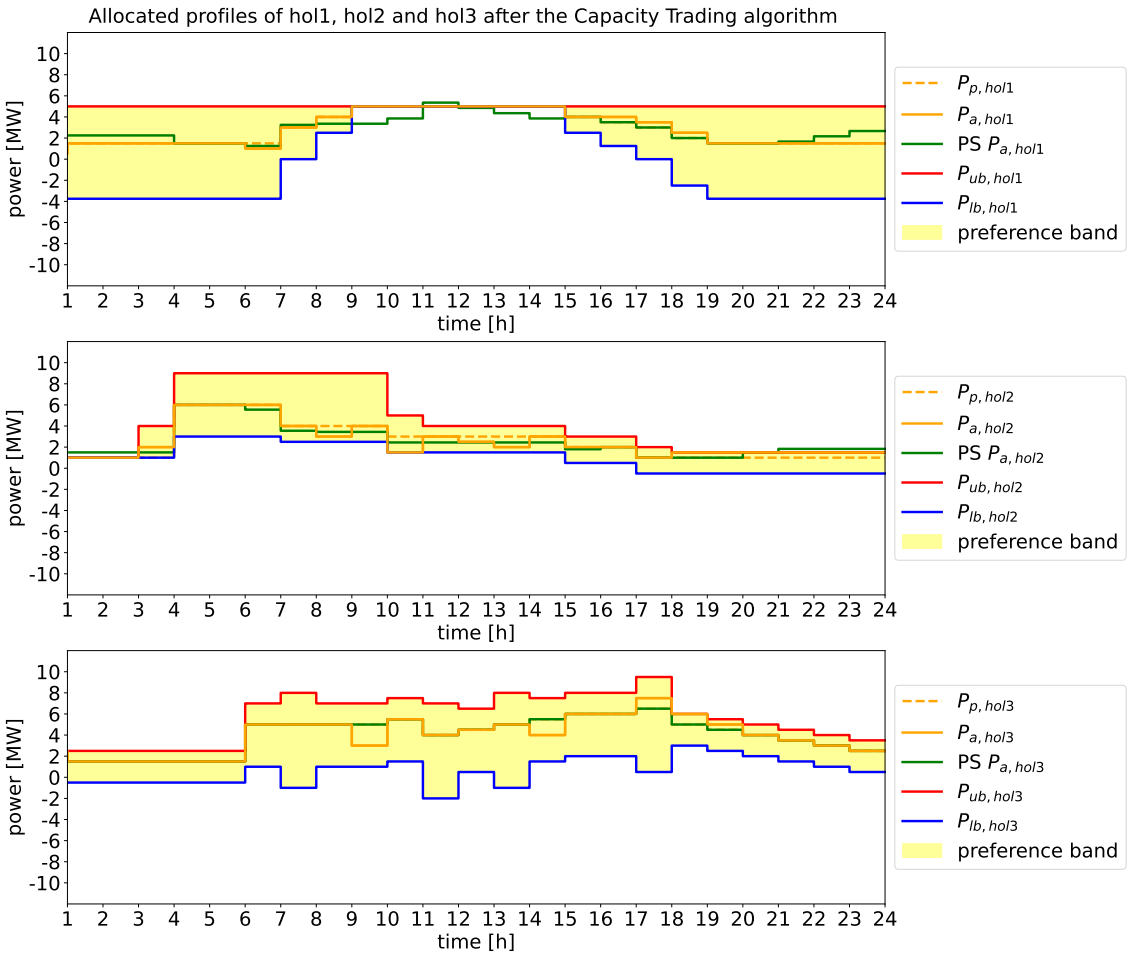


Figure 5.17: Allocated profiles after the Capacity Trading algorithm for scenario 2, along with the profile steering algorithm results.

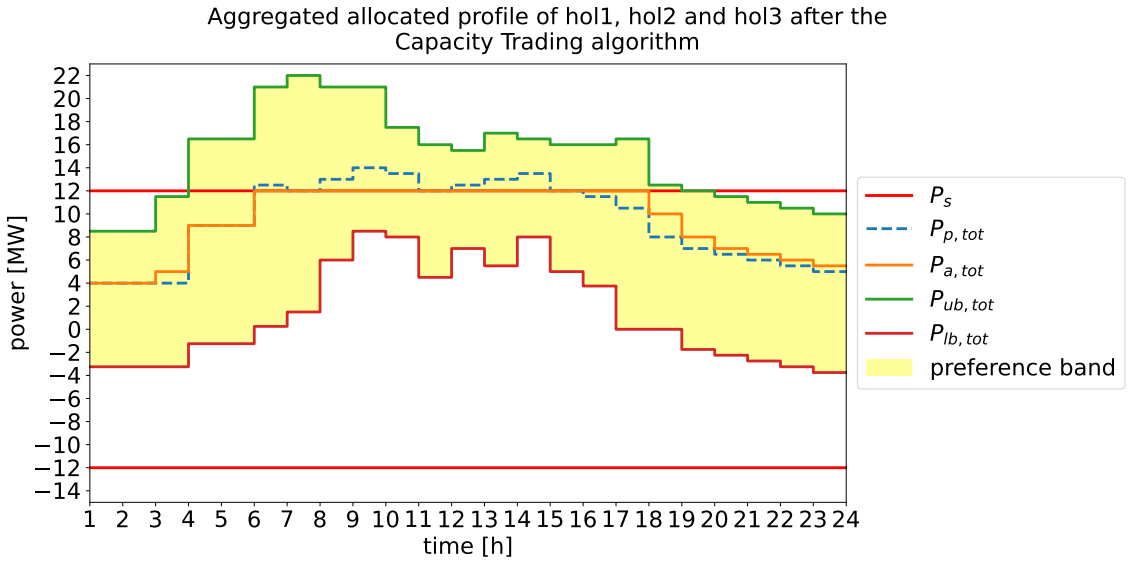


Figure 5.18: Aggregated power profile after the Capacity Trading algorithm for scenario 2.

Table 5.4: Results of the Developed algorithms for scenario 2 compared to profile steering.

		% preference met	% out of bounds	% deficit
Developed Algorithms	**hol1**	95.6	0.0	0.0
	hol2	86.7	0.0	0.0
	hol3	92.6	0.0	0.0
Profile Steering	**hol1**	83.4	0.5	0.0
	hol2	100	0.0	0.0
	hol3	89.4	0.0	0.0

Looking at the convergence for this scenario shown in Figure 5.19, the developed algorithms now fully converge in three iterations compared to the two iterations from scenario 1. This is due to the single trade that occurs between *hol*1 and *hol*3. This also results in the execution time to be slightly higher compared to the scenario 1: 0.68 seconds.

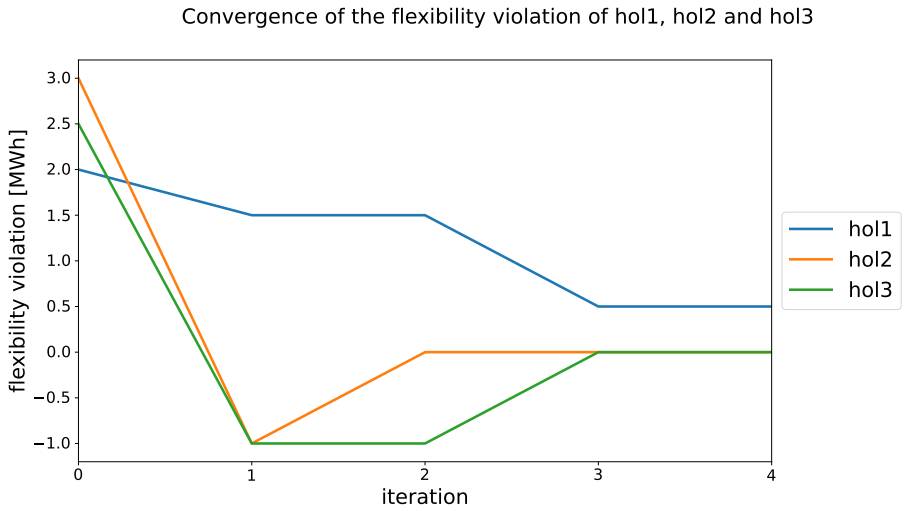


Figure 5.19: Convergence graph of scenario 2, where it takes one extra iteration to converge compared to scenario 1.

For the third scenario (see Table 5.2), Figure 5.20 still shows *hol*3 with a deficit. This time however, it is not able to mitigate its violation by only trading with one holon. By trading with both *hol*1 and *hol*2, *hol*3 is able to trade the full amount of the violation. The resulting $SoC_h(t)$ graph for the holons is shown in Figure 5.21, where it is also compared to the profile steering algorithm.

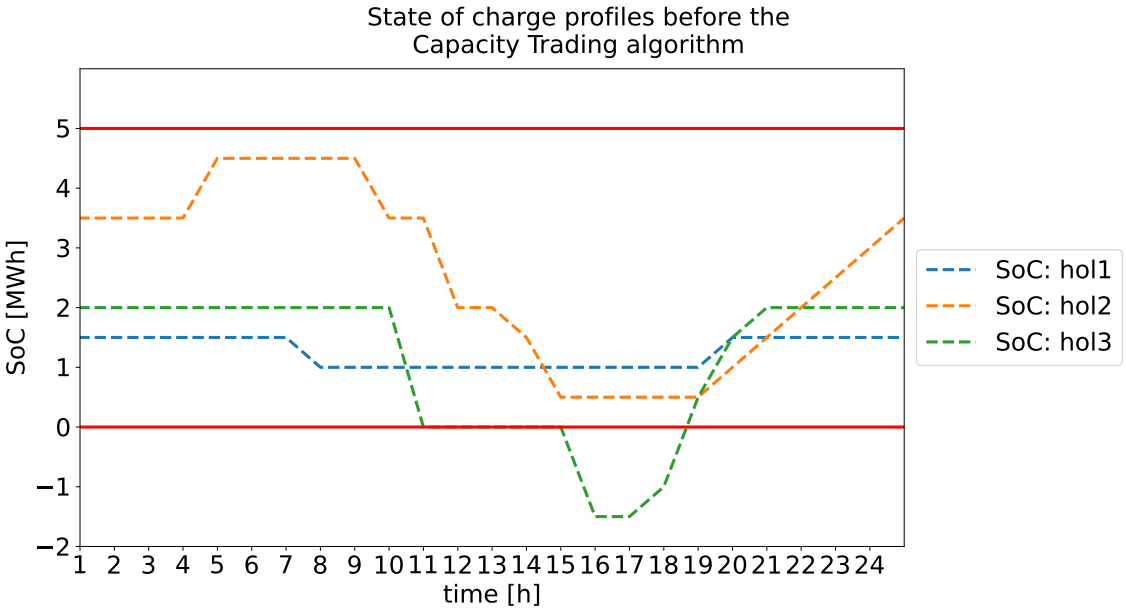


Figure 5.20: Profiles of the $SoC_h(t)$ for scenario 3 before the Capacity Trading algorithm.

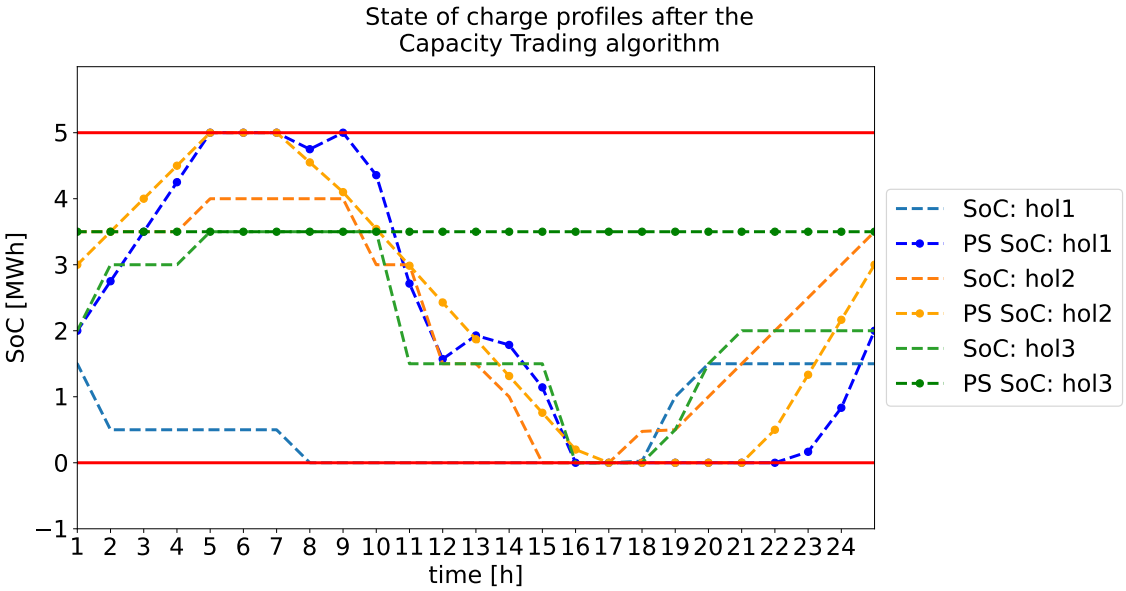


Figure 5.21: Profiles of the $SoC_h(t)$ for scenario 3 after the Capacity Trading algorithm (in filled lines) versus the $SoC_h(t)$ of the profile steering algorithm (in dashed lines).

The final allocated profiles for each holon, combined with the final allocated profiles from the profile steering algorithm are shown in Figure 5.22, as well as the aggregated power profile in Figure 5.23. Figure 5.23 indicates that the profiles produced are still feasible, however Table 5.5 shows that both algorithms now exceed the bounds at some point. However, the % preference met metric is still more grouped together for the developed algorithms compared to the profile steering algorithm.

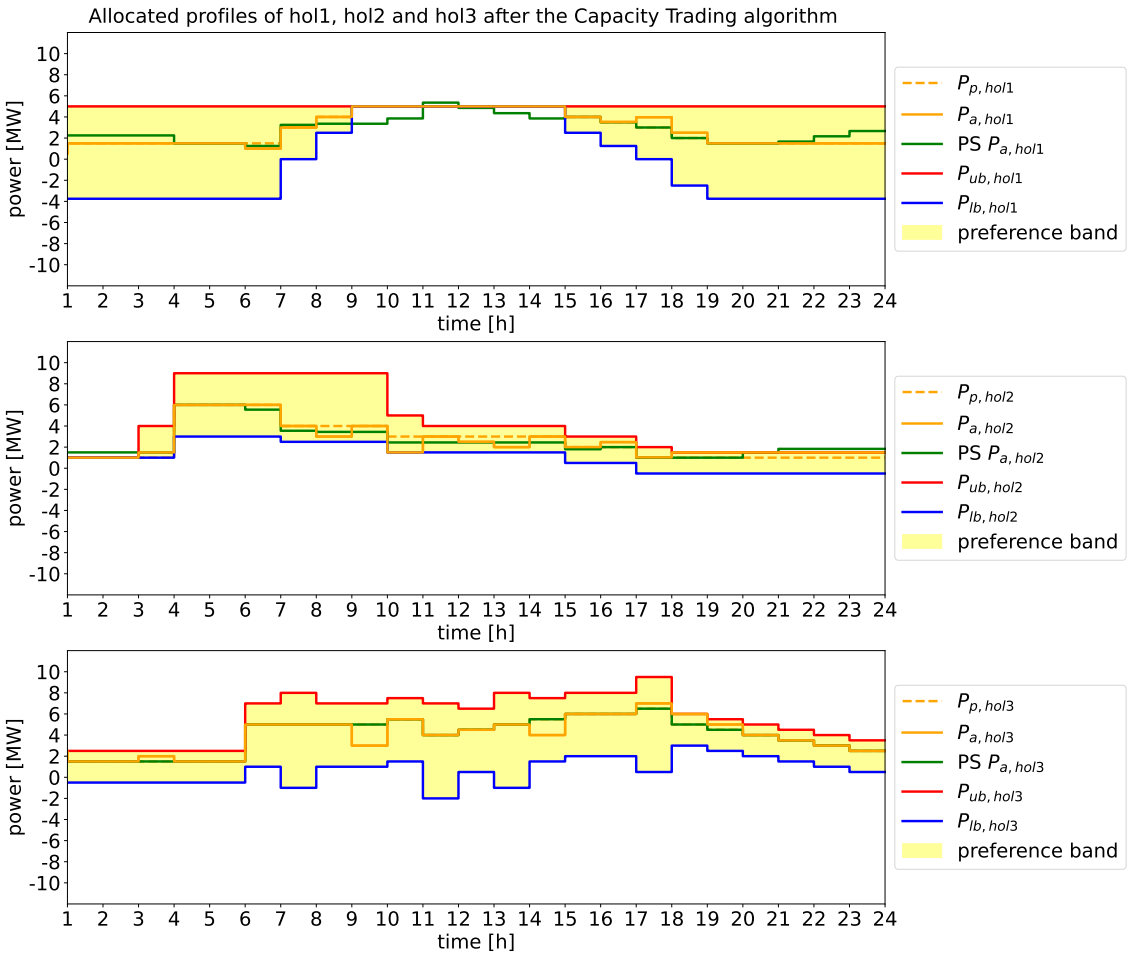


Figure 5.22: Allocated profiles after the Capacity Trading algorithm for scenario 3, along with the profile steering algorithm results.

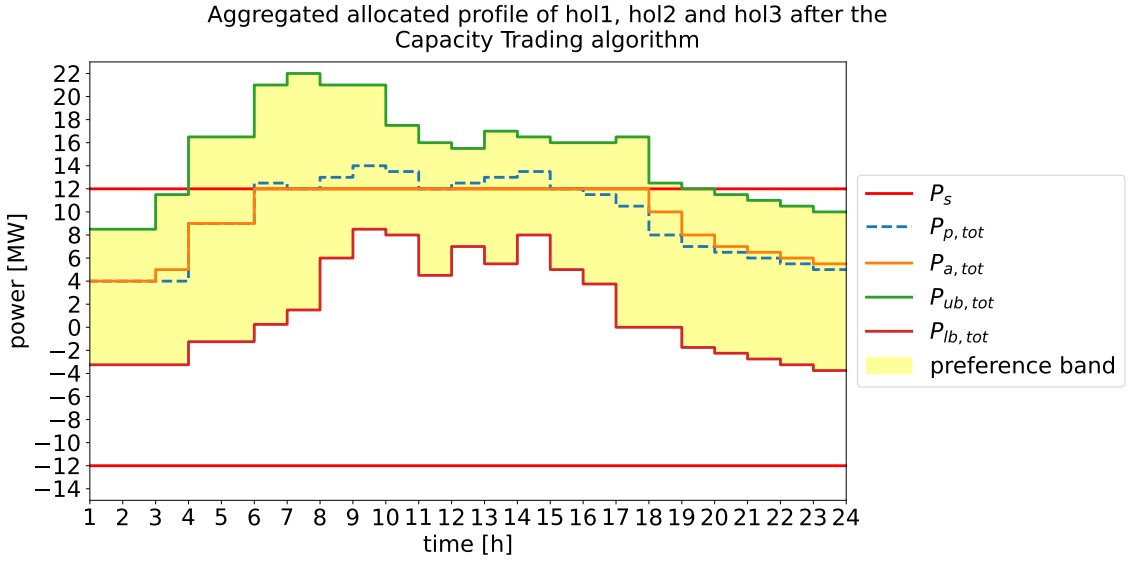


Figure 5.23: Aggregated power profile after the Capacity Trading algorithm for scenario 3.

Table 5.5: Results of the Developed algorithms for scenario 3 compared to profile steering.

		% preference met	% out of bounds	% deficit
Developed Algorithms	**hol1**	92.6	0.0	0.0
	hol2	86.7	0.9	0.0
	hol3	92.6	0.5	0.0
Profile Steering	**hol1**	100	0.0	0.0
	hol2	83.3	4.2	0.0
	hol3	88.1	0.2	0.0

The main downside with the developed algorithms is that the amount of iterations increases rapidly if holons need to trade in smaller quantities. This scenario illustrates that issue. Since *hol*3 needs to trade in small quantities, the amount of iterations it takes for the algorithms to converge increases. This is also represented in Figure 5.24, where nothing happens for example in between iterations 2 and 9, since the trade fails constantly and *hol*3 reduces its infraction (ϵ) to trade with in a stepwise manner. This results in the system converging in 14 iterations. This is also reflected in the execution time, since the average time to fully execute the program rose to 0.71 seconds.

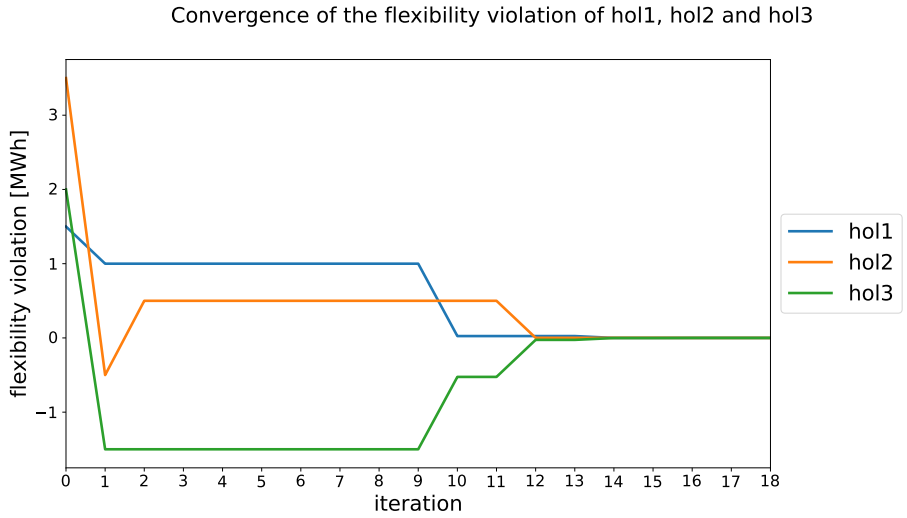


Figure 5.24: Convergence graph of scenario 3, where the amount of iterations to converge rapidly increased.

5.3.2 Ecofactorij

Curtailment and Reallocation Results

The aggregated power (P_{tot}) profile of the Ecofactorij after the Curtailment and Reallocation algorithm is shown in Figure 5.25.

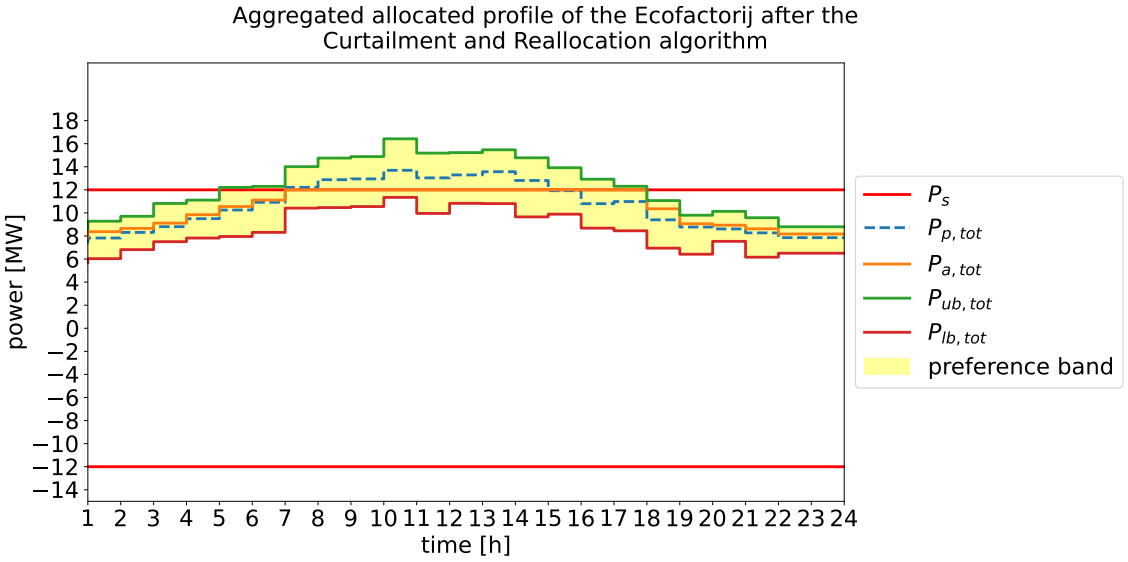


Figure 5.25: Aggregated allocated profile with the preference profile and the substation limits.

Since there are more holons compared to the small-scale scenario, the metrics are summarized in a mean and standard deviation. This results in showing both the accuracy and precision of the algorithms. The metrics are summarized in Table 5.6 and illustrate two observations.

The first observation is that the preference met for each holon is around the same region: the standard deviation is low with this increased amount of holons. The second observation is that the deficits are not 0. Even though it appears that there still is capacity in the grid to be allocated according to Figure 5.25, the holons with a deficit have reached their upper bounds during those moments where there is still capacity available in the grid. The functionality of the reallocation stage of the Curtailment and Reallocation algorithm only reallocate capacity of the holons if they have not reached their upper bound (which is used in (4.13)). Since their upper bounds have been reached during the moments where capacity can be reallocated, that capacity will remain unutilized.

Table 5.6: Results of the Curtailment and Reallocation algorithm for the Ecofactorij.

	% preference met	**% out of bounds**	**% deficit**
Mean	90.3	0.0	-3.1
Standard Deviation	5.9	0.0	3.8

In order to determine the flexibility violation, a new parameter sweep has been executed. However, the parameter sweeps are not executed for each holon, since the Ecofactorij consists of 24 holons compared to 3. Hence, only the two holons with the highest power usage have been selected to perform a parameter sweep, them being $hol1$ and $hol7$. The other holons have their initial state of charge, $SoC_h(t = 1)$, set to 0.5 MWh. As for the

capacity ζ_{max} of each holon, they are set to 3 MWh, since the individual power usage is lower compared to the individual power usage of the holons in the small-scale scenario. The results of this parameter sweep is shown in Figure 5.26, which include the minimum value of all the holons.

	$SoC_{hol7}(t=1)$ [MWh]										
$SoC_{hol1}(t=1)$ [MWh]	hol1\hol7	0	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3
	0	-2252.86	-1952.86	-1652.86	-1607.99	-1607.99	-1607.99	-1607.99	-1607.99	-1607.99	-1607.99	-1607.99
	0.3	-2252.86	-1952.86	-1652.86	-1352.86	-1307.99	-1307.99	-1307.99	-1307.99	-1307.99	-1307.99	-1307.99
	0.6	-2252.86	-1952.86	-1652.86	-1352.86	-1052.86	-1007.99	-1007.99	-1007.99	-1007.99	-1007.99	-1007.99
	0.9	-2252.86	-1952.86	-1652.86	-1352.86	-1052.86	-752.86	-707.99	-707.99	-707.99	-707.99	-707.99
	1.2	-2252.86	-1952.86	-1652.86	-1352.86	-1052.86	-752.86	-452.86	-407.99	-407.99	-407.99	-407.99
	1.5	-2252.86	-1952.86	-1652.86	-1352.86	-1052.86	-752.86	-452.86	-152.86	-107.99	-107.99	-107.99
	1.8	-2252.86	-1952.86	-1652.86	-1352.86	-1052.86	-752.86	-452.86	-152.86	-16.95	-16.95	-16.95
	2.1	-2252.86	-1952.86	-1652.86	-1352.86	-1052.86	-752.86	-452.86	-152.86	-16.95	-16.95	-16.95
	2.4	-2252.86	-1952.86	-1652.86	-1352.86	-1052.86	-752.86	-452.86	-152.86	-16.95	-16.95	-16.95
	2.7	-2252.86	-1952.86	-1652.86	-1352.86	-1052.86	-752.86	-452.86	-152.86	-16.95	-16.95	-16.95
	3	-2252.86	-1952.86	-1652.86	-1352.86	-1052.86	-752.86	-452.86	-152.86	-16.95	-16.95	-16.95

Figure 5.26: Heat map of the flexibility violation after the Curtailment and Reallocation algorithm. The values in the heat map are the minimum values of every holon.

Capacity Trading Results

The same parameter sweep has again been performed to evaluate the effect of the Capacity Trading algorithm on the feasibility of the power profiles. The result after trading is shown in Figure 5.27.

SoChol1(t=1) [MWh]	hol1\hol7	SoChol7(t=1) [MWh]										
		0	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3
	0	-2252.86	-1896.43	-1467.71	-1293.66	-894.26	-598.82	-272.08	0	0	0	0
	0.3	-1953.85	-1460.77	-1312.81	-918.04	-598.82	-272.08	0	0	0	0	0
	0.6	-1465.53	-1312.81	-942.17	-598.82	-272.08	0	0	0	0	0	0
	0.9	-1312.81	-881.63	-598.82	-272.08	0	0	0	0	0	0	0
	1.2	-904.24	-598.82	-272.08	0	0	0	0	0	0	0	0
	1.5	-598.82	-272.08	0	0	0	0	0	0	0	0	0
	1.8	-272.08	0	0	0	0	0	0	0	0	0	0
	2.1	0	0	0	0	0	0	0	0	0	0	0
	2.4	0	0	0	0	0	0	0	0	0	0	0
	2.7	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0

(a) Heat map of the flexibility violation.

SoChol1(t=1) [MWh]	hol1\hol7	SoChol7(t=1) [MWh]										
		0	0.3	0.6	0.9	1.2	1.5	1.8	2.1	2.4	2.7	3
	0	85.55	85.17	83.01	85.14	83.57	83.4	83.13	83.6	87.13	88.49	89.14
	0.3	85.63	82.97	85.18	83.79	83.62	83.17	83.57	86.75	88.42	89.2	89.54
	0.6	83.04	85.44	84.15	83.22	83.18	83.36	87.46	88.57	88.94	89.56	89.88
	0.9	85.35	83.29	83.78	83.19	83.9	87.27	88.48	89.21	89.58	89.9	90.19
	1.2	83.71	83.4	83.22	83.97	87.26	88.15	89.37	89.32	89.69	90.2	90.2
	1.5	83.36	83.1	83.38	86.96	87.9	88.84	89.61	89.45	89.82	89.82	89.82
	1.8	83.2	83.52	87.3	88.38	88.86	89.78	90.04	90.24	90.25	90.25	90.25
	2.1	83.08	86.91	88.28	88.99	89.66	89.74	90.12	90.15	90.16	90.16	90.16
	2.4	85.91	88.1	88.78	89.46	89.55	89.92	89.97	90	90.01	90.01	90.01
	2.7	88.02	88.7	88.18	89.11	89.22	89.27	89.32	90.05	90.04	90.01	89.97
	3	88.94	89.63	89.6	89.98	90.03	90.07	90.1	90.13	89.88	89.97	90.25

(b) Heat map of the preference met.

Figure 5.27: Heat maps for parameter sweep of $SoC_h(t = 1)$ for $hol2$ and $hol3$, where $SoC_h(t = 1)$ for the other holons is set to 2 MWh.

For the comparison with the profile steering algorithm, $SOC_{hol1}(t = 1)$ =0.9 MWh and $SoC_{hol7}(t = 1)$ =1.2 MWh, since they are as low as possible while still resulting in energetically feasible power profiles according to Figure 5.27. Figure 5.28 shows the profiles of the batteries connected to each of the holons before the Capacity Trading algorithm and Figure 5.29 shows the profiles of the batteries after the Capacity Trading algorithm. Figures 5.28 and 5.29 also show that trading occurs between several holons to successfully mitigate the flexibility violations.

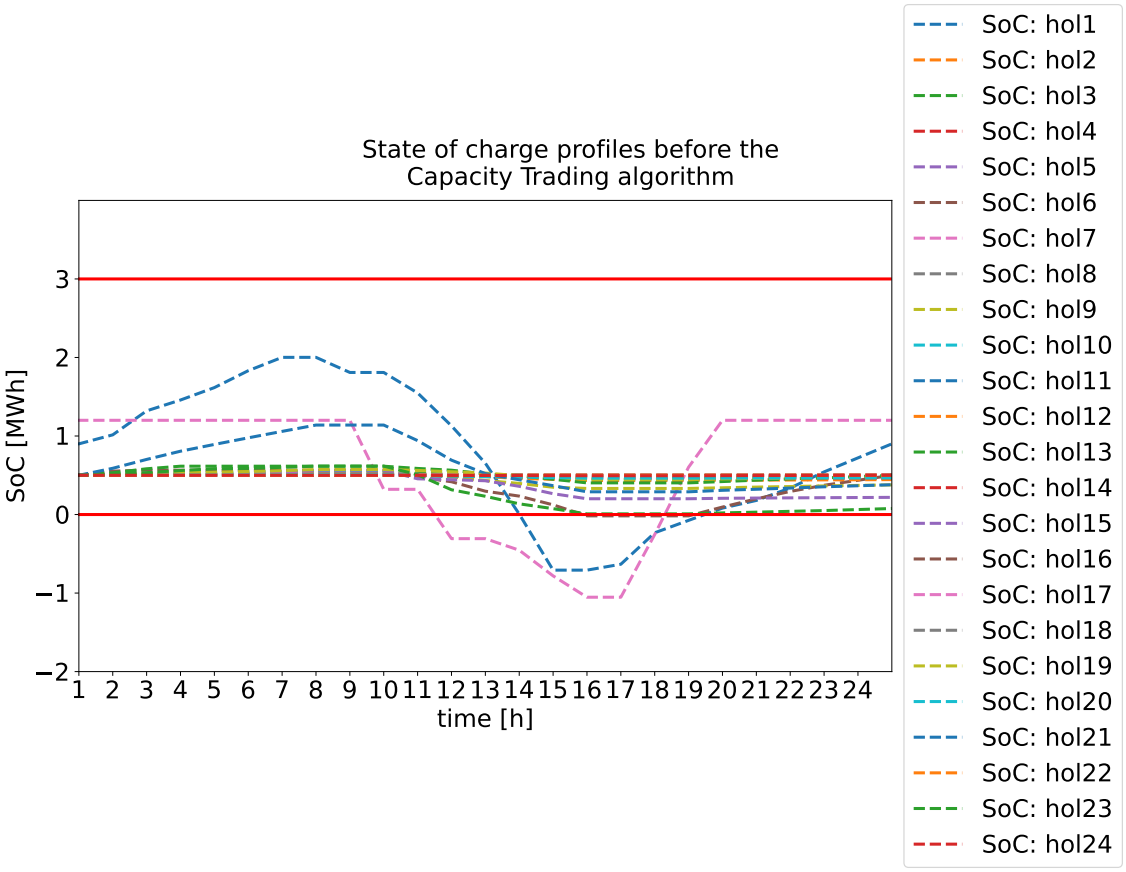


Figure 5.28: Profiles of the $SoC_h(t)$ of the Ecofactorij before the Capacity Trading algorithm.

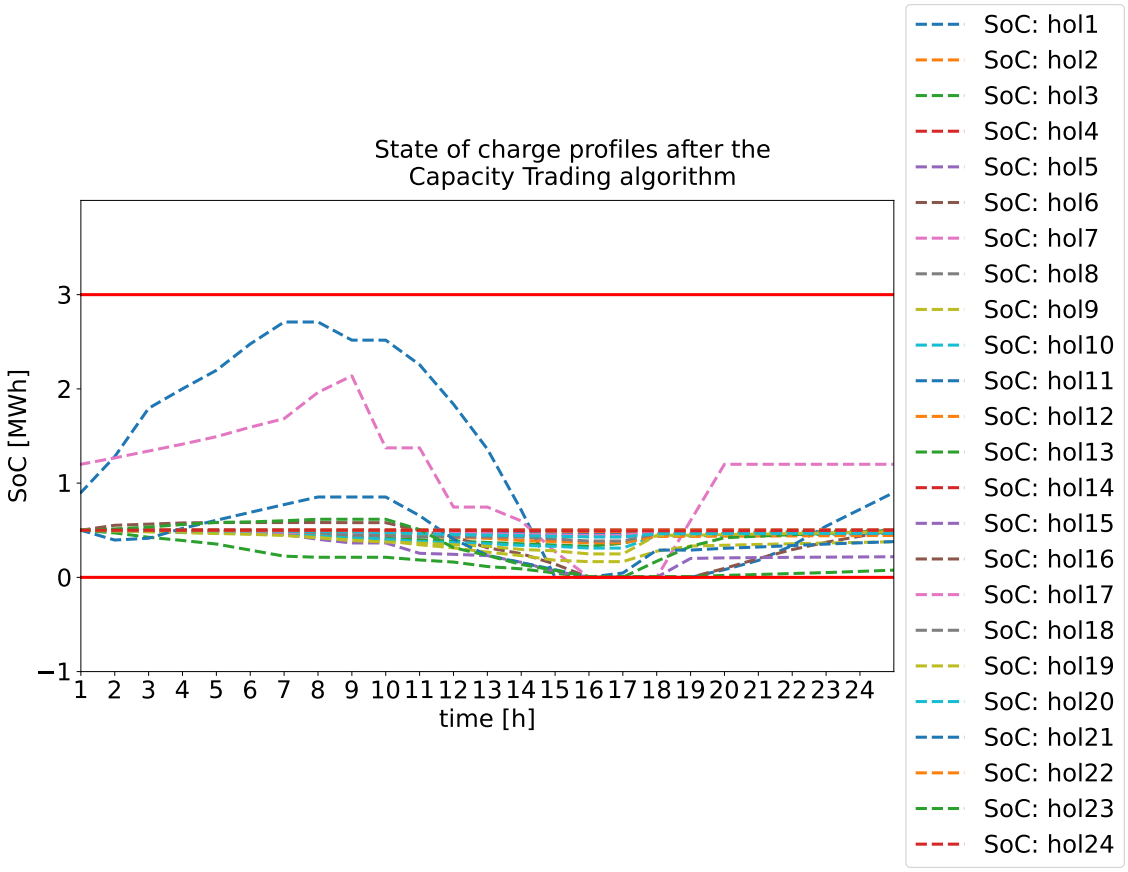


Figure 5.29: Profiles of the $SoC_h(t)$ of the Ecofactorij after the Capacity Trading algorithm.

With this scenario of the Ecofactorij, the metrics are again summarized in the mean and standard deviation and compared to the profile steering results in Table 5.7.

Table 5.7: Results of the Developed algorithms for the Ecofactorij compared to profile steering.

		% preference met	**% out of bounds**	**% deficit**
Developed Algorithms	**Mean**	83.9	4.3	-3.1
	Standard Deviation	11.4	4.4	3.8
Profile Steering	**Mean**	20.8	8.6	0.0
	Standard Deviation	41.5	28.2	0.0

Table 5.7 shows that the developed algorithms focused more on the user satisfaction aspect compared to profile steering. The main reason behind this is the nature of both algorithms: profile steering attempts to find the best solution in as little iterations as possible by asking one holon at a time to attempt to mitigate the congestion as much as possible. The developed algorithms on the other hand take more time to find a solution where it attempts to treat each holon the same. This is mainly shown in both the % preference met and % out of bounds. The mean for both metrics are in favor of the developed algorithms, with a way higher % preference met and a % out of bounds half

of the % out of bounds of profile steering. Furthermore, the standard deviation for the developed algorithms is much lower compared to the standard deviations of the profile steering algorithm, indicating a higher precision and thus a more evenly distributed solution for the grid congestion. On the other hand, the % deficit is worse compared to profile steering. Since profile steering does not take the bounds into account, the limit imposed by (4.13) on the developed algorithms results in a worse performance regarding that metric.

When we look at the amount of iterations needed in Figure 5.30, it indicates that the developed algorithms require way more iterations (179) to fully converge compared to the small-scale scenarios. It also shows the reason behind the amount of iterations: several holons are only able to trade a limited amount of capacity (mainly $hol7$) with several other holons, resulting in many iterations where they are unable to trade. In total, 20 trades have successfully commenced. Of these 20 trades, 3 have been initialized by $hol1$, 1 by $hol6$ and the remaining 16 by $hol7$, which are the three companies with the highest power profile in the Ecofactorij scenario. Since they have the same batteries as the other holons and their upper and lower bounds ($P_{ub,h}(t)$ and $P_{lb,h}(t)$ respectively) are further apart, they utilize their batteries more in case of congestion and can thus result in violations more often. The holons they traded with are mainly the companies with lower power profiles, since they have the most amount of charge in their batteries: companies with lower power profiles have smaller preference bounds, hence they have less deviation for the battery to compensate for. This results in the battery being utilized less, hence resulting in a higher SoC (since each battery is the same). This allows for more trading with other holons.

Due to the increase in number of trades and trading in smaller quantities, the execution time drastically increased compared to the small-scale scenarios: 1.61 seconds.

5.4 Conclusion

The main takeaway from these results is that the developed algorithms, the Curtailment and Reallocation and the Capacity Trading algorithms, have a main goal of providing for each holon equally. I.e., each holon has around the same amount of preference met and the same amount within the preference bounds percentage wise. As a result, it does perform better in those areas compared to the profile steering algorithm, which attempts to solve as much as possible with the least amount of holons involved in the process. On the other side, especially the Capacity Trading algorithm requires a lot of iterations to fully execute if trading in smaller quantities is needed to solve for the flexibility violations. This is illustrated well with the Ecofactorij use case, where the combined amount of iterations from the Curtailment and Reallocation and the Capacity Trading algorithms was 179 in order to converge. This is also represented in the execution time, where the Ecofactorij scenario took approximately 1 second longer compared to the small-scale scenario 1.

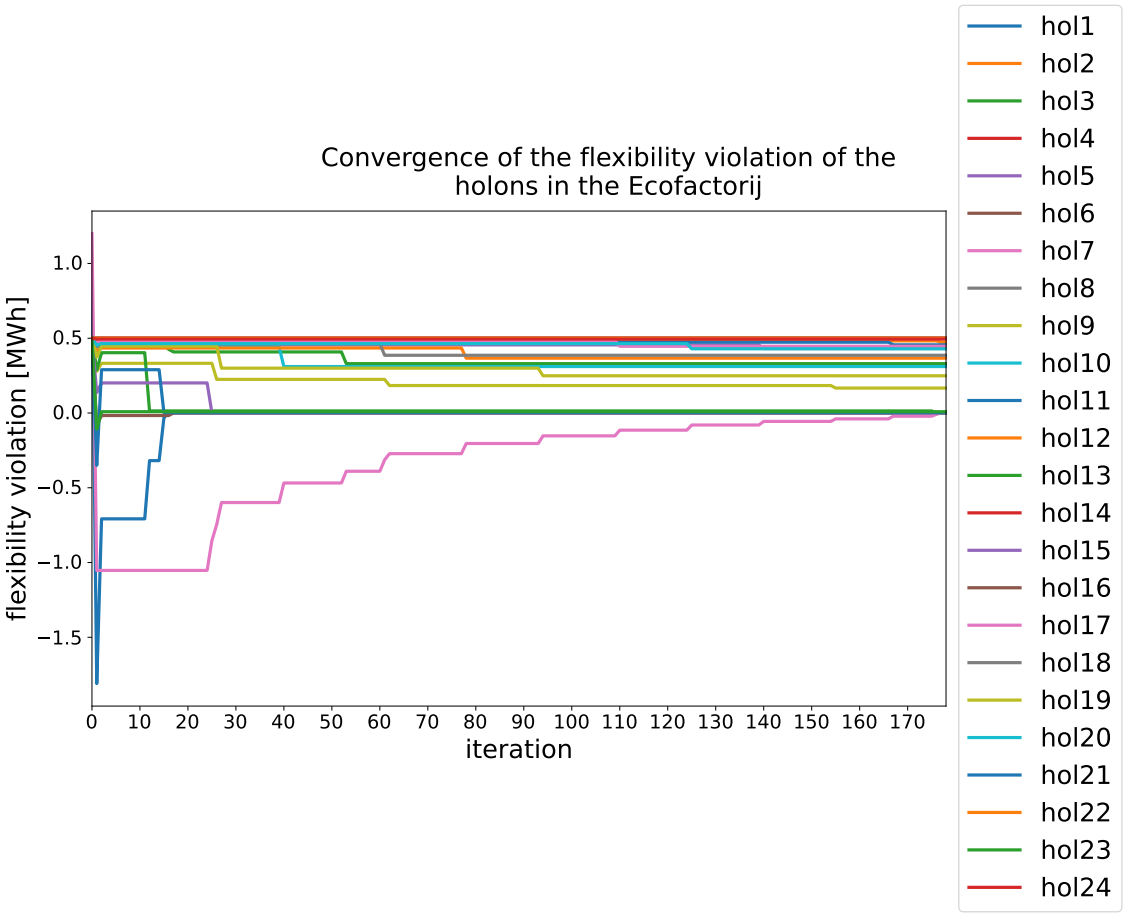


Figure 5.30: Convergence graph of the Ecofactorij, where the amount of iterations greatly increases mostly due to the trading of *hol*7.

Chapter 6

Conclusions and Future Work

The work in this thesis can be used to fully answer the research questions formulated in Chapter 1. This chapter starts with answering each of the research questions, after which possible future work is discussed.

6.1 Research Questions

In order to answer the main research question, we first answer the sub-questions. These answers will then form the foundation in order to answer the main question.

Which algorithms can be used for the decentralized allocation of resources in a holarchy and how do they perform?

Chapter 2 described a variety of different resource allocation algorithms that can be utilized in a decentralized manner. These include algorithms already incorporated in smart grids, such as price steering, profile steering and model predictive control. However, we have also taken a look at resource allocation algorithms from different fields, such as First-Come, First-Serve, Shortest Job First and Round Robin from the field of real-time systems and a branch-and-bound algorithm from supply chain management. These algorithms vary from centralized to decentralized and even distributed. Regarding congestion management for decentralized systems, the centralized algorithms quickly become unusable due to their poor scalability.

Furthermore, the incentives for reallocation algorithms need to be carefully considered. The main reason behind this is that it can lead to peak shifting (for example with price steering) instead of peak shaving. Peak shifting does not solve for grid congestion at all, hence it has to be avoided.

What are the benefits and drawbacks of implementing a holarchy in a smart grid compared to the state of the art?

The main benefit of implementing a holarchy is the combination between a decentralized and distributed system, which is highly modular and hence also future-proof. Literature has shown that this modularity also results in a more scalable and robust communication and coordination topology. However, setting up a holarchy with a high modularity can quickly evolve in a complex system. Furthermore, a holarchy brings along more coordina-

tion challenges when compared to centralized approaches, since each decentralized aspect has to communicate with each other instead of a centralized controller making all the decisions. This was also proven in the results in Section 5.3.2, where the amount of iterations the algorithm took to converge for the Ecofactorij scenario increased rapidly compared to the small-scale scenario.

How much flexibility is needed in order for the holarchy paradigm to have a beneficial effect?

The answer is that it depends on the environment. For the small-scale scenario, it appears that all holons equally contribute to solving for the flexibility violations. However, varying the initial state of charge for hol1 (especially for lower initial state of charges for hol2 and hol3) decreases the % preference met metric the most. So even though the flexibility violation might be solved, it is better to have a higher initial state of charge for hol2 and hol3 compared to hol1. The resulting profiles are energetically feasible if one of the holons has an initial state of charge of 5 MWh, one has an initial state of charge of 2 MWh and the final holon is drained. If that final holons has a higher initial state of charge, the initial state of charge of the other holons can be reduced and still result in a energetically feasible power profile.

For the Ecofactorij, this is not the case, since varying hol1 and hol7 does not result in a clear difference. It does show that the batteries do help mitigating the flexibility violations. For example, the flexibility violations are fully mitigated when the battery of either hol1 or hol7 has an initial state of charge of 2.1 MWh and the initial state of charge of the battery of the other holon is 0 MWh.

How can decentralized resource allocation be implemented in a holarchical smart grid, in which flexibility can be exploited to guarantee a feasible power profile?

To answer this research question, two algorithms have been developed that work together to solve grid congestion. The Curtailment and Reallocation algorithm has the goal of preventing grid congestion using the available flexibility as much as possible, whereas the Capacity Trading algorithm allows for capacity trading between holons in the holarchy. This capacity trading is used to mitigate flexibility violations caused by the first algorithm. Furthermore, the capacity trading is designed in such a way that the holons are able to share a limited amount of data with each other and still solve for their flexibility violations.

Additionally, profile steering attempts to solve the grid congestion is as few iterations as possible by asking one profile at a time to provide a new profile. This results in profile steering skewing more towards a single holon in the network to prevent grid congestion as much as possible, resulting in a lower user satisfaction. With the developed algorithms, the main priority is to treat each holon as evenly as possible. This is also reflected in the results of the Ecofactorij in Section 5.3.2, where the mean % preference met for the developed algorithms is 83.9% with a standard deviation of 11.4% for the developed algorithms. This is much better compared to profile steering, where the mean is 20.8% with a standard deviation of 41.5%.

6.2 Future Work

The main point to improve on is the scalability of the design. Since the amount of iterations the system needs to fully converge can increase rapidly with a slight increase in input complexity (the amount of iterations needed heavily depends on how many trades are needed and if the holons are unable to mitigate their violations in one trade), alternative design choices can be made. For example, if the holons returned how much flexibility they have to offer when a trade fails, the holon requiring the capacity is able to exactly compute with whom to trade a certain amount of capacity to solve for its flexibility violations. This does require more data to be transferred, hence resulting in the overall system being less privacy sensitive. Furthermore, the simulations only included batteries as their flexibility sources. In future research, different types of flexibilities can be introduced to evaluate their effect in a holarchy. For example with profile steering, where heat pumps and electric vehicles can also be used to prevent grid congestion.

Lastly, the holarchy structure can be expanded. The simulations revolved around a single group of holons, but future research can also evaluate the effect of having this at multiple levels in a holarchy. For example, if the holons at a higher level divide the capacity amongst each other first and the holons at a lower level are able to subdivide that capacity again. With this approach, the implementation can be made more scalable and more future proof.

Bibliography

[1] NASA. Extreme weather and climate change. https://science.nasa.gov/climate-change/extreme-weather/. Accessed: 26-04-2024.

[2] United Nations Environment Programme. Paris agreement. https://wedocs.unep.org/20.500.11822/20830, 12-12-2015. Accessed: 25-04-2024.

[3] Government of the Netherlands. Climate policy. https://www.government.nl/topics/climate-change/climate-policy. Accessed: 07-05-2024.

[4] Government of the Netherlands. Central government encourages sustainable energy. https://www.government.nl/topics/renewable-energy/central-government-encourages-sustainable-energy. Accessed: 25-04-2024.

[5] TenneT. Electricity grid under further pressure cabinet and grid operators take drastic measures. https://www.tennet.eu/news/electricity-grid-under-further-pressure-cabinet-and-grid-operators-take-drastic-measur 2023. Accessed: 07-05-2024.

[6] Léone Klapwijk and Noortje van Bergeijk. Alternative and flexible transmission capacity rights: a step closer to a more efficient use of the electricity grid in the netherlands. https://www.vandoorne.com/en/artikelen/alternative-and-flexible-transmission-capacity-rights-a-step-closer-to-a-more-efficien Accessed: 23-05-2024.

[7] TenneT. Security of supply. https://www.tennet.eu/nl/node/1905. Accessed: 22-05-2024.

[8] Kennisplatform. Uit welke onderdelen bestaat het elektriciteitsnetwerk? (Dutch) [What components make up the electricity network?]. https://www.kennisplatform.nl/uit-welke-onderdelen-bestaat-het-elektriciteitsnetwerk/. Accessed: 01-05-2024.

[9] Netbeheer Nederland. Basisdocument over energie-infrastructuur (Dutch) [Basic information about energy infrastructure]. https://www.netbeheernederland.nl/sites/default/files/Basisdocument_over_energie-infrastructuur_%2528oktober_2019%2529_161.pdf. Accessed: 01-07-2024.

[10] CFP. Grid congestion: what is it and how can you avoid it? https://cfp.nl/en/news-and-cases/grid-congestion-what-is-it-and-how-can-you-avoid-it/, 2023. Accessed: 14-05-2024.

[11] Netbeheer Nederland. Capaciteitskaart elektriciteitsnet (Dutch) [Capacity Map of the Electricity Grid]. https://capaciteitskaart.netbeheernederland.nl, 2024. Accessed: 14-05-2024.

[12] Editorial Noordoostpolder Nieuws. Pilots met tijdsgebonden contracten in Amsterdam, Alkmaar en Lelystad (Dutch) [Pilots with time-bound contracts in Amsterdam, Alkmaar and Lelystad]. https://noordoostpolder.nieuws.nl/nieuws/37035/pilots-met-tijdsgebonden-contracten-in-amsterdam-alkmaar-en-lelystad/, 07 2022. Accessed: 25-04-2024.

[13] Kiwatt. Wat houdt een aansluit- en transportovereenkomst (non-firm ATO) in? (Dutch) [What does a connection and transport agreement (non-firm ATO) entail]. https://www.kiwatt.nl/nieuws/non-firm-ato/. Accessed: 15-05-2024.

[14] quanalysus. Choosing between centralized, decentralized, and distributed networks. https://steemit.com/cryptocurrency/@quantalysus/choosing-between-centralized-decentralized-and-distributed-networks. Accessed: 22-05-2024.

[15] Ebisa Olana Negeri. *Smart Power Grid: A Holonic Approach*. PhD thesis, Technische Universiteit Delft, 2014.

[16] Arthur Koestler. *The ghost in the Machine: Arthur Koestler*. Hutchinson, 1967.

[17] TenneT. Congestion management. https://netztransparenz.tennet.eu/electricity-market/dutch-market/transmission-capacity/congestion-management/. Accessed: 21-05-2024.

[18] Shaojun Huang, Qiuwei Wu, Zhaoxi Liu, and Arne Hejde Nielsen. Review of congestion management methods for distribution networks with high penetration of distributed energy resources. In *IEEE PES Innovative Smart Grid Technologies, Europe*, pages 1–6, 2014.

[19] Marco E. T. Gerards, Hermen A. Toersche, Gerwin Hoogsteen, Thijs van der Klauw, Johann L. Hurink, and Gerard J. M. Smit. Demand side management using profile steering. In *2015 IEEE Eindhoven PowerTech*, pages 1–6, 2015.

[20] Killian McKenna and Andrew Keane. Discrete elastic residential load response under variable pricing schemes. In *IEEE PES Innovative Smart Grid Technologies, Europe*, pages 1–6, 2014.

[21] Thijs van der Klauw. *Decentralized Energy Management with Profile Steering: Resource Allocation Problems in Energy Management*. Phd thesis - research ut, graduation ut, University of Twente, Netherlands, May 2017. CTIT Ph.D. thesis series no. 17-424.

[22] Ioannis Kalogeropoulos and Haralambos Sarimveis. Predictive control algorithms for congestion management in electric power distribution grids. *Applied Mathematical Modelling*, 77:635–651, 2020.

[23] Benjamin Biegel, Palle Andersen, Jakob Stoustrup, and Jan Bendtsen. Congestion management in a smart grid via shadow prices. *IFAC Proceedings Volumes*, 45(21):518–523, 2012. 8th Power Plant and Power System Control Symposium.

[24] Rolf Egert, Tim Grube, Florian Volk, and Max Mühlhäuser. Holonic system model for resilient energy grid operation. *Energies*, 14(14), 2021.

[25] Adriano Ferreira and Paulo Leitão. Holonic self-sustainable systems for electrical micro grids. In *2016 IEEE 14th International Conference on Industrial Informatics (INDIN)*, pages 510–515, 2016.

[26] Anil Pahwa, Scott A. DeLoach, Bala Natarajan, Sanjoy Das, Ahmad R. Malekpour, S M Shafiul Alam, and Denise M. Case. Goal-based holonic multiagent system for operation of power distribution systems. *IEEE Transactions on Smart Grid*, 6(5):2510–2518, 2015.

[27] Stefan Dähling, Sonja Kolen, and Antonello Monti. Swarm-based automation of electrical power distribution and transmission system support. *IET Cyber-Physical Systems: Theory & Applications*, 3(4):212–223, 2018.

[28] Sonja Kolen, Timo Isermann, Stefan Dähling, and Antonello Monti. Swarm behavior for distribution grid control. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6, 2017.

[29] Alireza Ashrafi and S. M. Shahrtash. Dynamic wide area voltage control strategy based on organized multi-agent system. *IEEE Transactions on Power Systems*, 29(6):2590–2601, 2014.

[30] Alireza Ashrafi and S.M. Shahrtash. The effect of negotiation strategy on arrangement of agents in a multi-agent voltage control system. In *2013 Smart Grid Conference (SGC)*, pages 181–188, 2013.

[31] Siavash Valipour, Florian Volk, Tim Grube, Leon Böck, Ludwig Karg, and Max Mühlhäuser. A formal holon model for operating future energy grids during blackouts. In *2016 5th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*, pages 1–8, 2016.

[32] Mohamed F. Abdel-Fattah, Hannah Kohler, Peter Rotenberger, and Leonard Schöler. A review of the holonic architecture for the smart grids and the self-healing application. In *2020 21st International Scientific Conference on Electric Power Engineering (EPE)*, pages 1–6, 2020.

[33] Faizan Saeed, Nadeem Javaid, Muhammad Zubair, Muhammad Ismail, Muhammad Zakria, Muhammad Hassaan Ashraf, and Muhammad Babar Kamal. Load balancing on cloud analyst using first come first serve scheduling algorithm. In Fatos Xhafa, Leonard Barolli, and Michal Greguš, editors, *Advances in Intelligent Networking and Collaborative Systems*, pages 463–472, Cham, 2019. Springer International Publishing.

[34] Yanjun Shi, Hafiz Abdul Saboor, Lujun Wang, and Zihui Zhang. A shortest job first algorithm for minimizing the average delay of vehicles at the un-signalized intersection. In *2021 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pages 395–400, 2021.

[35] Hayatunnufus, Mardhani Riasetiawan, and Ahmad Ashari. Performance analysis of fifo and round robin scheduling process algorithm in iot operating system for collecting landslide data. In *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)*, pages 63–68, 2020.

[36] Morteza Rasti-Barzoki, Seyed Reza Hejazi, and Mohammad Mahdavi Mazdeh. A branch and bound algorithm to minimize the total weighed number of tardy jobs and delivery costs. *Applied Mathematical Modelling*, 37(7):4924–4937, 2013.

[37] Sajede Aminzadegan, Mohammad Tamannaei, and Morteza Rasti-Barzoki. Multi-agent supply chain scheduling problem by considering resource allocation and transportation. *Computers Industrial Engineering*, 137:106003, 2019.

[38] David R. Morrison, Sheldon H. Jacobson, Jason J. Sauppe, and Edward C. Sewell. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization*, 19:79–102, 2016.

[39] David Goldberg. What every computer scientist should know about floating-point arithmetic. In *ACM Computing Surveys*, volume 23, pages 5–48, 1991.

[40] Ecofactorij. Privaat Elektriciteitsnet (Dutch) [Private Electricity Grid]. https://ecofactorij.nl/eigen-elektriciteitsnet/, 2022. Accessed: 25-04-2024.

[41] ACM. Ontheffing netbeheer aanvragen (Dutch) [Request exemption network management]. https://www.acm.nl/nl/energie/elektriciteit-en-gas/netbeheer/eigen-netwerk-directe-lijn/ontheffing-netbeheer-aanvragen. Accessed: 20-06-2024.

[42] ACM. Eigen netwerk of directe lijn beheren (Dutch) [Regulate own network or direct line]. https://www.acm.nl/nl/energie/elektriciteit-en-gas/netbeheer/eigen-netwerk-directe-lijn/eigen-netwerk-directe-lijn-beheren. Accessed: 20-06-2024.

[43] Sparkling Projects. CO2 hoog temperatuur warmtepomp (Dutch) [CO2 high temperature heat pump]. https://www.sparklingprojects.nl/nieuws/co2-hoog-temperatuur-warmtepomp. Accessed: 20-06-2024.

[44] Sparkling Projects. Warmteaccu geplaatst (Dutch) [heat accumulator placed]. https://www.sparklingprojects.nl/nieuws/warmteaccu-geplaatst. Accessed: 20-06-2024.