Scheduling and Pricing Strategies for Prosumers in a Microgrid

Vom Promotionsausschuss der
Technischen Universität Hamburg
zur Erlangung des akademischen Grades

Doktorin der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation (Monografie)

von
Laura Codazzi

aus
Segrate, Italien

2026

Prof. Dr. Matthias Mnich
Prof. Dr. Marc Uetz

18 November 2025

ORCID iD: 0009-0004-2121-4347

ABSTRACT

We analyze fundamental scheduling and pricing problems with applications in energy trading. We consider agents in microgrids, wishing to optimize their cost by strategically buying, selling and storing energy in a battery, to satisfy their demand. They plan over a finite time horizon, where energy prices and demand may continuously change. We model several variants of this problem, and provide algorithms whose performance we analyze both theoretically and experimentally.

ZUSAMMENFASSUNG

Wir analysieren grundlegende Planungs- und Preisbildungsprobleme mit Anwendungen im Energiehandel. Wir betrachten Agenten in Mikronetzen, die ihre Kosten optimieren wollen, indem sie strategisch Energie kaufen, verkaufen und in einer Batterie speichern, um ihren Bedarf zu decken. Sie planen über einen endlichen Zeithorizont, in dem sich die Energiepreise und die Nachfrage ständig ändern können. Wir modellieren mehrere Varianten dieses Problems und stellen Algorithmen bereit, deren Leistung wir sowohl theoretisch als auch experimentell analysieren.

CONTENTS

INTRODUCTION

PROLOGUE

Imagine you are leaving for a multi-day hiking, which demands a high level of physical effort and the hot weather doesn't help either. One of the best pieces of advice you could get is to properly pack your backpack. On the one hand you want to make sure to bring with you objects and gear that are going to be helpful on the hike. On the other hand one of the most often repeated recommendation is to avoid overpacking, as a heavy backpack would certainly not help along the way. Many websites and guides suggest that you keep the weight of the knapsack below 10 kg and to make sure to do so, you—as a systematic person—decide to write down a table with all the objects you would like to bring with you and you report their weight as well as a value from 1 to 100 with the aim of evaluating how useful or necessary each object is. Now it should be easier to decide how to pack the backpack.

Everyone who is familiar with any Computer Science-related subject has surely already recognized by now the very famous KNAPSACK problem. Not only that, probably the word *object* has already been replaced with the one *item* and some notation is also emerging in their head. Most likely, if two people with these thoughts were to interact with each other about it, silently agreed words like *weights* and *capacity* would also be mentioned and they could make a rather long conversation about different heuristics, approximation algorithms or even some known extension of the problem.

Such rigor, or even codification if we want, is something extremely helpful in research, at least from my personal perspective. I have myself realized this once I started my PhD and I immersed myself in the broad field of *Energy Trading*. For reasons that are probably already clear to the reader, but will be nevertheless listed also later on in this work, such a field is intensively studied and significantly popular. The huge availability of data on the topic, as well as the emerging artificial intelligence technologies and the pressure from the industrial and political sides of the society, might have contributed to the continuous publication of these works. The result of this is that we do have an incredible amount of amazing research to learn and develop from, but a significant part of this literature has a very high dependency from the investigated setting and specific data of the given study.

This thesis arises from applying methods and techniques of the discipline of Theoretical Computer Science to the application domain of energy trading. We do not have the arrogance to affirm that the problems tackled in this work have the power to solve the current global energy crisis, nor to be taken from inspiration in any energy company. Our goal is that the results presented here contribute to a more systematic understanding of the application domain and spark further interest and research in the direction of mitigating its most important challenges. We further contribute to strengthening the already existing, but rather weak connection between problems that are typical of the Computer Science field and Energy Trading-related problems. Ultimately, we are convinced that advancing research in energy trading with the rigour and structure of the principles of Theoretical can contribute, even to some small extent, to addressing the urgent global climate crisis.

This work is divided into three main parts. We continue this first introductory part in Chapter 1 with a very general overview of the background theory which is required for the proper understanding of the results of this work and few remarks on the notation used. The second part is then dedicated to the first big problem that we tackle, namely the scheduling of the battery of an agent under certain conditions that will be defined more precisely later. We start with a constrained version of this problem in Chapter 2, and then proceed in Chapter 3 with a full information version of the problem, that we refer to as the *offline* version. The *online* one follows in Chapter 4. Then we present in Chapter 5 some other versions of our problem as well as a connection with another classic problem of Computer Science. In the second part, we move to a multi-agent setting. In Chapter 6 we briefly leave the energy trading world and we focus on an allocation division problem. Here we introduce a new framework, which we partially use in our final Chapter 7 of the thesis where we illustrate some ideas about future research directions.

1 BACKGROUND

The tremendous importance of energy and electricity is probably clear to the majority of us. It is not only the easiest way to bring comfort in our privileged life with all the most creative appliances we could own. It is essential in refrigerating food, providing us with clean and drinkable water. It incredibly facilitates sophisticated surgeries or even keeps patients alive. It is so important that electrical generators and connecting nodes of the transmission systems are frequently targeted during wars and at the same time cutting areas off the grid is used as a weapon to oppress populations, as we frequently learn from media.

In the last decades electricity has gained even more importance, as it is often proposed as the means to improve our environmentally damaged situation without having to compromise too much on the productivity of the industry. It has indeed become clear that our current lifestyle is unsustainable for the earth and we do need to modify our behaviors globally if we wish to avoid even major consequences. The damages caused by our species are various, we range from polluting several bodies of water, contaminating land and clearing huge areas of forests to the release of harmful substances into the atmosphere, as well as an uncontrolled amount of greenhouse gases (GHGs).

The severe consequences of the emission of the last mentioned, also known as GHG, have been a widely debated topic, frequently brought to global attention by initiatives such as the Paris Agreement [71] and the Net Zero Emissions by 2050 Scenario (NZE) [67]. Carbon dioxide, responsible for over 80% of total global warming, is the primary component of GHGs [39]. Among the largest contributors to GHG emissions is the electricity industry [8]. A well-established approach to mitigate the electricity industry's impact is transitioning toward renewable energy resources (RES). In this regard we are facing two trends. On one hand there is the effort of substituting traditional power plants with new power plants, such as photovoltaic panels, and wind turbine parks for instance [60]. On the other hand, politics have been subsidized the private acquisition and usage of photovoltaic panels at home, in favour of the formation of Local Energy Markets (LEM) [81].

In both cases the main advantages are the deployment of natural resources, which are considered unlimited as the sun radiations or the wind power, and the fact that the energy production process causes significantly less emission compared to the traditional power plants. This last reason is behind the appellation *clean energy* and *green energy* that we often hear and read. Again in both cases, one of the main drawbacks is well known: the natural resources which are used to produce this clean energy are way beyond our control, meaning that we need to adapt to their natural availability and elaborate systems and strategies to deal with their intrinsic unpredictability.

It is crucial to make sure that these modern power plants are reactive to the changes in the grid, and therefore we need to come up with new methods and strategies to ensure its stability. At the same time, the non continuous availability of these natural resources leads us–or maybe even forces us–to store the energy produced and use it as back up when needed. This is definitely not a trivial task to do, since it involves lots of studies and where several factors play a key role, such as the geographical composition of a territory. It is not surprising that all these challenges are also reflected in the research field [3]. Not to be forgotten is also that these technologies are rather recent. For instance, the production of photovoltaic panels (PVs) is still under an improvement phase, which results in the uncertainty for the future trends for costs and prices of PVs [23]. Effective and relatively fast solutions are indeed needed to proceed with the modernization of our electrical system, and therefore several disciplines need to put their best effort into this.

1.1 OUR CONTRIBUTIONS

In this work we focus on the private scenario above introduced: among all the possibilities which the energy related field has to offer, we decided to investigate some of the implications of PVs in private contexts. This choice was partially motivated by the rapid increase of installed photovoltaic panels, and by the conviction that such a trend will be also to be observed in the future, probably worldwide, but for sure here in Germany [80]. Household-operated photovoltaic panels provide an effective means for individuals to contribute to the energy transition, as already observed by Hockenos [50], Mengelkamp et al. [65], and Mehta and Tiefenbeck [63]. A common setup for PVs is to connect them in microgrids, which are "electricity distribution systems containing loads and distributed energy resources that can be operated in a controlled, coordinated way, either while connected to the main power network or while islanded", as stated in Shahbazitabar et al. [77]. In our work, we consider a microgrid where agents are equipped with PV panels. As a result, agents are not only passive consumers in the grid but also active participants, able to produce their own energy. Hence, we refer to them as *prosumers*. In particular we follow the definition given in Schill et al. [75], where they are denoted as grid-connected electricity consumers who own small-scale PV generators as well as batteries, and use these installations to produce their own electricity at times, draw electricity from the grid at other times, and feed electricity to the grid at yet other times.

Microgrids have been extensively investigated from various perspectives in the literature. In particular, peer-to-peer (P2P) interactions among prosumers provide a fertile ground for embedding numerous mathematical models. Moreover, the wide range of assumptions and specific contexts has resulted in a large body of related scientific work. This is evidenced by the significant number of scientific reviews in this area, where the investigated microgrids are often classified based on different design parameters. For reference we list some of the works we have encountered in these years, such as the works of Capper et al. [24], Cramton [32], Hönen et al. [51], Mengelkamp and Diesing [64] and Tushar et al. [83].

Our work mainly contributes to the body of prosumer-centric research. A considerable portion of related studies are characterized by a high dependency on the environment, which affects both the performance and analysis of these microgrids as in the works of Barbour and González [8] and Mehta and Tiefenbeck [63]. In contrast, we maintain a high level of abstraction to ensure that the model we design is as general as possible, and thus versatile. To do so, we study problems which are relatively simple, but still interesting–at least from our theoretical computer science perspective. The setting proposed in this thesis is depicted in Figure 1.1. The main idea is to observe

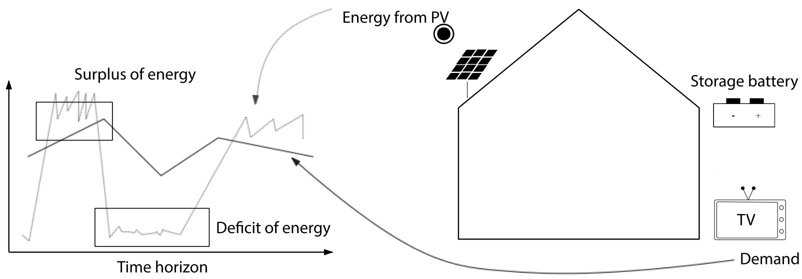


Figure 1.1: General high level setting

the energy produced by prosumers, as well as their demand during a given time period. We will refer to this as their *energy profile*. The energy profile showed in the figure is already underlining the peculiarity of the setting, namely that there are times where the prosumer is self-sufficient, or even with surplus of energy, and others where energy is instead needed to fulfill the demand. A battery here could be useful to shift the peaks of the excess, and use the energy later when needed. If on top of that we add that the prices for buying energy and those for selling are not always constant but do fluctuate over time, we get a very interesting problem with a real impact on electric bills of prosumers.

1.2 NOTATION

We now introduce the main notation used throughout the thesis. On a general note, we denote the set of *natural* numbers by $\mathbb{N} = \{1, 2, 3, \dots\}$ and the set of *integers* by $\mathbb{Z}$, and we define $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Finally, *real* and *non-negative real* numbers sets are denoted by $\mathbb{R}$ and $\mathbb{R}_+$, respectively.

In our energy frameworks, we will mostly refer to quantities which are measured over several time periods for a given finite time horizon. This time horizon has length $n \in \mathbb{N}$ and it is divided into equivalent discrete time periods, so that it can be described as $\tau = \{1, \dots, n\}$. We initially developed our model referring to days, but such partition can be more general, as for instance intervals of 15 minutes, so we refer to each interval generically as period and we denote each period as p. During each period $p \in \tau$, a prosumer has a given demand of $d_p \in \mathbb{N}_0$ units of electrical energy. Similarly, they have an amount of $e_p \in \mathbb{N}_0$ units of electrical energy available as being generated from the PV panels in period p. For the moment we remark that for sake of simplicity we consider all these quantities to be dimensionless, hence we refer to those as *quantities*, *units* or *amount*. We observe that each period p in the time horizon falls into one of two classes. On one hand, we have periods where the amount of PV-generated energy supply is at most the demand from that period ($e_p \leqslant d_p$); on the other hand, we have *excess periods* where this supply exceeds the demand ($d_p < e_p$). We define two quantities $d'_p := (d_p - e_p)^+$, and $ex_p := (e_p - d_p)^+$. Hence, for each period p, we have that $d'_p \geqslant 0$ and $ex_p \geqslant 0$ (at most one of which is positive), and $d_p - e_p = d'_p - ex_p$.

In most of our discussions, agents will be provided with a battery of capacity C, which is charged with some $soc_0 \in \mathbb{N}_0$ units of energy at the beginning of the time horizon. The amount of energy stored in the battery is the denoted with the term *state of charge*, often shorted as SOC or soc, depending on whether the quantity we are referring to is a variable or a parameter. As a clarification of this let us consider the initial state of charge of the battery, which is a known quantity, something that the prosumer can measure or just read on the battery itself, so a parameter soc_0. In this case we can initialize the variable corresponding to the state of charge at time zero with its value, so in our notation this will be denoted as $SOC_0 = soc_0$. The main goal of most of the investigated problems is to find a suitable strategy for prosumers, ensuring that their demand is satisfied, and that the energy is stored correctly in the battery without exceeding the natural constraints given by the battery characteristics. We assume the microgrid is connected to the main grid from which prosumers can buy energy, so that the demand of prosumers is for sure always satisfied. On top of that, we also consider that prosumers are indeed able to buy energy to exchange energy with each other being connected in a microgrid. This aspect allows the prices to fluctuate over time, as it is indeed reasonable that the prices for energy changes depending on the demand and on the self-produced energy available, or at least at time moment we assume so. To model this, we use a vector of prices both for buying energy and to sell energy. The parameters b_p and s_p are used to indicate respectively the price for buying a unit of energy at time p within the microgrid.

We give an overview of the parameters listed so far in Table 1.1.

$\tau = \{1, \ldots, n\}$	time horizon with n periods
b_p	price per unit for buying energy in period p
s_p	price per unit for selling energy in period p
d_p	demand in period p
e_p	units of PV-generated energy in period p
C	capacity of the battery
soc_0	initial state of charge of the battery
d'_p	net demand in period p
ex_p	excess of energy at time period p

Table 1.1: Parameters used in our work.

We conclude this chapter with two small remarks on the prices of the microgrid.

The vector prices are not to be intended general for the whole microgrid, but specific for a single prosumer: we will mostly consider a single prosumer in this work and we have just explained how in the microgrid there are different prices at which one can buy energy or sell energy. With the vector prices b_p and s_p we are already considering the best option for the prosumer, that could be the most convenient option in general or just the most convenient among the feasible exchanges.

Then, a natural assumption about electricity prices, which can be observed in real markets, is that typically buying is more expensive than the revenue obtained from selling to the grid[1]; so throughout the thesis, we assume that $b_p \geqslant s_p$ for $p = 1, \ldots, n$. This does not mean that selling is in general not convenient; instead, degenerated behavior of a prosumer buying infinite amounts of energy and selling them in the same period is avoided.

As already mentioned in the prologue, one of the main goals of this thesis was to contribute to bridging the gap between the field of Theoretical Computer Science and the more practical, application-driven research typically found in energy-related journals. Unlike many works that start from well-established problems in the literature and then improve the current results, in our case a major challenge was to identify meaningful research questions from scratch. The modelling phase played a central role in the research process underlying this thesis. Once a good model had been established, the literature was thoroughly reviewed to ensure that our newly formulated prob-

1 Germany's Federal Network Agency (Bundesnetzagentur) reports for 1 April 2023 an average household price of 45.19 ct/kWh [35] for electrical energy, whereas the feed-in tariff for PV-generated energy was at most 13.40 ct/kWh [37].

lems—or closely related ones—had not already been solved. The results of this research are discussed in each chapter, where we also point out the similarities with existing work while highlighting the differences. In the following chapters, we present the questions we developed and the methods we used to investigate them.

ENERGY SCHEDULING

2 A CONSTRAINED SETTING

We start now the first chapter of the thesis describing results and we also present what–from our point of view–is the most emblematic overlapping between the applied energy trading world and the more theoretical background we stamm from. Namely, we are going to introduce a new KNAPSACK problem; the original Knapsack problem is one of the the Karp's 21 NP-complete problems [56].

Surprisingly, the knapsack interpretation of the energy related problem we were tackling appread almost spontaneously during our research discussions. Then we discovered that the fuzzy problem we were looking at was a rather interesting one on its own, and we present it here, starting from the background in which it originated.

This chapter is based on joint work with Gergely Csáji and Matthias Mnich [28].

2.1 DECRIPTION OF THE PROBLEM

We focus on a single prosumer whose household is equipped with PVs and consider their *energy profile* over a given time horizon of length n. This profile is defined by their demand d_p, and the amount e_p of produced energy, over all time periods $p = 1, \dots, n$. Typically, such households are equipped with a Home Energy Management System (HEMS), which coordinates energy demand (e.g., domestic appliances, EVs) with PV generation and battery storage, as we as discussed in Beaudin and Zareipour [11] and Angenendt et al. [4]. The presence of a HEMS, or more generally of advanced metering infrastructure (e.g., smart meters) [83], is commonly assumed in the literature on local energy markets (LEMs) among prosumers, where it enables monitoring, control, and participation in decentralized energy exchanges.

As already stated, a key property of our model is that the prosumer has a private battery with capacity $C \in \mathbb{N}$, which can be charged and discharged according to the model by Hosseini et al. [52]. Under this setting, a prosumer within a fixed microgrid can satisfy their daily energy demands through a combination of self-generated electricity and energy exchanges with other prosumers within the microgrid. We introduce market dynamics within the microgrid by assuming that the prices for trading energy fluctuate daily. To simplify the analysis, we assume that each prosumer's full energy demand is always met; this assumption is justified by allowing one prosumer to be a large company operating the main grid, with zero energy demand.

The battery allows prosumers to act strategically, offsetting their energy costs by selling excess energy on days when prices are favorable. The goal of any prosumer is to minimize their overall energy acquisition costs over the fixed time horizon of length n. We refer to this as the Prosumer Energy Acquisition Cost Minimization (PEAC-min) problem, which forms the central focus of this chapter.

PEAC-min is, fundamentally, a scheduling problem: for each period, the prosumer decides (i) how much energy to consume from self-generated PV power, (ii) how much to store in the battery for future use or sale, (iii) how much to discharge from the battery, and (iv) how much to purchase from other prosumers.

A critical aspect of the problem is that decisions made at one point in time affect future outcomes by altering the state of the system, particularly the amount of energy stored. Although we analyze the actions of a single prosumer, their optimal strategies are shaped by the actions of other prosumers in the microgrid. These strategies influence not only the availability of energy for trading but also the buying and selling prices, which depend on demand and energy storage decisions.

Studying and optimizing local energy markets is important because individual prosumers may not wish or be able to coordinate their actions with others. Thus, understanding the optimal selfish strategy for each prosumer is crucial. Important related examples of this use are provided by Norbu et al. [69] and Zhang et al. [87], where a similar model is used to optimize the single strategy of prosumers before considering them collectively. In particular they use a heuristic approach to find a feasible scheduling which could instead be optimal with the approach presented in this work. More in general, further assumptions about energy prices can lead to efficient local energy markets, as seen in the overview of Capper et al. [24].

In this chapter we focus on a variant of the problem where prosumers either use their battery or they buy energy. This assumption is fundamental for the discussion of this chapter, as it makes the studied problems quite unique. In the next chapter we will see instead what happens when we remove such assumption. The key idea in solving this problem is to strategically use the battery to maximize the savings for buying energy, given we have a battery of finite capacity (which corresponds to the knapsack). We must choose to satisfy demands from the battery (so save the associated buying costs) optimally, and each period p with demand d_p corresponds to an item with weight $(d_p - e_p)^+$. We are not the first using a Knapsack approach for an energy related problem. Kumaragurparan et al. [59] and Sianaki et al. [79] use the Multiple Knapsack problem to schedule the usage of home appliances to minimize the energy costs while ensuring some costraints are met. Arikiez et al. [5] use instead a chance constrained knapsack to solve a similar problem. In our approach however, we have a different variant of the Knapsack problem, as we have a *sequence* of knapsacks (instead of a single one), one for each period in which we have some excess of energy. The subset of days selected for packing a knapsack influences the capacity of all the following knapsacks. This can be properly formalized as a unique knapsack whose capacity is non decreasing overtime and, peculiarly, has a minimum capacity requirement as well. This peculiarity is due to the capacity C of the battery and the free energy that prosumers obtain on excess periods. Consequently, we have a time-dependent upper bound U_p

on how much total weight we can have chosen so far (depending on the previous excess periods). Then, the excess periods also force us to have a time-dependent lower bound L_p on the minimum item weight we must have packed at any period p. We also have the option to sell energy to other prosumers in the microgrid, which corresponds to having items that we can pack multiple times (and this quantity is not necessarily polynomial in the input size). In that sense, our problem variant more resembles a *temporal* KNAPSACK problem, for which Clautiaux et al. [27] provide a successive sublimation dynamic programming algorithm. Other versions of KNAPSACK, where the capacity of the knapsack increases over time as in Bartlett et al. [10], are also similar to our setting. However, none of these already studied problems are capturing the innovative feature of our setting. Indeed, while the capacity of the battery is giving an upper bound on the capacity of each of the knapsacks, by storing and then freeing space with usage, we may end up consuming more energy than the capacity itself. We call this new problem the TEMPORAL KNAPSACK WITH ITEM COPIES AND LOWER AND UPPER BOUNDS (TKICLUB). Before formally defining TKICLUB, I would like to offer a very informal interpretation of it, much like how I would explain it in a casual conversation. One interpretation in my opinion, is that our knapsack is in a kind of "reverse Matryoshka doll", those wooden nesting dolls where each one contains a smaller replica of itself till we reach an unsplittable tiny doll. Similarly, we pack one knapsack into another and so on. We have a sequence of knapsacks, each fitting inside the other. At first, we are given a knapsack whose capacity is known and a set of items for which we have to decide whether to pack them or not. Then, we are given a second knapsack, into which we must pack the whole first knapsack and where we may also choose to add additional items, which might differ from the previous ones. This process continues iteratively. Effectively, we deal with a sequence of knapsacks, where the capacity of each depends on the items that have already been packed. The analogy with energy is as follows: The items represent time periods and packing an item p means that on period p the prosumer uses the energy stored in the battery to fulfill the demand. The capacity of the knapsack is clearly related to that of the battery. When there are periods with an excess of energy, we get a new knapsack with increased capacity. As will become clearer later, due to the finite capacity of the battery and the fact that we can have multiple days with a positive energy surplus, we might sometimes be forced to select a subset of days, or equivalently to pack some items. Now that we have given an intuitive idea of a part of our problem, we can move on to the formal formulation, which we will give first in terms of energy trading and then in terms of the knapsack analogy.

2.2 THE PROBLEM FORMULATION

As already stated, the peculiarity of the problems analyzed in this chapter is that if the prosumer buys energy from the grid, they are then forced to purchase the entire amount needed to satisfy their demand. The other alternative would be to use the energy stored in the battery to fulfill the demand. A combination of the two actions is not allowed and this implies that at every period p, the prosumer must either buy or use the energy stored in the battery. This is just one specific variant or scenario of the PEAC problem introduced in the previous section. In this thesis we will be introducing several variants of the PEAC problem. Here, to emphasize that the prosumer can both buy and sell, and that the choices of using the battery or buying energy are mutually *exclusive*, we call this variant PEAC-BS-Ex.

We give a model of our problem as an integer linear program (ILP), hence we introduce here some sets of variables, summarized in Table 2.1.

x_p	units of energy bought from other prosumers in period p
y_p	units of energy sold to other prosumers in period p
z_p	units of energy exchanged with battery in period p
SOC_p	state of charge of battery in period p
q_p	binary variable: $q_p = 1$ if we buy in period p

Table 2.1: Variables in our models.

For each period $p \in \tau$, let $x_p \in \mathbb{N}_0$ be the number of units of energy bought, $y_p \in \mathbb{N}_0$ be the number of units of energy sold, and z_p be the number of units stored in or discharged from the battery (so z_p can be either positive, negative, or zero). We recall that the prices of energy associated to the variables are expressed with the parameters b_p and s_p. We keep track of the current state of charge of the battery with the variables $SOC_p \in \{0, \dots, C\}$. Finally, the binary variable q_p takes value 1 in case the prosumer decides to buy at time p.

With these sets of variables, the problem can be modelled as

$$\min \sum_p (b_p x_p - s_p y_p) \tag{2.1}$$

$$\text{s.t.} \quad x_p - y_p - z_p = d_p - e_p, \quad p \in \tau, \tag{2.2}$$

$$x_p = (d_p - e_p)^+ q_p, \quad p \in \tau, \tag{2.3}$$

$$SOC_p = SOC_{p-1} + z_p, \quad p \in \tau, \tag{2.4}$$

$$SOC_0 = soc_0, \tag{2.5}$$

$$SOC_p \leqslant C, p \in \tau, \tag{2.6}$$

$$x_p, y_p, SOC_p \in \mathbb{N}_0, \quad p \in \tau, \tag{2.7}$$

$$q_p \in \{0, 1\}, \quad p \in \tau, \tag{2.8}$$

$$z_p \in \mathbb{Z}, \quad p \in \tau\ . \tag{2.9}$$

In this formulation we aim at minimizing the total costs, which are computed as the expenses incurred for buying energy discounted by the amount of energy sold. Constraint (2.2) ensures energy balance. On the left hand side we have the energy flows: the amount bought from the grid, the amonunt sold and the one exchanged with the battery, which can be either due to the charging or discharging process of the battery, depending on the sign of z_p. On the right hand side we have instead the difference of the demand d_p of the prosumer and the self-produced energy. Then in (2.3) we have the formulation of the main assumption of this chapter: if the prosumer decides to buy the whole amount, then this amount has to fulfill completely the actual demand needed, which is indeed the demand of the prosumer discounted by the energy from the PVs. Constraint (2.4) ensures that the state of charge of the battery is updated accordingly to the amount of energy stored in the battery, if z_p is positive, or taken from the battery, in case the variable is negative.Constraint (2.6) ensures that the state of charge never exceeds its capacity. If selling it is not allowed we can just add the constraints $y_p = 0, p \in \tau$, and we call the problem PEAC-B-Ex.

PEAC-BS-Ex

In: An instance $\mathcal{I} = (b_p, s_p, d_p, e_p, C, soc_0 \mid p \in \tau)$.

Out: An optimal solution to the ILP defined by objective (2.1) and constraints (2.2)–(2.9).

While discussing these variants, we considered several possible modifications. We found out that they can all be wrapped together by setting some bounds to the amount of energy that the prosumer can buy and sell at every time period. To bound them we introduced two sets of parameters α_p and β_p, which bound the number of units bought and sold at every time period, respectively. We can generalize these two problem variants by adding the following inequalities to the models:

$$x_p = \alpha_p q_p, \quad p \in \tau, \tag{2.10}$$

$$y_p \leqslant \beta_p, \quad p \in \tau, \tag{2.11}$$

$$q_p \in \{0, 1\}, \quad p \in \tau \ . \tag{2.12}$$

In Chapter 3 we will be discussing more about the parameters α and β and the relations among the variants when setting particular values. For the moment, we add these further constraints to obtain a more general version of the problem, which we call indeed PEAC-BS-Ex-General.

PEAC-BS-Ex-General

In: An instance $\mathcal{I} = (b_p, s_p, d_p, e_p, \alpha_p, \beta_p, C, soc_0 \mid p \in \tau)$.

Out: An optimal solution to the ILP defined objective (2.1) and constraints (2.2)–(2.9) and (2.10)–(2.12).

It is easy to see that the values to variables z_p and SOC_p for $p \in \tau$ are uniquely determined by the values of variables x_p, y_p for $p \in \tau$ in all the presented problem variants. Hence, we usually refer only to $(x_1, y_1, \ldots, x_n, y_n)$ as the solution.

Before proceeding with the results of the chapter, we want to make sure that the problem is clear, hence we provide a simple example.

Example 2.1. *Consider an instance $\mathcal{I}$ for* PEAC-B-Ex*, where the time horizon has length 3. Let the demand be $d_p = (4, 9, 6)$ and the energy produced with the photovoltaic panels be $e_p = (3, 7, 3)$. The battery has a capacity of $C = 10$ and an initial state of charge of $soc_0 = 5$. Assume now that the prices for buying energy are $b_p = (4, 3, 6)$. We compute the net demand over the time horizon, which is $d'_p = (1, 2, 3)$ for a total of 6 units, which is more than what we have in the battery. Then we need to buy at least on one on the periods. To decide when to buy, we can apply the following reasoning. If we had to buy energy at the first time period we would have a cost of 4, at the second a cost of 6 and 18 at the third. So, if we did not have the battery at all, we would have to buy all the units incurring into a cost of 28. Let us see now how much we can save each period. Given that we have 5 units of energy in the battery, we can use this energy to buy less energy. The cheapest solution here is to buy energy on the first time period and use the battery later. This is equivalent to saying that we are packing in our knapsack of capacity 5 the items corresponding to the second and third time peridos, as their value is of 24 which is the maximum we can get.*

So far there was only one knapsack. Let us expand the example to clarify what we meant earlier with a "matryoska" of knapsacks.

Example 2.2. *Consider now an instance $\mathcal{I}$ for* PEAC-B-Ex *with a time horizon of 5, where $d_p = (4, 9, 6, 1, 6)$ and $e_p = (3, 7, 3, 3, 3)$. Furthermore let $soc_0 = 5$, $C = 10$ and the prices $b_p = (4, 3, 6, 5, 8)$. We get a net demand of $d'_p = (1, 2, 3, 0, 3)$. We observe that at time period 4 not only we do not need to buy energy, since $d'_4 = 0$, but also we are producing 2 extra units of energy, which we can store in the battery. We can use the fourth time period to split the problem into two instances, each one corresponding to a knapsack problem as in Example 2.1. By doing so, the first knapsack is exactly in Example 2.1, for which we have already computed that it is optimal to pack the items of the second and third time periods. For the second knapsack instead, we can only decide whether to select the item at the 5-th time period or not. If these two knapsacks were completely independent, we would perfectly know how pack them in order to maximize their profit. However this is not the case, since the second knapsack has a capacity which is given by the remaining capacity of the first knapsack, if any, plus an extra capacity of 2, given by the excess period at time 4. If we pack the first knapsack as we did in Example 2.1, then the second knapsack would have a capacity of 2 and we would not be able to fit any new element and we would incur in a cost of 28. Then, it is convenient to only partially fill up the first knapsack by selecting the time periods 1 and 3. We are left now with a second knapsack of capacity 3 which is enough to pack the last element leading to a total cost of 6.*

Lastly, it is important to notice that sometimes we might be forced to select items in our knapsack, either by selling energy or discharging the battery to fulfill the demand.

Example 2.3. *Consider an instance $\mathcal{I}$ for* PEAC-B-Ex *with a time horizon of 2, where $d_p = (4, 7)$ and $e_p = (3, 9)$, battery with characteristics $soc_0 = 9$ and $C = 10$. Here we have $d'_p = (1, 0)$ and we have two units of extra energy at the second time period. Even if it is a bit of a degenerate* *example, we can notice that the prosumer is forced to use the battery at time* 1*, because otherwise it would not be possible to store the* 2 *units that will be received at time* 2 *leading to an infeasible instance (we remind that in* PEAC-B-Ex *we are not allowed to sell, but if that would have been the case, then we would have had to sell one unit either on period* 1 *or* 2*).*

2.3 RESULTS FOR PEAC-BS-EX

Our first result is an NP-hardness reduction.

Theorem 2.1. PEAC-B-Ex *and* PEAC-BS-Ex *are* NP*-hard, even if* $d_p \geqslant e_p$ *for all periods* $p = 1, \ldots, n$.

Proof. We present a reduction from the NP-hard KNAPSACK problem.

Consider an instance of KNAPSACK problem, consisting of a set I of n items, each with an associated value $v_i \in \mathbb{N}$ and weight $w_i \in \mathbb{N}$, a weight bound $W^\star \in \mathbb{N}$ and a target value $V^\star \in \mathbb{N}$. The task is to decide whether there is a subset $I^\star \subseteq I$ of items whose weight $\sum_{i \in I^\star} w_i$ is at most the weight bound $W^\star$ while its value $\sum_{i \in I^\star} v_i$ is at least $V^\star$. Let $W_\prod \in \mathbb{N}$ be the *product* of all item weights, and let $V_\sum$ be the *sum* of all item values.

From this instance, we create an instance $\mathcal{I}'$ of PEAC-B-Ex . Instance $\mathcal{I}'$ will have one period for each item of I. For each $i \in I$, let $v_i' = v_i \prod_{j \in I} w_j = v_i W_\prod$, and let

$$d_i = w_i, e_i = 0 \text{ and } b_i = \frac{v_i'}{w_i} \ .$$

The construction of the instance $\mathcal{I}'$ is completed by setting $soc_0 = W^\star$ and the battery capacity as $C = W^\star$.

We claim that the KNAPSACK instance admits a solution with weight at most $W^\star$ and value at least $V^\star$ if and only if there is a feasible solution for $\mathcal{I}'$ with cost at most

$$(V_\sum - V^\star) W_\prod \ .$$

In the forward direction, consider a solution $I^\star \subseteq I$ to the KNAPSACK instance with weight $\sum_{i \in I^\star} w_i \leqslant W^\star$ and value $\sum_{i \in I^\star} v_i \geqslant V^\star$.

Then we can create a feasible solution for $\mathcal{I}'$, where we set $x_i = 0$ for $i \in I^\star$ and $x_i = d_i^\star$ otherwise. Then, the cost of this solution is

$$\sum_{i \in I} b_i d_i' - \sum_{i \in I^\star} b_i d_i' = V_\sum W_\prod - W_\prod \sum_{i \in I^\star} v_i \leqslant (V_\sum - V^\star) W_\prod \ .$$

Also, it is a feasible solution, because $\sum_{i \in I^\star} d_i^\star = \sum_{i \in I^\star} w_i \leqslant W^\star = soc_0$.

Suppose now that PEAC-B-Ex has a feasible solution with cost at most $(V_\sum - V^\star) W_\prod$ and let I' be the set of time periods where the battery is unused, namely $x_i = 0$ if $i \in I^\star$ and $x_i = d_i'$ otherwise. This set $I^\star$ gives a solution to $\mathcal{I}$ with value

$$\sum_{i \in I^\star} v_i = \frac{1}{W_\prod} \sum_{i \in I^\star} b_i d'_i \geqslant \frac{1}{W_\prod}(V_\sum - (V_\sum - V^\star))W_\prod = V^\star \quad . \tag{2.13}$$

that $\sum_{i \in I^\star} d'_i = \sum_{i \in I^\star} w_i \leqslant soc_0 = W^\star$, the set $I^\star$ corresponds then to a solution to the knapsack problem with value at least $V^\star$. The hardness of PEAC-BS-Ex is analogous, with the addition that we enable selling, but set all selling prices to 0. Then, selling is allowed, but it holds that there exists a feasible solution with cost at most $(V_\sum - V^\star)W_\prod$ with selling allowed if and only if there exists one with only buying allowed, because from a solution which involves selling, we can obtain a solution of the same cost by setting all selling values to 0—this cannot lead to the capacity being overcharged, as there is no excess energy day in the construction. □

By Theorem 2.1, PEAC-B-Ex and PEAC-BS-Ex are NP-hard even when there are no excess periods. We complement this result by showing that if there can be periods with excess energy, then even deciding if there exists a feasible solution becomes NP-hard.

Theorem 2.2. *Checking if an istance of* PEAC-B-Ex *has a feasible solution is* NP*-hard, even if there is only a single period* p *with* $ex_p > 0$.

Proof. We reduce from the NP-hard SUBSET SUM problem. There, for a given set $S = \{s_1, \dots, s_n\}$ of n non-negative integers and a number $b \in \mathbb{N}$ one has to decide whether there exists some subset $S' \subseteq S$, such that $\sum_{i \in S'} s_i = b$.

Let (S, b) be an instance of SUBSET SUM; we create an instance $\mathcal{I}'$ of PEAC-B-Ex as follows.

Let $M = \sum_{i=1}^{n} s_i$. Let there be $n+2$ periods. We set $soc_0 = C = M+1$, $e_p = 0$, $d_p = s_p$ for $p \leqslant n$, and $e_{n+1} = 0$, $d_{n+1} = M + 1 - b$, $e_{n+2} = C$, $d_{n+2} = 0$. The prices are not relevant, as we only show hardness of feasibility in this proof.

We show that there exists a solution to (S, b) if and only if there exists a feasible solution to $\mathcal{I}'$.

In the forward direction, suppose there is a set $S' \subseteq S$ with $\sum_{s_i \in S'} s_i = b$. Then, we set $x_p = 0$ for $s_i \in S'$ and $x_p = d_p$ otherwise. Then, by period n the battery has exactly $M + 1 - b$ units of charge left. Hence, at period $n+1$, we can use the battery to satisfy the demand exactly, so we arrive at period $n+2$ with 0 charge. Hence, the excess of energy on the last period does not overcharge the battery, so the solution is feasible.

In the backward direction, suppose that there exists a feasible solution for $\mathcal{I}'$. Then, we must reach period $n+2$ with 0 state of charge. Hence, we either reach period $n+1$ with 0 or with $d_{n+1} = M + 1 - b$ charge. However, since $\sum_{p=1}^{n} d_p = M$, but $soc_0 = M+1$, only $M + 1 - b$ can be possible. Furthermore, this is only possible if and only if there is a set of periods $S' \subseteq S$ such that $\sum_{p \in S'} d_p = b$, where the battery is used instead of buying, which concludes the proof. □

As one can see from Theorem 2.2, it is indeed these excess periods which differentiate our problem from earlier studied temporal versions of Knapsack, where the knapsack capacity changes over time. Later, we introduce a new variant of Knapsack that captures the features of our problem, and we explain the connections between them.

First, we provide a dynamic programming algorithm for PEAC-BS-Ex-General, which also solves problems PEAC-B-Ex and PEAC-BS-Ex as a corollary. Here, instead of minimizing our cost, we change the framework and try to maximize our savings compared to a solution where each demand is satisfied by buying the required amount and no sells are made. This also helps to emphasize the similarities between PEAC-BS-Ex-General and temporal Knapsack problems.

The intuition of the algorithm is that we iteratively solve the subproblems $SP(p, C-k)$ for $k \in [0, C]$ and $p \in [0, n]$. Subproblem $SP(p, C-k)$ consists of finding the maximum of $\sum_{i=1}^{p}(y_i s_i - x_i b_i)$ (i.e., our savings compared to a solution which does no selling and buys for every demand) restricted to the first p periods for a solution $(x_1, y_1, \ldots, x_p, y_p)$ such that $SOC_p = C-k$ holds exactly. At the end of each iteration pt, we store the optimum values of subproblems $SP(p, C-k)$ for $k \in [0, C]$ in a vector R.

The algorithm's pseudocode is given in Algorithm 2.1. In the algorithm, the maximum over the empty set is set to be $-\infty$.

We suppose that $ex_p \leqslant C$ for any $p \in \tau$. Otherwise, we must sell at least $ex_p - C$ units in period p, so then we can reduce our instance by setting $\beta_p := \beta_p - (ex_p - C)$ (if this is negative, then there can be no feasible solution and we can stop) and $ex_p := C$ and solve the new instance. As mentioned, to align with the maximization goal of the corresponding Knapsack instance defined later, we equivalently change our objective function to $\max \sum_{p=1}^{n}(s_p y_p - b_p d_p)$.

We remark that we can compute the maxima in Algorithm 2.1 in a clever iterative way (as shown in the next proof) that leads to a significant speedup. Thereby, we obtain that:

Theorem 2.3. *Algorithm 2.1 outputs an optimal solution for* PEAC-B-General *in* $\mathcal{O}(nC \log C)$ *time. For* PEAC-B-Ex *and* PEAC-BS-Ex*, the run time is* $\mathcal{O}(nC)$.

Proof. It is easy to see that the algorithm correctly computes the optimum for the subproblems $SP(0, C-k)$ for $k = 0, \ldots, C$ at the initialization step, as we have no choice but to leave exactly soc_0 for the first period and our profits/savings so far are 0. Suppose that the algorithm correctly computed the optima for the subproblems $SP(p-1, C-k)$ for some p in R. If we fix x_p and y_p for $SP(p, C-k)$, then the problem is to find an optimal solution for the first $p-1$ periods that leaves exactly $C-k-x_p+y_p-ex_p+d'_p$ units of energy in the battery, which we have computed in $R[C-k-x_p+y_p-ex_p+d'_p]$ by induction. Furthermore, then we receive $s_p y_p - b_p x_p$ profit in period p. Also, then the bounds $a_0^k \leqslant y_p \leqslant a_1^k$ and $b_0^k \leqslant y_p \leqslant b_1^k$ (depending on if $x_p = 0$ or $x_p = \alpha_p$) must hold by $0 \leqslant C-k-x_p+y_p-ex_p+d_p \leqslant C$ and $0 \leqslant y_p \leqslant \beta_p$, so the algorithm checks each possibility for period p, hence finds the optimal solution for $SP(p, C-k)$. Feasibility is guar-

Algorithm 2.1 PEAC-BS-Ex-General

Input : An instance $\mathcal{I} = (b_p, s_p, d_p, e_p, \alpha_p, \beta_p, C, soc_0 \mid p \in \tau)$ of PEAC-BS-Ex-General.

Output : The optimum value for $\mathcal{I}$.

Initialize $ex_p := (e_p - d_p)^+$ for $p = 1, \dots, n$ and

$$R[C-k] := \begin{cases} 0 & \text{if} \quad C-k = soc_0 \\ -\infty & \text{otherwise} \end{cases}$$; `// We have no choice but // to leave` soc_0 `for the // S first period`

Let A and B be two empty arrays of C dimension ; `// we store the values of the subproblems we have when we respectively use and do not use the energy stored in the battery`

Set $Q := [0, \dots, 0]$; `// we store the optima for the new subproblems in Q before updating R`

for $p = 1, \dots, n$ **do**

  for $k = 0, \dots, C$ **do**

    $a_1^k = \min\{\beta_p, k + ex_p - d'_p\}$; `// Max amount possible to sell in case we choose` $x_p = 0$

    $a_0^k = \max\{0, k + ex_p - d'_p - C\}$; `// Min possible`

    $A[C-k] := \max_{a_0^k \leqslant y \leqslant a_1^k}\{R[C-k+y-ex_p+d'_p] + y \cdot s_p\}$ $b_1^k = \min\{\beta_p, k + \alpha_p + ex_p - d'_p\}$; `// Max amount possible to sell in case we choose` $x_p = \alpha_p$

    $b_0^k = \max\{0, k + \alpha_p + ex_p - d'_p - C\}$; `// Min possible`

    $B[C-k] := \max_{b_0^k \leqslant y \leqslant b_1^k}\{R[C-k+y-ex_p-\alpha_p+d'_p] + y \cdot s_p - \alpha_p \cdot b_p\}$

    $Q[C-k] = \max\{A[C-k], B[C-k]\}$

  Set $R := Q$ and $Q := [0, \dots, 0]$.

return $\max_{k=0,\dots,C}\{R[k]\}$

anteed by induction and the fact that we leave $0 \leqslant C - k \leqslant C$ battery at the end of period p.

As for the run time, we have $n(C+1)$ iterations in the two **for**-loops combined. In the case of PEAC-B-Ex, $\beta_p = 0$ for all p, so each maximum consists of at most 2 elements, hence each iteration requires only a constant number of steps. Hence, the run time in this case is $\mathcal{O}(nC)$.

Consider the case of PEAC-BS-Ex, where $\beta_p = C + ex_p$ and $\alpha_p = d'_p$. Suppose we are in the iteration for period p in the outer **for**-loop. Then, a_1^k is always $k + ex_p - d'_p = k + f_p$, for any k, where $f_p := ex_p - d'_p = e_p - d_p$. Assume that $a_0^k \leqslant a_1^k$ and $b_0^k \leqslant b_1^k$, otherwise the corresponding max is empty. Observe that, if $a_0^k = 0$, then $a_0^{k-1} = 0$ and

$$\begin{aligned}
A[C-k] &= \max_{0 \leqslant y \leqslant k+f_p} \{R[C-k-f_p+y] + ys_p\} \\
&= \max\left\{ \max_{1 \leqslant y \leqslant k+f_p} \{R[C-k-f_p+y] + ys_p\}, R[C-k-f_p] \right\} \\
&= \max\Bigg\{ s_p + \max_{0 \leqslant y' \leqslant k-1+f_p} \{R[C-(k-1)-f_p+y'] + y's_p\}, \\
&\qquad R[C-k-f_p] \Bigg\} \\
&= \max\left\{ s_p + A[C-(k-1)], R[C-k-f_p] \right\} \quad .
\end{aligned}$$

Otherwise, $0 \leqslant a_0^{k-1} = a_0^k - 1$ and

$$\begin{aligned}
A[C-k] &= \max_{a_0^k \leqslant y \leqslant k+f_p} \left\{ R[C-k-f_p+y] + ys_p \right\} \\
&= \max_{a_0^{k-1} \leqslant y \leqslant k-1+f_p} \left\{ R[C-k-f_p+(y+1)] + (y+1)s_p \right\} \\
&= s_p + \max_{a_0^{k-1} \leqslant y' \leqslant k-1+f_p} \left\{ R[C-(k-1)-f_p+y')] + y's_p \right\} \\
&= s_p + A[C-(k-1)] \quad .
\end{aligned}$$

Hence, in the **for**-loop $k = 0, \ldots, C$, the value $A[C-0]$ can be computed in $C + ex_p \leqslant 2C$ steps, and then each $A[C-k]$ can be computed by taking the maximum of at most two numbers.

Similarly, we can show that either

$$B[C-k] = \max\{s_p + B[C-(k-1)], R[C-k-ex_p] - b_p d'_p\},$$

or

$$B[C-k] = s_p + B[C-(k-1)],$$

depending on whether $b_0^k = 0$ or > 0), since b_1^k is always $k + ex_p$ for any k. Hence, after the iteration for $k = 0$, each maximum can be computed by taking the maximum of at most two numbers. Hence, the run time is $\mathcal{O}(nC)$ for PEAC-BS-Ex.

Finally, consider PEAC-BS-Ex-General. Here, each $A[C-k]$ and $B[C-k]$ is the maximum of

$$\begin{aligned}
&\{R[\eta_1] + (\eta_1 - \gamma)s_p - q_p b_p, \\
&R[\eta_1 + 1] + (\eta_1 + 1 - \gamma)s_p - q_p b_p, \ldots, R[\eta_2] + (\eta_2 - \gamma)s_p - q_p b_p\},
\end{aligned}$$

for some η_1, η_2, γ and $q_p \in \{0, \alpha_p\}$, which is just

$$-\gamma s_p - q_p b_p + \max\{R[\eta_1] + \eta_1 s_p, R[\eta_1 + 1] + (\eta_1 + 1)s_p, \ldots, R[\eta_2] + \eta_2 s_p\}.$$

Hence, we can compute the maxima for $k = 0, 1, \ldots, C$ by always maintaining an ordered list of

$$\{R[\eta_1] + \eta_1 s_p, R[\eta_1 + 1] + (\eta_1 + 1)s_p, \ldots, R[\eta_2] + \eta_2 s_p\},$$

where η_1 and η_2 are the current boundaries in the maximum. For $k = 0$, we can do this in $\mathcal{O}(C \log C)$ time. Note that η_1 and η_2 are monotone increasing during the inner **for**-loop. Then, inserting a new element, when η_2 increases and deleting an element, when η_1 increases can be done in $\mathcal{O}(\log C)$ time. As there are at most $2C$ insertions and deletions, all of them combined take at most $\mathcal{O}(C \log C)$ time. Furthermore, since we maintain this ordered list, each maximum in the iterations $k = 1, 2, \ldots, C$ can now be computed in $\mathcal{O}(1)$ time. Hence, the run time of the algorithm is bounded by $\mathcal{O}(nC \log C)$. □

2.4 INTRODUCING A NEW VARIANT OF THE TEMPORAL KNAPSACK PROBLEM

We now introduce a new variant of the KNAPSACK problem that closely relates to PEAC-BS-EX-GENERAL. It is formally defined as follows:

TEMPORAL KNAPSACK WITH ITEM COPIES AND LOWER AND UPPER BOUNDS (TKICLUB)

In: A time horizon $\tau = \{1, \ldots, n\}$, a set I of m items with weights $w_i \in \mathbb{N}$, values $v_i \in \mathbb{N}$, a number of copies N_i and an arrival time $p_i \in \tau$ for $i \in I$, and time-dependent lower and upper bounds $L_p \leqslant U_p$ for $p \in \tau$.

Out: An integral vector $x = (x_1, \ldots, x_m)$ with $x_i \in [0, N_i]$ for all $i \in I$ such that $L_p \leqslant \sum_{i \in I | p_i \leqslant p} w_i x_i \leqslant U_p$ for all $p \in \tau$, and $\sum_{i=1}^{n} v_i x_i$ is maximum.

Intuitively, the connection between PEAC-BS-EX-GENERAL and TKICLUB is that we can view PEAC-BS-EX-GENERAL as a problem, where we start from a configuration, where we buy the demand each period, and then we want to maximize our savings from there. Notice that such a configuration might not be feasible. That is, we choose the periods where we use the battery instead of buying d'_p and the periods and amounts that we sell. As our battery usage is limited by the excess energy up to that point on any given period, we have a time dependent upper bound on the corresponding KNAPSACK instance, but also because of the battery constraint, we have time dependent lower bounds on at least how much item weight must be in the knapsack. Also, as the amount we sell is not a binary choice, the KNAPSACK instance has items that we can put inside the knapsack multiple times (given by $N_i = \beta_p$), which we emphasize is fundamentally different from having β_p copies of the item, since in the former case, β_p may not be polynomial in the input size. In particular, we show:

Lemma 2.4. *Any instance $\mathcal{I}$ of* PEAC-BS-Ex-General *can be reduced to an instance $\mathcal{I}'$ of* TKICLUB *in $\mathcal{O}(|\mathcal{I}|)$ time such that an optimal solution to $\mathcal{I}$ can be constructed from an optimal solution to $\mathcal{I}'$ in polynomial time.*

Proof. From an instance $\mathcal{I} = (b_p, s_p, d_p, e_p, \alpha_p, \beta_p, C, soc_0 \mid p \in \tau)$ of PEAC-BS-Ex-General we create an instance $I' = (n', m, w_i, v_i, N_i, t_i, L_i, U_i \mid i \in I)$ of TKICLUB as follows.

Let $n' = n$. We define $U_i = soc_0 + \sum_{p=1}^{i}(\alpha_p + e_p - d_p)$ for $i = 1, \dots, n$, the amount of energy we would have (disregarding the battery capacity), if we bought on each period, and never sold on any period. Then, we define $L_i = \max\{0, U_{i+1} - C\}$ for $i = 1, \dots, n-1$ and $L_n = L_{n-1}$. For each period $p = 1, \dots, n$, we have two items $2p-1, 2p$ with $N_{2p-1} = \beta_p$,$v_{2p-1} = s_p$, $w_{2p-1} = 1$ and $N_{2p} = 1$, $v_{2p} = \alpha_p b_p$, $w_{2p} = \alpha_p$. Furthermore, we let the arrival times be defined by $p_{2p-1} = p_{2p} = p$ for $p \in \tau$.

Now, on the one hand, from a solution $(x_1, y_1, \dots, x_n, y_n)$ of I we create a solution $(z_1, z_2, \dots, z_{2n})$ to I' by setting $z_{2p-1} = y_p$ and $z_{2p} = \frac{\alpha_p - x_p}{\alpha_p} \in \{0, 1\}$ for $p = 1, \dots, n$.

Observe that

$$
\begin{aligned}
0 \leqslant soc_i &= soc_0 + \sum_{p=1}^{i}(e_p - d_p + x_p - y_p) \\
&= soc_0 + \sum_{p=1}^{i}(e_p - d_p + (\alpha_p - \alpha_p z_{2p}) - z_{2p-1}) \\
&= U_i - \sum_{j:p_j \leqslant i} w_j z_j,
\end{aligned}
$$

so the upper bounds are satisfied. Similarly,

$$
0 \leqslant C - soc_i = C - soc_0 - \sum_{p=1}^{i}(e_p - d_p + x_p - y_p) \leqslant -L_i + \sum_{j:p_j \leqslant i} w_j z_j,
$$

so lower bounds are satisfied too.

Using that $x_p = \alpha_p q_p$, $0 \leqslant y_p \leqslant \beta_p$, we have that these inequalities imply that $L_i \leqslant \sum_{j|t_j \leqslant i} w_j z_j \leqslant U_i$ and $0 \leqslant z_i \leqslant N_i$ for $i = 1, \dots, 2n$.

On the other hand, from a solution $(z_1, \dots, z_{2n})$ to I' we can create a solution $(x_1, y_1, \dots, x_n, y_n)$ by setting $y_p = z_{2t-1}$ and $x_p = \alpha_p(1 - z_{2p})$ (i.e., $q_p := (1 - z_{2p})$). Furthermore, the inequalities

$$
\begin{aligned}
soc_0 + \sum_{p=1}^{i}(\alpha_p + e_p - d_p) - C \leqslant L_i \leqslant & \sum_{j|p_j \leqslant i} w_j z_j \\
&= \sum_{p=1}^{i}(\alpha_p - x_p) + y_p \\
&\leqslant U_i = soc_0 + \sum_{p=1}^{i}(\alpha_p + e_p - d_p),
\end{aligned}
$$

for $i = 1, \dots, 2n$ together with $0 \leqslant L_i, 0 \leqslant z_i \leqslant N_i$ for $i = 1, \dots, 2n$ imply that for $p \in \tau$, $0 \leqslant soc_p \leqslant C$ and $x_p = \alpha_p q_p$ for $q_p = (1 - z_{2p}) \in \{0, 1\}$ and $0 \leqslant y_p \leqslant \beta_p$.

Again, as $\sum_{i=1}^{2n} v_i z_i = \sum_{i=1}^{n} b_p \alpha_p - \sum_{i=1}^{n} (b_p x_p + s_p y_p)$ the cost remains the same.

Hence, the feasible solutions of $\mathcal{I}$ and $\mathcal{I}'$ are in bijection with each other and the set of optimal solutions are too. Furthermore, $\mathcal{I}'$ can be constructed in $\mathcal{O}(|\mathcal{I}|)$ time and then the corresponding solution to $\mathcal{I}$ can be computed from the optimal solution to $\mathcal{I}'$ in $\mathcal{O}(|\mathcal{I}|)$ time. □

In the remainder of this section, we extend our dynamic programming algorithm to TKICLUB. First, we order the items $i = 1, \dots, m$ such that their arrival times are monotone increasing, that is $p_j \geqslant p_\ell$ if $j > \ell$. Then we have subproblems $SP(i, U_{p_i} - k)$ for each $i = 1, \dots, m$ and $0 \leqslant k \leqslant U_{p_i} - L_{p_i}$, which correspond to the restriction of the instance to the first p_i periods and the first i items, such that in the end exactly $U_{p_i} - k$ weight is in the knapsack. We store these optima in $R[k]$ for each iteration i. Beyond this, the idea of the algorithm remains as before.

Algorithm 2.2 TKICLUB

Data : A TKICLUB instance $\mathcal{I} = (n, m, w_i, v_i, N_i, t_i, L_i, U_i \mid i \in I)$.
Output : The optimum value for $\mathcal{I}$.

$$R[k] := \begin{cases} 0 & \text{if} \quad k = 0 \\ -\infty & \text{otherwise} \end{cases} \quad \text{for } k = 0, 1, \dots, \max_j\{U_j - L_j \mid j \in \tau\}. \ ;$$

```
// initialization for the empty subproblem
```
$L_0 := 0, U_0 := 0$
$Q := [0, \dots, 0]$; `// store the optima for the new subproblems // in Q before updating R`

for $i = 1, \dots, m$ **do**
  for $k = 0, \dots, U_{p_i} - L_{p_i}$ **do**
    $a_0 = \max\{0, \lceil \frac{U_{p_i} - U_{p_{i-1}} - k}{w_i} \rceil\}$
    $a_1 = \min\{N_i, \lfloor \frac{U_{p_i} - L_{p_{i-1}} - k}{w_i} \rfloor\}$; `// these bounds force that // the Q[k] only queries R-values // from the range` $[0, U_{p_{i-1}} - L_{p_{i-1}}]$
    $Q[k] = \max_{a_0 \leqslant x_i \leqslant a_1} \{R[U_{p_{i-1}} - U_{p_i} + k + w_i x_i] + v_i x_i\}$
  Set $R := Q$ and $Q := [0, \dots, 0]$.
return $\max_{k=0,\dots,C}\{R[k]\}$

Theorem 2.5. *Algorithm 2.2 solves* TKICLUB *optimally in time* $\mathcal{O}(mW \log W)$, *where* $W = \max_{i \in [m]}\{U_i - L_i\}$.

Proof. We show that at the end of iteration for item i, we have stored the optimal values for subproblems $SP(i, U_{p_i} - k)$ in $R[k]$ for $k = 0, 1, \dots, U_{p_i} - L_{p_i}$.

Consider $i = 1$. Then one must pack $0 \leqslant x_1 \leqslant N_1$ copies of item 1 into the knapsack. Hence, $Q[k]$ must be set to $v_1 x_1$, whenever $U_1 - k = w_1 x_1$ and $0 \leqslant x_i = \frac{U_1 - k}{w_1} \leqslant N_1$ (so $a_0 = a_1$ in Algorithm 2.2), otherwise $-\infty$. It is easy to see that this is indeed how the algorithm sets Q in the first iteration $i = 1$.

Suppose that the algorithm correctly computes the optimum for the subroblems $U(i-1, U_{p_{i-1}} - k)$ for $i-1$. As $p_1 \leqslant \cdots \leqslant p_m$, we also have that for the currently processed item, the only lower and upper bounds, where its weight matters are L_{p_i} and U_{p_i}, hence feasibility holds because of induction and the fact that in the end, we leave $L_{p_i} \leqslant W \leqslant U_{p_i}$ total weight in the knapsack. If we must have $U_{p_i} - k$ weight in the knapsack in the end, and we put $0 \leqslant x_i \leqslant N_i$ copies of item i in, then from the first $i-1$ items, we must have weight exactly $U_{p_i} - k - w_i x_i$ from the first $i-1$ items. Hence, in this case we need to pack the first $i-1$ items optimally such that in the end we have exactly this much weight. By induction, we stored this optimum in $R[\ell]$, where $U_{p_{i-1}} - \ell = U_{p_i} - k - w_i x_i$, which yields $\ell = U_{p_{i-1}} - U_{p_i} + k + w_i x_i$, which gives total profit $R[\ell] + v_i x_i$ as we have in Algorithm 2.2.

Then we have that $L_{p_{i-1}} \leqslant U_{p_i} - k - w_i x_i \leqslant U_{p_{i-1}}$, hence to have a feasible solution, the weight $U_{p_{i-1}} - \ell$ must be valid for the bounds for p_{i-1}. Hence, we must have that $L_{p_{i-1}} \leqslant U_{p_i} - k - w_i x_i \leqslant U_{p_{i-1}}$, so

$$\frac{U_{p_i} - U_{p_{i-1}} - k}{w_i} \leqslant x_i \leqslant \frac{U_{p_i} - L_{p_{i-1}} - k}{w_i},$$

leading to the bounds $a_0 \leqslant x_i \leqslant a_1$ in the algorithm. Therefore, the algorithm checks each possible x_i value and correctly computes to optimum for the subproblem $SP(i, U_{p_i} - k)$ for each k in iteration i. The run time follows from the same arguments as for Algorithm 2.1. □

2.5 EXPERIMENTAL RESULTS

We implemented all combinatorial algorithms in the Python programming language. To measure their experimental efficiency, we also implemented the ILP formulations of Section 2.2 and solved them with Gurobi 10.0.2 on simulated instances. We ran all experiments on an Intel iCore5 CPU at 2.3 GHz with 4 GB of memory. We took a systematic approach to simulate instances. To ensure variability in the instances [55], we generated them randomly from different distributions. Firstly, the battery capacity C is simulated from a uniform distribution with support $[0, 1\,000]$. Then, C is taken as upper bound for generating the initial state of battery soc_0. The values d'_p, ex_p were generated by first sampling X_p from a normal distribution with mean $C/4$ and then adding another uniform random variable Y_p with support $[-C/6, C/2]$ and setting $ex_p = \max\{X_p - Y_p, 0\}$ and $d'_p = \max\{Y_p - X_p, 0\}$. The prices for buying and selling are simulated by means of two Gaussian distributions tuned such that their probability density functions overlap. Such a choice is an effort to simulate real market conditions. Indeed, even if on average the prices for buying energy are higher than those for selling, if we consider prices on different periods, it could be that the price for selling is higher.

We start by comparing the performance of Algorithm 2.1 with the Gurobi solver for the corresponding ILP. We simulated 20 different instances with the number of periods ranging from 100 to 600. The upper cutoff of 600 was chosen experimentally, as beyond this value the solver was often unable to produce a solution. For the capacity we set 1,000. Figure 2.1 shows the average experimental run times for each time horizon selected.

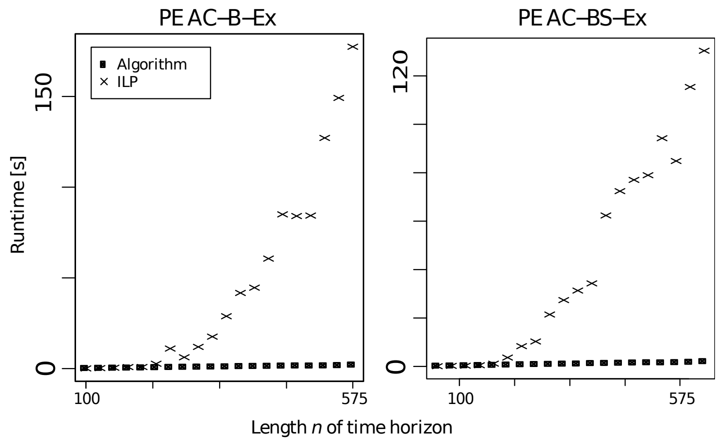


Figure 2.1: Experimental run times for solving PEAC-B-Ex and PEAC-BS-Ex-General on the simulated instances.

As expected, the times for solving instances with Algorithm 2.1 were significantly shorter and remained consistently within seconds even for the largest simulated values of C and n, whereas Gurobi often exceeded the time limit of 180 seconds to solve them.

3 PROSUMER OFFLINE PROBLEM

In this chapter we discuss the variants of the PEAC model introduced in Chapter 2 where the prosumer is allowed to buy a smaller amount of energy than the whole demand from the main grid. In other words, the net demand d'_p of a day can now be satified partially by buying energy and partially by using the energy stored in the battery. This results in relaxing the model (2.1)–(2.9) presented in the last chapter by removing the constraints (2.3) and (2.8).

This chapter is based on joint work with Gergely Csáji and Matthias Mnich [28].

3.1 PROBLEM VARIANTS

We report here the new model obtained when removing constraints (2.3) and (2.8):

$$\min \sum_p (b_p x_p - s_p y_p) \tag{3.1}$$

$$\text{s.t.} \quad x_p - y_p - z_p = d_p - e_p, \quad p \in \tau, \tag{3.2}$$

$$SOC_p = SOC_{p-1} + z_p, \quad p \in \tau, \tag{3.3}$$

$$SOC_0 = soc_0, \tag{3.4}$$

$$SOC_p \leqslant C, p \in \tau, \tag{3.5}$$

$$x_p, y_p, SOC_p \in \mathbb{N}, \quad p \in \tau, \tag{3.6}$$

$$z_p \in \mathbb{Z}, \quad p \in \tau\ . \tag{3.7}$$

We have already explained the meaning of all the variables and the constraints in Chapter 2, so we limit ourselves to remark that in this problem, the prosumer can fill up the battery by buying energy, and can buy or sell an arbitrary amount of energy in each period. Furthermore, the prosumer could also buy more energy of than they need and store some of it in the battery, as indeed the amount they can buy in each period x_p is not limited. The same applies for the variables related to the selling, thus we refer to this model as the UNBOUNDED version of PEAC and we call it PEAC-BS-U. As in the models of the last chapter, BS refers to the fact that both buying and selling is allowed:

PEAC-BS-U

In: An instance $\mathcal{I} = (b_p, s_p, d_p, e_p, C, soc_0 \mid p \in \tau)$.

Out: An optimal solution to the ILP defined by objective (3.1) and constraints (3.2)–(3.7).

The problem variant, where selling is not allowed, is called PEAC-B-U and it is defined by adding the constraints $y_p = 0, p \in \tau$, to PEAC-BS-U:

> PEAC-B-U
>
> **In:** An instance $\mathcal{I} = (c_p, d_p, e_p, C)$.
>
> **Out:** An optimal solution to the ILP (3.1) – (3.7), together with the constraints $y_p = 0, p \in \tau$.

As in the previous chapter, we generalize our model so that it can take into account also possible restrictions from the microgrid, the HEMS or the battery itself. Another aspect which could play a role here is fairness. Indeed, it could be that—for fairness reasons—prosumers are not allowed to buy more than the amount they need on a given period p. The same type of restriction is already to be observed in the literature, as in Hafiz et al. [47], where indeed the battery can only be charged with self-generated energy from PVs. In order to add this feature in our model, we bound the amount of energy which can be bought by adding the following constraint to PEAC-BS-U:

$$x_p \leqslant (d_p - e_p)^+, \quad p \in \tau \ . \tag{3.8}$$

This allows us to formally define our next problem variant:

> PEAC-BS
>
> **In:** An instance $\mathcal{I} = (b_p, s_p, d_p, e_p, C, soc_0 \mid p \in \tau)$.
>
> **Out:** An optimal solution to the ILP defined by (3.1)–(3.8).

Analogously, we define the variant PEAC-B, where selling is not allowed, by adding the constraints $y_p = 0$ for $p \in \tau$ to PEAC-BS.

We find it convenient to parameterize the last variants defined, so that the different variations of PEAC-BS-U and PEAC-BS can be linked together. To do so, we define common generalization of these models and we call it PEAC-BS-General. Here, instead of (3.8) we add inequalities

$$x_p \leqslant \alpha_p, \quad p \in \tau, \tag{3.9}$$

$$y_p \leqslant \beta_p, \quad p \in \tau, \tag{3.10}$$

where $\alpha = (\alpha_1, \ldots, \alpha_n)$ and $\beta = (\beta_1, \ldots, \beta_n)$ are given in the input.

Clearly, PEAC-BS-U corresponds to the case when we set each α_p and β_p sufficiently large, e.g., $\alpha_p = (C + d'_p - ex_p)^+$ and $\beta_p = (C + ex_p - d'_p)^+$ or simply set them to infinity. Similarly, PEAC-BS corresponds to the case where we set $\alpha_p = d'_p = (d_p - e_p)^+$ and $\beta_p = C + ex_p$ for each $p \in \tau$.

The variant PEAC-B-General is defined similarly, by adding constraints $y_p = 0$ for all $p \in \tau$; equivalently, we can set $\beta_p = 0$.

Hence, it is trivial that PEAC-B-General is a special case of PEAC-BS-General, and so are PEAC-B-U and PEAC-BS-U. However, we will show that PEAC-BS-GENERAL also reduces to PEAC-B-GENERAL in $\mathcal{O}(|\mathcal{I}|)$ time for a given instance $\mathcal{I}$, and hence it is enough to give an algorithm for PEAC-B-GENERAL to solve all the above problem variants, meaning that we can assume that only buying is allowed.

Before proceeding to our main results, we mention another approach often used in literature to deal with these type of problems. An important aspect of these types of scheduling problems is the intrinsic uncertainty of the parameters [62]. A popular method to deal with this stochasticity is to use a simulation approach. Closely related to the models in this paper is the work of Hafiz et al. [47], which extends the *stochastic dual dynamic programming* approach, already extremely popular in hydrothermal scheduling problems, as in Morton [66]. The main idea of this approach is to treat the right term of Equation (2.2) as stochastic and use a Bender decomposition scheme to approximate the solution. For sake of the aforementioned generality, we decided to introduce variability also on the prices for buying and selling energy. In this new setting, the use of a Monte Carlo scheme becomes very difficult as the scenario tree grows exponentially in several parameters. Thus, the decision relies on predictions about the expected amount of PV-generated energy in future periods. Motivated by the rapid advance of HEMS and predictor technologies [70], we study the offline version relying on a good predictor. Even though in practice, decisions regarding how much to buy and sell in a given period are made in an online fashion, a good offline strategy can lead to good decisions, if in each period, we base the online decision on solving an offline version for the next period according to some predicted prices, especially if these predictions are highly accurate.

This is especially the case for the variants PEAC-B-GENERAL and PEAC-B-EX-GENERAL, for which we show later in Theorem 3.6 that if predictions have small multiplicative errors, then a solution for the offline problem remains near-optimal even in the worst case. Therefore, for any instance of the studied version of PEAC-B-GENERAL and PEAC-B-EX-GENERAL where the precise prices are unknown, but we have accurate predictions for the buying prices, our optimal solution for the predicted prices remains near-optimal.

Finally, we will relax the assumption of perfect conservation of energy. Namely, we will enhance our models (in Section 3.3.2) so that they take into account losses due to the charging and discharging, which are functions of the charged and discharged power [21].

3.2 STRUCTURAL RESULTS

In this section we analyze the structure of the ILPs. Recall that a matrix $A \in \mathbb{R}^{m \times n}$ is *totally unimodular* if each subdeterminant of A has value $\{0, \pm 1\}$. It is well-known (cf. [76, Theorem 19.1]) that any linear program whose constraint matrix A is totally unimodular admits an integer optimum solution, whenever the bounding vector is integral.

Theorem 3.1. *The constraint matrix of the ILP* (3.2)–(3.7) *with* (3.9)–(3.10) *is totally unimodular.*

Proof. Let $A \in \mathbb{R}^{2n \times 4n}$ be the constraint matrix of the ILP (3.2)–(3.7), and let the columns of A correspond to the variables

$$[x_1 \quad y_1 \quad z_1 \quad SOC_1 \quad \dots \quad x_n \quad y_n \quad z_n \quad SOC_n] \ .$$

To show that A is totally unimodular, we show that we can iteratively find one row or one column with only one non-zero entry, and this entry is equal to ± 1. We then remove this row or column from the matrix, and repeat, until we eventually arrive at the empty matrix. As adding such a row or column to a totally unimodular matrix preserves the total unimodularity property, this implies that our matrix is totally unimodular.

The columns corresponding to the x_p and y_p variables have only one non-zero entry (which is a ± 1), so we can remove them, amd for showing the result it is enough that the rest of the matrix is totally unimodular. Now, the rows corresponding to (2.2) have only one nonzero entry (which is a -1), so they can be removed too.

Then, in the remaining matrix, we only have rows of the form $SOC_p - SOC_{p-1} - z_p = 0$ for $p = 1, \dots, n$. The variables z_p for $p = 1, \dots, n$ only appear in one constraint, hence in one equation, so we can delete their columns. Note that $SOC_0 = soc_0$ is a constant, not a variable. Hence, if we remove the rows and columns in an alternating sequence in the order $SOC_1 - SOC_0 - z_p = 0$, SOC_1, $SOC_2 - SOC_1 - z_2 = 0$, $SOC_2, \dots, SOC_p - SOC_{p-1} - z_p = 0$, SOC_p, then in each step, there is at most one nonzero (± 1) entry in the deleted row or column. Therefore, we reduced to matrix to the empty matrix by such steps, so it is indeed totally unimodular.

Finally, the constraints $SOC_p \leqslant C$, $x_p, y_p, SOC_p \geqslant 0$ as well as the constraints $x_p \leqslant \alpha_p$, $y_p \leqslant \beta_p$ in (3.9)–(3.10) only add rows with at most one nonzero element to A, so they preserve total unimodularity, too. □

Theorem 3.1 has the following corollary.

Corollary 3.2. PEAC-BS-General *can be solved in polynomial time.*

Even if total unimodularity implies that one can solve the problem by means of an LP solver in polynomial time, we provide a combinatorial algorithm that solves it in $\mathcal{O}(n^2)$ time. On the one hand, the combinatorial algorithm gives a better run time guarantee compared to solving the problem via its LP formulation; on the other hand, the combinatorial algorithm provides a more thorough insights into the structure of the problem.

We conclude this section with a general remark regarding the feasibility of the instances. In case the parameters β_p are all zero, which corresponds to the impossibility of selling for the prosumer, feasibility might be an issue, as shown by the following toy example.

Example 3.1. *Consider an instance with* $n = 2$, $d_p = [0, 0]$, $e_p = [1, 1]$, $SOC_0 = 0$ *and* $C = 1$. *As the prosumer cannot sell, and there is no demand to satisfy, the excess energy exceeds the battery capacity in the second period.*

Let $K = \{p \in \tau \mid e_p - d_p > 0\}$ be the subset of periods with positive excess, whose elements we label increasingly as $k_1, \ldots, k_m$. Let $k_0 = 0$ correspond to the 0^{th} period, where we get an excess energy of soc_0 to begin with. Then, we obtain the following characterization of feasible instances:

Lemma 3.3. *Any instance $\mathcal{I}$ of* PEAC-B-GENERAL *has a feasible solution if and only if*

$$\sum_{p=k_i}^{k_\ell} d'_p \geqslant \sum_{j=i}^{\ell} ex_{k_j} - C, \quad \text{for all } k_i < k_\ell \in K \ . \tag{3.11}$$

Proof. If there is a pair $k_i < k_\ell$ for which the relation (3.11) is violated, then starting at period k_i, we get $\sum_{j=i}^{\ell} ex_{k_j}$ units of excess energy up until period k_ℓ, so even if we satisfy all energy demands by taking energy from the battery, not enough storage remains to store ex_{k_ℓ} units energy in period k_ℓ.

In the other direction, if an instance is infeasible, then it means that even if we always prioritize using the battery whenever it is not empty to satisfy a demand, and only buy energy when the battery is empty, we cannot keep it under its capacity. Hence, there is a period k_ℓ where the excess energy would make the battery overfull. Choosing k_i to be the last excess period before k_ℓ such that the battery had 0 charge before k_i (if it was never 0, then $k_i = k_0 = 0$), we obtain that from period k_i we only use the battery and never buy energy. Hence, the total battery drain in the interval is exactly $\sum_{p=k_i}^{k_\ell} d'_p$ and the excess energy obtained in the interval – with the battery being empty before that – is $\sum_{j=i}^{\ell} ex_{k_j}$ (note we assumed that soc_0 is also obtained on an artificially created 0-th period), so the inequality is violated for the pair $k_i < k_\ell$. $\square$

One can also think of an infeasible instance in such cases as one where the prosumer must let some amount of energy go to waste.

3.3 ALGORITHMIC RESULTS

We illustrate now our analysis for PEAC-B-GENERAL.

3.3.1 *A polynomial-time algorithm for* PEAC-B-GENERAL

In this section we provide a polynomial-time algorithm to solve problem PEAC-B-GENERAL and show that this also allows to solve PEAC-B-U, PEAC-BS-U, PEAC-B and PEAC-BS efficiently. For this, we first show that an instance of PEAC-BS-GENERAL can be reduced to an instance of PEAC-B-GENERAL in linear time. When analyzing run time, each addition and multiplication is assumed to have unit cost.

Theorem 3.4. *Any instance $\mathcal{I}$ of* PEAC-BS-General *can be reduced to an instance $\mathcal{I}'$ of* PEAC-B-General *in $\mathcal{O}(|\mathcal{I}|)$ time, such that if $\mathcal{I}'$ has an optimal solution, we can construct an optimal solution for $\mathcal{I}$ in polynomial time.*

Proof. Let $\mathcal{I} = (b_p, s_p, d_p, e_p, \alpha_p, \beta_p, C, soc_0 \mid p \in \tau)$ be an instance of PEAC-BS-General. From $\mathcal{I}$ we create an instance $\mathcal{I}'$ of PEAC-B-General as follows. We create two periods for each original period, so $2n$ periods in total. We keep $\hat{C} = C$ and $\hat{soc}_0 = soc_0$. For $p \in \tau$ let $\hat{b}_{2p-1} := b_p$, $\hat{b}_{2p} := s_p$, $\hat{d}_{2p-1} := d'_p, \hat{d}_{2p} = \beta_p$ and $\hat{e}_{2p-1} = 0$, $\hat{e}_{2p} = ex_p$. Finally, set $\hat{\alpha}_{2p-1} = \alpha_p$ and $\hat{\alpha}_{2p} = \beta_p$. Clearly, we can construct $\mathcal{I}'$ in $\mathcal{O}(n)$ time.

In the forward direction, let $S = (x_1, y_1, \dots, x_n, y_n)$ be a solution to $\mathcal{I}$ such that $x_p y_p = 0$ for $p \in \tau$ (at least one optimal solution satisfies this condition, by the assumption that $s_p \leqslant b_p$). Let $\hat{S} = (\hat{x}_1, \dots, \hat{x}_{2n}) = (x_1, \beta_1 - y_1, x_2, \beta_2 - y_2, \dots, x_n, \beta_n - y_n)$.

We claim that $\hat{S}$ is feasible for $\mathcal{I}'$. First of all, $y_p \leqslant \beta_p$ holds, hence $\hat{S} \geqslant 0$. Also, $y_p \geqslant 0$ so $\beta_p - y_p \leqslant \hat{\alpha}_{2p} = \beta_p$. Let SOC_p denote the state of charge of the battery at the end of period p in S, so $0 \leqslant SOC_p \leqslant C$ for $p = 1, \dots, n$.

First, consider the case of an even period $p \in \{2, 4, \dots, 2n\}$. Then, using induction on p and the fact that $\hat{SOC}_0 = soc_0$,

$$\hat{SOC}_p = SOC_{(p-2)/2} + x_{p/2} + \beta_{p/2} - y_{p/2} - d'_{p/2} + ex_{p/2} - \beta_{p/2},$$

which is equal to $SOC_{p/2}$, so $0 \leqslant \hat{SOC}_p \leqslant C$.

Second, consider the case that $p \in \{1, \dots, 2n\}$ is odd. Then we end the period p with a battery state of charge of

$$\hat{SOC}_p = SOC_{(p-1)/2} + x_{(p+1)/2} - d'_{(p+1)/2} \geqslant 0,$$

because if $d'_{(p+1)/2} > 0$, then $ex_{(p+1)/2} = 0$, so

$$\begin{aligned} SOC_{(p+1)/2} &= SOC_{(p-1)/2} + x_{(p+1)/2} - d'_{(p+1)/2} \\ &- y_{(p+1)/2}(+ex_{(p+1)/2}) \geqslant 0 \ . \end{aligned}$$

Also, $\hat{SOC}_p \leqslant C$, because if $x_{(p+1)/2} > 0$, then $y_{(p+1)/2} = 0$, so

$$\hat{SOC}_p \leqslant SOC_{(p+1)/2} - ex_{(p+1)/2} \leqslant C \ .$$

Hence, $\hat{S}$ is feasible.

It is also clear that $\sum_{p=1}^{2n} \hat{x}_p = \sum_{p=1}^{n} x_p + \sum_{t=1}^{n} \beta_p s_p$ and therefore the two solutions S and $\hat{S}$ attain the same optimum value.

In the other direction, if $\hat{S} = (\hat{x}_1, \dots, \hat{x}_{2n})$ is a solution to $\mathcal{I}'$, then we create

$$S = (\hat{x}_1, \hat{\alpha}_2 - \hat{x}_2, \dots, \hat{x}_n, \hat{\alpha}_{2n} - \hat{x}_{2n}) = (x_1, y_1, \dots, x_n, y_n) \ .$$

Then $0 \leqslant x_p \leqslant \alpha_p$ and $0 \leqslant y_p = \beta_p - \hat{x}_{2p} \leqslant \beta_p$, as $0 \leqslant \hat{x}_{2p} \leqslant \beta_p = \hat{\alpha}_{2p}$. Furthermore, $SOC_p = \hat{SOC}_{2p}$ for any $p = 1, \dots, n$, so S is feasible. Also the solutions S and $\hat{S}$ attain the same optimum value, since the value attained by S is $\sum_{p=1}^{n} x_p b_p - \sum_{p=1}^{n} y_p s_p$, which can be decomposed into $\sum_{p=1}^{2n} \hat{x}_{2p-1} \hat{b}_{2p-1} - \sum_{p=1}^{2n} \hat{x}_{2p} s_{2p}$, so as the costs of $\hat{S}$ minus the term $\sum_{t=1}^{n} \beta_p s_p$ as before.

Therefore, $\mathrm{OPT}_{\mathcal{I}'} = \mathrm{OPT}_{\mathcal{I}} + \sum_{p=1}^{n} \beta_p s_p$, and we can construct an optimal solution for $\mathcal{I}$ from an optimal solution for $\mathcal{I}'$ in $\mathcal{O}(|\mathcal{I}|)$ time. □

We proceed to informally describe our algorithm for PEAC-B-General. The algorithm iterates through periods $p = 1, \ldots, n$, and at each point it computes a feasible solution for the first couple of periods, which is maintained by updating the amounts of energy bought on the previous periods. The algorithm keeps track of the latest period f where the battery is full in the current solution, initialized to 0. It also keeps track, for each period $p' < p$, of the largest possible amount $R_{p'}$ which we can add to $x_{p'}$ in our current solution, without creating a period $p'' \geqslant p'$ where the state of charge would exceed C. We initialize the variables $R_{p'}$ to α_p for each period, so that we can handle the possibility of buying even the maximum amount $C + d'_p$ on period p, if feasible (buying more can never be feasible). After period p, the value of R_p gets updated to $C - SOC_p$, representing the space still free in the battery and it weakly decreases from this point depending on the later actions.

In a given iteration corresponding to a period p, the algorithm does the following. If $ex_p > 0$ or $SOC_{p-1} \geqslant d'_p$, then it means that we do not need to buy anything to maintain a feasible solution for the first p periods, we just update the variables R_p, f and SOC_p according to this. Otherwise, we have to satisfy the additional demand $d'_p - SOC_{p-1}$ as follows. We always compute the (latest) period i^* with minimum buying price among periods where we can still buy on (i.e., those with $p > f$ and $x_p < \alpha_p$). Then, we look at the maximum amount we can add to x_{i^*} in a feasible solution, and if x_{i^*} is at least the remaining demand, then we update this value to the remaining demand, update the variables and proceed to the next period. Otherwise, we increase x_{i^*} by as much as we can by maintaining feasibility, update the R_p, f and SOC_p variables, and then finally update i^* and iterate with until the demand is fully satisfied. The algorithm's pseudocode is given as Algorithm 3.1.

Theorem 3.5. *Algorithm 3.1 outputs an optimal solution for* PEAC-B-General *in* $\mathcal{O}(n^2)$ *time if such exists.*

Proof. First of all, we show that if the algorithm outputs a solution, then it is feasible. To see this, observe that when we are at the end of period p, then for all $f < t \leqslant p$, the variable R_t correctly measures the maximum amount such that $SOC_j + R_j \leqslant C$ for $j = t, t+1, \ldots p$, which is $\min_{j \in \{t,\ldots,p\}}\{C - SOC_j\}$. Indeed, at the end of "period 0", it is correctly initialized as $R_0 = C - SOC_0$. Then, by induction on p, in case (1), $R_p = R_{p-1} - ex_p = C - SOC_{p-1} - ex_p = C - SOC_p$, and for $t < p$, it only changes is $C - SOC_t > C - SOC_p$, in which case the algorithm correctly updates it. In case (2), namely $R_p = R_{p-1} + d'_p = C - SOC_{p-1} + d'_p = C - SOC_p$ and also in this case

Algorithm 3.1 Buying backwards

Data : An instance $\mathcal{I} = (b_t, d_p, e_p, \alpha_p, C, soc_0 \mid p \in \tau)$ of PEAC-B-GENERAL.

Output : The schedule corresponding to an optimal solution $(x_1, x_2, \ldots, x_n)$ and its optimum value cost.

Initialize cost $:= 0, SOC_0 = soc_0, f = 0, R_0 = C - SOC_0$; `// f is latest period when the battery is full`

$x_p := 0, d'_p = (d_p - e_p)^+, ex_p = (e_p - d_p)^+$ for $p \in \tau$

$R_p = C + d'_p$ for $p = 1, \ldots, n$; `// smallest remaining capacity // on periods p,...,n, // monotone decreasing, after` R_p `becomes 0, // we never update it or any previous one`

for $p = 1, \ldots, n$ **do**

- $i^* :=$ (latest) $\arg\min\{b_i \mid f < i \leqslant p, x_i < \alpha_i\}$
  `// i* is latest cheapest period when we can still buy`
- **if** *(1)* $ex_p > 0$ **then**
  - $R_p := R_{p-1} - ex_p, SOC_p := SOC_{p-1} + ex_p$
  - **for** $j = f+1, \ldots, t-1$ **do**
    - $R_j := \min\{R_j, R_p\}$; `// update remaining capacities // considering` ex_p
- **else if** *(2)* $R_p = R_{p-1} + d'_p$, $SOC_{p-1} \geqslant d'_p$ **then**
  - $SOC_p := SOC_{p-1} - d'_p$; `// we can satisfy` d'_p `without // buying`
- **else**
  - (3) $d'_p := d'_p - SOC_{p-1}$; `// units of energy that need to // be bought to satify the demand`
  - **while** $d'_p > 0$ **do**
    - $incr := \min\{R_{i^*}, \alpha_{i^*} - x_{i^*}, d'_p\}$; `// amount of` d'_p `// we can satisfy on` i^*
    - $x_{i^*} = x_{i^*} + incr$
    - cost $:=$ cost $+ b_{i^*} incr$
    - **for** $j = p-1, \ldots, i^*$ **do**
      - $SOC_j := SOC_j + incr$
      - $R_j := R_j - incr$
      - **if** $R_j = 0$ **then**
        - $f := j$
        - **break** ; `// update f`
    - **for** $j = f+1, \ldots, i^* - 1$ **do**
      - $R_j := \min\{R_j, R_{i^*}\}$
    - $d'_p := d'_p - incr$; `// amount of demand still to be satisfied`
    - $i^* :=$ (latest) $\arg\min\{b_i \mid f < i \leqslant p, x_i < \alpha_i\}$
  - $SOC_p := 0, R_p := C$
- **if** $SOC_p > C$ **then**
  - **return** "No feasible solution!"
- **else if** $SOC_p = C$ **then**
  - $f := p$

return cost; $x_1, \ldots, x_n$

the minimum $\min_{j\in\{t,\ldots,p\}}\{C - SOC_j\}$ is not influenced by $C - SOC_p$ for any j, so we do not have to update R_j-values for values $j < p$. Finally, in case (3), we end with 0 units of energy in the batteries and R_p is correctly set to C. Furthermore, if we buy on the previous period i^*, then it is also straightforward to verify that R_j gets correctly updated by the inner for loop.

Hence, as we do not proceed to the next period until the demand is fully satisfied, and if we increase x_i on some period, then we increase with at most $\min\{R_i, \alpha_i - x_i, d'_i\}$; it follows that we never exceed the capacity of the battery, nor do we violate a constraint of the problem imposing $x_p \leqslant \alpha_p$.

If the algorithm returns that there is no feasible solution, then there cannot be one, as the algorithm only buys energy from other prosumers if no more energy is left in the battery to satisfy a demand. Hence, if the battery becomes overfull in period p_2, then by choosing p_1 to be the last period before p_2, where the battery was empty (or 0 if it never happened), we get

$$\sum_{p\in[p_1,p_2]:ex_p>0} ex_p - C > \sum_{p=p_1}^{p_2} d'_p, \tag{3.12}$$

so there cannot be a feasible solution by Lemma 3.3.

We prove optimality of the computed solution by induction on $\sum_p \alpha_p$. In case that $\sum_p \alpha_p = 0$, then the only possible feasible solution is $(0,\ldots,0)$ and the algorithm's output is clearly optimal, if it is feasible.

Suppose that $\sum_p \alpha_p > 0$ and the instance has a feasible solution. If the algorithm never buys energy, then the output is clearly optimal. Otherwise, let x_j be the coordinate of the output that is first increased during the algorithm and let $p \geqslant j$ be the last iteration where it is increased. Then, until that point, the algorithm only increased x_j, and none of the other x_i's. Denote the output by $Sol = (x_1,\ldots,x_n)$. As x_j is defined as the first x variable increased and the algorithm only perform positive increments on this set of variables, it holds that $x_j \geqslant 1$.

Let $O^* = (x^*_1,\ldots,x^*_n)$ be an optimal solution for which $|x^*_j - x_j|$ is minimum. We claim that $x^*_j = x_j$.

Suppose first for the contrary that $x^*_j < x_j$. The fact that the algorithm needed to increase buying on j to x_j in iteration p implies that in order to satisfy all the demands of the first p periods, O^* must also buy on some periods $(j \neq) j' \leqslant p$, and choose j' to be the first such period. Furthermore, by the choice of the algorithm, we had that $f < j = i^*$ in iteration p and note that $f < j'$ too, because the only way that $f = \ell > 0$ at the start of iteration p in the algorithm is if $\sum_{i=1}^{\ell}(d_i - e_i) + soc_0 = C$, so no feasible solution can buy in period $i \leqslant \ell$. Hence, we have $\alpha_j > 0, \alpha_{j'} > 0$, and $b_j \leqslant b_{j'}$, because b_j was minimum among those with $\alpha_i > 0$ and $f < i$.

Create a solution $S = (x'_1,\ldots,x'_n)$ from O^* by decreasing $x^*_{j'}$ by 1 and increasing x^*_j by 1. Then, we still have $x_{j'} \in [0,\alpha_j]$ and $x^*_j \in [0,\alpha_j]$, as Sol and O^* are feasible.

Suppose that $j < j' \leqslant p$. Then, $SOC_i^S \geqslant SOC_i^{O^*} \geqslant 0$ for any period i. Furthermore, as $j' \leqslant p$ was the first period other than j that O^* bought energy on, and as Sol is feasible, buying 1 more unit of energy on period j cannot make the battery go above capacity until period j', so $SOC_i^S \leqslant SOC_i^{Sol} \leqslant C$ for $i < j'$. From period j' on, it holds that $SOC_i^S = SOC_i^{O^*} \leqslant C$ for $i \geqslant j'$. Hence, we conclude that S is feasible, but either S has smaller cost than O^*, or S is optimal but $|x_j^S - x_j| < |x_j^* - x_j|$, a contradiction.

Suppose now that $j' < j$. Then $SOC_i^S \leqslant SOC_i^{O^*} \leqslant C$ for any period i. Furthermore, $SOC_i^S \geqslant 0$ for $i < j$ (as Sol did not buy anything before j up until iteration j, but still all demands up to period $j-1$ could be satisfied). Finally, $SOC_i^S = SOC_i^{O^*} \geqslant 0$ for $i \geqslant j$. Hence, S is feasible in this case too, but either S has smaller cost, or the same cost but $|x_j^S - x_j|$ is strictly smaller, contradiction.

For the other direction, suppose that $x_j < x_j^* (\leqslant \alpha_j)$. Then, there must be some some other period j', where the algorithm buys more than $x_{j'}^*$ (otherwise, O^* is not optimal or not the closest optimal solution by $b_j \geqslant 0$). Let $x_{j'}$ for $j' \neq j$ be the coordinate of the output that is first increased strictly above $x_{j'}^*$ in the algorithm and suppose it happened in iteration $p' \geqslant p$. This implies that i^* has changed by that point. However, it was still true that $f < j$ and $x_j < \alpha_j$ (before this increase in $x_{j'}$, we had that $x_i \leqslant x_i^*$ for all i and $x_j < x_j^*$, so if $f \geqslant j$ or $x_j \geqslant \alpha_j$, then this would mean that O^* is infeasible). Hence, $b_{j'} \leqslant b_j$ and $j' > j$ (if $b_{j'} = b_j$, then by the definition of i^*, otherwise if $b_{j'} < b_j$ and $j' < j$, then j' would have been chosen as i^* before j, contradiction).

Create a solution S from O^* by increasing $x_{j'}^*$ by 1 and decreasing x_j^* by 1. Then $0 \leqslant x_{j'}^* + 1 \leqslant x_{j'} \leqslant \alpha_{j'}$ and $0 \leqslant x_j \leqslant x_j^* - 1 \leqslant C$. As $j < j'$, we also have $SOC_i^S \leqslant SOC_i^{O^*} \leqslant C$ for all $i \in \tau$. Also, $SOC_i^S \geqslant 0$ for $i < j'$, as by the end of iteration $j' - 1$, the algorithm bought at most as much as S on each period and still satisfied all demands up until period $j' - 1$. For $i \geqslant j'$, we have that $SOC_i^S = SOC_i^{O^*} \geqslant 0$ too.

So S is feasible, and either S has smaller cost than O^*, or the same cost but $|x_j^S - x_j| < |x_j^* - x_j|$, a contradiction. We conclude that $x_j = x_j^*$.

Now create a new instance $\mathcal{I}'$ from $\mathcal{I}$ by increasing e_j by 1 and decreasing α_j to $x_j - 1$. Then $(x_1^*, \dots, x_j^* - 1, \dots, x_n^*)$ is feasible for $\mathcal{I}'$. Also, one sees that the algorithm outputs $(x_1, \dots, x_j - 1, \dots, x_n)$ for this new instance (until iteration p, if it buys, then it still only buys on period j and in iteration p, it increases this amount only to $x_j - 1$, but since e_j is increased by 1, from this point every R_i, SOC_i and $x_i (i \neq j)$ is the same, so it runs the same way from this point as for $\mathcal{I}$). By induction, this is optimal for $\mathcal{I}'$, so $\sum_p b_p x_p - b_j \leqslant \sum_p b_p x_p^* - b_j$, hence $\sum_p b_p x_p \leqslant \sum_p b_p x_p^*$, so the output is optimal for $\mathcal{I}$, too.

To analyze the run time, note that the outer **for**-loop has n iterations. In the first two cases ($ex_p > 0$ or $SOC_{p-1} \geqslant d'_p$), we make $\mathcal{O}(n)$ steps per iteration. In the third case ($d'_p > SOC_{p-1} \geqslant 0$), in each step, either the demand of period p gets satisfied and we move on to the next period, or period i^* gets saturated ($x_{i^*} = \alpha_{i^*}$), or f gets updated to a later period. Hence, the number of all such updates over all iterations of the outer **for**-loop is at most $3n$, which together with the inner

for-loops to update SOC_j, R_j and f gives $\mathcal{O}(n^2)$ steps. Thus, the overall run time is bounded by $\mathcal{O}(n^2)$. □

3.3.2 *Extensions to our models*

In the models listed above, and in particular in Section 3.4.2, the assumption of perfect conservation of energy is always assumed. However, this hypothesis is not truly realistic and in this section we show how it can be relaxed to enhance our models. To do so, we assume now that there are losses due to the charging and discharging which are functions of the charged and discharged power [21] and we consider them to be linear in this power. The battery energy storage system is then governed by the following dynamics:

$$SOC_p = SOC_{p-1} + \eta_p^{Ch} P_p^{Ch} \Delta p - \eta_p^{Dis} P_p^{Dis} \Delta t, \tag{3.13}$$

where P_p^{Ch} and P_p^{Dis} is the charged and discharged power respectively, whereas η_p^{Ch} and η_p^{Dis} are the coefficients of loss when charging and discharging. Note that for sake of generality we model the coefficients depending on time, as it is reasonable that the batteries deteriorates over time resulting in loss of efficiency, however for small windows of time, such as months, they can be assumed to be constant. Finally, we mention that if needed the objective function can also be extended to

$$\min \sum_{p \in \tau} \left(\sum_{j \in J} b_{p,j} x_{p,j} - \sum_{k \in K} s_{p,k} y_{p,k} \right), \tag{3.14}$$

where there is a set of J different sources to buy energy and K to sell it.

To take these modifications into account, we modify Algorithm 3.1 in the main **for**-loop, as follows. In case we have a positive excess of energy, when storing the energy we have to take into account the loss we incur in for charging, hence we consider a value $ex_p = \eta_p^{Ch} ex_p$. If instead we need energy but we are able to fully fulfill the demand with the stored energy, we have to check whether the energy in the battery is enough to also compensate for the discharge loss, which means we need to check whether $\eta_p^{Dis} SOC_p \geqslant d'_p$ or equivalently we set $d''_p := d'_p / \eta_p^{Dis}$, so that the battery is also modified accordingly. Lastly, the same procedure needs to be applied in the case when the energy stored is not enough and we use the one stored in the battery.In this case the energy needed is $d''_p := d'_p - SOC_{p-1} \eta_p^{Dis}$ as in the previous case, and if the current minimum time period selected i^* is different from the current period p, we also need to make sure to consider the discharge loss, namely the amount of energy which we charge on period i^* is

$$incr = \min\{\eta_{i^*}^{Ch} R_{i^*}, \eta_{i^*}^{Ch} (\alpha_{i^*} - x_{i^*}), d''_p\}, \tag{3.15}$$

given that $t \neq i^*$.

Lastly, to take into account the sets J and K of different sources, it is enough to compute the minimum time period i^* over these larger sets.

3.3.3 *Near optimality given accurate predicted prices*

We now prove formally that if we have an instance I of the studied version of PEAC-B-General and PEAC-B-Ex-General where the precise prices are unknown, but we have accurate predictions for the buying prices, then our optimal solution for the predicted prices remains near optimal. In particular, we have the following:

Theorem 3.6. *Let $\mathcal{I}$ be an instance of* PEAC-B-General *or* PEAC-B-Ex-General. *Let $\mathcal{I}'$ denote the instance with predicted prices b'_p for $p \in \tau$ such that $(1-\varepsilon) \leqslant \frac{b'_p}{b_p} \leqslant 1+\varepsilon$ for some $0 \leqslant \varepsilon < 1$. Then, it holds that*

$$(1-\varepsilon)\mathsf{OPT}(\mathcal{I}) \leqslant \mathsf{OPT}(\mathcal{I}') \leqslant (1+\varepsilon)\mathsf{OPT}(\mathcal{I}) \ .$$

Proof. Let $S = (x_1, \ldots, x_n)$ be an optimal solution for $\mathcal{I}$ and $S' = (x'_1, \ldots, x'_n)$ be an optimal solution for $\mathcal{I}'$. Observe that using the relation $b'_p \geqslant (1-\varepsilon)b_p$ we obtain

$$\mathsf{OPT}(\mathcal{I}') = \sum_{p\in\tau} b'_p x'_p \geqslant (1-\varepsilon)\sum_{p\in\tau} b_p x'_p \geqslant (1-\varepsilon)\mathsf{OPT}(\mathcal{I}),$$

where the last inequality follows from that the instances $\mathcal{I}$ and $\mathcal{I}'$ share the same feasible region, hence S' is a feasible solution for $\mathcal{I}$. Analogously, it holds

$$\mathsf{OPT}(\mathcal{I}) = \sum_{p\in\tau} b_p x_p \geqslant \frac{1}{(1+\varepsilon)}\sum_{p\in\tau} b'_p x_p \geqslant \frac{1}{(1+\varepsilon)}\mathsf{OPT}(\mathcal{I}') \quad .$$

□

The same results cannot be extended to the versions PEAC-BS-General nor PEAC-BS-Ex-General as Example 3.2 will now show.

Example 3.2. *Consider the following setting. The energy profile of the instance $\mathcal{I}$ and $\mathcal{I}'$ is*

$$d_{2p-1} = 0, e_{2p-1} = 1, d_{2p} = 1, e_{2p} = 0,$$

for $p = 1, \ldots, n$. Namely, on odd time periods the prosumer has an extra unit of energy, while one extra unit of energy is needed in the even ones. Suppose now the predicted prices are all unitary, $b'_p = s'_p = 1$ for $p = 1, \ldots, n$. The prosumer could store the extra unit of even periods and then use it for the next ones or keep selling and buying. Assuming that n is even, either strategy leads to a null objective function, so $\mathsf{OPT}(\mathcal{I}') = 0$. Consider now the real prices, which have a higher value in the odd time periods, $b_{2p} = s_{2p} = 1+\varepsilon$. In this case it is better for the prosumer to keep selling and buying as the associated objective function is $\mathsf{OPT}(\mathcal{I}) = \frac{n}{2}\cdot 1 - \frac{n}{2}(1+\varepsilon) = -\frac{n}{2}\varepsilon$ which would contradict Theorem 3.6.

3.4 EXPERIMENTAL RESULTS

As we did in Chapter 2, also for these variants we implemented all the algorithms to perform some sperimental analysis. First, we compared the actual runtime of our algorithms and the one of Gurobi (Section 3.4.1), then we also observed what the solutions of our algorithm looks like in a realistic microgrid of prosumers (Section 3.4.2).

3.4.1 *Simulations*

We conducted simulations regarding PEAC-B-U, PEAC-BS-U, PEAC-B and PEAC-BS with 50 different lengths n of time horizons, ranging from 100 to 10 000. For each time horizon considered, we simulated 100 instances and computed the average run time. The remaining parameters of the simulations are the one already described in Section 2.5. As expected from the $\mathcal{O}(n^2)$ run time guarantee of Theorem 3.5, the battery size had little impact on the algorithm performance. Our experiments showed that the run time grows linearly in n in the simulated instances both for Algorithm 3.1 and the ILP, probably due to the special structure of the created instances.. We observe that the ILP solver performs better when there are many periods with high demands and few periods with excess energy, otherwise Algorithm 3.1 performed better. In both cases, the run times were similar to each other. Finally, Figure 3.1 shows the results of our simulations for problems PEAC-B-General and PEAC-BS-General.

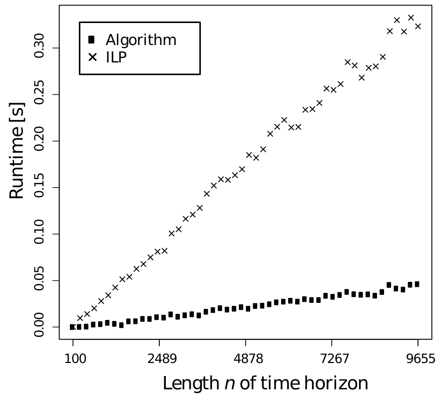
The simulations show that on average our algorithm is comparable with the solver. As as proven in Theorem 3.5 they always both find an optimal solution, from our perspective there is no need to adopt an external software instead of our algorithm.

3.4.2 *Case study*

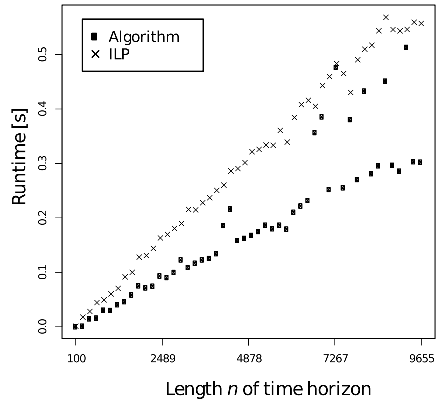
In this section we analyze the results obtained by appling our algorithm on a microgrid. For this purpose, we use the IEEE 15-bus distribution system by Chang et al. [26], represented in Figure 3.2.

This microgrid comprises 14 interconnected prosumers. The dataset spans 24 hours, with a 1-hour time step of, specifying the demand d_p and energy production e_p are specified for each time period.

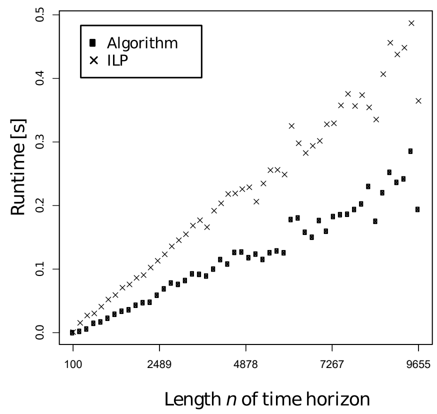
To facilitate energy exchange within this microgrid, we designed a pricing mechanism. The approach we followed is as outlined below, where the main idea is to keep the price for buying fixed for every prosumer and every time period, and model the price for selling depending on the total energy and total demand in the microgrid. To do so, we have first assigned a fixed buying price $b_p \equiv r$ for each time period p. To determine the value of r, we inferred electricity prices from Time-of-Use (ToU) data provided by an energy supplier in Hamburg during a summer day [74], and set r to the average of the values.



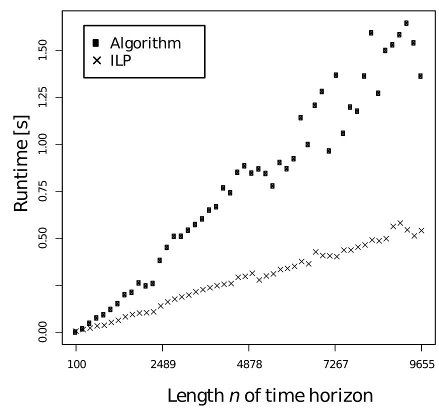
(a) PEAC-B-U



(b) PEAC-BS-U



(c) PEAC-B



(d) PEAC-BS

Figure 3.1: Comparison of the average run time for solving 100 instances with time horizon spacing from 100 and 10,000.

We then modeled the selling prices s_p. These are equal to the buying price r discounted by a factor ε_p, specifically $s_p = r - \varepsilon_p$, where ε_p is directly related to the total energy available in the microgrid relative to the total demand. Thus, we modeled this discount as a function of the ratio between the daily total supply and daily total demand. In particular, ε_p is assumed to be zero when this ratio is at its maximum and equal to r when the ratio is at its minimum. Equivalently this means that the price for selling energy are lower when the ratio of supply and demand is maximum and higher when the opposite happens, which is coherent with one of the central economic principles to describe the relation between price, supply and demand. The remaining values of ε_p are obtained through linear interpolation. The resulting selling prices s_p are shown in Figure 3.3.

Using this dataset, we conducted simulations for the 14 prosumers. A summary of the results for a subset of agents is provided in Figure 3.4. The simulations were repeated under four different conditions, with varying battery capacities of 8, 14, 20, and 26 units. In Figure 3.4 and 3.5 we present the resulting strategies. In particular, in the first sequence of plots we show the demand and the self-produced energy

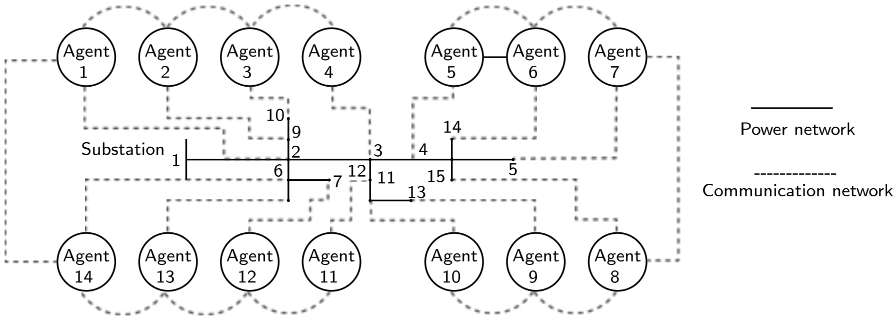


Figure 3.2: IEEE 15-bus distribution system by Chang et al. [26].

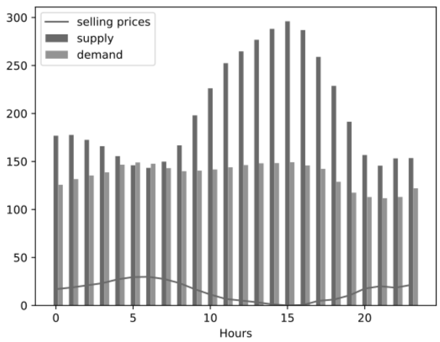


Figure 3.3: Prices for selling energy obtained for the IEEE-15 bus distribution system.

and then the energy bought and sold by each prosumer. In the modeled setting, it is never advantageous for prosumers to purchase more energy than required, which aligns with expected behavior. Energy is only sold when the battery is fully charged or when future periods of surplus are expected. In general, prosumers tend to charge the battery whenever possible and sell the surplus, independently of the battery size. This analysis confirms the expected behavior of the prosumers. We provide the complete code for the experimental analysis described in Section 5.6 in a publicly accessible Git repository at `https://github.com/lauracodazzi/CostMinimizationElectricalEnergyMicrogrid`.

3.5 Final remarks

We presented here the fundamental problem of scheduling battery usage over a finite time horizon in a general and concise manner. We formalize this problem mathematically as an Integer Linear Program (ILP) and provide an algorithm that computes the optimal solution for both the main problem and its variations, while remaining computationally efficient.

A key advantage of our algorithm is that it computes solutions without the need for external solvers, and the results are interpretable and meaningful. We provided an extensive analysis of the proposed algorithms, supported by a detailed case study. We discusseded how the generality of our framework allows our algorithm to be easily adapted to various scenarios.

Figure 3.4: Comparison of 7 prosumers from the microgrid in four setting with a battery of capacity $C = 8$ and $C = 14$. In the first line of plots we observe the demand and the energy from the PVs, in the second the energy bought and sold, respectively in dark and light gray. The third line of plots represents instead the state of charge of the battery.

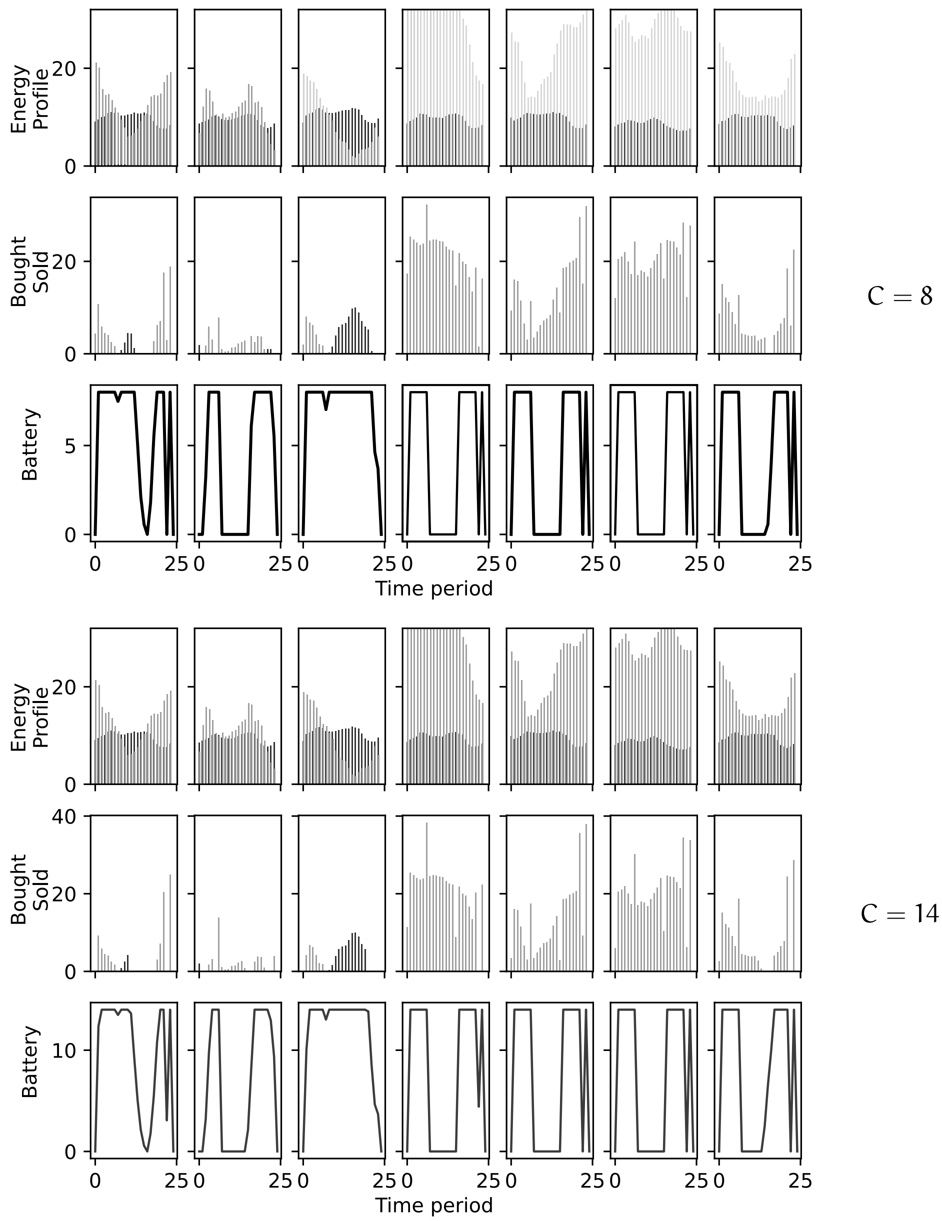
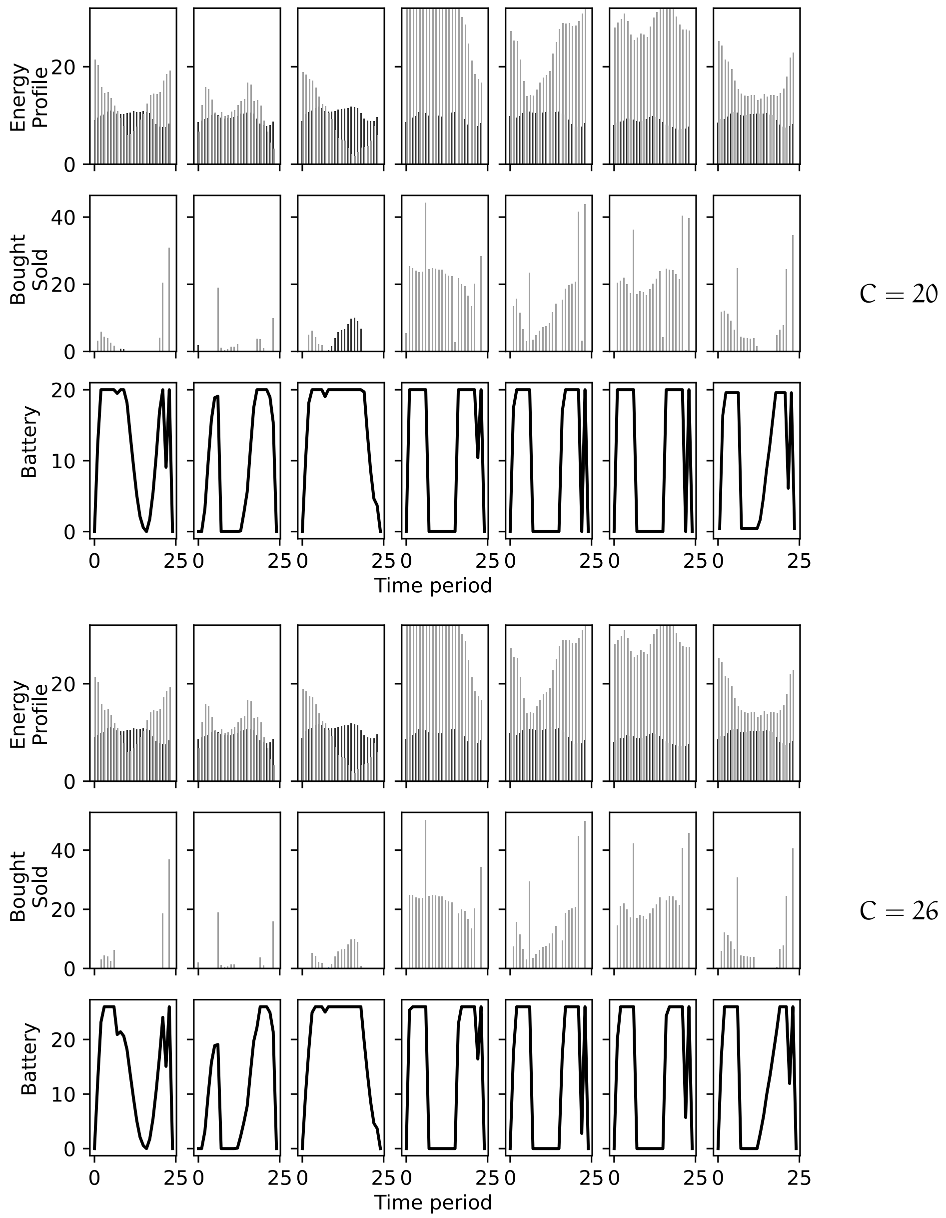


Figure 3.5: Comparison of the same 7 prosumers as in Figure 3.4 with a battery of capacity C = 20 and C = 26.



4 PROSUMER ONLINE PROBLEM

After studying the offline problem, a a natural question arises: is it possible to find a good strategy for our prosumers even when the parameters d_p, e_p and the prices b_p and s_p are unknown? Assuming these parameters are not known a priori introduces new challenges, mainly due to two key aspects. On one hand, there is uncertainty due to the unknown parameters. On the other hand, every decision made at a certain time period impacts on all subsequent ones, about which we also lack information. As a result, analyzing the problem was quite difficult for us, especially because existing analyses of similar problems in the literature fail to account for the physical constraints of the battery. However, it was precisely the battery constraints that allowed us to find a way to analyze this problem. Indeed, since the battery has both an upper and lower limit, we can focus on a subsection of the time horizon, for instance, by considering only what happens between two time periods in which the battery is full. In this chapter, we formally define the intuition described above and present the strategies we developed.

This chapter is based on joint work with Florian Schneider and Matthias Mnich.

4.1 FRAMEWORK

Let us start describing better what are the assumptions in this chapter, namely what is the setting by giving indeed a formal definition of it.

Definition 4.1. *A* setting *is a tuple* $(\mathcal{A}, \mathcal{B})$, *where the set* $\mathcal{A}$ *represents the parameters which are known in advance by the prosumer and* $\mathcal{B}$ *the ones which are instead revealed at runtime in an online fashion.*

So in the offline setting we have $\mathcal{A} = \{n, soc_0, C, b_p, s_p, d_p, e_p\}$ and $\mathcal{B} = \emptyset$. In the online settings considered in this paper, we either have $\mathcal{A} = \{soc_0, C\}$ and $\mathcal{B} = \{n, b_p, s_p, d_p, e_p\}$, or we have $\mathcal{A} = \{n, soc_0, C\}$ and $\mathcal{B} = \{b_p, s_p, d_p, e_p\}$.

Our goal is to determine strategies for the prosumers that provide guarantees on costs or benefits. By *strategy* (or *policy*), we mean the following.

Definition 4.2. *For a setting* $(\mathcal{A}, \mathcal{B})$ *and a time period* p, *given a state of the system* s_p, *which is made of the current state of charge* SOC_p, *the prices for selling and buying at time* p, *as well as the energy and the demand at time* p, *a strategy* π *is a triplet* $(x_{p,s}, y_{p,s}, z_{p,s}) \in \mathbb{N}^2 \times \mathbb{Z}$, *which indicates the amounts of energy that the prosumer is buying, selling, and discharging (or charging if* $z_p < 0$*) from the battery in period* p.

For ease of notation, in the rest of the chapter we will omit the state s every time we talk about a strategy.

We remark that we are using integers as we always refer to units of energy, but one may extend the definition to reals.

A strategy is called *feasible* if the demand d_p of each period is satisfied, and the battery constraints are never exceeded.

Definition 4.3. *For a feasible strategy* $\pi = (x_p, y_p, z_p)$*, its* costs *are defined as* $\mathsf{cost}(\pi) = \sum_{p=1}^{n} b_p x_p$*, whereas the* revenue *generated by it is defined as* $\mathsf{R}(\pi) = \sum_{p=1}^{n} s_p y_p$*.*

Furthermore, in case we need to specify which agent is performing a certain action following a strategy, we use superscript. In this way, the quantity x_p^i is the amount of energy bought by prosumer i at time period p. In principle, we are interested in the profit obtained by the prosumer following a certain strategy, which is defined as the difference of the revenue minus the costs. However, since in this particular the costs are typically higher than the revenue and the incomes are all likely to be negative, we analyze the discounted cost obtained at the end of the time horizon, namely, the bill a prosumer has to pay at the end of the time horizon.

Definition 4.4. *Given a strategy* $\pi = (x_p, y_p, z_p)$ *we define as* bill *the quantity* $\mathtt{bill}(\pi) = \mathsf{cost}(\pi) - \mathsf{R}(\pi)$*.*

Observe that the bill of a strategy it its profit negated.

On top of the variables already introduced and used in the previous chapters, we introduce the following. For each period p, we define the random variable X_p as the difference between the energy obtained in p and the demand of the same period, namely $X_p = e_p - d_p$. Further, we introduce the (non-negative) variables $X_p^+ = \max(0, X_p)$ and $X_p^- = \max(0, -X_p)$.

As the amount of information known apriori in the online setting is really limiting the planning ability of prosumers, elaborating a *good* strategy becomes quite challenging. We therefore start by devising very simple strategies, whose costs are easy to analyze. Thereafter, we make the strategies more and more sophisticated, with the goal of reducing their (expected) costs compared to simpler strategies, and always building our analysis on that of simpler ones.

A key notion which allows us to actually analyze the costs of strategies is to consider only well-structured subintervals of the time horizon, which we term "phases".

Definition 4.5. *Fix a setting* $(\mathcal{A}, \mathcal{B})$ *and a strategy* π *for it. Let an* event $\mathsf{E}(\pi_p)$ *be a binary function which given a strategy* π_p *(we recall this embeddd also a state* s*) returns* 1 *if certain conditions are met and* 0 *otherwise. If at time* p *we have* $\mathsf{E}(\pi_p) = 1$*, we say that the event occurs at time* p*.*

For a given event $\mathsf{E}(\pi_p)$*, a* phase Φ *is a time interval* $[i_P, f_P]$ *such that the event* $\mathsf{E}(\pi)$ *occurs at time* i_P*, and it either happens again only at time* $f_P + 1$ *or the time horizon is over.*

Typical examples of events we have in mind are "the selling price exceeds 40 ct/kWh" or "the prosumer sells 50 kWh stored in their battery". Here we evaluate the $\mathtt{bill}$ of prosumers over phases, namely $\mathtt{bill}_\Phi$ and then we extend the results over the time horizon. As we need a strategy to define a phase, we will now give such in Section 4.2.

4.2 EASY STRATEGY

We start by considering a setting where we only know the battery characteristics and that the prices b_p, s_p are realizations from two identically independently distributed random variables $B_1, \ldots, B_n$ and $S_1, \ldots, S_n$. That is, $\mathcal{A} = \{soc_0, C, B_1, \ldots, B_n iid, S_1, \ldots, S_n iid\}$, $\mathcal{B} = \{b_p, s_p, d_p, e_p\}$. As a first baseline, we consider the so-called dummy strategy, whose pseudocode is available in Algorithm 4.1. The name of this strategy arises from the fact that the prosumer essen-

Algorithm 4.1 Dummy strategy

```
Input : battery capacity C, initial state of charge soc_0.
Output : (x_p, y_p, z_p) dummy strategy strategy.
Initialize: (x_p, y_p, z_p) = (0, 0, 0) for p = 0, ..., n and SOC_0 = soc_0
for p = 1 ... n do
  if d_p ⩾ e_p then
    x_p = d_p − e_p
  else
    if SOC_p + e_p − d_p > C then
      y_p = SOC_p + e_p − d_p − C
      SOC_p = C

    else
      SOC_p = SOC_p + e_p − d_p
  z_p = SOC_p − SOC_{p−1}
return (x_p, y_p, z_p)
```

tially makes no use of the battery. If some amount of energy is needed they buy their energy as usual, and in the case that the produced energy exceeds the demand, they store the excess. Selling happens here only if there is an overflow of the battery.

As discussed in the introduction, when private PVs where first introduced, great focus was devoted to the possibility of selling energy back to the grid. One might erroneously think that selling a higher amount of energy back to the grid might be worth the initial investment for purchasing a battery. We take here a brief moment to show how this is not that convenient in expectation and we do so by defining the strategy only sell (Algorithm 4.2) as follows.

If in a given period p the prosumer needs energy to fulfill their demand, they simply buy the required amount from the main grid. If, instead, there is excess energy that they do not need, i.e. $X_p > 0$, they store this energy in the battery. Whenever the battery reaches its maximum capacity, the prosumer decides to sell a certain amount, for example, discharging $\varepsilon_p C$ of the battery. The newly introduced parameter ε_p express how risk adverse a player is. In case the paramter ε is set to 0 for each time period, then we reduce to the dummy player. For a positive quantity instead, we have that the player is willing to have less than $(1 - \varepsilon_p)C$ stored in the battery and use $\varepsilon_p C$ energy to trade rather than for self consume or backup. Note that, without loss of generality, the prosumer can vary the amount of energy discharged, meaning they sell a certain amount $\varepsilon_p C$, and the selling can occur at different battery loads, not necessarily when the battery is completely full.

Algorithm 4.2 Algorithm for the strategy only sell

Input : Desired percentage ε_p to sell at time p, battery capacity C, initial state of charge soc_0.
Output : (x_p, y_p, z_p) only sell strategy.
Initialize: $(x_p, y_p, z_p) = (0, 0, 0)$ for $p = 0, \dots, n$ and $SOC_0 = soc_0$

```
for p = 1 ... n do
    if d_p ⩾ e_p then
        x_p = d_p − e_p
    else
        if SOC_p + e_p − d_p > C then
            SOC_p = (1 − ε_p)C
            y_p = SOC_p + e_p − d_p − (1 − ε_p)C
        else
            SOC_p = SOC_p + e_p − d_p
    z_p = SOC_p − SOC_{p−1}
return (x_p, y_p, z_p)
```

We compare now two prosumers P_D and P_{OS} who follow the dummy strategy and only sell strategy, respectively on the same energy profile. This means that they have the same d_p, e_p, C and same set of prices for each time period p in the time horizon. As promised, we can now give a concrete example of a phase, and we do so by describing the one we used to compare the two prosumers. In particular consider the time interval between two consecutive times period when P_{OS} sells energy, as showed in Figure 4.1. Before proceeding with the computation, we stress that when estimating the bill of a phase, also the energy stored in the battery at the beginning and at the end of the phase plays a role, since it is the amount the prosumers would earn if he had to sell this amount at that particular period p, namely $SOC_p \cdot s_p$. The bill of P_D is given by the sum of the costs of the energy bought minus the excess sold. The one of P_{OS} is given by the same costs discounted by the initial contribution of energy sold at the beginning of the phase, and then adjusted by the value of the battery. Formally, we have

$$\text{bill}_\Phi(\pi^{P_{OS}}) = \sum_{p=i_P}^{f_P} b_p X_p^- - s_p \varepsilon_p C - s_{i_P} SOC_{i_P} + s_{f_P} SOC_{f_P} \quad . \tag{4.1}$$

By the definition of phase, we then have that

$$SOC_{i_P} + \sum_{p=i_P}^{f_P} X_p^+ = C = SOC_{f_P} - \varepsilon_{f_P} C, \tag{4.2}$$

from which we obtain $\sum_{p=i_P}^{f_P} X_p^+ = SOC_{f_P} - \varepsilon_{f_P} C - SOC_{i_P}$. Given that $\text{bill}_\Phi(\pi^{P_D}) = \sum_{p=i_P}^{f_P} b_p X_p^- - \sum_{p=i_P}^{f_P} s_p X_p^+$, by taking the expected value of the difference, since the prices for selling are identically distributed, we obtain that $\mathbb{E}\left[\text{bill}_\Phi(\pi^{P_D})\right] = \mathbb{E}\left[\text{bill}_\Phi(\pi^{P_{OS}})\right]$. We conclude that the bills of the two strategies are expected to be the same. From the considerations made so far, this implies that if

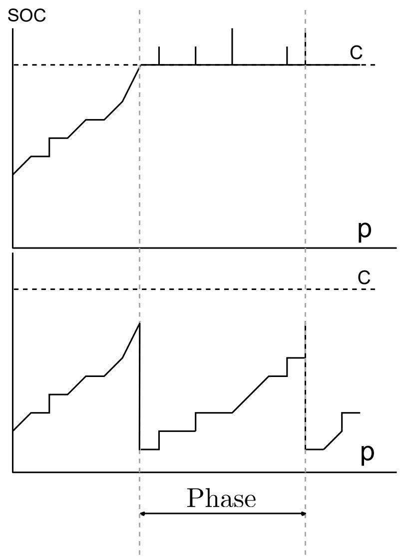


Figure 4.1: Representation of the comparison of dummy strategy and only sell and the defined Phase. In the uppper image we have the strategy of the dummy player. Please notice that here we are representing the strategies as continuous to help the visualization. As by definition of the dummy strategy, we have that the player never decreases their state of charge. In case there is an excess of energy the player just sell it. Even if the SOC does not change, we represent these excesses of energy as the vertical segments above the capacity C to show how the SOC of the only sell player is instead increasing meanwhile exactly because of those amounts.

a prosumer is selling without a well-defined strategy, they would actually have a better expected outcome by not having invested in a battery at all.

Since the dummy strategy has proved to be as good as the more complex only sell we keep considering the dummy strategy as a baseline and discard the other one. In particular we now develop a new strategy by partially altering the behaviour of the prosumer once the battery is full: according to the dummy strategy, prosumers only sell the excess. Now, they can also use this energy to (partially) satisfy their demand on that day. The decision on which action is to be taken once the battery is full, depends then on the sign of the difference $td_p - e_p$. Given the behavior of the prosumer who is both buying and selling, we call this the sell and buy strategy.

Algorithm 4.3 Sell and buy

Input : ε_p desired percentage to sell at time p, battery capacity C, initial state of charge soc_0.
Output : (x_p, y_p, z_p) only sell strategy.
Initialize: $(x_p, y_p, z_p) = (0, 0, 0)$ for $t = 0, \dots, n$ and $SOC_0 = soc_0$

for $t = 1 \dots n$ **do**
    if $d_p \geqslant e_p$ **then**
        if $SOC_p == C$ **then**
            if $\varepsilon_p C > d_p - e_p$ **then**
                $SOC_p = C - d_p - e_p$
            else
                $SOC_p = C - \varepsilon_p C$
                $x_p = d_p - e_p - \varepsilon_p C$
        $x_p = d_p - e_p$
    else
        if $SOC_p + e_p - d_p > C$ **then**
            $SOC_p = (1 - \varepsilon_p)C$
            $y_p = SOC_p + e_p - d_p - (1 - \varepsilon_p)C$
        else
            $SOC_p = SOC_p + e_p - d_p$
    $z_p = SOC_p - SOC_{p-1}$
return (x_p, y_p, z_p)

Lemma 4.6. *For the setting* $(\mathcal{A}, \mathcal{B})$ *with* $\mathcal{A} = \{soc_0, C, B_1, \ldots, B_n iid, S_1, \ldots, S_n iid\}, \mathcal{B} = \{b_p, s_p, d_p, e_p\}$ *the strategy sell and buy obtains a larger revenue in expectation than the dummy strategy.*

Proof. Let P_{SB} be a prosumer following the sell and buy strategy, and P_D be the one following dummy strategy. We define here a phase as the time interval between the first instance when the battery of P_{SB} is no longer full and the next time when it is full again. Notice that the phase can begin with two different types of actions from P_{SB}, since he can both discharge the batery to sell energy or to fulfill their demand. The former case has already been analyzed in above, the latter is the one we consider now. As P_{SB} discharges at most $\varepsilon_p C$, we have that at time i_P they will have to buy only the quantity $x_{i_P}^{P_{SB}} = \max\{0, X_{i_P}^- - \varepsilon_p C\}$ leading to a bill of

$$\text{bill}(\pi^{P_{SB}}) = b_{i_P} x_{i_P}^{P_{SB}} + \sum_{p=i_P+1}^{f_P} b_p X_p^- \ .$$

Then taking the expected difference of the bill of P_D and P_{SB} yields

$$\begin{aligned} &\mathbb{E}\left[\text{bill}(\pi^{P_D})\right] - \mathbb{E}\left[\text{bill}(\pi^{P_{SB}})\right] \\ &= \mathbb{E}[b]\,(x_{i_P}^{P_D} - x_{i_P}^{P_{SB}}) - \mathbb{E}[s]\,\mathbb{E}\left[\sum_{p=i_P}^{f_P} y_p^{P_D}\right] . \end{aligned} \tag{4.3}$$

By definition of phase we have then that

$$\sum_{p=i_P}^{f_P} X_p^+ = \sum_{p=i_P}^{f_P} y_p^{P_D} = \min\{X_{i_P}^-, \varepsilon_p C\},$$

which leads to

$$\mathbb{E}\left[\text{bill}(\pi^{P_D})\right] - \mathbb{E}\left[\text{bill}(\pi^{P_{SB}})\right] = (\mathbb{E}[b] - \mathbb{E}[s])\mathbb{E}\left[\sum_{p=i_P}^{f_P} X_p^+\right] > 0 \ .$$

□

4.3 THE AVOID BUYING STRATEGY

The result of Lemma 4.6 demonstrates that using the energy stored in the battery, rather than buying energy at certain times partially improves the bill of prosumers. To further enhance this behavior, we introduce a new strategy with the idea of maximizing the usage of the energy stored in the battery to satisfy the own demand. Under this strategy, energy is only sold when the stored energy exceeds the battery capacity. We refer to this as avoid buying strategy (Algorithm 4.4), and as before we consider the prosumer P_{AB}. In the following we designate the different battery states as $SOC_p^{P_{SB}}$ and $SOC_p^{P_{AB}}$ respectively. As expected, this new strategy outperforms the previous ones, which

Algorithm 4.4 Avoid buying strategy

Input : Battery capacity C, initial state of charge soc_0.
Output : (x_p, y_p, z_p) only sell strategy.
Initialize: $(x_p, y_p, z_p) = (0, 0, 0)$ for $t = 0, \dots, n$ and $SOC_0 = soc_0$

```
for t = 1 ... n do
    if d_t ⩾ e_t then
        if SOC_p ⩾ d_t − e_t then
            SOC_p = SOC_p − d_t + e_t
        else
            x_p = d_t + e_t − SOC_p
            SOC_p = 0
    else
        if SOC_p + e_t − d_t ⩽ C then
            SOC_p = SOC_p + e_t − d_t
        else
            y_p = SOC_p + e_t − d_t − C
            SOC_p = C
    z_p = SOC_p − SOC_{p−1}
return (x_p, y_p, z_p)
```

we will prove formally. Before though, we show a general result about *price-independent* strategies, which in period p decide regardless of the values of $b_{p'}, s_{p'}$ for all $p' \leqslant p$.

Lemma 4.7. *Consider the setting* $(\mathcal{A}, \mathcal{B})$ *with* $\mathcal{A} = \{soc_0, C, B_1, \dots, B_n iid, S_1, \dots, S_n iid\}$, *and* $\mathcal{B} = \{b_p, s_p, d_p, e_p\}$. *For any two prosumers* P *and* P' *following the price-independent strategies* π, π', *their expected difference of revenues is*

$$(\mathbb{E}[s] - \mathbb{E}[b]) \sum_{p=i_P}^{f_P} \mathbb{E}\left[x_p^P - x_p^{P'}\right] \quad .$$

Consequently, since $(\mathbb{E}[s] - \mathbb{E}[b]) < 0$, the strategy which buys the least in expectation will be optimal.

Proof. We start by analyzing the revenue of P_{SB} and P_{AB} in a generic phase. In this case we have no way to guarantee synchronization of the stored energy amount at the end, and as such a generic phase can also not assume that starting energy is the same. We consider this by again treating the value difference of the stored energy at the beginning and end of the phase. The revenue is then for any strategy defined as $s_{f_P} SOC_{i,f_P} - \sum_{p=i_P}^{f_P} b_p x_p + \sum_{p=i_P}^{f_P} s_p y_p - s_{i_P} SOC_{i,i_P}$. We note that we can always express SOC_{i,f_P} as a function of SOC_{i,i_P} as follows:

$$SOC_{f_P}^P = SOC_{i_P}^P - \sum_{p=i_P}^{f_P} (X_p^- - x_p^P) + \sum_{p=i_P}^{f_P} (X_p^+ - y_p^P) \quad . \tag{4.4}$$

It is now possible to write the general revenue as:

$$(s_{f_P} - s_{i_P})SOC_{i_P}^{P} - s_{f_P} \sum_{p=i_P}^{f_P} (X_p^-) + \sum_{p=i_P}^{f_P} ((s_{f_P} - b_p)x_p^P)$$
$$+ \sum_{p=i_P}^{f_P} ((s_p - s_{f_P})y_p^P) + s_{f_P} \sum_{p=i_P}^{f_P} (X_p^+) \ .$$

Note that as long as we keep our strategy independent on the prices, the x, y stay statistically independent from the prices, allowing us to write the expected revenue as follows:

$$-\mathbb{E}[s] \sum_{p=i_P}^{f_P} \mathbb{E}\left[X_p^-\right] + \sum_{p=i_P}^{f_P} (\mathbb{E}[s] - \mathbb{E}[b])\mathbb{E}\left[x_p^P\right] + \mathbb{E}[s] \sum_{p=i_P}^{f_P} \mathbb{E}\left[X_p^+\right] \ .$$

Note that the first and third terms are independent of the strategy, meaning a difference of two such strategies, can be written as:

$$\sum_{p=i_P}^{f_P} (\mathbb{E}[s] - \mathbb{E}[b])\mathbb{E}\left[x_p^P - x_p^{P'}\right] \ . \qquad \square$$

Using this approach, we can now also show that the avoid buying strategy is optimal under all price-independent strategies.

Theorem 4.8. *The revenue of strategy avoid buying is in expectation better or equal to any price-independent strategy.*

Proof. We follow again the same approach where prosumer P_{SB} follows sell and buy and P_{other} follows any other strategy. From Lemma 4.7 we know that, for strategies independent of the prices, the strategy which buys less has the higher expected revenue, so we only need to show that sell and buy buys less than any potential other strategy.

As avoid buying only buys in the case where the storage is unable to meet demand, we know P_{SB} only buys more than P_{other} on a day where the storage of P_{SB} is lower than that of P_{other}. However, as avoid buying never sells unless impossible not to, the only way P_{other} has more energy stored, is that it has bought that energy previously. So in total, P_{other} has to buy at least as much as P_{SB}. $\square$

4.3.1 *Comparison with the offline setting*

Since in Theorem 4.8 we have shown that the strategy avoid buying is the best we can possibly do in expectation if we do not have extra information regarding the price, we check now how far it is from the offline version of the problem. In particular, since the avoid buying strategy is the most effective when it comes to minimize the amount of energy bought, we ask ourselves how much do we still lose in doing this compared to an offline schedule. In this comparison we refer by OPT to the optimal solution of the offline version, which can be

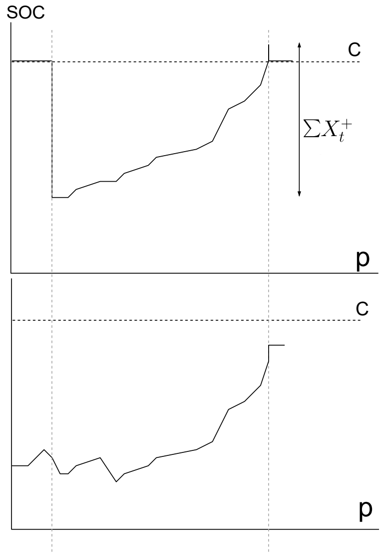


Figure 4.2: Phase comparison between sell and buy and avoid buying

computed with Algorithm 3.1, where all the prices for selling are set to zero. We give now an insight into the strategy OPT. According to OPT the battery usage is prioritized to fulfill the demand of the prosumer. Suppose now that at period p the energy stored is not enough to fulfill the demand. In the offline setting such situation would have been known ahead of time, resulting in the prosumer buying the needed energy at the most convenient time. Of course, this is not the case for real life prosumer who follow an online strategy such as our avoid buying, here denoted with ALG where the energy needed can be only bought at the current time. In Theorem 4.9 we give a bound by considering the worst case for ALG.

Theorem 4.9. *The expected ratio between the costs for buying energy in OPT and the online strategy ALG is bounded by the quantity* b_{max}/b_{min}.

Proof. Consider two prosumers P_{OPT} and P_{ALG} who are optimizing their schedule according to OPT and ALG respectively, where we assume that P_{OPT} is omniscient. We want to denote the set of times Γ where the prosumer is not able to fulfill the demand as follows. First we define the function $SOC(p)$ to keep track of the state of charge of the battery at every time period. Given the definition of ALG namely avoid buying, we have that $SOC(0) = soc_0$ and

$$(4.5)\qquad \begin{cases} SOC(0) & = soc_0, \\ SOC'(p) & = \min\{C, SOC(p-1) + e_p - d_p\}, \\ SOC(p) & = \max\{0, SOC'(p)\} \ . \end{cases}$$

Then we have that

$$(4.6)\qquad \Gamma = \{p \in \tau | SOC'_p < 0\},$$

and the quantity bought in these times is given by

$$\gamma_p = -\min\{0, SOC'(p)\} \ . \tag{4.7}$$

We remark that the value of $SOC(t)$ does not depend on the prices, but only on the quantities d_p and e_p and the two prosumers P_{OPT} and P_{ALG} actually do buy the same amount of energy but not necessary at the same time periods. Consider now the following worst case scenario, namely where we set the prices for buying energy at the highest possible value when ALG buys energy, namely $b_p = b_{max}$ for every $p \in \Gamma$ and all the other prices to b_{min}, in order to allow OPT to buy at cheapest prices. We get then

$$\frac{\mathsf{cost}(\pi^{ALG})}{\mathsf{cost}(\pi^{OPT})} \leqslant \frac{\sum_{p=1}^{n} |\gamma_p| b_{max}}{\sum_{p=1}^{n} |\gamma_p| b_{min}} = \frac{b_{max}}{b_{min}} \ . \qquad \square$$

We point out that in case there are more information regarding the distribution of the prices for buying, it could be possible to improve the bound on the costs. In particular already by knowing the mean $\bar{b}$ we have

$$\mathbb{E}[\mathsf{cost}(\pi^{ALG})] \leqslant \mathbb{E}[\sum_{p=1}^{n} |\gamma_p| b_p] = \sum_{p=1}^{n} |\gamma_p| \mathbb{E}[b_p] \leqslant \mathsf{cost}(\pi^{OPT}) \frac{\mathbb{E}[b]}{b_{min}},$$

from which we obtain the the ratio $\mathbb{E}[\mathsf{cost}(\pi^{ALG})]/\mathsf{cost}(\pi^{OPT})$ is bounded by $\frac{\mathbb{E}[b]}{b_{min}}$.

4.3.2 *Experimental comparison*

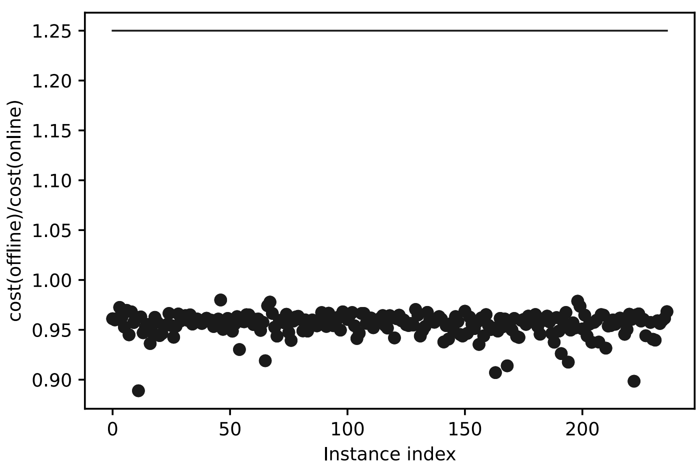


Figure 4.3: Ratio of the values of the offline solution of OPT and the online solution of avoid buying

We performed an experimental analysis on real data to investigate the ratio of the solution computed by avoid buying and the offline algorithm OPT as described in Section 4.3.1. A formal description of the data set and the simulation is provided in Section 4.5, where we do a more extensive experimental analysis. As expected the experiments have not shown differences with respect to the value of the capacity C of the battery, hence we just display the results obtained for $C = 20$.

In Figure 4.3 we take the ratio of the costs obtained by the two algorithms and we compare it to the worst case ratio obtained in our theoretical analysis (the continuous line at 1.25) we can notice that we are indeed far from the worst case scenario, meaning that in real life our strategy can on average be better of what we expected.

4.4 STRATEGIES WITH PARTIAL INFORMATION ON THE PRICES

We develop now a strategy which also includes prices in the decision process. Given the positive results of Theorem 4.8, we build on the strategy avoid buying. In particular, we ask ourselves whether it can be convenient to sell on some particular time periods. To answer this question we had to make the further assumption that prosumers are aware of the expected price for buying energy. Then, we develop the sell on very good days strategy where prosumers sell the amount εC of their stored energy if the price for selling on that time period p is higher than the price for buying, namely $s_t \geqslant \mathbb{E}[b]$. We give the pseudocode of the strategy in the Appendix and we show the following result.

Lemma 4.10. *Given the setting* $\mathcal{A} = \{soc_0, C, \mathbb{E}[b]\}$ *and* $\mathcal{B} = \{b_p, s_p, d_p, e_p\}$*, the strategy sell on very good days achieves a lower expected bill than the avoid buying strategy.*

Proof. We compare the bills of prosumers P_{AB} and P_S who respectively follow the avoid buying and sell on very good days strategies. We define the beginning of the phase i_P as each time periods when P_S sells. We have that

$$\begin{aligned}\text{bill}(\pi^{P_{AB}}) = \sum_{p=i_P}^{f_P} b_p x_p^{P_{AB}} - \sum_{t=i_P}^{f_P} s_p y_p^{P_{AB}} \\ - SOC_{f_P}^{P_{AB}} s_{f_P} + SOC_{i_P}^{P_{AB}} s_{i_P},\end{aligned} \tag{4.8}$$

where the values $x_p^{P_{AB}}$ and $y_p^{P_{AB}}$ are the amount of energy the prosumer P_{AB} might need to buy if the battery is empty and sell because of battery overflow. The bill of P_S is the same, plus the revenue obtained on the first day of the phase, namely $\varepsilon C s_{i_P}$. We now write the values $SOC_{f_P}^{P_{OS}}$ and $SOC_{f_P}^{P_S}$ as a function of the initial state of charges and the daily changes and we compute their difference:

$$\begin{aligned}& SOC_{f_P}^{P_S} - SOC_{f_P}^{P_{AB}} \\ &= SOC_{i_P}^{P_S} - \varepsilon C + (\sum X_p^+ - \sum y_p^{P_S}) - (\sum X_p^- - \sum x_p^{P_S}) \\ &- [SOC_{i_P}^{P_{AB}} + (\sum X_p^+ - \sum y_p^{P_{AB}}) - (\sum X_p^- - \sum x_p^{P_{AB}})] \\ &= SOC_{i_P}^{P_S} - \varepsilon C - \sum y_p^{P_S} \\ &- \sum x_p^{P_S} - SOC_{i_P}^{P_{AB}} + \sum y_p^{P_{AB}} - \sum x_p^{P_{AB}} \ .\end{aligned} \tag{4.9}$$

The difference of the bills in a phase is then:

$$
\begin{aligned}
&\mathrm{bill}(\pi^{P_{AB}}) - \mathrm{bill}(\pi^{P_S}) \\
&= s_{i_P}(\varepsilon C - SOC_{i_P}^{P_S} + SOC_{i_P}^{P_{AB}}) + s_{f_P}(-\varepsilon C + SOC_{i_P}^{P_S} - SOC_{i_P}^{P_{AB}}) \\
&+ \sum_{p=i_P}^{f_P} s_t y_p^{P_S} - \sum_{p=i_P}^{f_P} b_p x_p^{P_S} - s_{f_P} \sum y_p^{P_S} \\
&+ s_{f_P} \sum x_p^{P_S} - \sum s_p x_p^{P_{AB}} + \sum b_p x_p^{P_{AB}} \\
&+ s_{f_P} \sum y_p^{P_{AB}} - s_{f_P} \sum x_p^{P_{AB}} \\
&\geqslant (\mathbb{E}[b] - s_{f_P})(\varepsilon C - SOC_{i_P}^{P_S} + SOC_{i_P}^{P_{AB}}) + \sum_{p=i_P}^{f_P} s_t y_p^{P_S} \\
&- \sum_{p=i_P}^{f_P} b_t x_p^{P_S} - s_{f_P} \sum y_p^{P_S} + s_{f_P} \sum x_p^{P_S} \\
&- \sum s_p x_p^{P_{AB}} + \sum b_p x_p^{P_{AB}} \quad .
\end{aligned}
$$

Again, we work in expectation so we get

$$
\begin{aligned}
&\mathbb{E}\left[\mathrm{bill}(\pi^{P_{AB}})\right] - \mathbb{E}\left[\mathrm{bill}(\pi^{P_S})\right] \geqslant \\
&(\mathbb{E}[b] - \mathbb{E}[s])(\varepsilon C - SOC_{i_P}^{P_S} + SOC_{i_P}^{P_{AB}}) - \mathbb{E}[b] \sum \mathbb{E}\left[x_p^{P_S}\right] \\
&\mathbb{E}[s] \sum \mathbb{E}\left[x_p^{P_S}\right] + \mathbb{E}[b] \sum \mathbb{E}\left[x_p^{P_{AB}}\right] - \mathbb{E}[s] \sum \mathbb{E}\left[x_p^{P_{AB}}\right] \\
&= (\mathbb{E}[b] - \mathbb{E}[s])(\varepsilon C - SOC_{i_P}^{P_S} + SOC_{i_P}^{P_{AB}}) \\
&\mathbb{E}[b]\,(-\sum \mathbb{E}\left[x_p^{P_S}\right] + \sum \mathbb{E}\left[x_p^{P_{AB}}\right]) \\
&+ \mathbb{E}[s]\,(\sum \mathbb{E}\left[x_p^{P_S}\right] - \sum \mathbb{E}\left[x_p^{P_{AB}}\right]) \\
&= (\mathbb{E}[b] - \mathbb{E}[s])(\varepsilon C - SOC_{i_P}^{P_S} + SOC_{i_P}^{P_{AB}} - \sum \mathbb{E}\left[x_p^{P_S}\right] \\
&+ \sum \mathbb{E}\left[x_p^{P_{AB}}\right]) = (\mathbb{E}[b] - \mathbb{E}[s]) \cdot \\
&(\sum \mathbb{E}\left[y_p^{P_{AB}}\right] - \sum \mathbb{E}\left[y_p^{P_S}\right] + SOC_{f_P}^{P_{AB}} - SOC_{f_P}^{P_S}) \geqslant 0,
\end{aligned}
$$

now thanks to Equation 4.9 we can rewrite the term $\varepsilon C - SOC_{i_P}^{P_S} + SOC_{i_P}^{P_{AB}} - \sum \mathbb{E}\left[x_p^{P_S}\right] + \sum \mathbb{E}\left[x_p^{P_{AB}}\right]$ obtaining

$$
\begin{aligned}
&\mathbb{E}\left[\mathrm{bill}(\pi^{P_{AB}})\right] - \mathbb{E}\left[\mathrm{bill}(\pi^{P_S})\right] \geqslant \\
&(\mathbb{E}[b] - \mathbb{E}[s])(\sum \mathbb{E}\left[y_p^{P_{AB}}\right] - \sum \mathbb{E}\left[y_p^{P_S}\right] + SOC_{f_P}^{P_{AB}} - SOC_{f_P}^{P_S}) \geqslant 0 \quad .
\end{aligned}
$$

Finally the term $\sum \mathbb{E}\left[y_p^{P_{AB}}\right] - \sum \mathbb{E}\left[y_p^{P_S}\right] + SOC_{f_P}^{P_{AB}} - SOC_{f_P}^{P_S}$ is positive since by definition the state of charge of P_S cannot be higher than the one of P_{AB}. □

As in Theorem 4.9 we compare the stratgy sell on very good days with the offline algorithm, again by denoting with ALG the former and OPT the latter. We consider again the worst case scenario and we investigate the value of the parameters α and β such that

$$\alpha \mathbb{E}[\mathrm{bill}(\pi^{P_{OPT}})] + \beta \geqslant \mathbb{E}[\mathrm{bill}(\pi^{P_{ALG}})] \quad . \tag{4.10}$$

We remark that this choice is due to the fact that in the worst case analysis the bill of P_{OPT} could actually be negative in case the revenue obtained is particularly high. We know that in the worst case scenario described in Section 4.3.1 it holds

$$\mathbb{E}[\mathrm{bill}(\pi^{P_{OPT}})] \geqslant b_{\min} \sum_{p \in \tau} \mathbb{E}[b_p] - \frac{n}{2} C(s_{\max} - b_{\min}),$$

and we also have that $\mathbb{E}[\mathrm{bill}(\pi^{P_{OPT}})] \leqslant \mathbb{E}[\mathrm{bill}(\pi^{P_{AB}})]$ by definition and $\mathbb{E}[\mathrm{bill}(\pi^{P_{AB}})] \leqslant b_{max} \sum_{p \in \tau} \mathbb{E}[b_p]$ as seen already in Section 4.3.1. Hence, by substituting these into Equation 4.10, we then obtain

$$\begin{aligned}
&\alpha \mathbb{E}[\mathrm{bill}(\pi^{P_{OPT}})] + \beta \\
&\qquad \geqslant \alpha(b_{min} \sum_{p \in \tau} \mathbb{E}[b_p] - \frac{n}{2} C(s_{\max} - b_{\min})) + \beta \\
&\qquad \geqslant \alpha b_{\min} \sum_{p \in \tau} \mathbb{E}[b_p] - \alpha \frac{n}{2} C(s_{\max} - b_{\min}) + \beta \\
&\qquad \geqslant \alpha \frac{b_{\min}}{b_{\max}} \mathbb{E}[\pi^{P_{ALG}}] - \frac{\alpha n}{2} C(s_{\max} - b_{\min}) + \beta \ .
\end{aligned}$$

Hence, by setting $\alpha = \frac{b_{\max}}{b_{\min}}$ and $\beta = \frac{n}{2} \frac{C(s_{\max} - b_{\min}) b_{\max}}{b_{\min}}$, Equation 4.10 holds.

4.5 COMPARISON OF THE STRATEGIES IN A REAL SCENARIO

We analyze the performance of the strategies developed in this work in a real scenario. To do so, we used the open access data set of the UK Energy Research Centre (UKERC) available at [68], where the demand needed and the energy produced in 361 buildings in Bracknell, England were recorded every 30 minutes from October 2012 to March 2017.

From the nature of the problem, it seemed natural to us to compare it with some AI-related model. In this field, Reinforcement Learning (RL) models are particularly suitable, as we effectively have an agent who has to learn a policy and there is an instantaneous reward based on the current action taken. In case the reader is not familiar with these model, we take in the next subsection a brief moment to illustrate them.

4.5.1 *RL models*

In RL problems an agent is immersed in a given environment and has to solve a specific problem. They adopt a trial and error approach to explore the environment and understand what is a good strategy. In its nature, it is very common to lots of learning experiences in everyday life; for instance, it immediately comes to my mind a difficult bouldering problem, where one has to fall several times before understanding the best way, the so called Beta, to successfully climb the route. Going back to our agents, we still need to define how we can abstract this learning experience. In RLs this is done by modeling the environment as a Markov decision process, namely a 4-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}_A)$. In this definition $\mathcal{S}$ is the state space, so the set of all the possible states of the system, either discrete or continuous. The set $\mathcal{A}$ is the action space, namely all the actions which can be taken by the agent. The probability distribution $\mathcal{P}$ instead gives information regarding the result of an action. It is defined as $\mathcal{P}_A(S, S')$ and it expresses the probability of ending up in state S' performing action A in state S. The last ingredient of the model is the immediate reward $\mathcal{R}_A$, which quantifies the reward obtained by performing action A. The goal of RL models is to find a policy π, namely a function that indicates which actions are better to perform in a given state S, formally

$$\pi : \mathcal{S} \times \mathcal{A} \to [0, 1],$$

and in particular it aims at selecting one which maximizes the expected cumulative reward, defined as the sum of all the rewards discounted by a factor γ, in formulas $\sum_{p=1}^{n} \gamma^p \mathcal{R}_p$.

In this work we adopt a Q-learning reinforcement algorithm to compute the policy. Being from the family of RL models, the main framework is as described above. Q-learning assigns a value to the pair (S, A) which is an estimate of the revenue obtainable by performing action A from state S. All these values are collected in a table, the Q table, whose entries are constantly updated during the training phase. Then, the policy is simply computed by always selecting the action which maximizes the expected revenue starting from the current state. The updates of the entries of the matrix are performed following the Bellman equation. In particular, the following expression is derived from it:

$$Q^{p+1}(S_p, A_p) = (1-\alpha)Q^p(S_p, A_p) + \alpha\left(R_{p+1} + \gamma \max_A Q(S_{p+1}, A)\right),$$

which is a convex combination in the learning rate α of the current value $Q^t(S_t, A_t)$ and the newly observed reward. The second term $R_{t+1} + \gamma \max_A Q(S_{p+1}, A)$ is also given by two contributions, namely the immediate reward obtained by performing action A_p and the discounted expected future value obtainable from the next state. To ensure the convergence of the training of the matrices we tuned the learning rate parameter as $\alpha_p = 1/\sqrt{p+2}$. Furthermore, during the training, we have followed what is called an ε *greedy approach*. According to this, given a state S, the action A_p to perform is chosen in most of the cases as the current best expected action. With probability ε it is instead chosen randomly. This typically ensures that all the states of the Q matrix are explored and reduces the possibility of being trapped into local optima. The so called exploration is more frequent at the beginning of the training phase and it becomes almost null towards the end; this can be achieved by setting the parameter $\varepsilon_p = 1/\log(p+2)$. In our experiments we have implemented three different Q-learning methods, by changing the space of states and actions as we illustrate in the following subsection.

4.5.2 *Our approach*

We start by an easy version of Q-learning, called Q-easy, with action space $\mathcal{S} = \{0, \ldots, C\}$ and three possible actions, which we call $\mathcal{A} = \{\mathtt{buy}, \mathtt{sell}, \mathtt{usebattery}\}$. This method is easy to implement. It restricts the behavior of the agent by limiting the action space: for instance, it does not allow to buy more energy in a given time or to partially use the battery to fulfill a portion of the net demand and still interact with the market. To overcome this restriction, we implemented a second method, called Q-constant, where the $\mathcal{S}$ is the same, but we change the action space. In particular, for each state S we perform the action $A_{\rightarrow S'}$ = *transition to* S', so that the algorithm is not constrained in any of the chances. The algorithm is not limiting in any way the behavior of the agent. Its first drawback is that action $A_{\rightarrow S'}$ is not easy to interpret: to understand what the prosumer really did to go from state S to the transitioning state S' we need to know other quantities such as the demand and the energy on that day and then perform some computation. The second drawback is that this Q-matrix results in a one dimensional vector, meaning that the policy is constant and the prosumer always aims at keeping a constant level of battery, no matter the initial state nor the energy profile. We still implemented this strategy given the similarities with our dummy strategy, to which we will compare it indeed later. Lastly, we have developed the more sophisticated Q-complete strategy, for which the state space $\mathcal{S}$ as been codified as follows. At every time period p, the prosumer has a given demand d_p to satisfy and is able to produce the quantity e_p. We want these two quantities also to be taken into account when taking a decision, therefore we include their difference into the state space by taking the Cartesian product of the possible amount of energy in the battery and the difference $d_p - e_p$. However, we are only interested in the values of $d_p - e_p$ which are in the range $[-C, C]$, since other values would not influence the decision process of the algorithm. We take hence as state space $\mathcal{S} = \{0, \ldots, C\} \times \{-C, \ldots, C\}$. Given the more precise characterization of the states, this last method relies on a more precise underlying Markov Process. However, the matrix Q is now $(C+1)(2C+1)^2$ and we had to manually prohibit the access to some actions and state to guarantee the convergence of the method.

We can then compute the revenue by first computing the quantity $\Delta_p = d_p - e_p + S_p - S'_p$ and then obtaining the revenue as

$$R_p = -s_p \min(0, \Delta_p) + b_p \max(0, \Delta_p) \ . \tag{4.11}$$

4.5.3 *Results*

We completed the set of parameters e_p and d_p by simulating the prices. The prices for buying energy b have been randomly uniformly generated in the interval $[0.8, 1]$. The price for selling energy at time s_p is then computed by subtracting to the corresponding b_p a randomly uniformly distributed amount in interval $[0.1, 0.2]$. We have run the experiments for several values of the battery capacity C but since we have not noticed particular differences between the values we just report the results obtained for $C = 20$. We implemented the three methods and trained the respective Q-matrices. In this section we show the results obtained when training the matrices on 40% of the dataset, since these were achieving the best performance for what concerns the convergence of the matrices. The remaining entries of the dataset have been then used for the comparison of the strategies as follows. We have computed the schedule of the prosumers according to the policies found by the RL models as well as with our strategies avoid buying, sell on very good days, and dummy strategy. In particular, for each strategy π we have computed the quantity $\mathtt{bill}(\pi)$ in each instance. In our experiments were have recorded no differences among the strategies avoid buying and sell on very good days hence we report in the analysis only the results of the former. Let us start with the comparison of the strategies of RLs and avoid buying.

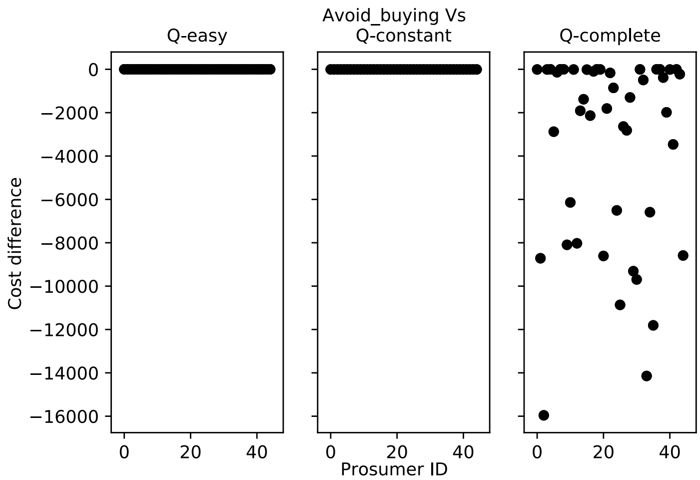


Figure 4.4: Comparison of avoid buying and our Q-learning policies

The quantities depicted in Figure 4.4 are $\mathtt{bill}(\pi^{P_{AB}}) - \mathtt{bill}(\pi^{P_Q})$, where the second term is–from left to right–Q-easy, Q-constantand Q-complete. Since here we are considering the bills, we would like to see observe that that the quantities $\mathtt{bill}(\pi^{P_{AB}}) - \mathtt{bill}(\pi^{P_Q})$ are not positive, and this is exactly what the plot shows.

Furthemore, we can observe that Q-easy and Q-constant show few points, hence few instances, where the costs are significantly larger, while Q-complete seems to incur much more in this problem. According to our investigation, the instances with a big difference are reporting energy profiles of prosumers who can produce a significant amount of energy. In some implementations, we have observed that the Q-matrix of Q-complete might push the prosumers to buy more energy of what they actually need and store it for later days, which is somewhat surprising to us, since the Q-learning has no information regarding the price, but it could also be to a too little exploration of the state space given the huge size of $\mathcal{S}$. In support of this thesis, there is also the fact that Q-easy nor Q-constant show this behavior.

One might think that the instances where Q-complete was exhibiting a poor performance are just outliers. As a last note we remove now

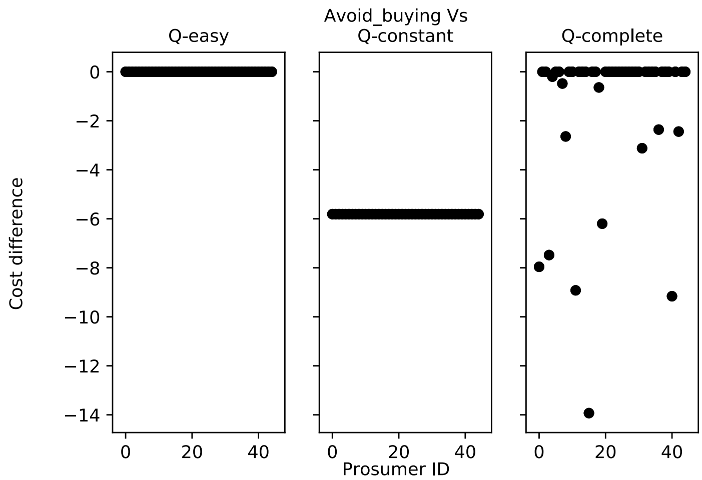


Figure 4.5: Avoid buying and our Q-learning policies without outliers

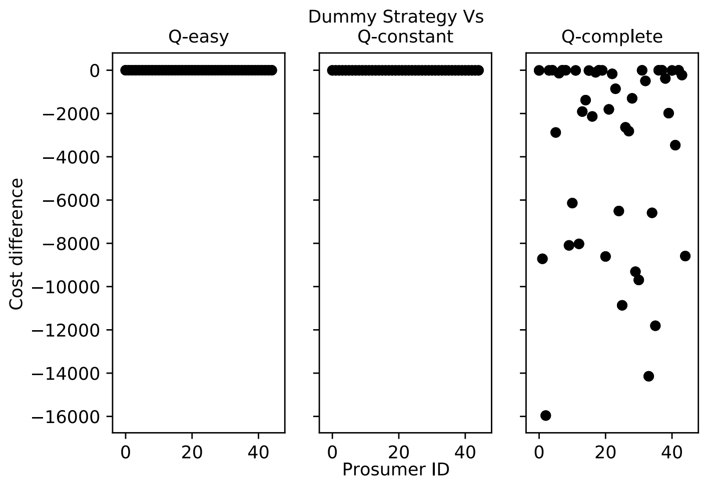


Figure 4.6: Comparison of dummy strategy and our Q-learning policies

these instances where there is a lot of energy produced, to show to the reader that the other results are comparable.

From Figure 4.5, it is clear that a more sophisticated Q-learning such as Q-complete performs way actually better than the other two, once the other instances are removed. We can conclude that the risky behavior of the strategy $\pi_{Q-\mathtt{complete}}$ can lead to bad outcomes, as it is the case for the instances that we removed in Figure 4.5 but otherwise it produces good results. Lastly, Q-constant shows also from experimental simulations that it has the same results of dummy strategy and the same holds for Q-easy as showed in figure Figure 4.6, where the value of the differences $\mathtt{bill}(\pi^{P_D}) - \mathtt{bill}(\pi^{P_Q})$ are represented.

We would like to remark that on top of the good experimental results there are two other main factors to be considered. The first is that our strategies do not need training and can be directly adopted in any new setting. The second is that in our strategies the decision taken at time period p do not depend on the value of the battery capacity. The same cannot be stated for the RL models, at least the ones used here: when the capacity of the battery changes, this affect also the dimension of the state space $\mathcal{S}$ and hence the dimension of the Q-matrix itself, so that the strategy for a new prosumer with a different battery is not straightforwards.

From this experimental analysis, we can conclude the chapters by observing that our strategies, in particular the strategy avoid buying, have a thorough theoretical analysis, are easy to be interpreted and perform well even when compared to the some more sophisticated RL models.

5 PROSUMER ONLINE PROBLEM: OPTIMIZING THE ENERGY ACQUISITION

In the previous chapter, we introduced several online strategies for our prosumers. For some of these strategies, we are able to make a comparison with the offline setting. This was particularly the case for the avoid buying strategy, in which the prosumer reasonably uses the energy stored in the battery to fulfill their demand as much as possible. In this chapter, we return to this analysis and further investigate the findings discussed in Section 4.3.1. As we have seen, when a prosumer attempts to use their battery to fulfill the demand rather than purchasing from the main grid,, the total amount of energy purchased is actually the same as in the offline setting. The key difference that we noticed in the analysis regards the costs associated to the amount of energy bought. According to our strategy, the prosumer waits until the battery is depleted and purchases energy only when necessary. In an offline setting instead the prosumer might buy earlier, if prices are more convenient. At this point, a natural question arose during our research discussions. Suppose the prosumer has a prediction of their demand and energy produced over the time horizon. If this is the case, then the function $SOC(p)$ defined in the previous chapter in Equation 4.5 as the $\max\{C, \min\{0, SOC(p-1) + e_p - d_p\}\}$ becomes fully deterministic, giving the prosumer the knowledge of the time periods when the battery becomes empty and it is necessary to buy energy. In other words, the set Γ defined in Equation 4.6 would be known. In this case, the prosumer is aware that they will need to buy energy at a certain point and how much energy they will need. In such a case, it might be advantageous for the prosumer to purchase energy in advance, rather than waiting until the exact moment it is needed.

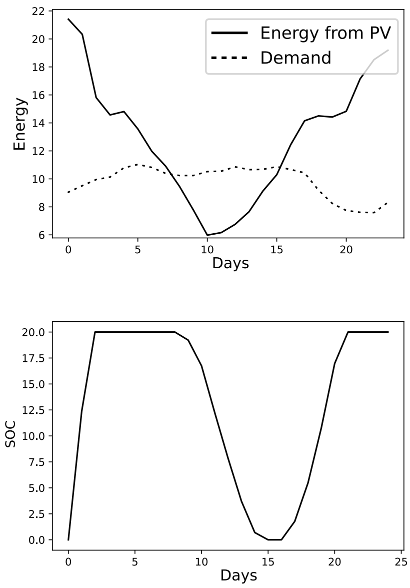
We have asked ourselves whether there is a good strategy to find the time for buying and we now formalize and show our analysis in this chapter.

This chapter is based on joint work with Matthias Mnich and Karolina Tammemaa.

5.1 PROBLEM FORMALIZATION

We formalize now the problem. Consider a prosumer who starts with a full battery and who is aware that they will run out of energy stored in the battery at a certain time point n. Such a time n it is known beforehand since we are assuming that the prosumer knows ahead of time all their demands and their energy produced. Because of the same assumption, they not only know the number K of units of energy needed at time n, but they also know how much space is free in the battery at every time point. Furthermore, the amount of

Figure 5.1: Example of the energy profile of a prosumer from Section 3.4.2: In the top panel the energy produced e_p and the demand d_p are displayed. From these quantities the function $SOC(p)$ is computed and portrayed in the panel below. From the second panel we see that the battery of the prosumer will be empty towards time 16. Suppose that it is not only empty, but they need one unit of energy: this unit could have been bought already from time 9 since the battery had space to store it. When would have been convenient then to buy?



free space left in the battery increases over time in K different time periods. Prosumers can then start from time 0 to purchase some of the K needed units and store them for the future. At each time period, the price for buying at that given time is revealed and the maximum number of units which can be bought is upper bounded by the space left in the battery. We ask ourselves whether at time p, knowning the price of energy at p and the space left in the capacity, it might be convenient for the prosumer to buy one or more of the K units of energy that they will need at time n.

We call this problem ONLINE ENERGY ACQUISITION WITH INCREMENTAL CAPACITY and we provide an analysis of it in this chapter. We remark that this problem is closely related to other optimal stopping problems, such as the Secretary Problem. This is a well-studied problem in the field of optimal stopping algorithms which dates back to the early 1950's and 1960's and it is first formalized in the work of Ferguson [40]. In the classic variant, a manager wants to hire the best secretary among n candidates. The hiring procedure is as follows: after interviewing each candidate, the manager has to decide immediately whether to hire or not the given candidate and they have no chances of recalling rejecting candidates. Hence, the decision of accepting or rejecting a candidate is only based on past information. For $K = 1$ we can even give an interpretation of it in terms of the secretary problem as follows. Consider a manager who has two options for the hiring process. Either they hire two secretaries with a part time contract of

50% or one single secretary with 100% working hours. As in the classic variants, they can interview only one secretary per time unit. Furthermore, there are funding constraints, such that until a known time $p^\star$, the manager can only hire a secretary with 50% position. After this time, they have further funding, either to hire a second secretary with part-time - if a first part-time secretary has already been hired - or to hire a secretary full time. Hence, we can encapsulate our online energy acquisition with incremental capacity as an extension of the secretary problem.

As already mentioned, we assume that the prosumer's demand for the energy is known over a time horizon and we try to provide strategies for buying energy in advance at a lower price than buying the needed energy at the last possible moment n. For sake of simplicity we assume that our prosumer is only consuming and not producing, which is equivalent to the prosumer always consuming more than producing. Since the prosumers considered in this chapter are exclusively buying energy, here we refer to them as *agents*.

5.2 RELATED LITERATURE

The problem considered in this chapter fits into the broader field of allocation resources. An example of it is given by Thielen, Tiedemann, and Westphal [82], where an online knapsack problem with incremental capacity is considered. In the same field is the work on Knapsack Secretary Problem by Babaioff et al. [6] to be counted. This is also contributing directly to the more standard extensions of the secretary problem, as they develop two algorithm with an asympthotical competitive ratio of e for the k secretary problem. In some sense related to our problem is also the work of Fiat et al. [41]. Here indeed the selection of the k secretaries has the additional constraint that the manager has to wait at least a given period of time after selecting the next candidate, so in some sense if γ is the amount of time the manager has to wait, selecting the candidate i implies rejecting the next γ.

Lastly we mention that there are variation of the secretary problem where some extra information is provided, as in the work of Gilbert and Mosteller [44] where the additional hypothesis of the candidates being independently identically distributed. Campbell and Samuels [22] extend this concept by adding some prior information to the problem in the form of a training sample of size m. Based on the size of the sample set with respect to n they derive different results which space from the limit $e^{-1} \simeq 0.37$ for the no information case to approximately 0.58 for the full information case. In the work of Correa et al. [30] the same result is achieved with another analysis.

5.3 PRELIMINARIES

An agent needs to plan when to buy energy over a finite discrete time horizon $\tau = \{1, .., n\}$, where $n \in \mathbb{N} = \{1, 2, \dots\}$. The agent has some energy stored in a battery of finite capacity and they are aware that they will run out of stored energy at time n. Hence, to completely fulfill the demand on that day, they need to buy an extra $\beta \in \mathbb{N}$ units of energy over the given time horizon.

We assume, that at any time $p \in \tau$ agent can buy any amount of energy from the main grid with fixed price of b_{MAX} per unit of energy, but agent can only store limited number of units for the future, other units need to be used during the time p. Therefore the β units can be bought at time n from the grid at cost $\beta \cdot b_{MAX}$. The agent could also buy some units ahead of time n from independent producers in the microgrid and store the energy in the battery without any loss. We assume that the amount of energy the agent purchases is small compared to what is offered and thus when energy is offered in the microgrid with cost c then the agent can buy all desired units with this price i.e. we do not assume the price to increase when we buy more than one unit of energy from the grid. Due to the capacity of the battery, the number of units that such an agent can buy on a given day are restricted. The price for buying energy from other independent producers fluctuates over time, thus the agent is interested in strategically purchasing energy to minimize costs.

We can see this as a maximization problem, where we are interested in maximizing our savings over the values $v_p = b_{MAX} - b'_p$, where b'_p is the buying price of a unit of energy in day p (note that $b'_p \leqslant b_{MAX}$ as one can always buy energy from the grid with fixed price b_{MAX}). The vector $\mathbf{v} = (v_1, \ldots, v_n)$ of *values* is not known apriori to the agent: only at time p the p-th entry of the vector is revealed. Now that we are considering a maximization problem, we are asking ourselves how can the agent select the periods with higher values. Intuitively the values v_p represent how much a prosumer is saving buying at time p rather than at the last time period n. We make no assumptions about the values in $\mathbf{v}$ other than that all values are different and each permutation of the values of $\mathbf{v}$ is equally likely to occur. The assumption on each permutation of the values being equally likely extends naturally to the rank, meaning that all possible $n!$ order are equally likely. Thus, our analysis holds if instead of the value v_p, only its rank $\mathrm{rank}(v_p)$ in the set $\{v_1, \ldots, v_p\}$ is revealed as in the classic secretary problem. However, in our analysis we use the value as it makes sense in the context of the problem, as one would always be able to see the energy prices.

We model the free space in the battery at a given time, assuming that the agent does not buy energy over the time horizon. We assume that there are tasks that prompt the agent to use some energy from the battery, thus decreasing the fullness of the battery and increasing the available free capacity. For simplicity we assume that all these tasks are independent of each other, have equal probability of happening in any of the n days and they all require 1 unit of energy. We assume that an agent knows at what time these tasks will take place already at the beginning of the time horizon τ. We model the free capacity function $r(p)$ in the battery as follows:

Definition 5.1. *Let $T_1, \dots, T_K$ be independent identically distributed (iid) uniform random variables on the discrete interval τ, which represent the time periods when the tasks occur. Let $Y_i(p) = 1$ if $p \geqslant T_i$ and 0 otherwise. We call $R_K(p) = 1 + \sum_{i=1}^{K} Y_i(p)$* ***remaining capacity sample space*** *or* ***free capacity sample space****.*

A realization $T_1 = p_1, \dots, T_K = p_K$ of the times of the tasks defines functions $y_i(p)$ for each $i \in \{1, \dots K\}$, where $y_i(p) = 1$ if $p \geqslant p_i$ and 0 otherwise. It also defines an instance of ***remaining space function*** *or* ***free capacity function****: $r_K(p) = 1 + \sum_{i=1}^{K} y_i(p)$.*

With definition 5.1, we are formally linking the space available in the battery at time p, so the value $r_K(p)$, with the defined tasks that have already occured. The intuition behind is that if already m tasks have occured at time p, then $r_K(p) = 1 + \sum_{i=1}^{K} y_i(p) = m + 1$, so that the prosumer can buy up to $m+1$ units of energy and store them in the battery. From the definition, we can see that $r_K(p)$ is monotonically increasing step-function with at most $K \in \mathbb{N}$ steps, which we also call increments, and it always has $r_K(n) = K + 1$. The $+1$ in the definition of $R_K(p)$ and $r_K(p)$ is added to ensure that the free capacity function is always at least 1.

For the rest of this work we assume that $\beta = K + 1$. We choose this, as this value captures the hardest part of the problem and it allows us to have fewer variables. For $\beta > K + 1$ we can not optimize for the units $\beta - (K + 1)$ as these are determined to be bought at time n. While for $\beta < K + 1$ the problem becomes easier because when purchasing energy we now have to potentially make fewer choices.

We formally define our problem as follows:

ONLINE ENERGY ACQUISITION PROBLEM WITH INCREMENTAL CAPACITY

In: An instance $\mathcal{I}$ of online energy acquisition with incremental capacity is defined with the tuple $(r_K(p), \mathbf{v})$ where $\mathbf{v}$ is revealed in an online fashion.

Out: A solution to the problem is a vector $x \in \mathbb{N}^n$ such that $\sum_{i=1}^{p} x_i \leqslant r_K(p)$ for every $p \in \tau$ and $\sum_{i=1}^{n} x_i = K + 1$. A solution x is optimal if for every other feasible solution $\bar{x}$, it holds $\bar{x} \cdot \mathbf{v}^T \leqslant x \cdot \mathbf{v}^T$.

We denote with x_{ALG} the solution returned by an algorithm ALG and we denote with V_{ALG} the value $x_{ALG} \cdot \mathbf{v}^T$. Similarly we denote with x_{OPT} the solution returned by the optimal offline algorithm OPT and with V_{OPT} the value $x_{OPT} \cdot \mathbf{v}^T$.

Classically, the competitive ratio for online problems is defined as $\mathbb{E}[V_{ALG}] \geqslant c \cdot V_{OPT}$ where c is the competitive ratio and $\mathbb{E}[V_{ALG}]$ is taken over the uniform distribution of all $n!$ input sequences. Importantly, in our online energy acquisition problem, the offline solution is not order independent as is the case in secretary problem and most of its variants. In other words, given a free capacity function $r_K(p)$ and a vector of values $\mathbf{v}$ and a permutation $\sigma(\mathbf{v})$ it is possible that $\mathrm{OPT}(r_K(p), \mathbf{v}) \neq \mathrm{OPT}(r_K(p), \sigma(\mathbf{v}))$:

Example 5.1. *Consider the following instance* $\mathcal{I} = (r_1(p), (1, 2, 1000, 3))$ *where* $r_1(p) = \begin{cases} 1 & \text{if } p < 2, \\ 2 & \text{if } p \geqslant 2 \end{cases}.$
If we consider the offline solution, the agent waits for the time when the value is highest and buys directly two units of energy. The solution vector is then $x_{OPT} = (0, 0, 2, 0)$ *and the solution has value* 2000.

For the permuted input value vector $\mathcal{I}' = (r_1(p), (1000, 2, 1, 3))$*, the agent still selects the time corresponding to the value 1000, but this time only one unit can be bought for such a value. The second unit has to be bought later. The solution is then* $x_{OPT'} = (1, 0, 0, 1)$ *with a value of 1003, which is different from the one obtained before.*

As we lack a unique optimal solution that would stay the same through the permutation of the input, we use an expected competitive ratio, like covered for example in Ezra et al. [38]. While giving the definition of the expected competitive ratio we use $\mathbb{R}_+ = \{r \in \mathbb{R} \mid r \geqslant 0\}$.

Definition 5.2. *The expected competitive ratio for an instance of a free capacity functions* $r_K(p)$ *is* d *if:*

$$\max_{v \in \mathbb{R}_+^n} \left(\mathbb{E}_{\sigma(v)} \left[\frac{V_{ALG}}{V_{OPT}} \right] \right) \geqslant d,$$

where the expected value is taken over all the permutations σ *of* v

The expected competitive ratio tells us what proportion of the optimal solution we expect to get. In Section 5.5 we are going to show the expected competitive ratio for instances $\bar{r}_K(p) \in R_K$ of our problem, where the increments p_i are at positions $1 + i \cdot \frac{n-1}{K+1} \simeq i\frac{n}{k}$. Function $\bar{r}_K(p)$ is in a sense the expected shape of the function $r_K(p)$ and also corresponds to real life scenarios where we need energy from the battery periodically (for instance when we need to cool down the refrigerator every hour and we are part of a microgrid where offers appear every 10 minutes).

5.4 ALGORITHMS FOR MAXIMIZING THE PROBABILITY OF SELECTING THE OPTIMAL SOLUTION

In an online setting it is natural to ask what can be said about the probability of finding the optimal solution; this was for example studied in the secretary problem. It is also natural to propose algorithms that try to maximize this probability. In this section we develop algorithms for maximizing an agent's probability of selecting the optimal solution for an instance of an online energy acquisition problem. We first develop a GENERAL-ONE-STEP-ALGORITHM for instances $\mathcal{I} = (r_1(p), \mathbf{v})$ and then show how we can use it as a subroutine in GENERAL-K-STEP-ALGORITHM for instances $\mathcal{I} = (r_K(p), \mathbf{v})$. We also give some non-trivial upper bounds for the probability of choosing the maximal solution.

5.4.1 *Algorithms for* $K = 1$

In this section we study the problem for $r_1(p)$, which is defined as

$$r_1(p) = \begin{cases} 1 & \text{if } p < p^\star, \\ 2 & \text{if } p \geqslant p^\star, \end{cases} \tag{5.1}$$

where $p^\star \in \tau$.

By the assumptions made in the beginning of this section, we know that we need to purchase 2 units of energy by time n.

Remark 5.3. *The free capacity function defined in this section has what might seem a very limiting image, since* $r_1(p) \in \{1, 2\}$*. However all the reasoning of this section including the results of Theorem 5.7 and Theorem 5.8 also hold when random variables* $Y_i(p)$ *in the definition of free capacity sample space are defined as:* $Y_i(p) = c_i$ *if* $p \geqslant T_i$ *and* 0 *otherwise for each* $i \in \{1, ..., K\}$*, where* $b_i \in \mathbb{N}$*. This generalization is possible thanks to the fact that the strategies developed in this section only depend on the values* v *and the fact that the free capacity function has only one increment. Suppose we have a more general function* $r_1(p)$ *which takes value* $m \in \mathbb{N}$ *if* $p \leqslant p^\star$ *and* $m + l$ *if* $p \geqslant p^\star$*, with* $l \in \mathbb{N}$*. The we can solve the problem for* $m = l = 1$*, obtain the optimal time(s)* p *to buy energy and buy exactly all the free space capacity available in that moment. Thus the challenge of this problem is finding the right times, which does not change when we let the size of increments vary. For sake of simplicity, the algorithms are easier to be explained when we talk about 1 or 2 units, hence our definition.*

We develop different strategies for this problem and based on the value of $p^\star$ we use the strategy that gives the highest probability of choosing x_{OPT}. The first result we show is based on the classic secretary problem.

Lemma 5.4. *Given an instance* $\mathcal{I}$ *of the online energy acquisition with incremental capacity with* $K = 1$*, if* $p^\star \leqslant n/e$*, then by using classic secretary problem algorithm, we have a probability* $1/e$ *of selecting the optimal solution.*

Proof. We sample the first $\lfloor n/e \rfloor$ values, without picking any of them. Then we select the first value higher than any value in the sample and we buy all available units which in this case is 2 as $p^\star \leqslant \lfloor n/e \rfloor$. As each permutation of the values in **v** is equally likely to occur, we get probability $1/e$ of selecting the optimal solutions as in the secretary problem. □

Recall that the classical $1/e$ probability in the secretary problem is obtained by maximizing the following expression:

$$\frac{1}{n} \sum_{i=h+1}^{n} \frac{h}{i-1} \ ,$$

which is the probbility of selecting the best candidate. The term $\frac{1}{n}$ represents the probability that a given position i contains the best candidate overall, and $\frac{h}{i-1}$, corresponds to the probability that the best candidate up to position $i-1$ was observed during the sampling phase. This holds for every day i from the end of the samples h to n. This equation is maximized for $h = n/e$, which asymptotically leads to a probability of success of $1/e$.

For the next two algorithms we need to define the events A, B_1 and B_2. We denote by A the event of ALG choosing same solution as OPT. As already mentioned in Section Section 5.3, OPT is not invariant to permutations: therefore OPT either selects one unit of energy in time interval $[1, p^\star - 1]$ and one unit of energy in time interval $[p^\star, n]$ or it directly selects two units of energy in the time interval $[p^\star, n]$. We refer to these two events as B_1 and B_2, respectively. Thus the events B_1 and B_2 form a partition of A and by the law of total probability we get $P(A) = P(A|B_1)P(B_1) + P(A|B_2)P(B_2)$.

For the rest of this section we assume that $p^\star > n/e$. We present a SAMPLE-AND-CHOOSE-ALGORITHM that works by waiting a time h without buying anything (Sampling) and after this phase, when the value seen in the online stream is greater than anything seen so far we buy as many units as possible at that time (Choosing). We give the algorithm here for the general $r_K(p)$ as we will also use it later, but for the case of $r_1(p)$ the algorithm simplifies to just sampling the interval $[1, p^\star - 1]$ and then buying all the required units at the first moment, when the value seen in the online stream is higher than anything seen so far.

Algorithm 5.1 SAMPLE-AND-CHOOSE-ALGORITHM

Input : $\mathcal{I} = (r_K(p), \mathbf{v})$, a sample size $h \in \tau$.
Output : Amount of energy bought at time $p \in \tau$.
$x = (0, \ldots, 0)$
$M = \max\{v_1, \ldots, v_h\}$
At the first time $p > h$ with $v_p > M$ buy $r_K(p)$ units, set $x_p = r_K(p)$
Buy the remaining units at day n, setting $x_n = K + 1 - \sum_{p=1}^{n} x_p$
return $(x_1, \ldots, x_n)$

Lemma 5.5. *The asymptotic probability of* SAMPLE-AND-CHOOSE-ALGORITHM *selecting the optimal solution is* $\frac{p^\star}{n} \log \frac{n}{p^\star}$ *when* $K = 1$ *and* $h = p^\star - 1$.

Proof. We sample the interval $[1, p^\star - 1]$ and pick the first value, starting from $p^\star$ which exceeds all the values seen so far. In the event B_1 we have zero probability of finding the optimal solution. In the event B_2 the global optimum is in the interval $[p^\star, n]$ and $P(B_2) = (n - p^\star + 1)/n$. In event B_2 each day $i \in [p^\star, n]$ has equal probability $1/(n - p^\star + 1)$ of being optimal, and the probability that we chose it is equal to the probability that we did not choose anything before, which is $\frac{p^\star-1}{i-1}$ Therefore,

$$\mathbb{P}(A) = \mathbb{P}(B_2)\cdot\mathbb{P}(A|B_2) \;=\; \frac{n-p^\star+1}{n}\cdot\frac{1}{n-p^\star+1}\sum_{i=p^\star}^{n}\frac{p^\star-1}{i-1} \tag{5.2}$$

$$= \frac{p^\star-1}{n}\sum_{i=p^\star-1}^{n-1}\frac{1}{i} \simeq \frac{p^\star}{n}(\log(n-1)-\log(p^\star-1)) \simeq \frac{p^\star}{n}\log\frac{n}{p^\star}\;. \quad \square$$

When $p^\star$ approaches n then $\frac{p^\star}{n}\log\frac{n}{p^\star} \to 0$, but we would like to have an algorithm which also for $p^\star \to n$ has some positive probability of finding the optimum. For that, we develop the algorithm Two-Secretary-In-Row where the name is due to the fact that we apply the classic algorithm of the secretary problem twice, as we now explain. More specifically, we observe the $h = p^\star(e^{\frac{n}{p^\star}-2})$ first values and choose the first element in interval $[h+1, p^\star-1]$ that is greater than any element in the sample $[1, h]$. Then, independently of weather something was chosen or not in the interval $[1, p^\star-1]$, we use a classic secretary problem in interval $[p^\star, n]$. We present the pseudocode for Two-Secretary-In-Row in Algorithm Algorithm 5.2.

Algorithm 5.2 Two-Secretary-In-Row

Input : $\mathcal{I} = (r_1(p), \mathbf{v})$ where $p^\star/n > 1/2$.
Output : Amount of energy bought at time $p \in \tau$.
$x = (0, \dots, 0)$ a n dimentional empty vector
$h = (p^\star-2)(e^{\frac{n}{p^\star}-2})$
Apply the secretary problem algorithm in the interval $[1, p^\star-1]$ with sample $[1, h]$ and buy one unit if a day p is selected: $x_p = 1$
Apply the secretary problem algorithm in the interval $[p^\star, n]$ and buy the remaining unit(s) on the selected day or at the last day if no units were bought before: $x_n = 2 - \sum_{p=1}^{n} x_p$
return $(x_1, \dots, x_n)$

Lemma 5.6. *For $p^\star/n \geqslant \frac{1}{2}$, the asymptotic probability of* Two-Secretary-In-Row *selecting the optimal solution is $\frac{p^\star}{en}e^{\frac{n}{p^\star}-2}$.*

Proof. We show that the h value used in Two-Secretary-In-Row is optimal, by analyzing the value $\mathbb{P}(A)$:

$$\mathbb{P}(A) = \frac{p^\star-1}{n}\mathbb{P}(A|B_1) + \frac{n-(p^\star-1)}{n}\mathbb{P}(A|B_2)\;. \tag{5.3}$$

We get

$$\mathbb{P}(A|B_1) = \left(\frac{1}{p^\star-1}\sum_{i=h+1}^{p^\star-1}\frac{h}{i-1}\right)\cdot\frac{1}{e} = \frac{h}{e(p^\star-1)}\sum_{i=h+1}^{p^\star-1}\frac{1}{i-1},$$

where the first terms expresses probability of choosing the optimum in interval $[1, p^\star-1]$ and the factor $1/e$ express the probability of choosing an optimal element in the interval $[p^\star, n]$.

For the event B_2 we get:

$$\mathbb{P}(A|B_2) = \frac{h}{(p^\star - 1)} \cdot \frac{1}{e} = \frac{h}{e(p^\star - 1)},$$

where the term $\frac{h}{p^\star - 1}$ is the probability that the algorithm did not choose anything in the first $p^\star$ days, and $1/e$ is the probability that the algorithm chooses optimally in interval $[p^\star, n]$.

Substituting the found values into ((5.3)), we get:

$$\begin{aligned}\mathbb{P}(A) &= \frac{h}{en} \sum_{i=h+1}^{p^\star - 1} \frac{1}{i-1} + \frac{h(n - p^\star + 1)}{en(p^\star - 1)} \\ &\simeq \frac{h}{en}(\log p^\star - \log h) + \frac{h(n - p^\star)}{enp^\star},\end{aligned}$$

which we want to maximize, by choosing an appropriate h. For that we take a derivative by h and solve

$$\mathbb{P}'(A) = \frac{1}{en}(\log p^\star - \log h) - \frac{1}{en} + \frac{n - p^\star}{enp^\star} = 0 \ .$$

From there we get $h = p^\star(e^{\frac{n}{p^\star} - 2})$, which is well defined for values of $p^\star \geqslant n/2$. By substituting in the optimal value for h, we get the corresponding probability:

$$\begin{aligned}\mathbb{P}(A) &\simeq \frac{h}{en}\left(\log(p^\star) - \log(p^\star(e^{\frac{n}{p^\star} - 2})) + \frac{n - p^\star}{p^\star}\right) \\ &= \frac{h}{en}\left(-\log(e^{\frac{n}{p^\star - 1} - 2}) + \frac{n - p^\star}{p^\star}\right) \\ &= \frac{h}{en}\left(-\frac{n}{p^\star - 1} + 2 + \frac{n - p^\star}{p^\star}\right) \simeq \frac{p^\star}{en} e^{\frac{n}{p^\star} - 2} \ . \quad \square\end{aligned}$$

We note that when $p^\star \to n$ then $\frac{p^\star}{en} e^{\frac{n}{p^\star} - 2} \to e^{-2}$.

By choosing the best intervals for each algorithm defined above, we present GENERAL-ONE-STEP-ALGORITHM which covers all values of $p^\star/n$. One can see the probability guaranteed by the algorithm for different values of $p^\star/n$ in Figure 5.2.

Algorithm 5.3 GENERAL-ONE-STEP-ALGORITHM

Input : $\mathcal{I} = (r_1(p), \mathbf{v})$.
Output : Amount of energy bought at time $p \in \tau$.
if $p^\star/n \leqslant 1/e$ **then**
  Set $h = 1/e$ and let $(x_1, \ldots, x_n) =$ SAMPLE-AND-CHOOSE-ALGORITHM
if $1/e < p^\star/n \leqslant 0.8511$ **then**
  Set $h = p^\star - 1$ and let $(x_1, \ldots, x_n) =$ SAMPLE-AND-CHOOSE-ALGORITHM
if $p^\star/n > 0.8511$ **then**
  $(x_1, \ldots, x_n) =$ TWO-SECRETARY-IN-ROW
return $(x_1, \ldots, x_n)$

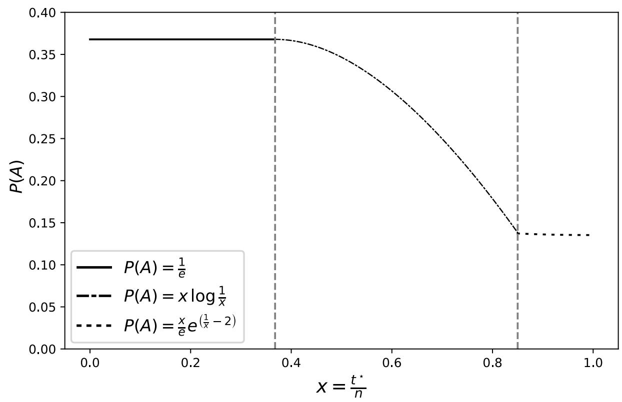


Figure 5.2: General-One-Step-Algorithm probabilities for different $\frac{p^\star}{n}$ values.

Theorem 5.7. *For any instance* $\mathcal{I} = (r_1(p), v)$, General-One-Step-Algorithm *finds an optimal solution with probability at least* $1/e^2$.

Proof. We split the time horizon in the three intervals $[1, 1/e]$, $[1/e, 0.8511]$ and $[0.8511, n]$. For each of this interval we have computed in Lemma 5.4, Lemma 5.5 and Lemma 5.6 respectively the probability of finding the optimal solution. Hence, General-One-Step-Algorithm finds an optimal solution with at least probability $1/e$. □

Theorem 5.8. *The* General-One-Step-Algorithm *expected probability of finding the optimal solution of a random instance* $\mathcal{I} = (r_1(p), v)$ *is asymptotically at least 0.29355.*

Proof. Using $x = p^\star/n$, we compute

$$\int_0^{\frac{1}{e}} \frac{1}{e} dx + \int_{\frac{1}{e}}^{0.8511} x \log\left(\frac{1}{x}\right) dx + \int_{0.8511}^{1} x e^{\frac{1}{x}-3} dx \geqslant 0.29355 \ . \quad \square$$

5.4.2 *Analysis for general* K

The step-function $r_K(p)$ consists of K steps, where each steps is an independent random variable, uniformly distributed on the interval $[1, n]$. Therefore it is possible that several steps are equal. We denote the steps with $p_1^\star \leqslant p_2^\star \leqslant \cdots \leqslant p_K^\star$, which are called *order statistics*. In particular, in case the variables are iid from a uniform continuous distribution $[0, 1]$ then $\mathbb{E}\left[p_i^\star\right] = i/(K+1)$, as showed by David and Nagaraja [34]. From this we get that $\mathbb{E}\left[p_i^\star/n\right] \simeq i/(K+1)$.

With more steps, finding an algorithm that works on the whole problem instance becomes increasingly difficult as we would generally have to do many independent choices and even just one of these choices being wrong means we are not going to get the optimal solution. Therefore our approach is to concentrate only on the tail end of the online stream, as if the optimal solution is indeed completely contained there, we at least have a good chance of finding it.

Our algorithm for a general class of functions $R_K(p)$ is the following: ignore everything that happens before time $p^\star_{K-1}$ and apply General-One-Step-Algorithm starting from time $p^\star_{K-1}$.

Algorithm 5.4 General-K-Step-Algorithm

Input : $r_K(p)$ and prices ν_p given in an online fashion
Output : Amount of energy bought at time $p \in \tau$
while $p \leqslant p^\star_{K-1}$ **do**
| Ignore the input
Call General-One-Step-Algorithm with the truncated function $r_{K-1}(p)|_{[p^\star_{K-1},n]}$. We require that when General-One-Step-Algorithm decides to buy on a day p, it buys an amount that fills the whole free available capacity of the battery.

Theorem 5.9. *The* General-K-Step-Algorithm *expected probability of finding the optimal solution for a random instance* $\mathcal{I} = (r_K(p), \boldsymbol{v})$ *is asymptotically lower bounded by* $0.29355 \cdot (2/(K+1)) = O(1/K)$.

Proof. The probability of the optimal solution being in $[p^\star_{K-1}, n]$ is $\frac{(n-p^\star_{K-1}+1)}{n}$ and the expected probability of finding the x_{OPT} in $[p^\star_{K-1}, n]$ is 0.29355.

$$\begin{aligned}\mathbb{E}\left[\mathbb{P}(A)\right] &= \mathbb{E}\left[0.29355 \cdot \frac{n-p^\star_{K-1}+1}{n}\right] \\ &= 0.29355 \cdot \frac{n-\mathbb{E}\left[p^\star_{K-1}\right]+1}{n} \\ &\simeq 0.29355 \cdot \left(\frac{n-\frac{(K-1)n}{(K+1)}}{n}\right) = 0.29355 \cdot \left(\frac{2}{K+1}\right) = O(1/K)\ . \quad \square\end{aligned}$$

5.5 COMPETITIVE ANALYSIS FOR ENERGY ACQUISITION PROBLEM WITH WELL STRUCTURED INPUTS

In this section we direct our attention to analyzing the expected value of our problem. We consider a special case where the free capacity function is $\bar{r}_K(p)$, which crucially has all the increments equally spaced at distance $(n-1)/(K+1)$ from each other.

The special case for $K = n-1$, so for a unitary increment at each time step is similar to the online knapsack problem with incremental capacity covered by Thielen, Tiedemann, and Westphal [82]. Similarly to our battery, they have a knapsack whose capacity increases by one unit each time period. But while we have an option to buy multiple units with the same price, they do not.

Lemma 5.10. *Using* $h = n/2$ *in* SAMPLE-AND-CHOOSE-ALGORITHM *the expected competitive ratio for* $\bar{r}_K(p)$ *is at least* $1/4$.

Proof. We show that the value $h = n/2$ maximizes the expected value for the SAMPLE-AND-CHOOSE-ALGORITHM. Let h be a variable in $[1, n]$ and let $M = \max\{v_1, \ldots, v_n\}$. Then

$$\begin{aligned}\mathbb{E}_\sigma[V_{\mathrm{ALG}}] &= \frac{1}{n}\left(\sum_{j=h+1}^{n} \frac{h}{j-1}\left\lceil\frac{j(k+1)}{n-1}\right\rceil M\right)\\ &\geqslant \frac{Mh(k+1)}{n(n-1)}\left(\sum_{j=h+1}^{n}\frac{j}{j-1}\right) \geqslant \frac{Mh(k+1)}{n(n-1)}(n-h)\ .\end{aligned}$$

This expression is maximized for $h = n/2$ which leads to $\mathbb{E}_\sigma[V_{\mathrm{ALG}}] \geqslant \frac{(k+1)nM}{4(n-1)}$. Since $\max_\sigma(V_{\mathrm{OPT}}) = (K+1)M$ we obtain

$$\mathbb{E}_\sigma\left[\frac{V_{\mathrm{ALG}}}{V_{\mathrm{OPT}}}\right] \geqslant \mathbb{E}_\sigma\left[\frac{V_{\mathrm{ALG}}}{(K+1)M}\right] = \frac{\mathbb{E}_\sigma[V_{\mathrm{ALG}}]}{(K+1)M} \geqslant \frac{n}{4(n-1)} > \frac{1}{4}\ . \qquad \square$$

5.6 EXPERIMENTAL ANALYSIS

In this section we show our experimental analysis for the algorithms developed in this chapter. We implemented GENERAL-ONE-STEP-ALGORITHM from Section Section 5.4 and SAMPLE-AND-CHOOSE-ALGORITHM from Section Section 5.5 on Python and ran them on simulated instances.

First, we checked experimentally whether the average number of instances in which we select the optimal solution is close to the probability 0.29355 computed in Theorem 5.8. To do so, we simulated instances with a time horizon n spanning from 1 to 200 and where the value of $p^\star$ was uniformly at random chosen from $\{1, \ldots, n\}$. The values $\mathbf{v}$ were instead uniformly at random sampled from the interval $\{10, \ldots, 10\,000\}$. In Figure 5.3 we see that experimentally we select the optimal solution with probability 0.292.

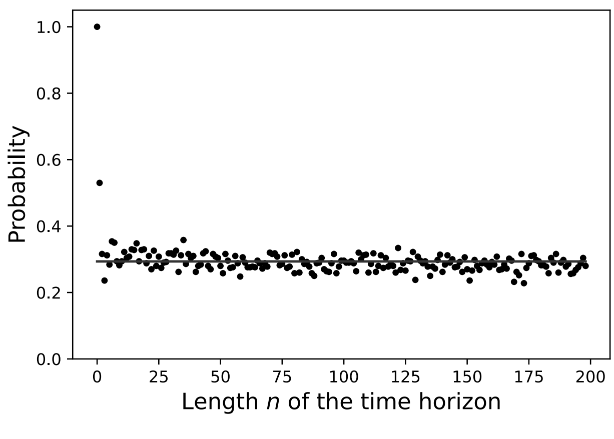


Figure 5.3: Rate at with GENERAL-ONE-STEP-ALGORITHM selects the optimal offline solution. With the continuous line we depict the predicted probability.

Lastly, we wanted to verify experimentally whether the empirical average of our the quantity V_{ALG}/V_{OPT} is close to the expected one computed in Lemma 5.10. We consider then 20 different instances with time horizon of length $n = 10$. We generate the vector of values $\mathbf{v}$ where each value is chosen uniformly at random from the interval $[0.1, 1]$ and we compute the ratio of V_{ALG} and V_{OPT} for all the possible permutations of the vector $\mathbf{v}$. We show in Figure 5.5 the results obtained for $K = \{1, 5, 7, 9\}$, from which we can conclude that even if our results is only in expectation, the values V_{ALG} our solution is not far from the value V_{OPT} of the optimal .

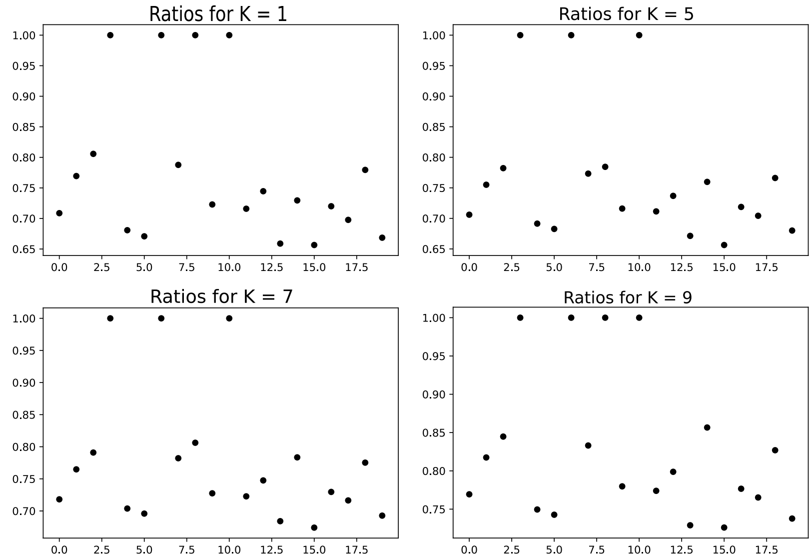


Figure 5.4: Ratio of the average over all the possible permutations of σ.

If instead of the average over all the permutations σ of the vector of values $\mathbf{v}$ we consider just one single among all the possible permutations, so that we can extend our experiments to instances with a time horizon up to 200 time periods.

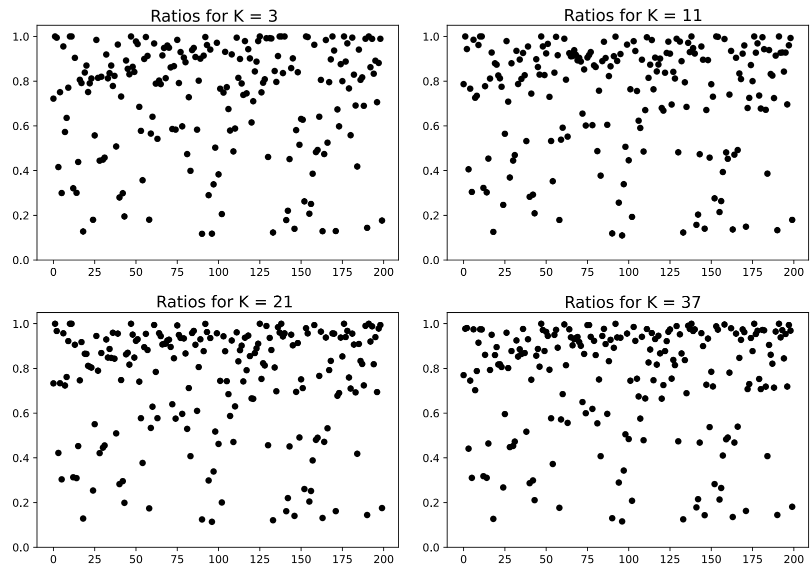


Figure 5.5: Ratio of the average over all the possible permutations of σ.

INTERACTION OF AGENTS

6 PRICING

In this chapter we leave for a moment the Energy Trading world and we focuse instead on a combinatorial market. Here, the goal is to find a pair of pricing and an allocation which maximizes some quantities, such as the social welfare or the revenue of some particular player. We consider dynamic pricing schemes, where players arrive in an unspecified sequential order. Between the arrive of two players, prices can be updated. This raises important questions, especially regarding the fairness of such markets. In this chapter, we investigate the existence of optimal dynamic prices under fairness in unit-demand markets.

This chapter is based on joint work with Kristóf Bérczi, Julian Golak and Alexander Grigoriev [15].

6.1 MOTIVATIONS AND BACKGROUND

The great availability of surveys and marketing researches help sellers to predict the interest of costumers in different products, which opens up the possibility to apply user specific pricing processes. While this helps customers in purchase decisions, it makes the pricing process even more challenging for sellers. In this context, some simple economic rules are straightforward: low prices typically attract more customers but have a small revenue per item, while high prices generate a greater revenue per item but attract less customers. As sellers aim at maintaining customers' satisfaction while achieving high revenue, effective pricing strategies are needed.

We consider combinatorial markets, in which a set of indivisible items is to be distributed among a set of agents. Each agent has a valuation for each subset of items that measures how much receiving the bundle would be worth to the agent. An allocation assigns a subset of items to each agent so that every item is assigned to at most one of them. In a posted price mechanism, the seller can set the prices of the items individually, and the utility of an agent for a given bundle of items is the agent's value for the bundle minus the total price of all contained items – the utility hence measures the agent's happiness when buying all the items of the bundle at the given prices. An allocation is considered to be envy-free if no agent would prefer a different bundle of items at the posted prices.

In this chapter, we study resource allocation problems in a dynamic environment from two perspectives on objectives. First, we focus on how to set prices so that a market equilibrium maximizes the overall social welfare, that is, the total sum of the agents' values. Second, we consider how to set prices that the seller's profit is maximized.

One of the main contributions of this chapter is the introduction and analysis of the fairness restrictions mitigating agents' enviousness in the dynamic price environment. Notice, under the realistic assumption that the agents arrive in unspecified sequential order, the concept of fairness of dynamic prices becomes non-trivial. For instance, in the markets with expected price deflation, e.g., of perishable products or deteriorating over time quality, a typical agent follows the prices prior to their purchase and become price-insensitive after. In the markets with expected price inflation, e.g., one-time or limited-time service, agents are price-insensitive before purchase and attentive after. To properly address these differences in fairness, we introduce ex-post and ex-ante types of agents' enviousness and envy-freeness in dynamic pricing.

6.2 PREVIOUS WORK

Achieving optimal social welfare through simple mechanisms has been the center of attention for a long time due to its far-reaching applications. In particular, posted price mechanisms became a key approach to allocate resources, hence finding optimal pricing schemes is a fundamental question in combinatorial markets. A pair of pricing and allocation is called a Walrasian equilibrium if all the items are assigned and each agent receives a bundle that maximizes their utility, so by definition this allocation maximizes the social welfare. The idea of Walrasian equilibria was introduced already in the late 1800s [85] and the existence of such an equilibria was verified for gross substitutes valuations by Kelso Jr. and Crawford [57]. However, it was pointed out by Cohen-Addad et al. [29] and independently by Hsu et al. [53] that the existence of Walrasian allocations strongly depends on the tie breaking process, usually carried out by a central coordinator. If the agents arrive one after the other and choose an arbitrary bundle of items that maximizes their utility, then the absence of a tie-breaking rule may result in a suboptimal allocation with respect the social welfare. To overcome these difficulties, Cohen-Addad et al. [29] introduced the notion of dynamic prices that proved to be a powerful tool in designing markets without a central tie-breaking coordinator. In the proposed model, agents arrive in an unspecified sequential order and the item prices can be updated before the arrival of the next agent. Their main result is a polynomial time dynamic pricing scheme that achieves optimal social welfare in unit-demand markets. This work initiated the study of dynamic pricing schemes, and the existence of optimal dynamic prices was settled for three agents with multi-demand valuations by Berger et al. [17], for bi-demand valuations by Bérczi et al. [14], for two agents with certain matroid rank valuations by Bérczi et al. [16], and recently for four agents with multi-demand valuations by Pashkovich and Xie [73].

The market clearing condition can lead to Walrasian equilibrium with low revenue for the seller, even if prices are as high as possible. In the seminal work of Guruswami et al. [46], envy-free pricing was introduced as a relaxation of Walrasian equilibrium by dropping the requirement on the clearance of the market, but keeping a fairness condition. In their model, the goal is to maximize the revenue when each agent has a demand and valuation for each bundle of items and each item has limited supply. The authors showed that maximizing the revenue is APX-hard already for unit-demand markets, and provided logarithmic approximations in the number of customers for the unit-demand and single-parameter cases. A similar hardness was proved independently by Aggarwal et al. [1]. Subsequently several versions of the problem have been shown to have poly-logarithmic inapproximability, see e.g. the works of Briest [19] and Chalersmook et al. [25]. Bansal et al. [7] adapted the concept of envy-freeness to a pricing over time scheme. In their model, there is a single item with unlimited supply, and each agent is associated with a time interval over which they will consider buying a copy of the item, together with a maximum value the agent is willing to pay for it. The seller's goal is to set the prices at every time unit so as to maximize revenue from the sale of copies of the item over the time period.

In a broader context, the considered pricing problems are related to fair division and prophet inequalities. In fair division problems, the goal is to allocate items among agents in a manner that ensures certain fairness criteria. More precisely, an allocation is considered to be envy-free if no agent desires another agent's bundle over their own. While the idea of dynamic settings in envy-free pricing has not been considered, it has been extensively studied within the realm of fair division. In particular, envy-freeness in a dynamic setting where agents arrive sequentially over time was considered e.g. in the works of Aleksandrov et al. [2], Barman et al. [9] , Benadè et al. [13], He et al. [49], Benadè et al. [12], Gorokh et al. [45] and Zeng and Psomas [86]. In other works, authors have considered fair allocations allowing subsidy payments to mitigate envy among participants, see Halpern and Shah [48] and Brustle et al. [20]. The prophet inequality is a classic theorem from the 70s that served as the starting point for several related optimal stopping problems, as for instance also the problem tackled in Chapter 5. Recently, a connection between prophet inequalities and posted price mechanisms was uncovered which gave the area a renewed interest. In the corresponding model, a seller is trying to maximize the revenue from a single indivisible item. Agents approach the seller sequentially, where each agent's value for the item is drawn from a distribution known by the seller. Based on this information, the seller sets prices before the arrival of each buyer with the aim of maximizing revenue. For a survey over the vast literature on prophet inequality, we refer the interested reader to the works of Lucier [61] and Correa et al. [31].

6.3 DEDICATED NOTATION

We add here some extra notation which is used in this chapter. We start with some extra notation for sets and opterations on sets. Let $\mathcal{I}$ be a ground set. For two subsets $X, Y \subseteq \mathcal{I}$, their *symmetric difference* is $X \triangle Y := (X \setminus Y) \cup (Y \setminus X)$. When Y consists of a single element y, then the *difference* $X \setminus \{y\}$ and *union* $X \cup \{y\}$ are abbreviated by $X - y$ and $X + y$, respectively. For a function $f \colon \mathcal{I} \to \mathbb{R}$, the total sum of its values over X is denoted by $f(X) := \sum_{s \in X} f(s)$. For $X = \emptyset$, we define $f(\emptyset) = 0$.

GRAPHS. We denote a *bipartite graph* by $G = (\mathcal{I}, \mathcal{A}; E)$, where $\mathcal{I}$ and $\mathcal{A}$ are the vertex classes and E is the set of edges. By *edge-weights*, we mean a function $w \colon E \to \mathbb{R}_+$. For a subset $X \subseteq \mathcal{I} \cup \mathcal{A}$, the *subgraph of* G *induced by* X is the graph obtained from G by deleting all the vertices not contained in X, together with edges incident to them. For ease of notation, we denote an edge of the graph going between $a \in \mathcal{A}$ and $i \in \mathcal{I}$ by ai.

By orienting the edges of a bipartite graph, we get a *directed graph* $D = (\mathcal{I}, \mathcal{A}; F)$, where F is the set of arcs. A directed graph is called *strongly connected* if every vertex is reachable from every other vertex through a directed path. A *strongly connected component* of a directed graph is a subgraph that is strongly connected and is maximal with respect this property. By contracting each strongly connected component of a directed graph to a single vertex, one obtains an acyclic directed graph. Therefore, the strongly connected components have a so-called *topological ordering* in which every arc going between components goes from an earlier component to a later one.

MARKET MODEL. A combinatorial market consists of a set $\mathcal{I}$ of *indivisible items* and a set $\mathcal{A}$ of *agents*. Throughout the chapter, we denote by $m := |\mathcal{I}|$ and $n := |\mathcal{A}|$ the numbers of items and agents, respectively. Note that since at every time a single agent arrives, we are also here preserving the same length n of the time horizon. An *allocation* $\mathbf{X}$ assigns each agent a a subset X_a of items so that each item is assigned to at most one agent.

In a *unit-demand market*, each agent $a \in \mathcal{A}$ has a valuation $v_a \colon \mathcal{I} \to \mathbb{R}_+$ over individual items and they desire only a single good, that is, we consider allocations $\mathbf{X}$ with $|X_a| \leqslant 1$ for $a \in \mathcal{A}$ – in such cases we denote the item obtained by agent a by x_a. If X_a is empty, we define $x_a = \emptyset$. We always assume that the agents' valuations are known in advance. Furthermore, we assume that $v_a(\emptyset) = 0$ for all agents $a \in \mathcal{A}$. As in the previous chapters, we denote prices with b, so that $b(i)$ is the price for item $i \in \mathcal{I}$. The *utility* of agent a for item i is then defined as $u_a(i) := v_a(i) - b(i)$. Then the *social welfare* corresponding to the allocation is $\sum_{a \in \mathcal{A}} v_a(x_a)$, while the *revenue* of the seller is $\sum_{a \in \mathcal{A}} b(x_a)$.

6.3.1 *Our market*

In combinatorial markets we are actually solving two problems connected to each other: we need indeed to find a pair of pricing vector $b \colon \mathcal{I} \to \mathbb{R}_+$ and an allocation $\mathbf{X}$ such that the social welfare or the revenue is maximized. In a *static pricing scheme*, the seller sets the price $b(i)$ of each item $i \in \mathcal{I}$ in advance. In contrast, in a *dynamic pricing scheme* the agents arrive one after the other, and the seller can update the prices between their arrivals based on the remaining sets of items and agents. The order in which agents arrive is represented by a bijection $\sigma\colon \mathcal{A} \to \tau$. The sets of agents, items and prices available at the arrival of the pth agent are denoted by $\mathcal{A}_p$, $\mathcal{I}_p$ and $b_p : \mathcal{I}_p \to \mathbb{R}_+$, respectively. The utility of agent a for item i at time step p is then defined as $u_{a,p}(i) := v_a(i) - b_p(i)$. The next agent always chooses an item that maximizes their utility. After the last buyer has left, the pricing scheme terminates and results in pricing vectors $\mathbf{b} = (b_1, \dots, b_n)$ and an allocation $\mathbf{X} = (x_1, \dots, x_n)$, where b_p is the price vector available at the arrival of the $p-$th agent and x_p is the item allocated to them. Note that x_p might be an empty set if the utility of the agent is non-positive for each item in $\mathcal{I}_p$.

In what follows, we define different variants of the model. Depending on whether ties between items are broken by the seller or the agents, we distinguish two cases:

(C1) *Seller-chooses.* If there are several items maximizing the utility of the current agent, then the seller decides which one to assign.

(C2) *Agent-chooses.* If there are several items maximizing the utility of the current agent, then the agent decides which one to take.

One might have the intuition that the two variants are equivalent, since the valuations are known in advance and hence the seller can simply subtract an epsilon from the price of the item she would like the agent to choose. However, such a solution would violate envy-freeness over some part of the time horizon. In fact, in terms of finding an optimal pricing, problem (C1) is easier. Indeed, given an optimal pricing for (C2), the seller can always decide to allocate the item that was chosen by the agent.

Previous works generally assumed that agents arrive in an unspecified order. Besides this, we consider two further variants based on the control and information of the arrival process:

(O1) *Unspecified.* The agents arrive in a fixed order that the seller has no information on.

(O2) *Predetermined.* The agents arrive in a fixed order that the seller knows in advance.

(O3) *Alterable.* The order of the agents is determined by the seller in advance.

Note that the alterable setting significantly differs from the offline problem. Indeed, a natural approach would be picking the order of the agents to be decreasing/increasing in the prices of their optimal bundle and then setting infinite prices for items that should not be bought at any given step. Again, such a solution would violate envy-freeness over some part of the time horizon.

FAIRNESS DEFINITION Our model differs from earlier ones mainly in that we are seeking for optimal pricing schemes under fairness constraints. In the static setting, a pair of pricing b and allocation $\mathbf{X}$ is *envy-free* if $u_{a,\sigma(a)}(x_a) = \max\{u_{a,p}(i) \mid p \in P_a, i \in \mathcal{I}_p\}$ holds for each agent $a \in \mathcal{A}$. The dynamic setting naturally suggests variants in which envy-freeness is defined over a subset of time steps. Let $P_a \subseteq \tau$ be a subset of time steps for each agent $a \in \mathcal{A}$. Then price vectors $\mathbf{b} = (b_1, \ldots, b_n)$ and allocation $\mathbf{X} = (x_1, \ldots, x_n)$ form an envy-free allocation if $u_{a,\sigma(a)}(x_a) = \max\{u_{a,p}(i) \mid p \in P_a, i \in \mathcal{I}_t\}$ for each agent $a \in \mathcal{A}$. We propose four possible notions of envy-freeness of different strength depending on the time period over which agents compare themselves to others:

(F1) *Strong envy-freeness.* Agents consider prices for the whole time horizon, that is, $P_a = \{1, \ldots, n\}$ for $a \in \mathcal{A}$.

(F2) *Ex-post envy-freeness.* Agents consider prices available before and at their arrival, that is, $P_a = \{1, \ldots, \sigma(a)\}$ for $a \in \mathcal{A}$.

(F3) *Ex-ante envy-freeness.* Agents consider prices available after and at their arrival, that is, $P_a = \{\sigma(a), \ldots, n\}$ for $a \in \mathcal{A}$.

(F4) *Weak envy-freeness.* Agents consider prices at their arrival, that is, $P_a = \{\sigma(a)\}$ for $a \in \mathcal{A}$.

Using this terminology, optimal dynamic pricing schemes discussed in already in the works of Bérczi et al. [14], Bérczi et al. [16], Berger et al. [17], Cohen-Addad et al. [29], and Pashkovich and Xie [73] provide weakly envy-free solutions.

As for the *objective function*, we either consider the *social welfare* $W(\mathbf{X}) = \sum_{a \in \mathcal{A}} v_a(x_a)$ or the *revenue* of all sold items $R(\mathbf{b}, \mathbf{X}) = \sum_{a \in \mathcal{A}} b_{\sigma(a)}(x_a)$.

These variants and the results for the social welfare and the revenue are summarized in Table 6.1. The results are split horizontally by the type of envy-freeness considered, while the columns are indexed by the type of the ordering of the agents. Algorithmic results hold irrespective of how agents break ties, while hardness results hold even if ties are broken by the seller. It is worth noting that the $O(\log(n))$-approximation algorithm of Guruswami et al. [46] extends to all of variants of envy-free pricing where the objective is to maximize the revenue.

Remark 6.1. *In strongly envy-free pricing models when ties are broken by agents, optimizing with respect to social welfare or revenue may lead to allocations where the outcome is drastically influenced by the choices of agents. In such cases, 'optimality' of a pricing scheme is not well-defined. This is well-illustrated by the classic example of Cohen-Addad et al. [29] with three items* i_1, i_2, i_3 *and three agents* a_1, a_2, a_3 *having valuations* $v_{a_j}(i_j) = v_{a_j}(i_{j+1}) = 1$ *and* $v_{a_j}(i_{j+2}) = 0$ *for* $j \in [3]$*, where indices are meant in a cyclic order. Since each item has the same value for two of the agents, strong envy-freeness implies that the seller should set the prices uniformly and cannot update them. Assume that* $b(i_1) = b(i_2) = b(i_3) = 1$ *and that agent* a_3 *arrives first who chooses item* i_3*. If* a_1 *arrives next, the agent is actually indifferent between* i_1 *and* i_2*, but the achieved social welfare or revenue heavily depends on their decision.*

Table 6.1: Complexity landscape of social welfare and revenue maximization under fairness constraints in unit-demand markets. Algorithmic results (green cells) hold even in the agent-chooses setting, while hardness results (red cells) hold already for the seller-chooses case. In each row, complexities of cells with light shade are implied by cells with darker shade. While maximizing welfare and in the variant of seller breaking ties and agents arrive in unspecified order is considered in Walras [85] and Kelso and Crawford [57] assuming strong envy-free and in Cohen-Addad et al. [29] assuming weak envy-free agents.

Welfare maximization

	Ties	Unspecified	Predetermined	Alterable
Strong	Agents	Not exists Rem. 6.1	P Thm. 6.5	P
	Seller	P [57, 85]		
Ex-post		P Thm. 6.10	P	P
Ex-ante		P Thm. 6.16	P	P
Weak		P [29]	P	P

Revenue maximization

	Ties	Unspecified	Predetermined	Alterable
Strong	Agents	Not exists Rem. 6.1	APX-hard	APX-hard Thm. 6.21
	Seller	APX-hard		
Ex-post		APX-hard	APX-hard Thm. 6.22	P Thm. 6.23
Ex-ante		APX-hard	APX-hard Thm. 6.22	P Thm. 6.24
Weak		Open	P Thm. 6.25	P

6.3.2 *Weighted coverings*

A unit-demand combinatorial market can be represented by a complete edge-weighted bipartite graph $G = (\mathcal{I}, \mathcal{A}; E)$, where vertex classes $\mathcal{I}$ and $\mathcal{A}$ correspond to the sets of items and agents, respectively. For any item $i \in \mathcal{I}$ and agent $a \in \mathcal{A}$, the weight of the edge ai is $w(ia) := v_a(i)$. Then there is a one-to-one correspondence between allocations maximizing social welfare and maximum weight matchings of G. Even more, the maximum weight of a matching is clearly an upper bound on the maximum revenue achievable through any pricing mechanism. These observations motivate to investigate dynamic pricing schemes through the lens of maximum weight matchings.

Let us denote the vertex set of G by $V := \mathcal{I} \cup \mathcal{A}$. A function $\pi\colon V \to \mathbb{R}$ is a *weighted covering* if $\pi(i) + \pi(a) \geqslant w(ia)$ holds for every edge $ia \in E$. The *total value* of the covering is $\pi(V) = \sum_{v \in V} \pi(v)$. A weighted covering of minimum total value is called *optimal*. A cornerstone result of graph optimization is due to Egerváry [36] who provided a min-max characterization for the maximum weight of a matching in a bipartite graph.

Theorem 6.2 ([36]). *Let $G = (\mathcal{I}, \mathcal{A}; E)$ be a bipartite graph and $w\colon E \to \mathbb{R}$ be a weight function on the set of edges. Then the maximum weight of a matching is equal to the minimum total value of a non-negative weighted covering π of w.*

Given a weighted covering π, item $i \in \mathcal{I}$ and agent $a \in \mathcal{A}$, the edge ai is called *tight with respect to* π if $\pi(i) + \pi(a) = w(ia)$. The *subgraph of tight edges* is then denoted by $G_\pi = (\mathcal{I}, \mathcal{A}; E_\pi)$. We call an edge $ia \in E$ *legal* if there exists a maximum weight matching containing it, and in such a case we say that i is *legal* for a. It is known that legal edges are always tight with respect to any optimal weighted covering, while the converse does not always hold, that is, a tight edge is not necessarily legal. However, Bérczi, Bérczi-Kovács, and Szögi [14] showed that a careful choice of π ensures the sets of tight and legal edges to coincide.

Lemma 6.3 ([14]). *The optimal π attaining the minimum in Theorem 6.2 can be chosen such that*

(a) an edge ai is tight with respect to π if and only if it is legal, and

(b) $\pi(v) = 0$ for some $v \in V$ if and only if there exists a maximum weight matching M in which v is exposed, that is, no edge of M is incident to v.

Furthermore, such a π can be determined in polynomial time.

Finally, we will use the following technical lemma, see Bérczi, Bérczi-Kovács, and Szögi [14].

Lemma 6.4. *Consider an instance of maximizing social welfare in a unit-demand combinatorial market, and let $G = (\mathcal{I}, \mathcal{A}; E)$ be the corresponding edge-weighted bipartite graph. We may assume that all items are covered by every maximum weight matching of G.*

6.4 MAXIMIZING THE SOCIAL WELFARE

Since a Walrasian equilibrium maximizes social welfare and ensures envy-freeness at the same time, the existence of optimal dynamic pricing schemes under fairness constraints is settled when ties are broken by the seller. The seminal paper of Cohen-Addad et al. [29] initiated the study of the agent-chooses case, and provided an algorithm for determining a weakly envy-free solution in unit-demand markets.

A different proof of the same result was later given by Bérczi, Bérczi-Kovács, and Szögi [14]. Their algorithm starts with a minimum weighted covering provided by Lemma 6.3 in the edge-weighted bipartite graph representing the market, and sets the initial prices according to the covering values. As a result, the first agent a chooses an item i such that ai is tight, hence i is legal for a. Based on this, it may seem that, keeping the same prices throughout, the resulting allocation will eventually be a maximum weight matching. However, after item i is taken, an edge that was legal before might become non-legal. To overcome this, the weighted covering needs to be updated in the remaining graph at each time step, which causes the price of an item fluctuating over time. For that reason, the algorithm does not extend to the ex-post and ex-ante envy-free cases.

To prevent the fluctuation of prices, we do not recompute the weighted covering at each time step from scratch. Instead, we fix a single weighted covering at the very beginning, and then we always slightly modify it to control the agents' choices in such a way that

1. no matter which agent arrives next, if they are covered by every maximum weight matching in the current graph then they pick an item that is legal for them, otherwise they either pick an item that is legal for them or do not take an item at all, and

2. the price-changes from time step $p-1$ to p are limited to non-increases in the ex-post and to non-decreases in the ex-ante case.

The first property implies that the final allocation corresponds to a maximum weight matching of G, hence it maximizes social welfare. The second property ensures that the resulting allocation automatically meets the requirements of ex-ante envy-freeness. However, non-increasing prices themselves are not sufficient for ensuring ex-post envy-freeness, since a newly arriving agent might envy an item that was already taken by someone in an earlier time step. Consequently, the two settings do not admit a straightforward symmetry, and achieving ex-post envy-freeness requires stronger arguments.

As a warm-up, we first prove the existence of an optimal strong envy-free pricing when the ordering of the agents is known in advance.

Theorem 6.5. *If the ordering of the agents is known in advance, then there exists a social-welfare-maximizing dynamic pricing scheme for the unit-demand strong envy-free pricing problem even if ties are broken by the agents. Furthermore, the optimal prices can be determined in polynomial time.*

Proof. Consider the edge-weighted bipartite graph $G = (\mathcal{I}, \mathcal{A}; E)$ representing the market and let σ denote the ordering of the agents. By Lemma 6.4, we may assume that all items are covered by every maximum weight matching of G. Let $M \subseteq E$ be an arbitrary maximum weight matching. For each agent a, define x_a as the item to which a is matched in M; if a is unmatched, x_a is defined as the empty set. Additionally, let π be a weighted covering provided by Lemma 6.3. Recall that for any agent a, the edge ax_a is tight if $x_a \neq \emptyset$, and $\pi(a) = 0$ if $x_a = \emptyset$.

Now we describe how to set the prices at each time step p. If agent a arrives next according to σ, then we set the price of each item $i \in \mathcal{I}$

to $\pi(i)$ if $i = x_a$ and to $\pi(i) + \varepsilon$ otherwise, where $\varepsilon > 0$. We show that the resulting allocation corresponds to the maximum weight matching M.

For any other item i different from x_a, we have

$$u_{a,\sigma(a)}(i) = v_a(i) - b_{\sigma(a)}(i) \leqslant (\pi(a) + \pi(i)) - (\pi(i) + \varepsilon) < \pi(a)$$

$u_{a,\sigma(a)}(i) = v_a(i) - b_{\sigma(a)}(i) \leqslant (\pi(a) + \pi(i)) - (\pi(i) + \varepsilon) < \pi(a)$. If $x_a = \emptyset$, then $\pi(a) = 0$ and so the agent takes none of the items because of having negative utility for any of those. If $x_a \neq \emptyset$, then $u_{a,\sigma(a)}(x_a) = v_a(i) - b_{\sigma(a)}(x_a) = (\pi(a) + \pi(x_a)) - \pi(x_a) = \pi(a)$, hence the agent prefers x_a to any other item. This shows that the final allocations is indeed M. Also, note that the utility received by any agent a is $\pi(a)$.

It remains to show that the allocation is strongly envy-free. To see this, observe that

$$u_{a,\sigma(a)}(x_a) = \pi(a) = (\pi(a) + \pi(i)) - \pi(i) \geqslant v_a(i) - b_p(i) = u_{a,p}(i)$$

for every $i \in \mathcal{I}$ and $p \in \tau$, concluding the proof of the theorem. $\square$

Consider the edge-weighted bipartite graph $G = (\mathcal{I}, \mathcal{A}; E)$ representing the market. By Lemma 6.4, we may assume that all items are covered by every maximum weight matching of G. Take a weighted covering π provided by Lemma 6.3. Throughout this section, tightness of an edge is always meant with respect to π. Recall that G_π denotes the subgraph of tight edges.

At time step p, the sets of remaining agents and items are denoted by $\mathcal{A}_p$ and $\mathcal{I}_p$, respectively. We denote the subgraph of G_π induced by vertices $\mathcal{I}_p \cup \mathcal{A}_p$ by $G_p = (V_p, E_p)$. For a maximum weight matching M_p of G_p, let x_a^p denote the item to which a is matched in M_p if such an item exists, otherwise define x_a^p to be the empty set. Note that the edges ax_a^p are obviously legal in G_p.

Given such a matching M_p, we construct a directed graph D_p as follows. We add another copy of every edge in M_p to G_p that we refer to as dummy edges. Then we orient the original copies in M_p from $\mathcal{I}_p$ to $\mathcal{A}_p$, and orient all the remaining edges – including the dummy ones – from $\mathcal{A}_p$ to $\mathcal{I}_p$. We denote the strongly connected components of the resulting directed graph by $C_1^p, \dots, C_{q_p}^p$ indexed according to a topological ordering, see Figure 6.1 for an example.

The following technical claims will be used later.

Claim 6.6. *Let $a \in \mathcal{A}_p$ and $i \in \mathcal{I}_p$ be such that they are in the same strongly connected component of D_p and ai is tight. Then i is legal for a in G_p.*

Proof. If the edge is oriented from i to a, then $ia \in M_p$ and the statement clearly holds. Otherwise, the edge is oriented from a to i. Since every arc of a strongly connected digraph is contained in a directed cycle, it suffices to show that any directed cycle C consists only of legal edges. This follows from the fact that all the edges of G_p are tight, hence $M_p \triangle C$ is also a maximum weight matching of G_p, implying that it consists of legal edges. $\square$

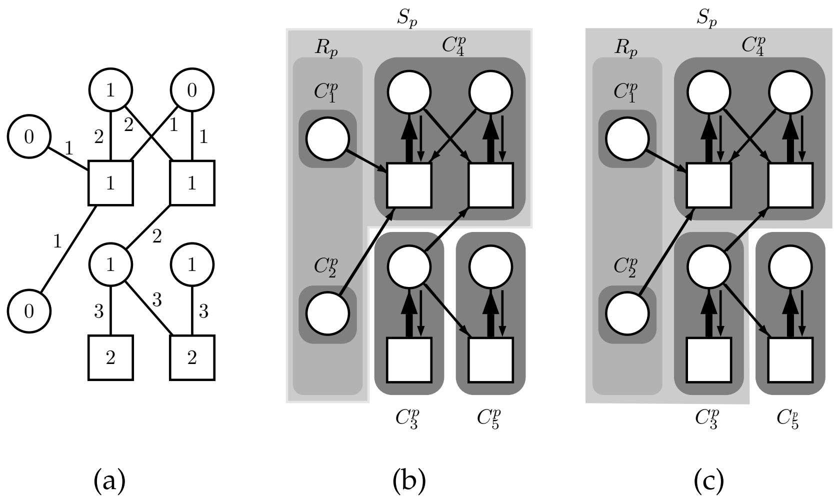


Figure 6.1: Illustration of the constructions for the ex-post and ex-ante cases. Circles and squares correspond to agents and items, respectively. Thick edges denote a maximum weight matching M_p, while $C_1^p, \ldots, C_5^p$ are the strongly connected components of the directed graph D_p. In (a) we have G_p with edge-weights and weighted covering values, in (b) the definition of the set S_p in the proof of Theorem 6.23 and in (c) the definition of the set S_p in the proof of Theorem 6.24.

In both the ex-post and ex-ante cases, our proof builds on the following idea. It is not difficult to check that if one sets the price of each item to its π value, then agents strictly prefer items to which they are connected through a tight edge over items for which the corresponding edge is not tight, see Claim 6.7. However, as time passes, a tight edge is no longer necessarily legal. In order to prevent the next agent to choose such an edge, we will slightly change the prices in a case-specific way. To do this, let $\delta > 0$ be a constant for which δ is strictly less than

$$\frac{1}{2}\min\left\{\min\{\pi(a)+\pi(i)-v_a(i) \mid ia \in E \text{ is not tight}\}, \min\{\pi(i) \mid i \in \mathcal{I}\}\right\}.$$

Note that such a δ exists by Lemma 6.3(b) and Lemma 6.4. We also choose a small constant $0 < \varepsilon < \delta/(n2^n)$ that will be used later on. The next claim shows that tight edges lead to greater utility than non tight ones.

Claim 6.7. *Let $a \in \mathcal{A}_p$ and $i, i' \in \mathcal{I}_p$ be such that ai is tight, ai' is not tight. If $b_p(i) \leqslant \pi(i) + \delta$ and $b_p(i') \geqslant \pi(i') - \delta$, then a strictly prefers i over i'.*

Proof. Assume that $b_p(i) \leqslant \pi(i) + \delta$ and $b_p(i') \geqslant \pi(i') - \delta$. Since ai is tight and ai' is not tight, we have $v_a(i) = \pi(a) + \pi(i)$ and $v_a(i') + 2\delta < \pi(a) + \pi(i')$ by the definition of δ. Thus we get

$$\begin{aligned} u_{a,p}(i') &\leqslant v_a(i') - (\pi(i') - \delta) < \pi(a) - \delta \\ &= v_a(i) - (\pi(i) + \delta) \leqslant u_{a,p}(i) \ . \end{aligned}$$

Hence the utility of a for item i is strictly larger than for item i', concluding the proof. $\square$

Finally, the following two technical claims follow easily from the definitions.

Claim 6.8. *Let* $a \in \mathcal{A}_p$ *and* $i \in \mathcal{I}_p$ *be such that* ai *is not tight. If* $b_p(i) \geqslant \pi(i) - \delta$*, then* $u_{a,p}(i) < \pi(a)$*.*

Proof. As ai is not tight, the definition of δ implies $u_{a,p}(i) = v_a(i) - b_p(i) < (\pi(a) + \pi(i) - \delta) - (\pi(i) - \delta) = \pi(a)$ as stated. □

Claim 6.9. *Let* $a \in \mathcal{A}_p$ *and* $i \in \mathcal{I}_p$ *be such that* ai *is tight. If* $b_p(i) = \pi(i) + \omega$ *for some real number* ω*, then* $u_{a,p}(i) = \pi(a) - \omega$*.*

Proof. As ai is tight, we get

$$u_{a,p}(i) = v_a(i) - b_p(i) = (\pi(a) + \pi(i)) - (\pi(i) + \omega) = \pi(a) - \omega \ ,$$

as stated. □

The set of agents not matched in M_p is denoted by $R_p \coloneqq \{a \in \mathcal{A}_p \mid x_a^p = \emptyset\}$. Note that $\pi(a) = 0$ for each $a \in R_1$ by Lemma 6.3(b). In fact, we will maintain this property for each set R_p throughout the pricing process.

From this point, we discuss the ex-post and ex-ante cases separately as their proofs differ in that prices must be set differently.

6.4.1 *Ex-post envy-free pricing*

The algorithm in Bérczi, Bérczi-Kovács, and Szögi [14] for the unit-demand case updates the prices at each time step using an arbitrary weighted covering in the remaining graph. In order to adapt a similar idea for the ex-post case, we do this in a controlled manner which ensures that prices do not increase over time.

Theorem 6.10. *There exists a welfare-maximizing dynamic pricing scheme for the unit-demand ex-post envy-free pricing problem even if ties are broken by the agents. Furthermore, the optimal prices can be determined in polynomial time.*

Proof. Throughout the algorithm, we maintain a maximum weight matching M_p in the graph G_p of remaining agents and items. Recall that R_p denotes the set of agents not covered by the matching M_p. Initially, $\pi(a) = 0$ for $a \in R_1$ by Lemma 6.3(b), and we will maintain this property for each $a \in R_p$ throughout. We describe a general phase $p \in \tau$ of the pricing process. Assume that

$$(\star) \qquad \pi(a) = 0 \text{ for } a \in R_p.$$

We define the set

$$S_p \coloneqq \{v \in V_p | \text{there exists a directed path in } Dp \\ \text{to } v \text{ from an agent } a \in R_p\},$$

see Figure 6.1 for an example.

In particular, $R_p \subseteq S_p$. Remember, that $C^p_1, \dots, C^p_{q_p}$ are indexed according to a topological ordering. Note that either $C^p_j \subseteq S_p$ or $C^p_j \cap S_p = \emptyset$ for each strongly connected component C^p_j of D_p. The prices are then updated as follows:

$$b_p(i) = \begin{cases} \pi(i) + \delta/2^p + j\varepsilon & \text{if } i \in C^p_j \text{ such that } C^p_j \subseteq S_p, \\ \pi(i) - \delta(1 - 1/2^p) + j\varepsilon & \text{if } i \in C^p_j \text{ such that } C^p_j \cap S_p = \emptyset. \end{cases}$$

By the choice of δ and ε, the prices are non-negative. Let $a \in \mathcal{A}_p$ denote the agent who arrives at time step p. The next three claims together show that the prices satisfy property 1.

Claim 6.11. *If $a \in V_p \setminus S_p$, or $a \in S_p \setminus R_p$ and $\pi(a) > 0$, then they choose an item that is legal for them in G_p.*

Proof. Since $a \in V_p \setminus R_p$, the matching M_p covers a and hence x_a is non-empty. Let $i \in \mathcal{I}_p$ be an arbitrary item distinct from x_a. If ai is not tight, then the agent strictly prefers x_a over i by Claim 6.7. If ai is tight but i is in a different strongly connected component than a, then the index of the component of i is strictly larger than that of a. Furthermore, by the definition of the set S_p, either $a, i \in S_p$, $a \notin S_p$ and $i \in S_p$, or $a, i \notin S_p$. These together imply that $b_p(x_a) - \pi(x_a) < b_p(i) - \pi(i)$, hence the agent strictly prefers x_a over i by Claim 6.9. Finally, if ai is tight and i is in the same strongly connected component as a, then ai is legal by Claim 6.6. The utility of a is non-negative for such items by the choice of δ and the assumptions on a, hence the claim follows. $\square$

Claim 6.12. *If $a \in S_p \setminus R_p$ and $\pi(a) = 0$, then she takes no item at all and there exists a maximum weight matching in G_p that does not cover a.*

Proof. Let $i \in \mathcal{I}_p$ be an arbitrary item. If ai is not tight, then the utility of a for i is negative by Claim 6.8. If ai is tight, then $i \in S_p$ by the definition of the set S_p. This implies $b_p(i) > \pi(i)$ by the definition of the prices, hence the utility of a for i is negative by Claim 6.9. However, in such cases there exists a directed path P to a from an agent $a' \in R_p$. The fact that P consists of tight edges together with $\pi(a) = \pi(a') = 0$ by ($\star$) imply that $M_p \triangle P$ is also a maximum weight matching in G_p. $\square$

Claim 6.13. *If $a \in R_p$, then they take no item at all.*

Proof. By assumption ($\star$), we have $\pi(a) = 0$. Let $i \in \mathcal{I}_p$ be an arbitrary item. If ai is not tight, then the utility of a for i is negative by Claim 6.8. If ai is tight, then $i \in S_p$ by the definition of the set S_p. This implies $b_p(i) > \pi(i)$ by the definition of the prices, hence the utility of a for i is negative by Claim 6.9. $\square$

The matching M_p, and implicitly the set R_p, is updated as follows. If $a \in V_p \setminus S_p$, or $a \in S_p \setminus R_p$ and $\pi(a) > 0$, then a takes an item i from her strongly connected component. If $ai \in M_p$, then set

$M_{p+1} := M_p \setminus \{ai\}$. Otherwise, let C be a directed cycle of D_p containing the arc ai, and set $M_{p+1} := (M_p \triangle C) \setminus \{ai\}$. In this case, we have $R_{p+1} = R_p \setminus \{a\}$. If $a \in S_p$ and $\pi(a) = 0$, then consider the directed path P to a from an agent $a' \in R_p$, and set $M_{p+1} := (M_p \triangle P) \setminus \{ai\}$, implying $R_{p+1} = R_p \setminus \{a'\}$. Finally, if $a \in R_p$, then $M_{p+1} := M_p$, hence $R_{p+1} = R_p$. These show that $(\star)$ remains valid for time step $p+1$ as well.

We now verify that the pricing scheme satisfies property 2, which is done by the following statement.

Claim 6.14. *The price of any item does not increase over time.*

Proof. At each phase of the algorithm, the price of an item i is obtained by shifting its original weighted covering value. Though the structure of the directed graph D_p and therefore the index of the strongly connected component containing i might change from phase to phase, the choice of ε ensures that $\pi(i) + \delta/2^p > \pi(i) + \delta/2^{p+1} + n\varepsilon$ and $\pi(i) - \delta(1 - 1/2^p) > \pi(i) - \delta(1 - 1/2^{p+1}) + n\varepsilon$. Hence, in order to verify the claim, it suffices to show that $S_{p+1} \subseteq S_p$.

Note that M_{p+1} is chosen in such a way that no arc of D_{p+1} leaves the set $S_p \cap \mathcal{I}_{p+1}$. Indeed, D_{p+1} is obtained from D_p by possibly reorienting a directed cycle or a directed path that lies completely in S_p, and then deleting an agent and possibly an item. These steps do not result in a directed arc leaving $S_p \cap \mathcal{I}_{p+1}$, implying $S_{p+1} \subseteq S_p$. $\square$

Finally, we show that the we get an ex-post envy-free solution.

Claim 6.15. *The resulting allocation is ex-post envy-free.*

Proof. Let a be an arbitrary agent, and first consider the case when she selects an item i upon arrival, that is,

$$u_{a,\sigma(a)}(i) = \max\left\{u_{a,\sigma(a)}(j) \mid j \in \mathcal{I}_{\sigma(a)}\right\} \geqslant 0.$$

Suppose that $u_{a,\sigma(a)}(i) < \max\{u_{a,p}(j) \mid p \in [\sigma(a)], j \in \mathcal{I}_p\}$, and let $u_{a,\sigma(a)}(i) = \max\left\{u_{a,\sigma(a)}(j) \mid j \in \mathcal{I}_{\sigma(a)}\right\} \geqslant 0' \in [\sigma(a)]$ and $i' \in \mathcal{P}_{p'}$ be a pair of time step and item attaining the maximum. Note that $p' < \sigma(a)$.

By Claim 6.11-6.13, ai is legal in $G_{\sigma(a)}$ and ai' is legal in $G_{p)}$ hence both ai and ai' are tight. We claim that $a, i' \notin S_{p'}$. Indeed, if $i' \in S_{p'}$, then Claim 6.14 and the choices of δ and ε imply $u_{a,p'}(i') < u_{a,p'+1}(i) \leqslant u_{a,\sigma(a)}(i)$, contradicting the choice of p' and i'. Since ai' is an arc of $D_{p'}$ and $i' \notin S_{p'}$, $a \notin S_{p'}$ also follows by the definition of $S_{p'}$. On the other hand, we claim that $i \in S_{\sigma(a)}$. Indeed, if $i \notin S_{\sigma(a)}$, then Claim 6.14 and the choices of the values of δ and ε imply $u_{a,p'}(i') < u_{a,\sigma(a)}(i)$, contradicting the choice of p' and i'. Finally, observe that $S_{\sigma(a)} \subseteq S_{p'}$ implies $a \notin S_{\sigma(a)}$. Concluding the above, a is covered in $M_{\sigma(a)}$ and its matching-pair is not in $S_{\sigma(a)}$ by construction, that is, $x_a^{\sigma(a)} \notin S_{\sigma(a)}$. This means that $u_{a,\sigma(a)}(x_a^{\sigma(a)}) > u_{a,\sigma(a)}(i)$, contradicting the assumption that a chooses i upon arrival.

Now assume that agent a selects no items upon arrival. By Claim 6.12 and Claim 6.13, this can only occur when $\pi(a) = 0$ and a belongs to $S_{\sigma(a)}$. As deduced in proof of Claim 6.14, $S_p \subseteq S_{p+1}$ and thus a belongs to S_p for $p \in [\sigma(a)]$. As deduced in the proofs of Claim 6.12 and Claim 6.13, the utility of agent a for all items is negative in each time step from 1 to $\sigma(a)$. This concludes the proof of the claim. □

By Claim 6.11-6.13, if the next agent is covered by the matching M_p then they either choose an item that is legal for them, or M_{p+1} is also a maximum weight matching of G_p. Otherwise, they do not take any of the items. This implies that the resulting allocation corresponds to a maximum weight matching of G and hence maximizes social welfare. By Claim 6.14, the prices do not increase over time. By Claim 6.15, this implies that the solution is ex-post envy-free, concluding the proof of the theorem. □

6.4.2 *Ex-ante envy-free pricing*

To give an algorithm for the ex-ante case, we adapt the proof of Theorem 6.10. However, to ensure that the final prices and allocation form an ex-ante envy-free solution, prices have to be updated differently.

Theorem 6.16. *There exists a welfare-maximizing dynamic pricing scheme for the unit-demand ex-ante envy-free pricing problem even if ties are broken by the agents. Furthermore, the optimal prices can be determined in polynomial time.*

Proof. Similarly to the ex-post case, we maintain a maximum weight matching M_p in the graph G_p of remaining agents and items, and denote by R_p the set of agents not covered by the matching M_p. Initially, $\pi(a) = 0$ for $a \in R_1$ by Lemma 6.3(b), and we will maintain this property for each $a \in R_p$ throughout. We describe a general phase $p \in \tau$ of the pricing process. Assume that

$$(\star\star) \qquad \pi(a) = 0 \text{ for } a \in R_p.$$

We define the set

$$S_p := \{v \in V_p \mid \text{there exists a directed path in } D_p \text{ from } v \text{ to an agent } a \in \mathcal{A}_p \setminus R_p \text{ with } \pi(a) = 0\},$$

see Figure 6.1 for an example. Remember, that $C_1^p, \ldots, C_{q_p}^p$ are indexed according to a topological ordering. Note that either $C_j^p \subseteq S_p$ or $C_j^p \cap S_p = \emptyset$ for each strongly connected component C_j^p of D_p. The prices are then updated as follows:

$$b_p(i) = \begin{cases} \pi(i) - \delta/2^p + j\varepsilon & \text{if } i \in C_j^p \text{ such that } C_j^p \subseteq S_p, \\ \pi(i) + \delta(1 - 1/2^p) + j\varepsilon & \text{if } i \in C_j^p \text{ such that } C_j^p \cap S_p = \emptyset. \end{cases}$$

By the choice of δ and ε, the prices are non-negative. Let $a \in \mathcal{A}_p$ denote the agent who arrives at time step p. The next three claims together show that the prices satisfy property 1.

Claim 6.17. *If $a \in V_p \setminus R_p$, then they choose an item that is legal for her in G_p.*

Proof. Since $a \in V_p \setminus R_p$, the matching M_p covers a and hence x_a is non-empty. Let $i \in \mathcal{I}_p$ be an arbitrary item distinct from x_a. If ai is not tight, then the agent strictly prefers x_a over i by Claim 6.7. If ai is tight but i is in a different strongly connected component than a, then the index of the component of i is strictly larger than that of a. Furthermore, by the definition of the set S_p, either $a, i \in S_p$, $a \in S_p$ and $i \notin S_p$, or $a, i \notin S_p$. These together imply that $b_p(x_a) - \pi(x_a) < b_p(i) - \pi(i)$, hence the agent strictly prefers x_a over i by Claim 6.9. Finally, if ai is tight and i is in the same strongly connected component as a, then ai is legal by Claim 6.6. The utility of a is non-negative for such items by the choice of δ, hence the claim follows. ☐

Claim 6.18. *If $a \in R_p \setminus S_p$, then she takes no item at all.*

Proof. By assumption ($\star\star$), we have $\pi(a) = 0$. Let $i \in \mathcal{I}_p$ be an arbitrary item. If ai is not tight, then the utility of a for i is negative by Claim 6.8. If ai is tight, then $i \notin S_p$ by the definition of the set S_p. This implies $b_p(i) > \pi(i)$ by the definition of the prices, hence the utility of a for i is negative by Claim 6.9. ☐

Claim 6.19. *If $a \in R_p \cap S_p$, then she either chooses an item that is legal for her in G_p, or takes no item at all.*

Proof. By assumption ($\star\star$), we have $\pi(a) = 0$. Let $i \in \mathcal{I}_p$ be an arbitrary item. If ai is not tight, then the utility of a for i is negative by Claim 6.8. If ai is tight but $i \notin S_p$, then $b_p(i) > \pi(i)$ by the definition of the prices, hence the utility of a for i is negative by Claim 6.9. If ai is tight and $i \in S_p$, then there exists a directed path P in D_p from a to an agent a' in D_p which is covered by M_p and $\pi(a') = 0$. The fact that P consists of tight edges together with $\pi(a) = \pi(a') = 0$ imply that $M_p \triangle P$ is also a maximum weight matching in G_p, therefore i is legal for a. ☐

The matching M_p, and implicitly the set R_p, is updated as follows. If $a \in V_p \setminus R_p$, then a takes an item i from her strongly connected component. If $ai \in M_t$, then set $M_{p+1} := M_p \setminus \{ai\}$. Otherwise, let C be a directed cycle of D_p containing the arc ai, and set $M_{p+1} := (M_p \triangle C) \setminus \{ai\}$. In this case, we have $R_{p+1} = R_p$. If $a \in R_p \setminus S_p$ or $a \in R_p \cap S_p$ but a takes no item, then set $M_{p+1} := M_p$, implying $R_{p+1} = R_p \setminus \{a\}$. Finally, if $a \in R_p \cap S_p$ and a takes an item i, then consider the directed path P from a to an agent a' which is covered by M_p and $\pi(a') = 0$, and set $M_{p+1} := (M_p \triangle P) \setminus \{ai\}$. Since we have $R_{p+1} = R_p \setminus \{a\} \cup \{a'\}$ and $\pi(a') = 0$, each agent in R_{p+1} has π value 0. These show that ($\star\star$) remains valid for time step $p + 1$ as well.

It remains to verify that the pricing scheme satisfies property 2, which is done by the following statement.

Claim 6.20. *The price of any item does not decrease over time.*

Proof. At each phase of the algorithm, the price of an item i is obtained by shifting its original weighted covering value. Though the structure of the directed graph D_p and therefore the index of the strongly connected component containing i might change from phase to phase, the choice of ε ensures that $\pi(i) - \delta/2^p + n\varepsilon < \pi(i) - \delta/2^{p+1}$ and $\pi(i) + \delta(1 - 1/2^p) + n\varepsilon < \pi(i) + \delta(1 - 1/2^{p+1})$. Hence, in order to verify the claim, it suffices to show that $S_{p+1} \subseteq S_p$.

Note that M_{p+1} is chosen in such a way that no arc of D_{p+1} enters the set $S_p \cap \mathcal{I}_{p+1}$. Indeed, D_{p+1} is obtained from D_p by possibly reorienting a directed cycle or a directed path that lies completely in S_p, and then deleting an agent and possibly an item. These steps do not result in a directed arc entering $S_p \cap \mathcal{I}_{p+1}$, implying $S_{p+1} \subseteq S_p$. □

By Claim 6.17-6.19, if the next agent is covered by the matching M_p then she chooses an item that is legal for her. Otherwise, she either chooses an item that is legal for her, or does not take any of the items. This implies that the resulting allocation corresponds to a maximum weight matching of G and hence maximizes social welfare. By Claim 6.20, the prices do not decrease over time. This implies that the solution is ex-ante envy-free, concluding the proof of the theorem. □

6.5 MAXIMIZING THE REVENUE

When it comes to revenue maximization in the static setting, the problem is not only hard to solve but also to approximate within a reasonable factor, and the difficulty stems from the lack of strong upper bounds. Clearly, the maximum weight of a matching is an upper bound on the total revenue achievable through pricing mechanisms, but the gap between optimal revenue and maximum weight of a matching may be $\Omega(\log(n))$. Indeed, consider a market with items $\mathcal{I} = \{i_1, \ldots, i_n\}$ and agents $\mathcal{A} = \{a_1, \ldots, a_n\}$. Let the valuations be defined as $v_{a_j}(i_k) := 1/j$ for $1 \leqslant j \leqslant n$ and $j \leqslant k \leqslant n$ and 0 otherwise, see Figure 6.2 for an illustration.

Then there is a unique maximum weight matching between agents and items that consists of the edges $i_j a_j$ for $1 \leqslant j \leqslant n$ with total weight $\sum_{j=1}^{n} 1/j$. For any pair of envy-free static pricing and allocation, if an agent a_j receives an item i_k at some price $b(i_k)$, then the price of all the other items must be at least $b(i_k)$ to ensure that agent a_j is not envious. On the other hand, the price $b(i_k)$ cannot be greater than $1/j$ as otherwise the utility of agent a_j for item i_k is negative. These observations together imply that the price of each item sold is at most $1/j$ where j is the largest index for which agent a_j receives an item. Hence the total revenue is at most $j \cdot 1/j = 1$, leading to an $\Omega(\log(n))$ gap as stated.

In what follows, we turn our attention to the revenue maximization problem in a dynamic environment with fairness constraints.

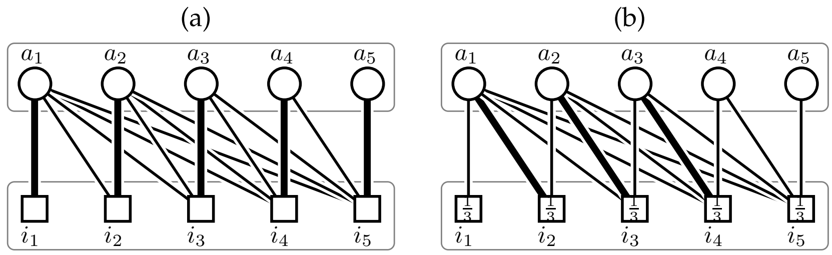


Figure 6.2: Example with an $\Omega(\log(n))$ gap between the optimal revenue and maximum weight of a matching in the case of envy-free static pricing. In (a) the weight of an edge $a_j i_k$ is $1/j$, while missing edges have weight 0. Thick edges correspond to the unique maximum weight matching of total weight $\sum_{j=1}^{n} 1/j$. In (b) values on items and thick edges correspond to a pair of envy-free pricing and allocation, respectively. The revenue is always less than or equal to 1.

6.5.1 *Hardness results*

When the solution is required to be strongly envy-free, then the dynamic setting does not make a difference compared to the static one, as shown by the following theorem.

Theorem 6.21. *Maximizing the revenue in the unit-demand strongly envy-free dynamic pricing problem is APX-hard, even if the agents' ordering is chosen and ties are broken by the seller.*

Proof. Let σ be an ordering of the agents, $b_1, \ldots, b_n$ be prices, and $\mathbf{X}$ be an allocation that maximizes the revenue. Recall that x_a denotes the item received by agent $a \in \mathcal{A}$ if exists, otherwise x_a is the empty set. We show that there exist static prices and an envy-free allocation resulting in the same revenue. Note that the reverse always holds, that is, any envy-free allocation with respect a static pricing can be seen as a strongly envy-free solution in the dynamic setting. Since determining the static prices is known to be APX-hard [46], this proves the theorem.

We define a static pricing as follows: for each agent $a \in \mathcal{A}$ with $x_a \neq \emptyset$, set $b(x_a) := b_{\sigma(a)}(x_a)$, that is, we define the price of an allocated item to be the price at which it was sold. For the remaining items, set the price to $+\infty$. We claim that the allocation $\mathbf{X}$ is envy-free with respect to the static pricing p, and hence has the same revenue as the dynamic solution. Indeed, this follows from the definition of strong envy-freeness, as $u_{a,\sigma(a)}(x_a) = \max\{u_{a,p}(i) \mid p \in P_a, i \in \mathcal{I}_p\}$ and $b(x_a) = b_{\sigma(a)}(x_a)$ imply $x_a \in \arg\max\{v_a(i) - b(i) \mid i \in \mathcal{I}\}$ for each $a \in \mathcal{A}$. $\square$

Unfortunately, the problem remains hard for weaker notions of envy-freeness. Our proof follows the main idea of the proof of Guruswami et al. [46] for the APX-hardness of revenue maximization.

Theorem 6.22. *Maximizing the revenue in the unit-demand ex-post and ex-ante envy-free dynamic pricing problems are APX-hard, even if the agents' ordering is known in advance and ties are broken by the seller.*

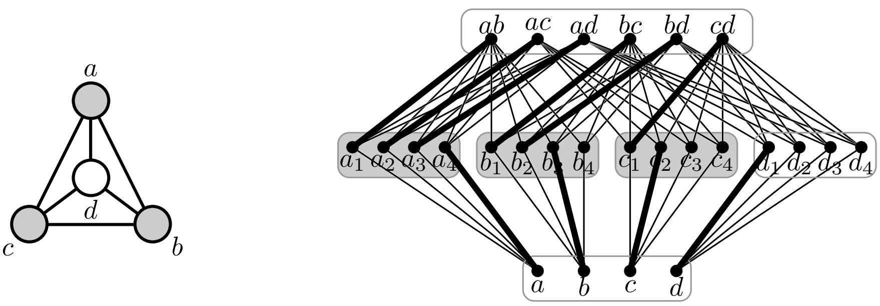


Figure 6.3: A 3-regular graph G with $n = 4$ and $m = 6$, where grey vertices form a minimum vertex-cover of size $k = 3$. In the corresponding pricing instance, edges incident to vertex-agents and edge-agents have weights 2 and 1, respectively. When edge-agents arrive first, then setting the prices to 1 on grey elements, 2 otherwise, and allocating the items according to thick edges results in an ex-post envy-free solution with revenue $11 = m + 2n - k$.

Proof. The proof is by reduction from Vertex Cover in 3-regular graphs. Vertex Cover asks, given a 3-regular graph $G = (V, E)$, for a minimum number of vertices that include at least one endpoint of every edge of the graph. This problem was shown to be APX-hard by Fleischner, Sabidussi, and Sarvanov [42].

Let $G = (V, E)$ be a 3-regular graph with n vertices and m edges. We construct a pricing instance with item set $\mathcal{I}$ and agent set $\mathcal{A}$ consisting of $4n$ items and $m + n$ agents, respectively. For each vertex $z \in V$, we add four vertex-items z_1, z_2, z_3, z_4 to $\mathcal{I}$ and one vertex-agent to $\mathcal{A}$ that, by abuse of notation, we also denote by z. The valuation of the agent is then defined as $v_z(z_i) = 2$ for $i \in [4]$ and 0 for any other item. Furthermore, for each edge $e = zw \in E$, we add an edge-agent e to $\mathcal{A}$ with valuation $v_e(z_i) = v_e(w_i) = 1$ for $i \in [4]$ and 0 for any other item; for an example, see Figure 6.3.

We first consider the ex-post case. Assume that the ordering of the agents is such that edge-agents arrive first, followed by vertex-agents. We claim that for such an ordering, there exists an ex-post envy-free pricing scheme and allocation that results in a total revenue of $m + 2n - k$ if and only if there exists a vertex cover of size k in G. Since $m = 3n/2$ and the minimum vertex cover has size at least $m/3 = n/2$, a constant factor gap in the size of a vertex cover translates into a constant factor gap in the optimal profit for the pricing instance, which yields the desired APX-hardness result.

To see the 'if' direction, let $C \subseteq V$ be a vertex cover of G of size k. For each vertex $z \in V$, let $b_p(z_i) \coloneqq 1$ if $z \in C$ and $b_p(z_i) \coloneqq 2$ otherwise for $i \in \{1, \ldots, 4\}$, which as before we short as $i \in [4]$, and $p \in \tau$ – note that the prices do not change over time, implying that the final solution is ex-post envy-free. According to the ordering, edge-agents arrive first, and each of them takes one of the vertex-items that correspond to one of its endpoints that lies in C for a price of 1. Then vertex-agents arrive, and take a copy of the vertex-items corresponding to them for a price of 2. Note that, since the graph is 3-regular and four vertex-items were added for each vertex, each agent receives an item, and hence the total revenue is $m + 2n - k$.

To see the 'only if' direction, consider dynamic prices $b_1, \dots, b_n$ and an ex-post envy-free allocation $\mathbf{X}$ that maximizes the revenue for the ordering considered. It is not difficult to check that the pricing vectors can be assumed to take values 1 and 2 only. Let $e = zw$ be the first edge-agent, if exists, who does not get any item. That is, all the remaining vertex-items from $z_1, \dots, z_4, w_1, \dots, w_4$ are priced at 2 upon the arrival of e. If we reduce the price of one of these items, say z_i, then we have to do the same modification for all the remaining vertex-items corresponding to z and for all the remaining time steps to ensure ex-post envy-freeness from the point of view of vertex-agent z. This way, we lose a revenue of 1 coming from vertex-agent z, but we gain this back by making a profit of 1 from edge-agent e. By this observation, we may assume that for each vertex $z \in V$ and time step $p \in [n+m]$, either $b_p(z_i) = 1$ for $i \in [4]$ or $b_p(z_i) = 2$ for $i \in [4]$, and that vertices belonging to the former class form a vertex cover of G. This implies that the revenue is at most $m + 2n - k$.

If the ordering of the agents is such that vertex-agents arrive first, followed by edge-agents, then a similar argument shows the hardness of the ex-ante case. □

6.5.2 *Algorithms*

In the previous section, we showed that if the seller has no control over the order in which agents arrive, then even the seemingly more flexible framework of dynamic pricing is not enough for maximizing the revenue in the ex-post and ex-ante envy-free settings. On the positive side, if the ordering can be chosen by the seller, then an optimal pricing scheme can be determined efficiently. In what follows, we give polynomial-time algorithms for determining an ordering of the agents together with the price vectors so that the final allocation is ex-post or ex-ante envy-free and maximizes the revenue irrespective of how ties are broken by the agents. In both cases, we compare the solution to the maximum weight of a matching in the corresponding edge-weighted bipartite graph, which is clearly an upper bound for the revenue.

Note that, in the agent-chooses case, an agent can decide not to take an item with utility 0 for them. For keeping the description of the algorithms simple, we assume that in such cases the agent decides to take the item. In the general case then one can decrease the prices of the items in each step by a small $\varepsilon > 0$, thus obtaining a revenue arbitrarily close to the maximum weight of a matching. First we consider the ex-post case.

Theorem 6.23. *If the ordering of the agents can be chosen by the seller, then there exists a revenue-maximizing dynamic pricing scheme for the unit-demand ex-post envy-free pricing problem even if ties are broken by the agents. Furthermore, the optimal ordering and prices can be determined in polynomial time.*

Proof. Consider the edge-weighted bipartite graph $G = (\mathcal{I}, \mathcal{A}; E)$ representing the market, where the weight of an edge ai is $v_a(i)$ for $i \in \mathcal{I}$,

$a \in \mathcal{A}$. By Lemma 6.4, we may assume that all the items are covered by every maximum weight matching of G. Let $M \subseteq E$ be an arbitrary maximum weight matching, and for each agent a, let x_a denote the item to which a is matched in M if such an item exists, otherwise define x_a to be the empty set. Furthermore, take a weighted covering π provided by Lemma 6.3. Note that M consists of tight edges.

We define the ordering σ of the agents as follows: agents not covered by M arrive first, and then the remaining agents arrive in a decreasing order according to their π values, where ties are broken arbitrarily.

Now we describe how to set the prices at each time step. Define the prices to be $+\infty$ until all the agents not covered by M have left. Then, at each time step, consider the next agent a and set the price of all remaining items to $+\infty$ except for x_a, for which we set the price to $v_a(x_a)$. Clearly, the agent will take the item x_a, hence the resulting allocation corresponds to M and has total revenue equal to the maximum weight of a matching.

It remains to verify that the pricing and the allocation thus obtained provide an ex-post envy-free solution. To see this, consider the arrival of an agent $a \in \mathcal{A}$. As all the remaining items has been priced at $+\infty$ so far except for x_a which is priced at $v_a(x_a)$, it is enough to show that a does not envy an item that was taken before her arrival. Those items were also priced at $+\infty$ except for the time step when they were taken by the corresponding agent. So let a' be an agent who arrived before a and took the item $x_{a'}$, that is, $x_{a'} \neq \emptyset$. Since π is a weighted covering, M consists of tight edges, and $\pi(a') \geqslant \pi(a)$, we get

$$\begin{aligned}
u_{a,\sigma(a')}(x_{a'}) &= v_a(x_{a'}) - b_{\sigma(a')}(x_{a'}) \\
&= v_a(x_{a'}) - v_{a'}(x_{a'}) \\
&\leqslant (\pi(a) + \pi(x_{a'})) - (\pi(a') + \pi(x_{a'})) \\
&= \pi(a) - \pi(a') \\
&\leqslant 0 \\
&= v_a(x_a) - v_a(x_a) \\
&= u_{a,\sigma(a)}(x_a),
\end{aligned}$$

which means that agent a does not envy the item $x_{a'}$. □

A similar proof works for the ex-ante setting as well. However, the proof is is slightly more complicated as maintaining ex-ante envy-freeness requires a careful choice of prices. As a result, the revenue of the final allocation is not exactly the maximum weight of a matching in the associated bipartite graph, but can be arbitrarily close to that. For simplicity, we still refer to such a pricing as 'optimal' in the statement of the theorem.

Theorem 6.24. *If the ordering of the agents can be chosen by the seller, then there exists a revenue-maximizing dynamic pricing scheme for the unit-demand ex-ante envy-free pricing problem even if ties are broken by the agents. Furthermore, the optimal ordering and prices can be determined in polynomial time.*

Proof. Consider the edge-weighted bipartite graph $G = (\mathcal{I}, \mathcal{A}; E)$, a maximum weight matching M, x_a for $a \in \mathcal{A}$, and weighted covering π as in the proof of Theorem 6.23. Let

$$0 < \delta < \min\{\min\{\pi(a) + \pi(i) - v_a(i) \mid ia \in E \text{ is not tight}\}, \min\{\pi(i) \mid i \in \mathcal{I}\}\} \ .$$

Note that, by Lemma 6.3 and the assumption that each item is covered by every maximum weight matching of G, such a δ exists. Furthermore, let $0 < \varepsilon < \delta/2^n$.

We define the ordering σ of the agents as follows: agents covered by M arrive first in an increasing order according to their π values, where ties are broken arbitrarily, followed by the remaining agents.

Now we describe how to set the prices at each time step. If an agent a arrives for which $x_a \neq \emptyset$, then for each item $i \in \mathcal{I}$ set its price to $\pi(a) + \pi(i) - \delta/2^{\sigma(a)} + \varepsilon$ except for x_a, for which we set the price to $\pi(a) + \pi(x_a) - \delta/2^{\sigma(a)}$. If $x_a = \emptyset$, then by the choice of the ordering there is no remaining item. By the definition of the ordering and the values δ and ε, the prices remain non-negative and do not decrease over time, hence the resulting allocation is automatically ex-ante envy-free.

It suffices to show that each agent a chooses x_a upon arrival. Indeed, if this holds, then that results in a profit of $\pi(a) + \pi(x_a) - \delta/2^{\sigma(a)} \geqslant v_a(x_a) - \delta$, where we used the fact that the edge $x_a a$ is tight by Lemma 6.3(a). By choosing δ small enough, the total revenue of the final allocation can be arbitrarily close to the weight of M. Consider any remaining item i distinct from x_a. As π is a weighted covering and $x_a a$ is tight, we get

$$\begin{aligned} u_{a,\sigma(a)}(i) &= v_a(i) - p_{\sigma(a)}(i) \\ &= v_a(i) - (\pi(a) + \pi(i) - \delta/2^{\sigma(a)} + \varepsilon) < \delta/2^{\sigma(a)} \\ &= v_a(x_a) - (\pi(a) + \pi(x_a) - \delta/2^{\sigma(a)}) = u_{a,\sigma(a)}(x_a) \ . \end{aligned}$$

This means that x_a is the unique maximizer of the utility of a at time step $\sigma(a)$ and has positive utility for a, hence agent a takes x_a as stated. □

Finally, we settle the existence of weakly envy-free solutions when the ordering of the agents is fixed but known in advance.

Theorem 6.25. *If the ordering of the agents is known in advance, then there exists a revenue-maximizing dynamic pricing scheme for the unit-demand weakly envy-free pricing problem even if ties are broken by the agents. Furthermore, the optimal prices can be determined in polynomial time.*

Proof. Let σ denote the fixed ordering of the agents. Define an edge-weighted bipartite graph $G = (\mathcal{I}, \mathcal{A}; E)$, maximum weight matching $M \subseteq E$, and x_a for $a \in \mathcal{A}$ as in the proof of Theorem 6.23. At the arrival of agent a, set the price of all remaining items to $+\infty$ except for x_a, for which we set the price to $v_a(x_a)$. The agent clearly takes x_a

at the maximum possible price, hence the resulting allocation and pricing are optimal. □

6.6 CONCLUSIONS

We studied the existence of optimal dynamic prices under fairness constraints in unit-demand markets. We proposed four possible notions of envy-freeness depending on the time period over which agents compare themselves to others, and settled the existence of optimal dynamic prices in various settings.

We would now like to mention a few open problems. While we concentrated on social welfare and revenue maximization problems, a natural question is to consider alternative objective functions such as the average or the max-min social welfare and revenue. Besides being interesting on their own, such functions may be used to overcome the difficulties mentioned in Remark 6.1. A recent line of research investigated the problem of balancing fairness and efficiency in markets, see e.g. [43]. It would be interesting to see how dynamic envy-free pricing behaves under such objective functions.

The inapproximability results showed by Briest [19] and Chalermsook et al. [25] suggest that the $O(\log(n))$-approximation of Guruswami et al. [46] for static envy-free pricing is tight. It is unclear if the results can be generalised to the dynamic case, which would then imply that the same $O(\log(n))$ approximation is tight for dynamic envy-free pricing when maximizing revenue for each variant found in a red cell in Table 6.1.

Finally, the complexity of weak envy-free revenue maximization with unspecified order remains open. This variant is of special interest, since it naturally connects revenue maximization with the recent popular strain of research on dynamic pricing schemes.

7 GENERAL OVERVIEW AND CONCLUSIONS

In this chapter, we go back to our Energy Trading problem and we take a brief moment to see what happens when we switch to a setting where there are multiple prosumers. So far indeed, we have always performed analysis on single prosumers while assuming a general microgrid built around them. The price fluctuations we have assumed where taken as a justification of the peers within the microgrid, but we have never really tackled the topic.

Our idea is then now to actually connect prosumers in a microgrid and to almost create a market. We would like to do so by putting together the pieces collected so far. Unfortunately, we succeeded to do so only in a simplified setting, as we will be soon explaining. We still decided to share these ideas, since we believe that it can be interesting for the reader and a future starting point for further developments.

As probably does not come as a surprise, we are not the first to try to connect prosumers into a microgrid. First of all, these concepts already exist in real life, as the real case examples in the work of Ustun et al. [84] show. From a research point of view instead, the ways of analyzing them are multiple. And this is exactly the type of multiplicity that I was referring to in the Prologue. On one hand, this setting can be easily embedded in a Game theoretical framework. Prosumers are players, who can act by themselves, cooperate, or participate in auctions. Taking a step further from the Game Theory field, we still have all the mechanism design settings, where players can act as buyers and sellers in some designed market.

In some sense, there is still some freedom in the design of microgrids, meaning that it is possible to structure them ensuring that a list of desired properties are fulfilled. We now illustrate our ideal microgrid and to do so we have to extend a bit the notation used so far.

Instead of considering a single prosumer, we do now have a set $\mathcal{M}$ of m prosumers, whom are connected within the microgrid. The distinctive parameters for prosumers already used in the previous chapter are repeated and now personalized. More specifically, the value d_p^a it is the demand at time p of player a. In the same way, e_p^a is the energy produced by player a and C^a is the size of their battery. As in Chapter 2, Chapter 3 and Chapter 5, we still assume that these quantities are known to each prosumer in advance. Then, as microgrids are always connected with a traditional retailer which can provide energy, we add the symbol R for it and we use the vector r for the prices at which the energy is sold.

Let us now switch to the properties that we are wishing for our microgrid. First of all, we recall that the main idea of organizing prosumers into a microgrid is actually to increase the local exchanges of energy and in particular to sell the individual surplus of energy within the microgrid. Therefore, our first design goal is to maximize the amount of renewable energy which is used in the microgrid. Of course we do not have power to increase the amount of sun which radiates on the PVs if the setting is given, but what we can do is to make sure that the exchanges of energy are maximized. This is in other words equivalent to minimize the total amount of energy that the microgrid buys from the retailer R.

We can easily formalize this problem by extending the (2.1)-(2.9) of Chapter 3. In particular we introduce the variables w_p^a for the amount of energy that prosumer a buys from the grid at time p and q_p^a for the one sold. At the same time we add a variable $v_p^{+;a,j}$ for the amount of energy prosumer a receives from prosumer j at time p and $v_p^{-;a,j}$ the amount they give to prosumer j at time p. Note that in this moment we do not care for the price at which these exchanges happen, as indeed we are focusing exclusively on our primary design goal, namely maximize the independence of the microgrid. We call this problem the PROSUMER EXCHANGE and we formalize it as follows.

$$\min \sum_p \sum_a (w_p^a - q_p^a) \tag{7.1}$$

$$\text{s.t.} \quad w_p^a - q_p^a + \sum_{j \neq a} v_p^{+;a,j} - \sum_{j \neq a} v_p^{-;a,j} - z_p^a = d_p^a - e_p^a, \quad p \in \tau, a \in \mathcal{M}, \tag{7.2}$$

$$v_p^{+;a,j} = v_p^{-;j,a}, \quad p \in \tau, a \in \mathcal{M}, j \in \mathcal{M}, \tag{7.3}$$

$$SOC_p^a = SOC_{p-1}^a + z_p^a, \quad p \in \tau, a \in \mathcal{M}, \tag{7.4}$$

$$SOC_0^a = soc_0^a, \quad a \in \mathcal{M}, \tag{7.5}$$

$$SOC_p^a \leqslant C^a, \quad p \in \tau, a \in \mathcal{M}, \tag{7.6}$$

$$x_p^a, y_p^a, SOC_p^a \in \mathbb{N}, \quad p \in \tau, a \in \mathcal{M}, \tag{7.7}$$

$$z_p^a \in \mathbb{Z}, \quad p \in \tau, a \in \mathcal{M} \ . \tag{7.8}$$

To ensure that our primary goal–maximizing self-consumption–is satisfied, we take advantage from another configuration used in the Energy Trading field: the Energy Community (EC). In particular, ECs are groups of users who typically share an energy renewable asset such as a photovoltaic system and a storage unit (e.g., a battery). Unlike the microgrids analyzed so far, where each prosumer was able to self-produce energy and store it independently, ECs are characterized by the collective ownership and use of these resources. As a result, in microgrids prosumers can trade with each other, while in ECs users are aggregated together and the community as a whole interacts with external entities or the energy market. ECs are quite common nowadays, as lots of newly built building are provided with these type of shared technologies. They are formally defined by the European Parliament in Directive 2018/2001 [72], and their implementation in everyday life is highly encouraged, as well as subsidized. In ECs the shared energy produced is used to meet the demand of members and if this is not enough the energy previously stored can be used. In case also this energy is not enough to fulfill the demand, then energy is bought from the retailer as it is normally done. Then, by definition itself of EC, it is clear that the energy bought from the retailer is minimum.

We want to transform our microgrid into a EC, and we do so by aggregating the demands d_p^a and energy e_p^a and battery capacities C^a of all the prosumers. We call this a Super Prosumer (SP) and we describe here the energy profile:

$$d_t^{SP} = \sum_{a \in \mathcal{M}} d_p^a, \tag{7.9}$$

$$e_t^{SP} = \sum_{a \in \mathcal{M}} e_p^a, \tag{7.10}$$

$$C^{SP} = \sum_{a \in \mathcal{M}} C^a \quad . \tag{7.11}$$

Consider now the problem depicted in Chapter 3 with $\alpha_p = \beta_p = \infty$ for every $p = 1, \dots, n$ for the SP and assume that $r_p = r$. We call this prosumer the super prosumer (SP) and we report its scheduling problem here.

$$\min \sum_p (r_p x_p^{SP} - s_p y_p^{SP}) \tag{7.12}$$

$$\text{s.t.} \quad x_p^{SP} - y_p^{SP} - z_p^{SP} = d_p^{SP} - e_p^{SP}, \quad p \in \tau, \tag{7.13}$$

$$SOC_p^{SP} = SOC_{p-1}^{SP} + z_p^{SP}, \quad p \in \tau, \tag{7.14}$$

$$SOC_0^{SP} = soc_0^{SP}, \tag{7.15}$$

$$SOC_p^{SP} \leqslant C^{SP}, \quad p \in \tau, \tag{7.16}$$

$$x_p^{SP}, y_p^{SP}, SOC_p^{SP} \in \mathbb{N}, \quad p \in \tau, \tag{7.17}$$

$$z_p^{SP} \in \mathbb{Z}, \quad p \in \tau \quad . \tag{7.18}$$

Notice that since selling is allowed, all the instances are feasible and let $OPT(SP)$ be an optimal solution of this problem. We now show the following two results.

Lemma 7.1. *Given an instance of the* PROSUMER EXCHANGE*, this can be transformed in an instance for the* SUPER PROSUMER *problem.*

Proof. Let us set the parameters of the EC as the parameters of the SP depicted above. Since all the parameters are obtained as transformations of the parameters of PROSUMER EXCHANGE according to Equation 7.9, the result follows immediately. □

Given a microgrid and a PROSUMER EXCHANGE problem, we want to transform it into an instance of SUPER PROSUMER. As we have seen in Chapter 3, it might be the case that in some days it is more convenient to sell more than the simple overflow of the battery, even if this might means that the prosumer has to buy more energy later on. To make sure that this is not the case–as this would be not in line with our primary design goal–we set the prices for selling in the SUPER PROSUMER model to null. Notice that, since selling still remains allowed, this does not make the instance infeasible, which might be the case if we decide to use PEAC-B-U, but instead the selling is discouraged and interpreted as a waste of energy (as already observed in Chapter 3). With this modification, the schedule of the SP is exactly the one maximizing the independence of the microgrid, under the assumption $r_t = r$. It is left then to show that if we find a solution of SUPER PROSUMER we can retrieve a solution of PROSUMER EXCHANGE. To do so, we will use a simple trick. We assume only a random prosumer, say prosumer 1, to be the one with interaction with the retailer, then all the other exchanges happen within prosumers. In order to get a polished algorithm then, similarly to what we do with prosumer 1, we assume that the energy stored is stored in order, meaning it is all in the battery of prosumer 1, and if this is not enough in the one of prosumer 2 and so on. In our algorithm we need to virtually transfer the stored energy several times, and to improve the general readability of it, we present in Algorithm 7.1 a subroutine to perform this allocation. In particular, we remark that to ensure that the balance of energy of prosumers express in Equation 7.3 still holds, every time a prosumer a sells an amount E of energy to another prosumer j, not only the variable $\nu_p^{+;j,a}$ needs to be incremented by the value E, but also the variable z_p^j has to be increased of the same value to ensure the balance of energy still holds. The same happens for SOC_p^j. Even if all these variables are modified, for sake of simplicity we treat the subroutines in the spirit of programming and we assume that their values are accessible to the algorithm subroutine and can be modified. Hence, we omit these variables in the input and output of the algorithms.

Algorithm 7.1 *Move energy to the first $a^\star$ batteries*

Input : A prosumer a and the positive amount of energy E to redistribute from battery $a^\star$.

Output : The same instance with energy allocated only in the first $a^\star$ batteries.

Let $a^\star < a$ be the last prosumer whose battery is still non empty

while $SOC_p^{a^\star} < E$ **do**

- $E = E - SOC_p^{a^\star}$
- $\nu_p^{+;a^\star,a} = \nu_p^{+;a^\star,a} + SOC_p^{a^\star}$
- $\nu_p^{-;a,a^\star} = \nu_p^{-;a,a^\star} + SOC_p^{a^\star}$
- $z_p^a = z_p^a - SOC_p^{a^\star}$
- $SOC_p^{a^\star} = 0$
- $a^\star = a^\star + 1$

$SOC_p^{a^\star} = SOC_p^{a^\star} - E$

$\nu_p^{+;a^\star,a} = E$

$\nu_p^{-;a,a^\star} = E$

$z_p^a = z_p^a - E$

return $a^\star$

As it is probably already clear from the Algorithm 7.1, the high number of variables to be updated when performing an action such as an exchange of energy between two prosumers requires a significant number of updates. For this reason we include here a subroutine, for exchanging the excess of energy to another prosumer. Notice that if the effect of the subroutine is just changing the variables that we do not pass explicitly, our algorithm is not returning any variable. We call it *Give energy to player* and we give its pseudocode here.

Algorithm 7.2 *Give energy to player*

Input : A prosumer a and the positive amount of energy E to give to prosumer j.
Output : The same instance with the exchange performed.

$\nu_p^{-;a,a+1} = \nu_p^{-;a,a+1} + E$
$\nu_p^{+;a+1,a} = \nu_p^{+;a+1,a} + E$

Finally we do the same for the last time of actions needed, namely when the prosumers need to get some of the energy stored to fulfill their demand. They will start getting energy from the last battery with some energy inside and then, if needed, go backwards till the demand is fully satisfied. In the subroutine Algorithm 7.3 we take care of these actions, as well as updating all the related variables.

Algorithm 7.3 *Take energy from battery*

Input : A prosumer a and the positive amount of energy E to be taken starting from the battery of prosumer $a^\star$.
Output : The same instance with the actions performed.

while $SOC_p^{a^\star} < E$ **do**
  $E = E - SOC_p^{a^\star}$
  $\nu_p^{+;a,a^\star} = \nu_p^{+;a,a^\star} + SOC_p^{a^\star}$
  $\nu_p^{-;a^\star,a} = \nu_p^{-;a^\star,a} + SOC_p^{a^\star}$
  $z_p^{a^\star} = z_p^{a^\star} - SOC_p^{a^\star}$
  $SOC_p^{a^\star} = 0$
  $a^\star = a^\star - 1$
$\nu_p^{+;a,a^\star} = \nu_p^{+;a,a^\star} + E$
$\nu_p^{-;a^\star,a} = \nu_p^{-;a^\star,a} + E$
$z_p^{a^\star} = z_p^{a^\star} - E$
return $a^\star$

We now show that from a solution of the SP problem, we can obtain a solution for the whole microgrid.

Lemma 7.2. *Given a solution* (x_p^{SP}, y_p^{SP}) *of the SP problem under the assumption* $r_p = r$ *and* $s_p = 0$ *for every time period* p*, we get a solution of the* Prosumer Exchange *problem where the independence of the grid is maximized.*

Proof. Since the SP solution is the one maximizing the independence of the microgrid, to prove the lemma we just need to show that it is possible to build a solution which buys the same amount of energy from the retailer. In particular, we can do so by applying Algorithm 7.4. We start by observing that the amount of energy bought is indeed the same as the amount bought by the SP, since

$$\sum_{p\in\tau}\sum_{a\in\mathcal{M}}(w_p^a - q_p^a) = \sum_{p\in\tau}(w_p^1 - q_p^1) = \sum_{p\in\tau}(x_p^{SP} - y_p^{SP}) \ .$$

Then, we have to show that the solution obtained by the Algorithm 7.4 is actually feasible, which means that all the demand are satisfied and that the capacity constraint are satisfied at every time period for each battery.

Suppose now that it exists a time period p such that a battery overflows. Since in our algorithm all the excess is sent to the next prosumer, this implies that here we are assuming that the battery of the last prosumer is overflown and all the other batteries are full, namely $a^\star = m$ and $SOC_p^{a^\star} > C_{a^\star}$. From these it follows immediately $\sum_{a\in\mathcal{M}} SOC_p^a > \sum_{a\in\mathcal{M}} C^a$ and we rewrite the term $\sum_{a\in\mathcal{M}} SOC_p^a$ using the relation of (7.4), namely

$$SOC_p^a = SOC_{p-1}^a + z_p^a = SOC_{p-2}^a + z_{p-1}^a + z_p^a = soc_0^a + \sum_{p\in\tau} z_p^a,$$

which holds for every time period and every prosumer, included the super prosumer SP and from which we obtain

$$\sum_{a\in\mathcal{M}} SOC_p^a = \sum_{a\in\mathcal{M}} soc_0^a + \sum_{a\in\mathcal{M}} \sum_{p\in\tau} z_p^a \ . \tag{7.19}$$

We now derive the value of $\sum_{a\in\mathcal{M}} \sum_{p\in\tau} z_p^a$ by summing (7.8) for all the prosumers, namely

$$\begin{aligned}\sum_{a\in\mathcal{M}} z_p^a = \sum_{a\in\mathcal{M}} \left(d_p^{SP} - e_p^{SP} - w_p^a + q_p^a - \sum_{j\neq a} v_p^{+;a,j} + \sum_{j\neq a} v_p^{-;a,j} + z_p^a \right) \\ = d_p^{SP} - e_p^{SP} - x_p^{SP} + y_p^{SP},\end{aligned}$$

which by substituting in Equation 7.19 we obtain

$$\begin{aligned} SOC_p^{SP} =& SOC_0^{SP} + \sum_{p\in\tau} z_p^{SP} = \\ & SOC_0^{SP} + \sum_{p\in\tau} \left(x_p^{SP} + y_p^{SP} - d_p^{SP} + e_p^{SP} \right) = \\ & SOC_0^{SP} + \sum_{p\in\tau} \sum_{a\in\mathcal{M}} z_p^a = \\ & \sum_{a\in\mathcal{M}} SOC_p^a > \sum_{a\in\mathcal{M}} C^a = C^{SP}, \end{aligned} \tag{7.20}$$

where the inequality $SOC_p^{SP} > C^{SP}$ would contradict the feasibility of the SP solution. Suppose now than that one of the batteries run out of energy and some energy is asked there to discharge. Since in our algorithm we start getting energy from battery $a^\star$ and we proceed than back wards once it is empty, so basically once the battery of prosumer $a^\star$ is empty we update $a^\star$ by decreasing its value of 1. Under this scheme, supposing that there is a battery which runs out of battery implies supposing that there exists a time period p such that $SOC_p^1 < 0$, which also implies that $SOC_p^a = 0$ for all the prosumers $a \neq 1$. It follows than that the sum of energy stored is negative and by the relations obtained above in Equation 7.20 we get that there exists a time period p such that

$$\sum_{a \in \mathcal{M}} SOC_p^a = SOC_p^{SP} < 0, \tag{7.21}$$

which is of course not compatible with the feasibility of the SP solution. Lastly we observe that if a net demand $d_p'^a = d_p^a - e_p^a > 0$ is not satisfied, then this imply that there is no more possibility of getting stored energy from the batteries, reducing to the case seen before. The same, with the positive excess $ex_p^a = e_p^a - d_p^a > 0$ would mean that the battery are all full and still some energy has to be stored and we have already showed that both cases are impossible. □

Algorithm 7.4 From Super Prosumer to Prosumer Exchange

Input : A solution (x_p^{SP}, y_p^{SP}) of a Super Prosumer problem.
Output : A solution $(w_p^a, q_p^a, v_p^{+;a,j}, v_p^{-;a,j})$ of the Prosumer Exchange problem.
$w_p^a, q_p^a = 0, 0 \quad \forall a \in \mathcal{M}, p \in \tau$
$v_p^{+;a,j}, v_p^{-;a,j} = 0, 0 \quad \forall a, j \in \mathcal{M}, p \in \tau$
Set $a^\star = \min\left\{a \in \mathcal{M} \mid \sum_{a \in \mathcal{A}} soc_0^a < \sum_{a \leqslant i} C_a\right\}$; `// the last prosumer with some energy in the battery`
for $a = 2, \ldots, m$ **do**
  Apply Move energy to the first $a^\star$ batteries $(a, SOC_0^a, a^\star)$; `// Move the energy in the first batteries`
Set $w_p^a = \begin{cases} x_p^{SP} & \text{if} \quad a = 1 \\ 0 & \text{otherwise} \end{cases}$ and $q_p^a = \begin{cases} y_p^{SP} & \text{if} \quad a = 1 \\ 0 & \text{otherwise} \end{cases}$; `// Only the first prosumer interacts with the main grid`
for $p = 1, \ldots, n$ **do**
  $SOC_p = SOC_{p-1}$; `// Report the amount of energy in the battery from the previous period`
  for $a = 1, \ldots, m$ **do**
    if $e_p^a > d_p^a$ **then**
      Move energy to the first $a^\star$ batteries $(a, e_p^a - d_p^a)$
    else if $w_p^a - q_p^a + \sum_j v_p^{+;a,j} - \sum_j v_p^{-;a,j} \geqslant d_p^a - e_p^a$ **then**
      if $a + 1 \in \mathcal{M}$ **then**
        $E := w_p^a - q_p^a + \sum_j v_p^{+;a,j} - \sum_j v_p^{-;a,j} - d_p^a + e_p^a$
        Give energy to player$(E, a, a+1)$
      else
        $E := w_p^a - q_p^a + \sum_j v_p^{+;a,j} - \sum_j v_p^{-;a,j} - d_p^a + e_p^a$
        Move energy to the first $a^\star$ batteries (a, E)
    else
      while $d_p^a - e_p^a > w_p^a - q_p^a + \sum_j v_p^{+;a,j} - \sum_j v_p^{-;a,j}$ **do**
        $E := d_p^a - e_p^a - (w_p^a - q_p^a + \sum_j v_p^{+;a,j} - \sum_j v_p^{-;a,j})$
        Take energy from battery$(a, E, a^\star)$
return $(w_p^a, q_p^a, v_p^{+;a,j}, v_p^{-;a,j})$

We have now seen that it is possible to have an allocation in which the independence of the microgrid is maximized. Indeed, it is possible to have multiple of that, for instance by switching the order of the prosumer and applying our Algorithm 7.4. However, the main problem of these is that the allocation is not exactly fair. It is enough to consider how such an allocation was obtained: we have made the first prosumer responsible for the exchanges and then distributed the energy that they have bought. Clearly no prosumer would want to be the so called, prosumer 1 as they are indeed disadvantaged by the system.

This brings up to the second design criteria that we would like to model in our microgrid, namely a fair allocation of the resources. This important aspect is what it is majorly studied also for the ECs in the literature. Additionally to the costs for provisioning energy, ECs also need to take into account the costs related to the battery. Even if we assume that somehow the initial investment is already allocated and so that there is no need to fairly split the amount, there are still the costs related to the battery degradation. All these costs needs to be shared somehow fairly and different approaches have been already explored in literature. In the brief overview of Kulmala et al. [58], some very natural criteria are presented. In particular, they provide an overview of these criteria and list pros and cons for what concern the how understandable a criteria is, its computation and the stability. In the list there are some criteria which are based on the energy profile of the single members of the community. In other option instead the partition is based on fixed shares, which means that prosumers are really buying portions of the shared resource like if they were assets. Another quite popular method is inherited directly from Game Theory, more precisely from Coalitional Games. Without entering too much into details, the high level idea of these types of games is that players act together as a coalition, rather than each player having a selfish behavior. In this case we can assume that the members of the EC are forming the so called grand coalition and win the game, which for us means getting the least amount possible of energy from the retailer. This still opens one of the most basic questions of cooperative games as stated in the book *Models in cooperative game theory* [18], namely how to divide the costs or profit of the grand coalition once it is formed? One of the main solution concept is then the Shapley Value, named after Lloyd Shapley who introduced it in 1951 [78], and it is particularly popular, since it is considered to be a fair criteria given that it splits costs by measuring how much a player contributes in each coalition. From its closed formula it is also possible to prove a list of desirable properties guarantees a list of properties such as efficiency, symmetry, additivity, the null player property. For these reasons, such a criteria would be perfect for splitting the costs of ECs, however because the Shapley value considers all the possible coalition, it is quite impractical from a computational point of view. Some relaxations of this concept are now used, as for example in the work of Cremers et al. [33], but still a lot of research is done in the field. Being in a microgrid, and not in a ECs, there is a bit less of pressure in splitting the costs, given that the main assets are mainly used individually, hence our goal is a bit easier here. However, it is still left to understand how to fairly allocate the exchanges, and we illustrate here our idea. Being this a place where we are allowed to dream of a better society, we try to avoid speculations at every cost. Prosumers with a larger battery already cut their costs by using the energy previously stored and do not necessary need to sell the overflow of their battery at a high price. We model then the selling price as $\varepsilon > 0$, a small amount which is just enough to guarantee individual rationality, namely the requirement that each individual weakly prefers participation in the mechanism rather than not participating [54]. As already discussed, the prices for selling back to the main grid are extremely low, so this ensures also low rates for our desired ε. Then, the main idea is that prosumers figuratively split the remained units. Let us first see a couple of easy examples, featuring only prosumers A, B and C, where prosumer A has some units of energy they would like to sell and the others need to buy energy. To simplify further the setting we consider an extremely short time horizon.

Example 7.1. *Consider the following energy profile:*

$$\begin{aligned} d_A &= (1), \quad e_A = (2), \quad C_A = 1, \quad soc_0^A = 0, \\ d_B &= (1), \quad e_B = (0), \quad C_B = 1, \quad soc_0^B = 0 \ . \end{aligned}$$

According to our model, prosumer A stores one unit, but then they cannot store the second one which would be sold. In this case we want prosumer B to acquire the unit from prosumer A and pay them ε.

Let us now add a third prosumer C with the same profile of B. Then we want to assign the unit to one of the two prosumer and have them split the unit bought from the retailer, so they both pay $(\varepsilon + r)/2$.

In the setting we are considering, we typically do not have to worry about sharing the costs for batteries among prosumers, since each one owns their own battery and no batteries are shared between multiple prosumers. However, there is still the possibility that a prosumer has to use their battery to share some units of energy for another prosumer. In this case, our idea is that the prosumer for whom this energy is kept should pay the battery usage of the prosumer storing it for them, since the prosumer storing the energy is providing a service to the community. We call λ the cost degradation of the battery incurred for storing one unit, and we clarify what we mean with the following example.

Example 7.2. *Consider the following energy profile over a time horizon of two periods:*

$$\begin{aligned} d_A &= (2, 1), \quad e_A = (3, 0), \quad C_A = 1, \quad soc_0^A = 1, \\ d_B &= (0, 0), \quad e_B = (0, 0), \quad C_B = 1, \quad soc_0^B = 0, \\ d_C &= (0, 2), \quad e_C = (1, 0), \quad C_C = 1, \quad soc_0^C = 0 \ . \end{aligned}$$

Now, if we consider the super prosumer SP whose energy profile is

$$d_{SP} = (2, 3) \quad e_{SP} = (4, 0) \quad C_{SP} = 3 \quad soc_0^{SP} = 1,$$

then only one unit needs to be bought from the retailer. So to achieve the same independence of the microgrid, prosumer B *has to store the surplus energy of prosumer* A *at time 1, for prosumer* C *to use it. Hence,* C *will only buy one unit from the market and pay* $\varepsilon + \lambda$ *for the other, where* ε *will go to prosumer* A *and* λ *will compensate the action of* B*.*

Now, we consider a last example, to explain the last cardinal point in our system.

Example 7.3. *Consider the following energy profile over a time horizon of two periods:*

$$\begin{aligned} d_A &= (1,0) \quad e_A = (2,0) \quad C_A = 1 \quad soc_0^A = 0 \\ d_B &= (1,0) \quad e_B = (0,0) \quad C_B = 1 \quad soc_0^B = 0 \\ d_C &= (0,1) \quad e_C = (0,0) \quad C_C = 1 \quad soc_0^C = 0 \ . \end{aligned}$$

The corresponding SP would have to buy one unit. This example is similar to the one in Example 7.1, with the difference that now the demand of prosumer C *is delayed of one unit. Still, we want to split the extra unit equally between prosumers* B *and* C*. The reason behind it is that we do not want to encourage prosumer* C *to lie about their demand, as they could indeed claim to have a demand* $d_C = (0,1)$ *and store the unit of energy in the battery to use it later.*

We generalize the idea described by figuratively splitting each extra unit among all the prosumers who could actually use it. If there is an excess of u units to be split at time p, we define the set of prosumers who will receive this unit as I_p^u and we define it as

$$\begin{aligned} I_p^u = \{ a \in \mathcal{M} \mid \exists p' \geqslant p \text{ s.t. } d_{p'}^a - e_{p'}^a < SOC_{p'}^a - u \\ \wedge\ SOC_s^a \leqslant u \forall s \in [p, p'] \} \ . \end{aligned}$$

The definition of the set I_p^u has also to do with the nature of the market. In this type of market indeed, prosumers can "shift their demand towards left", meaning that if they need one unit of energy at time p but have the space to buy it already at time p', they can easily declare the unit at time p'. The other way around is not possible: if they needed a unit at time p and do not receive it by time p, then their chance to get it is lost. In this sense then we wanted to minimize the envy of a prosumer towards the other who arrive earlier in the market. In some sense this is similar to minimize the *ex-post* envy-freeness, as defined in Chapter 6.

So in conclusion, the allocation resulting from Algorithm 7.5 is actually designing our desired microgrid. We remark that in Algorithm 7.5 we assume that all the units of energy are immediately absorbed by the market. To generalize it however to the more general case of Example 7.2, it would be enough to add the value λ to the costs. The problem however, arises ones the prices for buying energy from the retailer r_p is no longer constant, but changes over time. In this case, the solution obtained for the SP is no longer necessary the once maximizing the independence of the microgrid and hence our approach is no longer suitable. We believe our algorithm would produce an allocation which is somewhat similar to the desired one, but giving an approximation of it is quite challenging, hence we leave it for future research.

Algorithm 7.5 Microgrid exchanges

Input : A Prosumer Exchange instance $\mathcal{I}$, $r_p = r$.
Output : An allocation of energy over the time horizon.
Set $b_p = r$ and $s_p = 0 \, \forall p \in \tau$.
From the Prosumer Exchange instance $\mathcal{I}$ together with b_p and s_p obtain the instance $\mathcal{I}^{SP}$.
Use Algorithm 3.1 to find the schedule of SP minimizing the energy imported form the retailer.
Compute the schedule of each single prosumer with the same prices again with Algorithm 2.1.
Let u_p be the number of units of energy sold within the market at time p and $u_p = \max(0, \sum_{a \in \mathcal{M}} y_p^a - q_p)$.
Initialize $\gamma_p^a = 0 \quad \forall p \in \tau, a \in \mathcal{M}$.
Initialize $\mathsf{cost}_p^a = 0 \quad \forall p \in \tau, a \in \mathcal{M}$.
for $p = 1, \dots, n$ s.t. $u_p > 0$ **do**
  for $\nu = 1, \dots, u_p$ **do**
    $S = 1$
    while $S > 0$ **do**
      Let I_p be the set of prosumers a such that $\exists p' \geqslant p$ such that
        - $d_{p'}^a - e_{p'}^a < SOC_{p'}^a + \gamma_p^a$
        - $R_u^a > 0 \, \forall u \in [p, p']$

      $\gamma_p^a = \min \{R_p^a, S/|I_p^1|\}$
      $S = S - \sum_{a \in \mathcal{M}} \gamma_p^a$
      Select randomly a prosumer a' in I_p and set
      $x_p^{a'} = x_p^{a'} - \sum_{a \in \mathcal{M}} \gamma_p^a$
      $\mathsf{cost}^{a'} = \mathsf{cost}^{a'} + \varepsilon(\gamma_p^a / \sum_{a \in \mathcal{M}} \gamma_p^a) + \sum_{a \neq a'} \gamma_p^a$
      For all the remaining prosumer $a \in I_p$ set
      $\mathsf{cost}^a = \mathsf{cost}^a + \varepsilon(\gamma_p^a / \sum_{a \in \mathcal{M}} \gamma_p^a) - \gamma_p^a$
Compute again the schedule of each single prosumer with the modifications and discounting the costs of prosumers of ε for each unit of energy sold by them.
return An allocation for Prosumer Exchange, namely $(x_p^{a'}, y_p^{a'}, \gamma_p^a)$ for each prosumer *a*.

BIBLIOGRAPHY

[1] Gagan Aggarwal, Tomás Feder, Rajeev Motwani, and An Zhu. "Algorithms for multi-product pricing." In: *International Colloquium on Automata, Languages, and Programming*. 2004, pp. 72–83. DOI: https://doi.org/10.1007/978-3-540-27836-8_9 (cit. on p. 85).

[2] Martin Aleksandrov, Haris Aziz, Serge Gaspers, and Toby Walsh. "Online fair division: Analysing a food bank problem." In: *arXiv preprint* (2015). DOI: https://arxiv.org/abs/1502.07571 (cit. on p. 85).

[3] S. Ould Amrouche, Djamila Rekioua, Toufik Rekioua, and Seddik Bacha. "Overview of energy storage in renewable energy systems." In: *International Journal of Hydrogen Energy* 41.45 (2016), pp. 20914–20927. DOI: https://doi.org/10.1016/j.ijhydene.2016.06.243 (cit. on p. 6).

[4] Georg Angenendt, Sebastian Zurmühlen, Hendrik Axelsen, and Dirk Uwe Sauer. "Comparison of different operation strategies for PV battery home storage systems including forecast-based operation strategies." In: *Applied Energy* (2018), pp. 884–899. DOI: https://doi.org/10.1016/j.apenergy.2018.08.058 (cit. on p. 13).

[5] Mohamed K. Arikiez, Peter Gatens, Floriana Grasso, and Michele Zito. "Smart domestic renewable energy management using knapsack." In: *IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*. 2013, pp. 1–5. DOI: 10.1109/ISGTEurope.2013.6695345 (cit. on p. 14).

[6] Moshe Babaioff, Nicole Immorlica, David Kempe, and Robert Kleinberg. "A knapsack secretary problem with applications." In: *International Workshop on Approximation Algorithms for Combinatorial Optimization*. 2007, pp. 16–28. DOI: https://link.springer.com/chapter/10.1007/978-3-540-74208-1_2 (cit. on p. 69).

[7] Nikhil Bansal, Ning Chen, Neva Cherniavsky, Atri Rurda, Baruch Schieber, and Maxim Sviridenko. "Dynamic pricing for impatient bidders." In: *ACM Transactions on Algorithms* 6.2 (2010), pp. 1–21. DOI: https://doi.org/10.1145/1721837.1721851 (cit. on p. 85).

[8] Edward Barbour and Marta C. González. "Projecting battery adoption in the prosumer era." In: *Applied Energy* 215 (2018), pp. 356–370. DOI: https://doi.org/10.1016/j.apenergy.2018.01.056 (cit. on pp. 5, 7).

[9] Siddharth Barman, Arindam Khan, and Arnab Maiti. "Universal and tight online algorithms for generalized-mean welfare." In: *AAAI Conference on Artificial Intelligence* 36.5 (2022), pp. 4793–4800. DOI: https://doi.org/10.1609/aaai.v36i5.20406 (cit. on p. 85).

[10] Mark Bartlett, Alan M. Frisch, Youssef Hamadi, Ian Miguel, S. Armagan Tarim, and Chris Unsworth. "The temporal knapsack problem and its solution." In: *Constraint Programming for Combinatorial Optimization Problems*. 2005, pp. 34–48. DOI: https://link.springer.com/chapter/10.1007/11493853_5 (cit. on p. 15).

[11] Marc Beaudin and Hamidreza Zareipour. "Home energy management systems: A review of modelling and complexity." In: *Renewable and Sustainable Energy Reviews* (2015), pp. 318–335. DOI: https://doi.org/10.1016/j.rser.2015.01.046 (cit. on p. 13).

[12] Gerdus Benadè, Daniel Halpern, and Alexandros Psomas. "Dynamic fair division with partial information." In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 3703–3715. DOI: https://proceedings.neurips.cc/paper_files/paper/2022/hash/17bb0edcc02bd1f74e771e23b2aa1501-Abstract-Conference.html (cit. on p. 85).

[13] Gerdus Benadè, Aleksandr M. Kazachkov, Ariel D. Procaccia, and Christos-Alexandros Psomas. "How to make envy vanish over time." In: *ACM Conference on Economics and Computation*. 2018, pp. 593–610. DOI: https://doi.org/10.1145/3219166.3219179 (cit. on p. 85).

[14] Kristóf Bérczi, Erika R. Bérczi-Kovács, and Evelin Szögi. "A dual approach for dynamic pricing in multidemand markets." In: *SIAM Journal on Discrete Mathematics* 37.3 (2023), pp. 1771–1787. DOI: https://doi.org/10.1137/22M1535474 (cit. on pp. 84, 88, 90, 94).

[15] Kristóf Bérczi, Laura Codazzi, Julian Golak, and Alexander Grigoriev. "Envy-free dynamic pricing schemes." In: *Operations Research* (2025). DOI: https://doi.org/10.1287/opre.2023.0568 (cit. on p. 83).

[16] Kristóf Bérczi, Naonori Kakimura, and Yusuke Kobayashi. "Market pricing for matroid rank valuations." In: *SIAM Journal on Discrete Mathematics* 35.4 (2021), pp. 2662–2678. DOI: https://doi.org/10.1137/20M1386335 (cit. on pp. 84, 88).

[17] Ben Berger, Alon Eden, and Michal Feldman. "On the power and limits of dynamic pricing in combinatorial markets." In: *International Conference on Web and Internet Economics*. 2020, pp. 206–219. DOI: https://doi.org/10.1007/978-3-030-64946-3_15 (cit. on pp. 84, 88).

[18] Rodica Branzei, Dinko Dimitrov, and Stef Tijs. *Models in cooperative game theory*. Vol. 556. 2008 (cit. on p. 114).

[19] Patrick Briest. “Uniform budgets and the envy-free pricing problem.” In: *International Colloquium on Automata, Languages, and Programming*. 2008, pp. 808–819. DOI: https://doi.org/10.1007/978-3-540-70575-8_66 (cit. on pp. 85, 105).

[20] Johannes Brustle, Jack Dippel, Vishnu V. Narayan, Mashbat Suzuki, and Adrian Vetta. “One dollar each eliminates envy.” In: *ACM Conference on Economics and Computation*. 2020, pp. 23–39. DOI: https://doi.org/10.1145/3391403.3399447 (cit. on p. 85).

[21] Lisa Calearo, Charalampos Ziras, Kristian Sevdari, and Mattia Marinelli. “Comparison of smart charging and battery energy storage system for a pv prosumer with an ev.” In: *IEEE PES Innovative Smart Grid Technologies Europe*. 2021, pp. 1–6. DOI: 10.1109/ISGTEurope52324.2021.9640120 (cit. on pp. 31, 39).

[22] Gregory Campbell and Stephen M Samuels. “Choosing the best of the current crop.” In: *Advances in Applied Probability* 13.3 (1981), pp. 510–532. DOI: 10.2307/1426783 (cit. on p. 69).

[23] Chiara Candelise, Mark Winskel, and Robert JK. Gross. “The dynamics of solar PV costs and prices as a challenge for technology forecasting.” In: *Renewable and Sustainable Energy Reviews* 26 (2013), pp. 96–107. DOI: https://doi.org/10.1016/j.rser.2013.05.012 (cit. on p. 6).

[24] Timothy Capper, Anna Gorbatcheva, Mustafa A. Mustafa, Mohamed Bahloul, Jan Marc Schwidtal, Ruzanna Chitchyan, Merlinda Andoni, Valentin Robu, Mehdi Montakhabi, Ian J. Scott, et al. “Peer-to-peer, community self-consumption, and transactive energy: A systematic literature review of local energy market models.” In: *Renewable and Sustainable Energy Reviews* 162 (2022), p. 112403. DOI: https://doi.org/10.1016/j.rser.2022.112403 (cit. on pp. 7, 14).

[25] Bundit Chalermsook Parinya and Laekhanukit and Danupon Nanongkai. “Independent set, induced matching, and pricing: Connections and tight (subexponential time) approximation hardnesses.” In: *Annual Symposium on Foundations of Computer Science*. 2013, pp. 370–379. DOI: 10.1109/FOCS.2013.47 (cit. on pp. 85, 105).

[26] Xinyue Chang, Yinliang Xu, Qinglai Guo, Hongbin Sun, and Wai Kin Chan. “A Byzantine-resilient distributed peer-to-peer energy management approach.” In: *IEEE Transactions on Smart Grid* 14.1 (2022), pp. 623–634. DOI: 10.1109/TSG.2022.3185140 (cit. on pp. 41, 43).

[27] François Clautiaux, Boris Detienne, and Gaël Guillot. “An iterative dynamic programming approach for the temporal knapsack problem.” In: *European Journal of Operational Research* 293.2 (2021), pp. 442–456. DOI: https://doi.org/10.1016/j.ejor.2020.12.036 (cit. on p. 15).

[28] Laura Codazzi, Gergely Csáji, and Matthias Mnich. "Efficient cost-minimization schemes for electrical energy demand satisfaction by prosumers in microgrids with battery storage capabilities." In: *International Joint Conference on Artificial Intelligence*. 2024. DOI: `https://doi.org/10.15480/882.13195` (cit. on pp. 13, 29).

[29] Vincent Cohen-Addad, Alon Eden, Michal Feldman, and Amos Fiat. "The invisible hand of dynamic market pricing." In: *ACM Conference on Economics and Computation*. 2016, pp. 383–400. DOI: `https://doi.org/10.1145/2940716.2940730` (cit. on pp. 84, 88–90).

[30] José Correa, Andrés Cristi, Laurent Feuilloley, Tim Oosterwijk, and Alexandros Tsigonias-Dimitriadis. "The secretary problem with independent sampling." In: *Management Science* (2024). DOI: `https://doi.org/10.1287/mnsc.2021.01580` (cit. on p. 69).

[31] Jose Correa, Patricio Foncea, Ruben Hoeksma, Tim Oosterwijk, and Tjark Vredeveld. "Recent developments in prophet inequalities." In: *ACM SIGecom Exchanges* 17.1 (2019), pp. 61–70. DOI: `https://doi.org/10.1145/3331033.3331039` (cit. on p. 85).

[32] Peter Cramton. "Electricity market design." In: *Oxford Review of Economic Policy* 33.4 (2017), pp. 589–612. DOI: `https://www.jstor.org/stable/48539475` (cit. on p. 7).

[33] Sho Cremers, Valentin Robu, Daan Hofman, Titus Naber, Kawin Zheng, and Sonam Norbu. "Efficient methods for approximating the Shapley value for asset sharing in energy communities." In: *ACM International Conference on Future Energy Systems*. 2022, pp. 320–324. DOI: `https://dl.acm.org/doi/pdf/10.1145/3538637.3538861` (cit. on p. 114).

[34] Herbert A David and Haikady N Nagaraja. *Order statistics*. John Wiley & Sons, 2004 (cit. on p. 77).

[35] *Durchschnittlicher Preis Haushaltsstrom zum 1. April 2023*. May 2024. DOI: `https://www.bundesnetzagentur.de/DE/Vportal/Energie/PreiseAbschlaege/Tarife-table.html` (cit. on p. 9).

[36] Jeno Egerváry. "Matrixok kombinatorius tulajdonságairól." In: *Matematikai és Fizikai Lapok* 38.1931 (1931), pp. 16–28 (cit. on p. 90).

[37] *Einspeisevergütung Solarstrom 2023 nach EEG*. May 2024. URL: `https://www.bundesnetzagentur.de/DE/Fachthemen/ElektrizitaetundGas/ErneuerbareEnergien/EEG_Foerderung/Archiv_VergSaetze/start.html` (cit. on p. 9).

[38] Tomer Ezra, Stefano Leonardi, Rebecca Reiffenhauser, Matteo Russo, and Alexandros Tsigonias-Dimitriadis. "Prophet inequalities via the expected competitive ratio." In: *ACM Transactions on Economics and Computation* 13.2 (2025), pp. 1–32. DOI: `https://doi.org/10.1007/978-3-031-48974-7_16` (cit. on p. 72).

[39] Zigeng Fang, Jiayi Yan, Qiuchen Lu, Long Chen, Pu Yang, Junqing Tang, Feng Jiang, Tim Broyd, and Jingke Hong. "A systematic literature review of carbon footprint decision-making approaches for infrastructure and building projects." In: *Applied Energy* 335 (2023), p. 120768. DOI: https://doi.org/10.1016/j.apenergy.2023.120768 (cit. on p. 5).

[40] Thomas S Ferguson. "Who solved the secretary problem?" In: *Statistical Science* 4.3 (1989), pp. 282–289. DOI: 10.1214/ss/1177012493 (cit. on p. 68).

[41] Amos Fiat, Ilia Gorelik, Haim Kaplan, and Slava Novgorodov. "The temp secretary problem." In: *Annual European Symposium*. 2015, pp. 631–642. DOI: https://doi.org/10.1007/978-3-662-48350-3_53 (cit. on p. 69).

[42] Herbert Fleischner, Gert Sabidussi, and Vladimir I. Sarvanov. "Maximum independent sets in 3-and 4-regular Hamiltonian graphs." In: *Discrete mathematics* 310.20 (2010), pp. 2742–2749. DOI: https://doi.org/10.1016/j.disc.2010.05.028 (cit. on p. 101).

[43] Jugal Garg, Edin Husić, and László A. Végh. "Approximating Nash social welfare under Rado valuations." In: *ACM Symposium on Theory of Computing*. 2021, pp. 1412–1425. DOI: https://doi.org/10.1145/3406325.3451031 (cit. on p. 105).

[44] John P Gilbert and Frederick Mosteller. "Recognizing the maximum of a sequence." In: *Journal of the American Statistical Association* 61.313 (1966), pp. 35–73. URL: https://www.jstor.org/stable/2283044 (cit. on p. 69).

[45] Artur Gorokh, Siddhartha Banerjee, and Krishnamurthy Iyer. "From monetary to nonmonetary mechanism design via artificial currencies." In: *Mathematics of Operations Research* 46.3 (2021), pp. 835–855. DOI: https://doi.org/10.1287/moor.2020.1098 (cit. on p. 85).

[46] Venkatesan Guruswami, Jason D. Hartline, Anna R. Karlin, David Kempe, Claire Kenyon, and Frank McSherry. "On profit-maximizing envy-free pricing." In: *ACM-SIAM Symposium on Discrete Algorithms*. Vol. 5. 2005, pp. 1164–1173. DOI: http://dl.acm.org/citation.cfm?id=1070432.1070598 (cit. on pp. 84, 88, 100, 105).

[47] Faeza Hafiz, Anderson Rodrigo de Queiroz, Poria Fajri, and Iqbal Husain. "Energy management and optimal storage sizing for a shared community: A multi-stage stochastic programming approach." In: *Applied Energy* 236 (2019), pp. 42–54. DOI: https://doi.org/10.1016/j.apenergy.2018.11.080 (cit. on pp. 30, 31).

[48] Daniel Halpern and Nisarg Shah. "Fair division with subsidy." In: *International Symposium on Algorithmic Game Theory*. 2019, pp. 374–389. DOI: https://doi.org/10.1007/978-3-030-30473-7_25 (cit. on p. 85).

[49] Jiafan He, Ariel D. Procaccia, CA. Psomas, and David Zeng. "Achieving a fairer future by changing the past." In: *International Joint Conference on Artificial Intelligence* (2019). DOI: `https://par.nsf.gov/servlets/purl/10113972` (cit. on p. 85).

[50] Paul Hockenos. In: *Wired* (2019). DOI: `https://www.wired.com/story/in-germany-solar-powered-homes-are-catching-on/` (cit. on p. 6).

[51] Jens Hönen, Johann L. Hurink, and Bert Zwart. "A classification scheme for local energy trading." In: *OR Spectrum* 45.1 (2023), pp. 85–118. DOI: `https://doi.org/10.1007/s00291-022-00697-6` (cit. on p. 7).

[52] Seyed Mohsen Hosseini, Raffaele Carli, and Mariagrazia Dotoli. "Robust day-ahead energy scheduling of a smart residential user under uncertainty." In: *European Control Conference*. 2019, pp. 935–940. DOI: `10.23919/ECC.2019.8796182` (cit. on p. 13).

[53] Justin Hsu, Jamie Morgenstern, Ryan Rogers, Aaron Roth, and Rakesh Vohra. "Do prices coordinate markets?" In: *ACM Symposium on Theory of Computing*. 2016, pp. 440–453. DOI: `https://doi.org/10.1145/2897518.2897559` (cit. on p. 84).

[54] Matthew O. Jackson. "Mechanism theory." In: *Available at SSRN 2542983* (2014) (cit. on p. 115).

[55] David S. Johnson. "A theoretician's guide to the experimental analysis of algorithms." In: *Data Structures, Near Neighbor Searches, and Methodology* 5 (1999), pp. 215–250. URL: `http://archive.dimacs.rutgers.edu/Challenges/TSP/papers/experguide.pdf` (cit. on p. 27).

[56] Richard M. Karp. "Reducibility among Combinatorial Problems." In: ed. by Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger. Springer US, 1972, pp. 85–103 (cit. on p. 13).

[57] Alexander S. Kelso Jr. and Vincent P. Crawford. "Job matching, coalition formation, and gross substitutes." In: *Econometrica: Journal of the Econometric Society* (1982), pp. 1483–1504. DOI: `https://doi.org/10.2307/1913392` (cit. on pp. 84, 89).

[58] Anna Kulmala, Marius Baranauskas, Amir Safdarian, Jussi Valta, Pertti Järventausta, and Tomas Björkqvist. "Comparing value sharing methods for different types of energy communities." In: *IEEE PES Innovative Smart Grid Technologies Europe*. 2021, pp. 1–6. DOI: `10.1109/ISGTEurope52324.2021.9640205` (cit. on p. 114).

[59] Nallaperumal Kumaraguruparan, Harihara Sivaramakrishnan, and Sachin S. Sapatnekar. "Residential task scheduling under dynamic pricing using the multiple knapsack method." In: *IEEE PES Innovative Smart Grid Technologies*. 2012, pp. 1–6. DOI: `10.1109/ISGT.2012.6175656` (cit. on p. 14).

[60] JA. Pecas Lopes, Nikos Hatziargyriou, Joseph Mutale, Predrag Djapic, and N. Jenkins. “Integrating distributed generation into electric power systems: A review of drivers, challenges and opportunities.” In: *Electric Power Systems Research* 77.9 (2007), pp. 1189–1203. DOI: https://doi.org/10.1016/j.epsr.2006.08.016 (cit. on p. 5).

[61] Brendan Lucier. “An economic view of prophet inequalities.” In: *ACM SIGecom Exchanges* 16.1 (2017), pp. 24–47. DOI: https://doi.org/10.1145/3144722.3144725 (cit. on p. 85).

[62] Georgios Mavromatidis, Kristina Orehounig, and Jan Carmeliet. “Design of distributed energy systems under uncertainty: A two-stage stochastic programming approach.” In: *Applied Energy* 222 (2018), pp. 932–950. DOI: https://doi.org/10.1016/j.apenergy.2018.04.019 (cit. on p. 31).

[63] Prakhar Mehta and Verena Tiefenbeck. “Solar PV sharing in urban energy communities: Impact of community configurations on profitability, autonomy and the electric grid.” In: *Sustainable Cities and Society* 87 (2022), p. 104178. ISSN: 2210-6707. DOI: https://www.sciencedirect.com/science/article/pii/S2210670722004917 (cit. on pp. 6, 7).

[64] Esther Mengelkamp, Julius Diesing, and Christof Weinhardt. “Tracing local energy markets: A literature review.” In: *it-Information Technology* 61.2-3 (2019), pp. 101–110. DOI: https://doi.org/10.1515/itit-2019-0016 (cit. on p. 7).

[65] Esther Mengelkamp, Johannes Gärttner, Kerstin Rock, Scott Kessler, Lawrence Orsini, and Christof Weinhardt. “Designing microgrid energy markets: A case study: The Brooklyn Microgrid.” In: *Applied Energy* 210 (2018), pp. 870–880. DOI: https://doi.org/10.1016/j.apenergy.2017.06.054 (cit. on p. 6).

[66] David P Morton. “An enhanced decomposition algorithm for multistage stochastic hydroelectric scheduling.” In: *Annals of Operations Research* 64 (1996), pp. 211–235. DOI: https://doi.org/10.1007/BF02187647 (cit. on p. 31).

[67] *Net Zero Emission by 2050 Scenario (NZE)*. https://www.iea.org/reports/global-energy-and-climate-model/net-zero-emissions-by-2050-scenario-nze. May 2024 (cit. on p. 5).

[68] *New Thames Valley Vision - End Point Monitors*. [Accessed 02-05-2025]. URL: https://ukerc.rl.ac.uk/cgi-bin/dataDiscover.pl?Action=detail&dataid=f3dd1d3b-41f6-4980-b38c-a80d4d0fb7b9 (cit. on p. 60).

[69] Sonam Norbu, Benoit Couraud, Valentin Robu, Merlinda Andoni, and David Flynn. “Modelling the redistribution of benefits from joint investments in community energy projects.” In: *Applied Energy* 287 (2021), p. 116575. DOI: https://doi.org/10.1016/j.apenergy.2021.116575 (cit. on p. 14).

[70] Alex Nutkiewicz, Zheng Yang, and Rishee K Jain. "Data-driven Urban Energy Simulation (DUE-S): A framework for integrating engineering simulation and machine learning methods in a multi-scale urban energy modeling workflow." In: *Applied Energy* 225 (2018), pp. 1176–1189. DOI: `https://doi.org/10.1016/j.apenergy.2018.05.023` (cit. on p. 31).

[71] *Paris Agreement*. `https://unfccc.int/files/essential_background/convention/application/pdf/english_paris_agreement.pdf`. May 2024 (cit. on p. 5).

[72] European Parliament. "Directive (EU) 2018/2001 of the European Parliament and of the Council on the promotion of the use of energy from renewable sources." In: *Official Journal of the European Union* vol. 2018 (2018), 82–209 (cit. on p. 108).

[73] Kanstantsin Pashkovich and Xinyue Xie. "A two-step approach to optimal dynamic pricing in multi-demand combinatorial markets." In: *arXiv preprint arXiv:2201.12869* (2022) (cit. on pp. 84, 88).

[74] *Preis Rechner tado*. `https://energy.tado.com/price?zipcode=21075&yearlyConsumption=2400&includeHourlyTariffSavings=true&city=Hamburg`. May 2024. URL: `https://figshare.com/articles/dataset/Modified_IEEE_15-bus_system_parameters_docx/19289642` (cit. on p. 41).

[75] Wolf-Peter Schill, Alexander Zerrahn, and Friedrich Kunz. "Prosumage of solar electricity: pros, cons, and the system perspective." In: *Economics of Energy & Environmental Policy* 6.1 (2017), pp. 7–32. DOI: `https://www.jstor.org/stable/26189569` (cit. on p. 6).

[76] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1998 (cit. on p. 31).

[77] Maryam Shahbazitabar, Hamdi Abdi, Hossein Nourianfar, Amjad Anvari-Moghaddam, Behnam Mohammadi-Ivatloo, and Nikos Hatziargyriou. "An introduction to microgrids, concepts, definition, and classifications." In: *Microgrids: Advances in Operation, Control, and Protection* (2021), pp. 3–16. DOI: `https://doi.org/10.1007/978-3-030-59750-4_1` (cit. on p. 6).

[78] LS Shapley. "Notes on the N-Person Game mdash." In: *II: The Value of an N-Person Game (Santa Monica, CA: RAND Corporation)* (1951) (cit. on p. 114).

[79] Omid Ameri Sianaki, Omar Hussain, and Azadeh Rajabian Tabesh. "A knapsack problem approach for achieving efficient energy consumption in smart grid for endusers' life style." In: *2010 IEEE Conference on Innovative Technologies for an Efficient and Reliable Electricity Supply*. 2010, pp. 159–164. DOI: `10.1109/CITRES.2010.5619873` (cit. on p. 14).

[80] *Szenariorahmen zum Netzentwicklungsplan Strom 2037/2045, Version 2025*. 2024. DOI: https://www.bundesnetzagentur.de/DE/Fachthemen/ElektrizitaetundGas/NEP/DL_Szenariorahmen/SR_Entwurf2025Strom.pdf?__blob=publicationFile&v=2 (cit. on p. 6).

[81] Falti Teotia and Rohit Bhakar. "Local energy markets: Concept, design and operation." In: *2016 national power systems conference (NPSC)*. 2016, pp. 1–6. DOI: 10.1109/NPSC.2016.7858975 (cit. on p. 5).

[82] Clemens Thielen, Morten Tiedemann, and Stephan Westphal. "The online knapsack problem with incremental capacity." In: *Mathematical Methods of Operations Research* 83 (2016), pp. 207–242. DOI: https://doi.org/10.1007/s00186-015-0526-9 (cit. on pp. 69, 78).

[83] Wayes Tushar, Chau Yuen, Tapan K Saha, Thomas Morstyn, Archie C Chapman, M Jan E Alam, Sarmad Hanif, and H Vincent Poor. "Peer-to-peer energy systems for connected communities: A review of recent advances and emerging challenges." In: *Applied Energy* 282 (2021), p. 116131. DOI: https://doi.org/10.1016/j.apenergy.2020.116131 (cit. on pp. 7, 13).

[84] Taha Selim Ustun, Cagil Ozansoy, and Aladin Zayegh. "Recent developments in microgrids and example cases around the world—A review." In: *Renewable and Sustainable Energy Reviews* 15.8 (2011), pp. 4030–4041. DOI: https://doi.org/10.1016/j.rser.2011.07.033 (cit. on p. 107).

[85] Leon Walras. *Elements of pure economics: Or the theory of social wealth*. Routledge, 2013 (cit. on pp. 84, 89).

[86] David Zeng and Alexandros Psomas. "Fairness-efficiency tradeoffs in dynamic fair division." In: *ACM Conference on Economics and Computation*. 2020, pp. 911–912. DOI: https://doi.org/10.1145/3391403.3399467 (cit. on p. 85).

[87] Ying Zhang, Valentin Robu, Sho Cremers, Sonam Norbu, Benoit Couraud, Merlinda Andoni, David Flynn, and H Vincent Poor. "Modelling the formation of peer-to-peer trading coalitions and prosumer participation incentives in transactive energy communities." In: *Applied Energy* 355 (2024), p. 122173. DOI: https://doi.org/10.1016/j.apenergy.2023.122173 (cit. on p. 14).