TECHNISCHE
UNIVERSITÄT
WIEN
Vienna University of Technology

MASTER'S THESIS

Stochastic Optimization of a Battery Storage System

to obtain the academic degree

Master of Science

as part of the Master's programme

Statistics and Mathematics in Economics

submitted by

Sarah Wimmeder

Matriculation Number: 01406064

Carried out at the Institute of Statistics and Mathematical Methods in Economics, at the Faculty of Mathematics and Geoinformation at the Vienna University of Technology, in cooperation with the Austrian Institute of Technology.

Supervisor:
Univ. Ass. Priv.-Doz. Dr. Raimund Kovacevic

Vienna, September 2021

Abstract

Due to the increasing share of renewable energies the demand for power storage has been growing significantly in recent years. This development opens a wide variety of possible battery applications at distribution as well as at household or consumption level.

The focus of this thesis is on electricity consumers who are concerned about optimally operating an energy storage system (ESS) to reduce their electricity costs. This is achieved by storing self- produced photovoltaic (PV) energy and low-cost energy from the grid. Energy consumers can dynamically adjust their electricity storage decisions in response to randomly evolving electricity demand, unpredictable photovoltaic generation and varying electricity prices.

To quantify the value of integrating a battery storage into the system, a multistage stochastic programming model is formulated with the objective to minimize the expected total electricity costs over a finite planning horizon. It provides optimal charging and discharging decisions under uncertainty at each stage of the decision horizon.

Due to the so called 'curse of dimensionality', stochastic dynamic decision problems are really challenging. The curse implies that the complexity of the problem increases exponentially in the number of state variables, and that in general no solution algorithm which converges towards an exact solution in polynomial time exists. To guarantee computational tractability, the problem is solved by a combination of stochastic dual dynamic programming (SDDP) and a quantization method which approximates the input parameters by a discrete scenario lattice. This method is referred to as approximate dual dynamic programming (ADDP).

The proposed method is analysed numerically via conducting a case study. Thereby, an econometric model including actual price, PV generation and consumption data is examined.

To assess the added value of the stochastic solution, the results obtained by the ADDP method are compared to the results assessed by a deterministic approach, where the stochastic part of the optimization is ignored, and the expected values of the input parameters are used instead. Furthermore, we model different risk preferences of the consumers by coherent acceptability functionals and determine their influence on the optimal solution.

Kurzfassung

Aufgrund des immer größer werdenden Anteils an erneuerbaren Energien hat in den letzten Jahren die Nachfrage nach Energiespeichern erheblich zugenommen. Diese Entwicklung eröffnet eine Vielzahl möglicher Batterieanwendungen, sowohl auf Verteiler- als auch auf Haushalts- bzw. Verbraucherebene.

Die vorliegende Arbeit widmet sich der Betrachtung eines Stromverbrauchers, dessen Ziel es ist, ein Energiespeichersystem (ESS) optimal zu betreiben, um seine Stromkosten zu minimieren. Dies wird durch die Speicherung von selbst erzeugtem Photovoltaik-(PV) Strom und kostengünstigem Strom aus dem Netz erreicht. Es wird davon ausgegangen, dass Energieverbraucher ihre Einspeicherentscheidungen dynamisch anpassen können, um auf zufälligen Strombedarf, unvorhersehbare PV-Erzeugung und variierende Strompreise zu reagieren.

Um den Wert eines in das System integrierten Batteriespeichers zu qunatifizieren, wird ein mehrstufiges stochastisches Modell formuliert, dessen Ziel es ist, die erwarteten Gesamtkosten über einen begrenzten Planungshorizont zu minimieren. Dieses bietet optimale Lade- und Entladeentscheidungen unter Einbezug von Unsicherheiten in jeder Phase des Entscheidungshorizonts.

Aufgrund des sogenannten "Fluchs der Dimensionalität" kann sich die Lösung stochastisch-dynamischer Entscheidungsprobleme sehr herausfordernd gestalten. Grund dafür ist, dass die Komplexität des Problems exponentiell in der Anzahl der Zustandsvariablen steigt. Außerdem existiert im Allgemeinen kein Lösungsalgorithmus, der zu einer exakten Lösung konvergiert. Um die numerische Lösbarkeit des behandelten Problems zu gewährleisten, wird eine Kombination aus stochastischer dualer dynamischer Programmierung (SDDP) und einer speziellen Quantisierungsmethode, welche die Eingangsdaten durch ein diskretes Szenariogitter approximiert, zur Lösung des Problems verwendet. Diese Methode bezeichnet man als approximierte dynamische duale Programmierung (ADDP). Unter anderem werden wir feststellen, dass die so ermittelte, angenäherte Lösung gegen eine Obergrenze der optimalen Lösung konvergiert.

Die untersuchte Vorgehensweise wird anhand einer Fallstudie numerisch analysiert. Es wird ein ökonometrsiches Modell aufgesetzt und analysiert, das Daten zu tatsächlichen Preisen, PV-Erzeugungen und Stromverbräuchen enthält.

Um den Mehrwert der stochastischen Lösung aufzuzeigen, werden die mit der ADDP-Methode ermittelten Ergebnisse mit Ergebnissen verglichen, die mit einem deterministischen Ansatz ermittelt wurden. Bei diesem deterministischen Ansatz wird der stochastische Teil der Optimierung ignoriert und stattdessen werden die Erwartungswerte der Inputdaten verwendet. Darüber hinaus werden unterschiedliche Risikopräferenzen der Verbraucher durch koheränte Akzeptanzfunktionale modelliert und deren Einfluss auf die optimale Lösung ermittelt.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt bzw. die wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht habe.

Wien, im September 2021

Name des Autors

Contents

1 Introduction

The Paris Agreement marks a major breakthrough in international climate policy. As it is summarized in Bundesministerium für Klimaschutz (2015), one of its major goals is to limit the global warming to a maximum of two degrees Celsius compared to pre-industrial values and to decrease the global greenhouse gas emissions to (net) zero by 2050. In numbers, the shared purpose of the EU member states is to cut greenhouse gas emissions by 80-95% by 2050.

The authors of European-Comission (2011) and Directorate-General for Energy (2012) claim that due to the possibility of reducing emissions through implementation of low-carbon technologies and increased energy efficiency, the impact on our energy systems will be huge. In the Austrian government program 2020-2024, the goal of increasing the share of electricity from renewable energy sources up to 100% by 2030 is set. As outlined by Neubarth (2020), the plan is to add an annual amount of around 27 TWh electricity generation from renewables. In contrast to the common centralized generation structures in large power plants, the expansion of renewable energies will tend to result in increasingly decentralized structures. However, the decentralization of our power supply system will not be limited to generation.

Electricity storage as well as direct on-site consumption of generated or stored electricity in new areas of application, such as e-vehicles and heat pumps (keyword *sector coupling*), will play a major role. An increasing number of individual citizens, municipalities or commercial and industrial companies are forming a direct part of the changing new energy system. Neubarth (2020) and numerous other authors are certain that in this context energy communities can play an important role, as they enable the population to actively participate in the initiation and implementation as well as in the operation of projects and thus be able to contribute to the necessary acceptance for the expansion of renewables. A study conducted by the authors of res (2016) found, that half of the EU citizens- including local communities, schools and hospitals- could produce their own renewable electricity by 2050, covering 45% of their energy needs.The European Union has also recognized the fundamental importance of energy communities for achieving the overarching energy policy goals and has issued a number of regulations as part of its Clean Energy Package in order to strengthen the role of energy communities at the level of both, renewable energy and community energy. The next step, however, is to convert the European framework into national law in such a way that the necessary requirements for the successful establishment of energy communities in Austria are actually created.

The increasing share of renewable energy systems like photovoltaic systems or wind power plants has significant consequences on the energy sector, not only in Europe but also in the U.S., where the future power industry generation portfolio will include a strong mix of renewable technologies as well. Mokrian and Stephen (2006) find that recent trends in new renewable generation facilities have shown positive growth, especially in the wind sector, where installed capacity within the U.S. has grown by over 300% over the past six years.

The most significant drawback of renewable sources is their intermittency. Due to their volatile, weather dependant electricity production future supplies are very difficult to predict. The output of a solar power plant for example naturally depends on the geographical placement of the plant. It has one component, which is depending on time and another one which depends on the actual weather.

Those features of renewable sources are challenging to integrate in power systems where it is important that at any time demand meets supply. Many authors, for example Mokrian and Stephen (2006), Bhattacharya et al. (2018) or Delong et al. (2019) to name a few, agree, that batteries could be used to mitigate the impacts of this characteristics.

A key component of a more flexible, smarter grid is the ability to store electricity and thereby to decouple electricity generation from electricity consumption. During recent years energy storage technologies have been rapidly evolving. Lithium-ion chemistry (Li-ion) battery technology has advanced significantly over the past two decades. According to Stenclik et al. (2017), Li-ion chemistry accounted for at least 97% of grid-scale battery energy storage deployment in 2016. However, the current price of ESS is still relatively high.

The use of batteries in renewable energy systems has been investigated as part of the Benchmarking project by Tselepis et al. (2004). Data from many power systems were analysed and similar profiles of the batteries have been identified. Donnellan et al. (2015) review the need for various forms of energy storage within power systems. They focus on the potential applications, operation and benefits of intermediate storage.

Following for example Ghofrani et al. (2013) or Baker et al. (2017), distributed energy storage systems can be used to enhance grid performance and reliability with services like managing peak demand, reducing the ramping of generators, providing ancillary services such as load following and regulation, resolving transmission and distribution upgrades and supporting demand response resources.

Batteries are also useful in the field of electricity consumers. The offset in time between power generation and consumption provides a clear potential for arbitrage profits as an example. Prices determined in electricity markets are also extremely volatile and highly dependent on time of day and seasonal effects. By purchasing electricity during off-peak hours (typically at night) when the price is low, storing that power, and then selling the stored power during peak hours at a significantly higher price it is possible to generate profit. Roberts and Sandberg (2011) propose the operation of batteries to hedge against renewable uncertainty and shift local energy consumption from peak-demand to low-demand periods.

To facilitate those electricity transactions numerous approaches exist, such as demand response (DR) programs or smart home energy management systems (SHEMS).

Rahimi and Ipakchi (2010) and Chen et al. (2012) define DR as a flexible mechanism that enables consumers to demand modulation in response to signals from the system operator. Keerthisinghe et al. (2014) for example claim that besides the ability to reduce the electricity consumption during periods of high energy prices, consumers are also encouraged to reduce their loads during periods of critical network congestion. Through special devices consumers are provided with access to near real-time information and can benefit from technologies such as two-way communication, distributed generation and schedulable assets.

Currently, proposed SHEMS require accurate weather forecasts and an exact knowledge about the behavioural patterns of the inhabitants to work properly. In the real world making such precise predictions is often very difficult. Furthermore, as explained by Pedrasa et al. (2011), unprecise predictions can result in additional costs and may have negative consequences on the comfort of the inhabitants. For this reason, including uncertainties by using probabilistic models is a promising approach. As an example Keerthisinghe et al. (2014) propose a SHEMS that assist residential energy users to schedule and coordinate their energy demand. The stochastic nature of the household's energy consumption and the intermittent nature of its distributed generation are considered.

Most prevalent are models that devise optimal demand-response schemes for consumers with elastic loads, as for example the models of Gouveia et al. (2013) and Atzeni et al. (2012). Habib et al. (2017) propose an emery storage system optimization configuration method to improve the reliability of consumer electricity supply in case consumers lose their power supply. The application scenario includes consumers, PV, and energy storage. It does not take the uncertainty into consideration, which has an important effect on the optimal result. Kovacevic et al. (2017) investigate how batteries can provide flexibility to a welfare or profit optimizing aggregator. Chen et al. (2012) divide the operation tasks of appliances into three categories: interruptible and deferrable, non-interruptible and deferrable, and non-interruptible and non-deferrable.

There is far less literature on cases where consumers do not have the possibility to adapt their load, as it would be the case for office buildings, for instance.

One attempt often made concerning this setting, is to optimize the charging and discharging behavior of the electricity storage. Understanding the impact of optimal storage management under economic and operational constraints is of practical importance for various businesses, from consumers to system operators to investors in smart grid technologies. There exist numerous studies on optimizing the operation of a single storage connected to a general linear memoryless system in the presence of ramp constraints. However, in most cases, the objective function is deterministic and known a priori. As stated by Faghih et al. (2011) there is a strong need for development of econometric models and characterization of the response of a storage system to real-time price signals. To start with, Faghih et al. (2011) provide a model for optimal utilization of ramp-constrained storage regarding stochastically varying electricity prices. The problem is formulated in a finite- horizon dynamic programming framework.

Baker et al. (2017) propose a model predictive control method to optimize the ESS. The errors in the prediction of the wind forecasts are taken into account and the problem is directly solved by using stochastic optimization. van de Ven et al. (2013) examine a demand response problem under real-time pricing uncertainty using a finite-horizon Markov decision process (MDP) model which prescribes the amount of energy to procure, store and discharge in each decision stage of the horizon. In contrast, Koutsopoulos et al. (2011) use an infinite-horizon MDP model to derive an optimal threshold policy under price and supply uncertainty.

However, none of these MDP models account for simultaneous uncertainty in demand, supply, and pricing, and all of them use relatively few scenarios to keep the problem dimension low.

A widespread approach to represent a stochastic problem is to consider the decision problem as a multi- stage stochastic program. The prerequisite for the representation as a stochastic program is that the problem has a finite number of stages and that the parameters can be described by an exogenous stochastic process. If the stochastic processes are discrete, it is possible to represent them by using a scenario tree. The disadvantage of this representation is that the number of nodes increases exponentially as the number of stages increase. This leads to the so called curse of dimensionality.

However, multistage SP models are significantly hard to solve, and are intractable for even a moderate number of stages. To date, no reasonably fast generic solution method is known which is able to solve stochastic dynamic decision problems to optimality.

The stochastic dual dynamic programming (SDDP) method proposed by Pereira (1989) became popular in many applications. This strategy can handle problems with a large number of stages, as long as the optimization problem at each stage is convex and the stochastic process is stagewise independent. This assures the independence of the cost-to-go functions of the data process. Bhattacharya et al. (2018) model a microgrid assuming that demand, renewable supply and prices are random variables and customize the SDDP algorithm.

In many applications the data process is not completely memoryless but exhibits a Markovian type dependence structure. Like any stochastic-dynamic decision problem MDP's do also have the property of getting increasingly difficult to solve as the number of decision states increases.

Löhndorf and Wozabal (2017) developed a promising methodology to solve such problems in a scalable way and break the curse of dimensionality. This method is called approximate dynamic dual programming (ADDP). It is used to approximately solve a Markov decision problem by discretization of the Markov Process using optimal quantization. To avoid a complete enumeration of the state space Monte Carlo simulation is used to sample relevant states. In the next step the value function of the real problem is approximated by a function of much lower complexity. In the same way as the SDDP approach the decision problem is solved iteratively using forward simulation to sample potential decisions and backward recursion to construct an approximate value function. However, the advantage of ADDP is that the method does not require stagewise independence of the stochastic process, instead so called scenario lattices are used.

Gorski (2017) compares the performance of tree-based models for solving stochastic programs to the performance of lattice-based approaches. The author comes to the conclusion that especially for high-dimensional problems with increasing dependency between the stages, lattice- based solution methods offer a higher expected payoff. Löhndorf et al. (2013) use the ADDP method to optimize the operation of hydro storage systems with multiple connected reservoirs and show that the approximate solution converges towards an upper bound of the optimal solution.

The research focus of this thesis is to better understand the various potential benefits of stochastic optimization in combination with an energy storage system. For this reason an optimal stochastic optimization model for electricity consumers with installed PV plants and an integrated Lithium-Ion battery is proposed. The consumers are not able to adopt their consumption patterns, hence it is necessary to minimize consumers' expenditures to find an optimal charge and discharge strategy for managing the installed storage device. Electricity prices, demand and PV output are assumed to be random. The problem is stated as a multi-stage stochastic dynamic program and it is solved by adapting an ADDP algorithm. Furthermore different risk preferences of the decision maker are modeled, including risk neutral, risk averse and risk loving.

As far as we are concerned, this is the first work which simultaneously assumes randomness in price, demand and PV generation, as well as considering different risk measures in the objective function.

The thesis is organized as follows. In Section 2, the underlying deterministic base case of the problem is formulated and motivation for the stochastic approach is given. Section 3 introduces the stochastic formulation of the battery operating problem, after giving a general overview of stochastic programs. In Section 4 the construction of scenario lattices and their advantages over scenario trees are discussed. Furthermore an overview of different methods for scenario construction is given. Section 5 takes a deeper look at solution methods for stochastic programs, focusing on approximate dual dynamic programming. In Section 6 the approach is applied to a real world case study and some computational results are described. Section 7 concludes the thesis and gives an overview of research questions that could be addressed in future work.

2 The Deterministic Case

In this section, we want to introduce the deterministic base case of a battery storage systems. Section 2.1 outlines the general motivation and setting and section 2.2 gives an overview of the most commonly used methods to generate profit with batteries. Physical constraints of the battery model are discussed in Section 2.3 and the mathematical model is introduced in Section 2.4.

2.1 Motivation and Setting

Following Breeze (2018), the history of batteries being used at different levels of the power section dates back to the 1960s. In Japan, intensive research on Sodium Sulfur (NaS) battery systems was conducted in order to demonstrate the use of large batteries as a tool to manage power demand in a utility grid.

The focus of this thesis lies on economic benefits that can occur through the installment of a battery at the residential level. In more detail, we will focus on a union of office buildings, whose aim is to operate a battery storage system in an optimal way. The goal of the union is to maximize profits from buying and selling electricity with respect to the planning horizon of a whole year.

The existing literature on battery storage systems covers most of the application scenarios in power systems. A reason for the popularity of optimal operation problems for batteries is, first of all, their relation to classic inventory control problems, which have a much longer history. Furthermore, the amount of application scenarios in the real world has been growing rapidly in the last few years. Batteries can be found in an increasing amount of devices, such as power grids, electronic vehicles or the residential sector. As renewable power sources such as wind and solar do not have the natural means to control their energy output, additional investments on storage capacity are required.

To start with, we want to introduce the term ’smart neighborhood’, which is widely applicable. Usually it is used for a union of individual smart homes located in the same geographical area, which are connected through electricity and communication networks. The word ’smart’ normally refers to the fact that devices, which are able to control the energy demand, exist in the different buildings.

The (office) buildings considered in this work are connected by direct electricity lines.

However, it is often impossible for office buildings to adapt their load profiles because of fixed office hours, for example. This means that the considered office buildings are not "smart" by definition, but in the absence of a better alternative we will still refer to this setting as smart neighborhood.

Celik et al. (2017) state that there are two main approaches for modeling residential load. We extend these approaches for our setting:

Definition. (top-down methods)
In *top-down* approaches all energy consumption units are aggregated in one spot. Therefore, the advantage of this method is that only the total energy consumption of a house needs to be known. In most cases, this data is commonly available. The arising drawback is that information about individual peaks, types of loads, load factors and customer behavior are not included. Load profiles generated with top-down approaches are often dependent on historical data.

Definition. (bottom-up methods)
In *bottom-up* approaches, the consumption pattern of each individual household appliance is investigated. Then the load curve for each single home is obtained by aggregating these different patterns. Obtaining such concrete data is still very difficult these days. The benefit of bottom-up methods is that, by examining every device in detail, potential for improvement is easier to detect.

To satisfy the determined electricity demand, we assume that electricity is delivered by either a generating or a distribution company. The electricity providing company charges a time varying price $p(t)$ for the electricity demanded by the smart neighborhood. Prices are known in advance and are just depending on time t and not on the purchased quantity.

Bindner et al. (2005) classify the existing battery model approaches into three categories:

First, there are so termed *performance or charge models*, where the focus lies on modeling the state of charge of the battery.

The second type of models are *voltage models*, modeling the terminal voltage of a battery system. Those models are often used for more detailed modeling of battery management systems.

The last category are *lifetime models*. They are used for surveying the impact of a particular operating scheme on the expected lifetime of a battery.

Each of these models has advantages and disadvantages and the nature of the system in which the battery model is implemented is a very crucial part of the decision process.

2.2 Generating Profit with Batteries

In this thesis, we focus on a scenario where the battery installed by the consumers is used to minimize the total energy costs. In general, there are many ways to use electricity storage facilities to generate profit. We will give a short introduction of a few commonly used methods.

Arbitrage

A widely known and examined approach is generating profit through *arbitrage*. The idea is to take advantage of the fluctuating electricity prices throughout the day. By purchasing inexpensive off-peak power and selling electricity during peak hours at a higher price, profit is generated.

Alternatively a costumer who needs to consume at peak hours, because there is no way to shift the demand, could use a battery storage system to charge the battery at low electricity prices and use the stored electricity during peak hours, when the price is high.

As stated by van de Ven et al. (2013), the consumer is usually not exposed to these price fluctuations, but rather pays a fixed monthly tariff. However, there is an increasing trend towards the use of dynamic price models, even in the consumer sector. In Austria, there are already some providers which enable the consumer to buy electricity at an hourly varying price (see for example aWATTar GmbH (2021)).

As to positive effects of dynamic pricing, Borenstein (2005b) predicts better reflection of the prices on the wholesale market, lower demand peaks as well as lower and less volatile wholesale prices.

In general, consumers have the opportunity to benefit from arbitrage even if they have no battery installed. The drawback here surely is that consumers have to shift their demand to off-peak hours. In practice, as has been investigated by Allcott (2011), consumers are only willing to shift a very small part of their demand.

Application of storage in combination with renewable energy sources

Another field of interest is the application of electricity storage in combination with intermittent renewable energy generation.

The electricity output of a renewable energy plant such as wind and solar is very intermittent. A storage facility can be used to match generation with demand by storing redundant energy. By operating a battery these time shifts can be used to delay the consumption or generation of electrical energy by temporarily storing the electricity generated by the power plant.

Donnellan et al. (2015) focus their work on integrating energy storage on economic benefits at different levels of the power network. We will adopt a few of their concepts and use them in our consumer-sided setting. Amongst others, Roberts and Sandberg (2011), Coppez et al. (2010) and Oudalov et al. (2006) identified the same fields of battery storage application:

First, there is the option of using a battery for reducing the peak demand, also known as peak shaving. In this case, the storage device stores energy when the demand is lower than a specified threshold and the device is unloading when the demand is above a certain threshold.

Another application is the intermittent renewable generation shift or simply time shift, where surplus electricity from intermittent generation sources is stored and energy is provided, when electricity demand exceeds the output of the power plant. The most common application at household level is to use the storage to time-shift the energy consumption. The reduced costs for the households arise from the value of renewable energy used less the income that would have been generated if this power had been supplied to the grid. In case of a PV plant, the feed- in tariff plays an important role. In this work, we do not discuss different types of feed- in tariffs, but merely refer to (Klein et al., 2010), where the authors evaluate different feed-in tariff design options, mostly in an European context.

Another way to profit from time-shifting is through arbitrage where the benefit lies in discharging the storage when the electricity price provided by the grid is higher than the costs of providing energy from the storage.

Reliability

Furthermore, *reliability* is an important factor. In an end-user context reliability means that consumers experience fewer events where electricity supply is lost. With the increasing share of renewable power plants in the energy sector, blackout- protection is more and more demanded by end users. It is still very difficult to put a price on reliability, as customers do have different expectations of network dependability.

Reliability also covers the case of storage providing an opportunity for increasing the flexibility of renewables and the widely admitted approach to improve self-reliance of electricity consumers.

Ancillary Services

The last point we want to list here is the option to use batteries for ancillary services such as frequency regulation and black start capability.

Again, it is very difficult to put a price on such services, but there already exist different ideas on how consumers could benefit, if they provided their private storage applications for such matters.

2.3 Physical Constraints

Various physical properties can be used to store energy and in particular electricity. For example, Schainker (2004) gives an overview of different energy storage technologies.

We agree with Coppez et al. (2010)'s claim that chemical storage or battery are the most popular and frequently used methods of energy storage. The two most common types within this sector are flow batteries and normal cell batteries. As maintained by Tselepis et al. (2004), lead acid batteries are the most common used types of storage in combination with renewable energy sources. Besides their low costs and their maturity other advantages are their high efficiency, little self-discharge and their long life span.

As a special type of cell battery, we will have a closer look at *lithium ion batteries*. To understand later discussed ageing phenomena inside such a battery cell, it is important to gain knowledge about the cell performance in general.

Major components of a Li-Ion battery cell are an anode, cathode, separator, electrolyte and two current collectors (positive and negative). The anode and cathode are used to store the lithium and the electrolyte carries positively charged lithium ions from the anode to the cathode through the separator and vice versa. During discharge, Li-Ions de-intercalate from the negative electrode to the positive electrode. At the same time, electrons travel in the same direction through the external circuit. The opposite reactions take place during charging process, as provided schematically in Figure 1.

In order to get a precise representation of the reality, development of physically correct models and characterization of the associated operational policies are of very high importance. Mokrian and Stephen (2006) as well as Tan et al. (2013) claim that all energy storage systems can be modeled in the same way, using some essential physical criteria:

- Power rating, which corresponds to the maximum rate at which the battery can be charged or discharged.

- Energy capacity, which means the amount of energy that can be stored within the system.

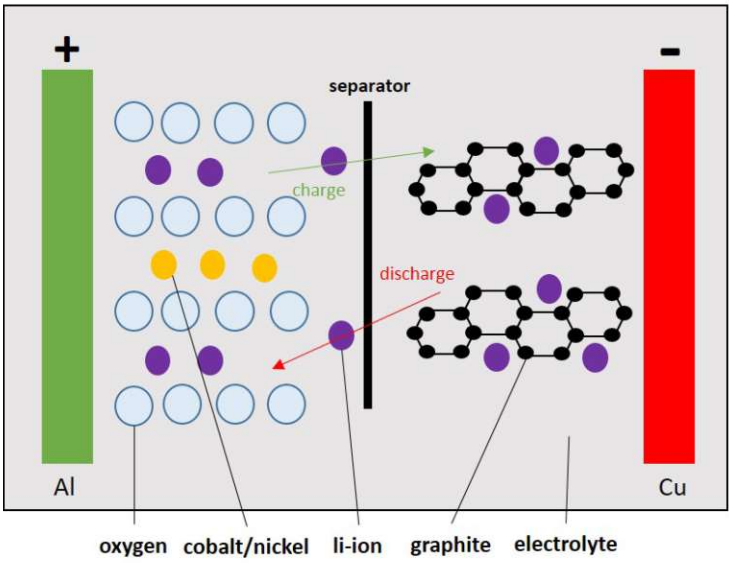


Figure 1: Generalized schematic of a Li-Ion cell

- Efficiency, which equals the ratio of energy that can be drawn from the battery to the energy supplied. Efficiency can be split into conversion efficiency, which refers to losses occurring when converting power input into a storage, and storage efficiency. This term refers to time-based losses during storage.

Based on a Figure in (Mokrian and Stephen, 2006), Figure 2 shows the basic components of power input, the storage medium and power output.

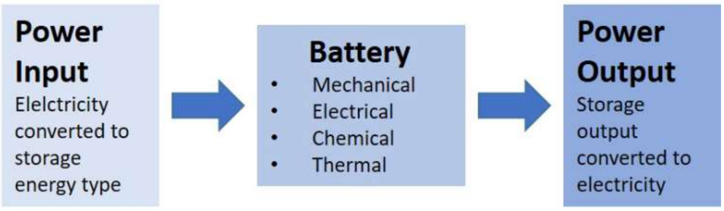

Power Input
Elelctricity converted to storage energy type
Battery
• Mechanical
• Electrical
• Chemical
• Thermal
Power Output
Storage output converted to electricity

Figure 2: Generalized schematic of a storage system

2.4 The Deterministic Battery Model

In the deterministic base case we follow the approach of Borenstein (2005b) and study a finite horizon model where electricity prices are known in advance. We base our model on the work of Haunschmied et al. (2016) and Tan et al. (2013). Keep in mind that for -now all- relevant parameters are deterministic and known in advance.

The battery operator, which in our case is the consumer, decides on the amount of electricity that is charged or discharged in each time period. We consider the decision is made periodically over a finite time horizon, denoted by $\mathcal{T} = \{0, 1, \dots T\}$. $t \in \mathcal{T}$ denotes the discrete time index corresponding to the decision period for time interval $(t, t+1]$.

In the model, $C(t) \geq 0$ [kW] denotes the amount of electricity which the operator charges into the battery at timestep t and $D(t) \geq 0$ [kW] denotes the amount of energy the operator discharges. If both those variables are Zero the battery is referred to as *idle*. Due to operating costs and efficiencies we will introduce later on, it will never be optimal to charge and discharge the battery at the same time.

The first physical constraints we include are power rating constraints. Those are often referred to as *ramping constraints*. We assume that with $\overline{C}$ and $\overline{D}$ the upper bounds for the possible amount of kW which can at most be charged or discharged are given. This results in the following constraints on the amount of electricity that can be charged or discharged at time t:

$$\begin{aligned} 0 \leq C(t) \leq \overline{C} \\ 0 \leq D(t) \leq \overline{D} \end{aligned} \tag{2.1}$$

By $q(t)$ we denote the consumers electricity demand. This demand has to be fulfilled for every timestep $t \in [0, T]$ and for now it is assumed to be known in advance. Operating a battery implies that discharged electricity reduces the necessary amount of electricity which has to be bought from the grid, whereas charged electricity increases the demand. Considering those properties, the electricity demand can be modeled as

$$Q(t) = q(t) + C(t) - D(t). \tag{2.2}$$

In case $Q(t)$ is negative, the battery operator feeds power to the grid, whereas positive $Q(t)$ accounts for buying electricity from the grid in order to fulfill the demand.

By $B(t)$, we denote the battery's charge level, or *state of charge* in kWh at time t. $B(t)$ is limited by the upper and lower bound of electricity that can be stored in the battery. We refer to those bounds as $\underline{B}(t)$ and $\overline{B}(t)$:

$$\underline{B}(t) \leq B(t) \leq \overline{B}(t) \tag{2.3}$$

Based on the charging and discharging behavior $\{C(t), D(t)\}$, we can describe the charge level of the battery through the following storage dynamics:

$$\begin{aligned} B(t+1) &= B(t) + C(t) - D(t) \\ B(0) &= B_0 \end{aligned} \tag{2.4}$$

2.4.1 Battery Efficiencies and Self-discharge

Following Mokrian and Stephen (2006), important physical parameters we have to consider are the charge and discharge efficiencies $\eta_C, \eta_D \in [0, 1]$. By including separate variables for charging and discharging, as well as the regarding efficiencies, it is possible to incorporate conversion losses into the model. Therefore, we extend the battery model by including the effects of efficiencies and self-discharge. Following Sun (2010), the efficiency of a battery can be calculated as the amount of power discharged by the battery divided by the amount of power delivered to the battery. These losses can be explained by the loss of energy through heat, which warms up the battery.

Hence, we assume that in every charging or discharging process, a fraction η_C or η_D of the energy dissipates.

Self-discharge refers to a chemical reaction in batteries, in which the stored electricity is reduced without external influence. To capture this phenomenon, we introduce a variable $\kappa \in [0, 1]$ which accounts for storage losses other than charging or discharging losses.

Including efficiencies and self-discharge in our model, we have to replace equation 2.4 with

$$B(t+1) = (B(t) + \eta_C \cdot C(t) - \frac{1}{\eta_D} \cdot D(t)) \cdot \kappa \tag{2.5}$$

2.4.2 Battery Lifetime

The conditions under which a battery is operated influence the so called *state of health* (SOH) of the battery. All types of batteries suffer from a varying amount of damage mechanisms, but to different extents. In the Benchmark project examined by Bindner et al. (2005), a clear distinction between damage mechanisms of ageing processes, which the authors defined as irreversible changes of the components of the battery or the material, and stress factors, which are defined as characteristic features of the operating conditions of the battery that alter the rate of action of the damage mechanisms, is made. The most important damage mechanisms Bindner et al. (2005) identified are:

- Corrosion of positive grid
  This phenomenon influences the internal resistance and available capacity of the battery. As the corrosion layer increases due to the reduced conductivity of the corroded material and the reduced cross section of the grid, the internal resistance increases. The reduction in capacity results from the fact that, as part of the grid corrodes, some of the active mass has reduced electrical connection to the terminals.

- Hard/irreversible sulphation
  The fundamental chemical reaction in the battery implies the production of sulphate crystals at both electrodes when the battery is discharged. During charging, those crystals dissolve. If the battery is left at low SoC for a long time, those cristals grow and lead to hard or irreversible sulphation, which implies a loss of capacity.

- Shedding
  Shedding is a process influenced by sulphation or overcharging of the battery, whereby some of the active material detaches from the electrodes and falls to the bottom of the battery.

- Active mass degradation
  Also known as softening of the electrodes, this phenomenon primarily is a change in the mechanical structure of the electrodes and the active material. This leads to a decrease in porosity and surface area of the electrolyte and active material boundary.

By those four mechanisms, the main ageing processes of a battery are listed. All these factors lead to a reduction of the battery's capacity.

Furthermore, recognized stress-factors are:

- Discharge rate

- Time at low state of charge

- Ah-throughput

- Charge factor

- Time between full charge

- Partial cycling

- Temperature

Due to high replacement costs of a battery storage system, it is important to preserve the battery life time where possible.

Ways of doing so are for example ensuring that the battery always remains within the maximum and minimum boundaries of the battery charge level, not allowing the battery to stay at low charge level for long periods or avoiding high frequencies of partial charging and discharging.

Following Bindner et al. (2005), there exist two main types of lifetime models for lead acid batteries, which can be generalized for Li-Ion batteries:

- Post-processing models
- Performance degradation models

Post processing models are pure lifetime models, when in contrast, performance degradation models combine a performance model with a lifetime model, thereby the aspect of battery degradation depending on the utilization of the battery is captured.

Tselepis et al. (2004) claim that the lifetime of a battery in cycling applications is mostly determined by its cycle life, which means the number of cycles a battery can undergo before the end of its life. To proceed on this assumption that battery capacity decreases with cycles, the end of life of a battery is mostly assumed to be reached when the capacity falls below a given value, usually 80% of the initial capacity (see 2.4.3). For more information on different Li-ion battery degradation models and their advantages and disadvantages consult Jafari et al. (2018) for example.

Following Tan et al. (2013) and Haunschmied et al. (2016), we now formulate the battery model as a lifetime-constrained model. The widespread approach of the *Ah-throughput Model*, which is explained in detail in Bindner et al. (2005) is used to connect the lifetime of the battery to its charge and discharge policy.

The Ah-throughput model pursues the approach of simply counting the amount of electricity cycled through the battery. It assumes that there exists a fixed amount of energy which can be cycled through the battery before it has to be replaced. This fixed amount is assumed to be independent of the way the battery is operated, concerning depth of the cycles or any other mentioned before stress factors.

By θ_{max}, we denote the amount of energy that can be charged or discharged over the lifetime of the battery. Consequently we have

$$\theta_{max} = \sum_{t=0}^{\tau} \eta_C \cdot C(t) + \frac{1}{\eta_D} \cdot D(t) \tag{2.6}$$

with stopping time $\tau = \inf\{t : \sum_{t=0}^{t} \eta_C \cdot C(t) + \frac{1}{\eta_D} \cdot D(t) > \theta_{max}\}$ when the battery has to be replaced. Following the Ah-throughput approach, we assume that θ_{max} is given and does not depend on the operations.

Because the planning horizon of the battery operator is much shorter than the whole lifetime of the battery, we simply ignore the stopping time problem proposed in Bindner et al. (2005). The interested reader is referred to Tan et al. (2013) where the solution of the stopping-time problem was computed.

2.4.3 Battery Decaying

The capacity of the battery will decay over time. To model the decaying process, we assume that the battery starts with an initial capacity $\hat{B}$. This state is dependent on time t. We assume that at the beginning of the planning horizon the battery is new and we have $\hat{B}(0) = \hat{B}$. At the end of the battery's lifetime, $\hat{B}(T) = \rho \cdot \hat{B}$ of the batteries' original capacity will be left. ρ is a constant and in most case studies it equals 80%.

Taking into account the decaying process over time, maximal and minimal battery levels are now related to the capacity and can be written as

$$\begin{aligned} \underline{B}(t) &= \omega_1 \hat{B}(t) \\ \overline{B}(t) &= \omega_2 \hat{B}(t). \end{aligned} \tag{2.7}$$

Coefficients ω_1, ω_2 are dependent on the depth of charge and discharge of the battery. Therefore, equation 2.3 can be rewritten as

$$\omega_1 \hat{B}(t) \leq B(t) \leq \omega_2 \hat{B}(t). \tag{2.8}$$

Assumption 2.1. *The battery's capacity will degrade when the battery is either charging or discharging. It will remain unchanged in idle mode.*

Following Haunschmied et al. (2016), under assumption 2.1 and a linear relationship between the accumulated use of the battery and the capacity, the dynamics of capacity decaying is given by

$$\hat{B}(t) = \hat{B} \cdot (1 - \frac{1-\rho}{\theta_{max}}[\eta_C C(t) + \frac{1}{\eta_D} D(t)]) \tag{2.9}$$

.

2.4.4 Battery Costs

When considering buying a battery storage system, another important aspect to study are its costs. The potential economic value and the capital costs vary depending on size and different types of models.

We consider the following types of costs for operating a battery: First, considering fixed costs M, which are directly related to the battery, like initial investments, it is possible to derive the costs per kWh charged or discharged as

$$\alpha = \frac{M}{\theta_{max}}. \tag{2.10}$$

Furthermore, we include investment costs for investments whose amortization period extends beyond the life of an individual battery, such as housing, wiring or replacement of general equipment. Therefore, we add depreciated costs δ in period t. Thus, the overall costs for operating a battery over the planning horizon T can be summarized as

$$C_B = \sum_{t=0}^{T} \alpha[\eta_C C(t) + \frac{1}{\eta_D} D(t)] + \delta T \tag{2.11}$$

2.4.5 Photovoltaic Production

Solar generation is one of the key technologies in de-carbonizing and decentralizing the energy system. The demand for photovoltaic power is constantly increasing. Therefore, it is a obvious choice to extend the model introduced by Haunschmied et al. (2016) and Tan et al. (2013) by including the possibility for end users to produce electricity via a PV plant. We denote the amount of electricity produced by the PV plant by $pv(t)$ and for now we assume that the amount of provided electricity is known in advance.

As a result of the demand and generation fluctuations over the day, batteries in combination with PV systems often operate on a daily cyclic basis. Therefore, to restrict battery degradation to a low level, batteries with good cycling performances are required.

When we adapt our model, we have to consider different possibilities for the electricity demand, resulting from operating PV panels. In general, there are two options for the system; First, it is possible that produced electricity exceeds demand at a certain timestep t. In this case, either the surplus electricity is fed to the grid, or the battery (if it is not already fully loaded) is charged.

Second, the situation of demand exceeding the electricity provided by the PV system might occur. Then, we have to decide whether the lacking electricity should be bought from the grid or be discharged from the battery (if it is not empty already).

The consumer, now called prosumer, has three different possibilities concerning the use of the produced energy. The energy can either be used to satisfy demand, stored in the battery or sold back into the main grid.

After including PV production, the electricity demand given in equation 2.2 extends to

$$Q(t) = q(t) + C(t) - D(t) - pv(t). \tag{2.12}$$

2.4.6 The Objective Function

The objective of the battery operator is to minimize the costs arising from buying electricity from the grid at price $p(t)$. This means that our goal is to determine an optimal policy for storing, selling and buying electricity. These components can be combined in the objective function in the following way:

$$\mathcal{J} = C_B + \sum_{t=0}^{T} Q(t) \cdot p(t) \tag{2.13}$$

where $Q(t)$ is derived from equation 2.12 and C_B are the battery costs obtained from 2.11.

The overall battery charging model we have derived step by step in this section can be summarized as

$$min_{C,D} \quad \mathcal{J} = C_B + \sum_{t=0}^{T} Q(t) \cdot p(t) \tag{2.14}$$

$$\begin{aligned} s.t. \quad & Q(t) = q(t) + C(t) - D(t) - pv(t) \\ & C_B = \sum_{t=0}^{T} \alpha[\eta_C C(t) + \frac{1}{\eta_D} D(t)] + \delta T \\ & B(t+1) = (B(t) + \eta_C \cdot C(t) - \frac{1}{\eta_D} \cdot D(t))\kappa \\ & B(0) = 0 \\ & \underline{B} \leq B(t) \leq \overline{B} \\ & 0 \leq C(t) \leq \overline{C} \\ & 0 \leq D(t) \leq \overline{D} \\ & \hat{B}(t) = \hat{B} \cdot (1 - \frac{1-\rho}{\theta_{max}}[\eta_C C(t) + \frac{1}{\eta_D} D(t)]) \\ & \hat{B}(0) = \hat{B} \end{aligned} \tag{2.15}$$

The model proposed above will accompany us in the rest of this thesis. As far as we are concerned, it is unique in its structure and absolutely applicable on real-world battery storage problems.

In the next chapter, we will extend the battery model to its stochastic counterpart. Naturally, the final goal is to propose a solution and in order to do so, we will develop the necessary tools step by step.

3 The Stochastic Case

The goal of this chapter is to introduce the basic concepts of stochastic optimization. It is structured as follows.

Section 3.1 outlines the general motivation about why stochastic optimization is used to model real-world problems. In Sections 3.2 and 3.3, a general introduction of stochastic programs is given. We define the classic two-stage and multi-stage model and introduce the stochastic version of our battery model. Then we take a closer look at stochastic-dynamic programming, especially Markov decision problems. This knowledge is further used to embed the battery model from 2.4 in a stochastic-dynamic programming framework. Afterwards, different measures for evaluating stochastic solutions are introduced and the risk formulation of the battery model is given. At the end of this chapter, the reasons for undergoing the effort of performing stochastic optimization are explained.

3.1 Motivation and Setting

In general, deterministic decision problems like the one we introduced in Chapter 2 are widely used in the field of optimization. However, a wide range of real-world problems occurring for example in finance, marketing, operations or industrial engineering, involve uncertainty. If a statistical model can be used to describe this uncertainty, the decision problem can be modeled as stochastic optimization problem. In the case of our battery charging problem, there are in fact several parameters which cannot be predicted with absolute certainty.

First and most obviously there is electricity output of PV-panels. The significant drawbacks of PV technology is the intermittency and variability of the electricity generation. Currently, most of the proposed software used to optimize the charging behavior of a battery, for instance in smart home energy management systems (SHEMS), requires exact weather forecasts. The problem is that weather predictions are not always accurate and if appearing uncertainties are not taken into account when solving an optimization problem, those forecasting errors can lead to substantial losses. This fact was, for example, verified by Pedrasa et al. (2011). To give an example of the variability of PV production, Figure 3 shows the PV-production profile of Germany in 2012 and Figure 4 visualizes the day/night discrepancies which occur in solar production.

Another variable one must consider when operating a battery in an optimal way is the exact behavioral pattern of the electricity consumers. Despite the difficulty of accurate

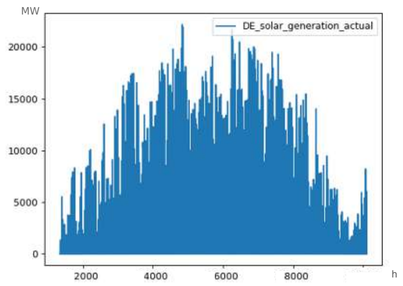


Figure 3: PV production data, Germany, 2012

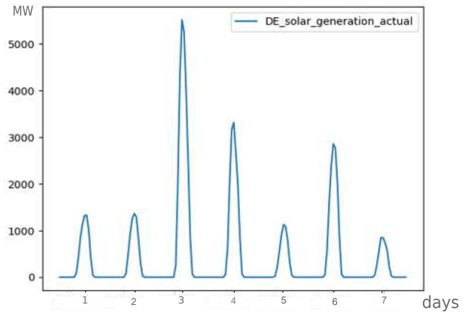


Figure 4: PV production data, Germany, first week of January 2012

prediction of different load profiles, consumption patterns evolve due to a strong periodicity. If we think for example at an office building, electricity demand is much lower at weekends or at night than it is during regular office hours.

The third uncertain variable we must deal with is the development of electricity prices. Uncertainty of the electricity price is the most studied uncertain variable in the respective literature. This is due to the significant volatility of electricity prices and their great impact on different existing market structures. It is nearly impossible to predict the prices accurately. A typical development of the day-ahead market price in Germany is shown in Figure 5.

The solution of stochastic problems is usually aimed at making cost-optimized decisions which are robust to the effect of chance. In most cases, this is done by minimizing the total costs over a certain number of scenarios, while accounting for various outliers which might occur. The existence of an optimal solution depends on either a known or at least a reasonably well estimated probability distribution of the random variables.

Generally speaking, a decision maker has the ability of influencing the behavior of a probabilistic system, either by making decisions or by choosing actions. When making decisions, the goal is to choose a sequence of actions in an optimal way, which in most cases implies maximizing or minimizing a given function. Since the system which is modeled is ongoing it is important to keep in mind that a decision made today affects the opportunities and costs associated with future system states.

The benefits of stochastic optimization were studied in great detail and are versatile.

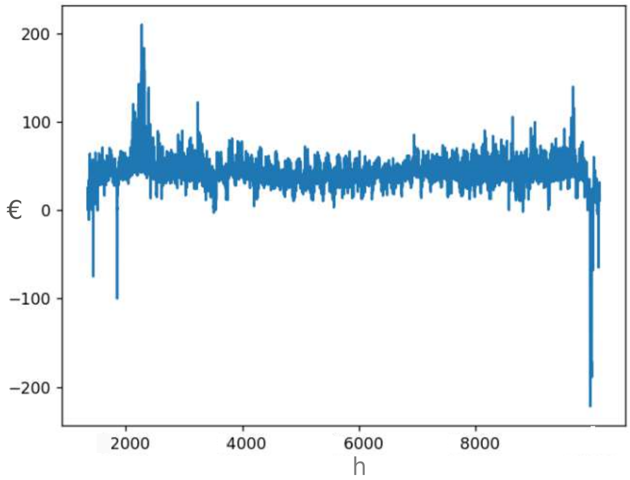


Figure 5: day ahead electricity price, Germany, 2012

It has been proven that stochastic decision-making can improve the decision in various areas. Hereinafter, a few applications are given:

Mokrian and Stephen (2006) examined a probabilistic model for the derivation of the arbitrage value of a storage over a 24 hours period. They used the deterministic approach in a scenario setting, representing the prices by their expected value and found a lower bound on profits. Moreover, they also proposed multi-stage stochastic programming, and a dynamic programming approach.

van de Ven et al. (2013) addressed the problem of organizing energy storage purchases in order to minimize long-term energy costs under variable demands and prices. They modeled the problem as a Markov decision process.

Keerthisinghe et al. (2014) analysed PV output depending on uncertain weather patterns in combination with electrical demand, which is assumed to be directly related to the number of occupants in the household at a given time. In their study, they included probabilities for occupancy transitions (i.e. all home, some home, all away).

Rahmani-andebili and Shen (2017) investigated a distributed energy resources scheduling problem of a set of smart homes. Their stochastic approach included forecasting solar irradiances and modeling the uncertainty of predictions by defining some effective scenarios.

Löhndorf and Wozabal (2017) examined the indifferent pricing of natural gas storage. The authors formulated the problem of future trading and storage operation as a Markov decision process, modeling risk preferences using the nested conditional value-at-risk.

Delong et al. (2019) proposed a stochastic optimization method based on an expected value model for electricity consumers with a PV plant. A method for storage sizing based on the expected value model is proposed, which considers the uncertainty of load and PV.

We are not aware of any existing battery charging model including all three of the above stated stochastic variables, namely electricity demand, electricity price and PV-production. However, before we adapt our model to the stochastic setting, we have to introduce a few basic definitions of stochastic programs.

3.2 Stochastic Programming

Usually, the components of a decision problem are the objective function, decision variables, constraints and problem data. As has been claimed before, in many cases, several parameters of a problem can be considered uncertain and are thus represented as random variables. Then the goal is to find optimal decisions with respect to this uncertainty.

3.2.1 Two-stage Problems

The most common way to represent stochastic programs is formulating the problem as a two-stage program. For such programs one has to divide the set of decisions into two groups:

- *first-stage decisions*, which have to be made before the random event realizes and
- *second-stage decisions* which can be used to react to the outcome of the first-stage decision, i.e. they have to be made after the realization of the random event.

There is plenty of literature on solving two-stage problems in low dimensions (see for example Birge and Louveaux (2011) or Shapiro et al. (2009)). The most common approach is to consider the so-called deterministic equivalent of the problem and to then solve it using standard LP-methods. The part of finding an appropriate deterministic equivalent is not always expedient, hence, there are other approaches to solving stochastic problems which avoid the given issue. One prominent approach is, for example, the L-shaped method, based on finding an outer linearisation of the recourse function. Then, one optimizes the given objective function with respect to a first-stage solution (see for example Birge and Louveaux (2011)).

Following Shapiro et al. (2009), a two-stage stochastic program can be defined as follows:

Definition. (Two-stage stochastic program)

$$\begin{aligned} min_{x\in\mathbb{R}^n} \quad & c^T x + \mathbb{E}[Q(x,\xi)] \\ & s.t.\ Ax = b,\ \ x \geq 0 \end{aligned} \tag{3.1}$$

where $Q(x, \xi)$ is the optimal value of the second-stage problem

$$\begin{aligned} min_{y\in\mathbb{R}^m} \quad & q^T y \\ & s.t.\ Tx + Wy = h,\ \ y \geq 0. \end{aligned} \tag{3.2}$$

The vector x represents the first stage decision and y or $y(\omega)$ or even $y(\omega, x)$ the second stage decision. The vector c represents the weights of first-stage decisions x. $\xi := (q, h, T, W)$ is the data of the second-stage problem. We assume that some or all elements of vector ξ as random and the expectation operator at the first-stage problem 3.1 is taken with respect to the probability distribution of ξ. To increase readability we often use the notation ξ for both, the random vector and its particular realization. However, the meaning will be clear from the context. If there is doubt, we write $\xi = \xi(\omega)$ to stress that ξ is a random vector defined on a corresponding probability space. By $\Xi \subset \mathbb{R}^d$, we denote the support of the probability distribution of ξ.

If for some x and $\xi \subset \Xi$ the second-stage problem 3.2 is infeasible, then $Q(x, \xi) = +\infty$ by definition. Another special case occurs if the second-stage problem is unbounded from below, hence $Q(x, \xi) = -\infty$. We try to avoid models with such properties.

Often $\mathcal{Q}(x) = \mathbb{E}_\xi(Q(x, \xi(x))$ is referred to as *expected second-stage value function* of the problem.

Considering the notation of the definition, we can summarize the sequence of events as follows:

$$x \to \xi(\omega) \to y(\omega, x)$$

Note that the definitions of first and second stage problems are only related to before and after the random experiment and may contain sequences of decisions and events.

3.2.2 Multi-stage Problems

Multi-stage problems are a natural extension of the already discussed two-stage models. In multi-stage models, the decision maker has the problem of making a whole sequence of decisions, instead of one. The uncertain data $\xi_1, \ldots, \xi_T$ is revealed gradually over time in

T periods and the decisions should be adapted to this process. In a multi-stage decision problem, the sequence of events can be displayed as

$$x_1 \to \xi_2 \to x_2 \to \cdots \to \xi_T \to x_T.$$

Following the notation of Shapiro et al. (2009), we view the sequence $\xi_t \in \mathbb{R}^{d_t},\ t = 1, \dots, T$ of data vectors as a stochastic process. With $\xi_{[t]} := (\xi_1, \dots, \xi_t)$ we denote the history of the process up to time t.

Definition. (Nonanticipativity)
The values of the decision vector x_t, chosen at stage t may depend on the information $\xi_{[t]}$, available up to time t, but not on future observations.

remark. Note that, as x_t may depend on $\xi_{[t]}$, the sequence of decisions is also a stochastic process.

Definition. (Stagewise Independence)
The process $\{\xi_t\}$ is *stagewise independent* if ξ_t is stochastically independent of $\xi_{[t-1]},\ t = 2, \dots, T$.

Definition. (Markov Process)
If for every $t = 2, \dots, T$ the conditional distribution of ξ_t given $\xi_{[t-1]}$ is the same as the conditional distribution of ξ_t given ξ_{t-1} the process is called *Markovian.*

remark. If the process is stagewise independent, then it is Markovian.

Following Shapiro et al. (2009) a T-stage stochastic programming problem can be defined as follows.

Definition. (Multi-stage Stochastic Program- Nested Formulation)

$$min_{x_1 \in \mathcal{X}_1} f_1(x_1) + \mathbb{E}[\inf_{x_2 \in \mathcal{X}_2(x_1, \xi_2)} f_2(x_2, \xi_2) + \mathbb{E}[\cdots + \mathbb{E}[\inf_{x_T \in \mathcal{X}_T(x_{T-1}, \xi_T)} f_T(x_T, \xi_T)]]] \tag{3.3}$$

where $\xi_1, \xi_2, \dots, \xi_T$ is the random data process. $x_t \in \mathbb{R}^{n_t},\ t = 1, \dots, T$ are decision variables, $f_t : \mathbb{R}^{n_t} \times \mathbb{R}^{d_t} \to \mathbb{R}$ are continuous functions and $\mathcal{X}_t : \mathbb{R}^{n_{t-1}} \times \mathbb{R}^{d_t} \rightrightarrows \mathbb{R}^{n_t},\ t = 2, \dots, T$ are measurable closed valued multifunctions. The first-stage data, ξ_1, the function $f_1 : \mathbb{R}^{n_1} \to \mathbb{R}$ and the set $\mathcal{X}_1 \subset \mathbb{R}^{n_1}$ are deterministic.

The multistage problem is called *linear*, if the objective functions and the constrain functions are linear. A typical formulation would be

$$f_t(x_t, \xi_t) := c_t^T x_t,\ \mathcal{X}_1 := \{x_1 : A_1 x_1 = b_1,\ x_1 \geq 0\},$$

$$\mathcal{X}_t(x_{t-1}, \xi_t) := \{x_t : B_t x_{t-1} + A_t x_t = b_t,\ x_t \geq 0\},\ t = 2, \dots, T.$$

$\xi_1 := (c_1, A_1, b_1)$ is the non-random first stage and $\xi_t := (c_t, B_t, A_t, b_t) \in \mathbb{R}^{d_t},\ t = 2, \dots, T$ are data vectors including random elements.

To make this formulation precise, we need to consider the following definition:

Definition. (Policy)
A sequence of measurable mappings $x_t : \mathbb{R}^{d_1} \times \cdots \times \mathbb{R}^{d_t} \to \mathbb{R}^{n_t},\ t = 1, \dots, T$, where $x_t = x_t(\xi_{[t]}),\ t = 1, \dots, T$ is understood as function of the data process $\xi_{[t]}$ up to time t, is called an *implementable policy* or just *policy*.

An implementable policy is called *feasible* if it satisfies the feasibility constraints, i.e.

$$x_t(\xi_{[t]}) \in \mathcal{X}_t(x_{t-1}(\xi_{[t-1]}), \xi_t),\ t = 2, \dots, T,\ w.p.\ 1. \tag{3.4}$$

Now, we can formulate the multistage problem 3.3 in the form

$$\begin{aligned} min_{x_1, x_2, \dots, x_T} \mathbb{E}[f_1(x_1) + f_2(x_2(\xi_{[2]}), \xi_2) + \cdots + f_T(x_T(\xi_{[T]}), \xi_T)] \\ s.t.\quad x_1 \in \mathcal{X}_1,\ x_t(\xi_{[t]}) \in \mathcal{X}_t(x_{t-1}(\xi_{[t-1]}), \xi_t),\ t = 2, \dots, T \end{aligned} \tag{3.5}$$

remark. The optimization in 3.5 is performed over implementable and feasible policies. Policies $x_2, \dots, x_T$ are functions of the data process, hence are elements of appropriate functional spaces, while $x_1 \in \mathbb{R}^{n_1}$ is a deterministic vector. Unless the data process $\xi_1, \dots, \xi_T$ has a finite number of realizations, 3.5 results in an infinite dimensional optimization problem.

3.2.3 The Stochastic Battery Model

With the tools of stochastic programming at hand, we will now embed our battery charging model from Section 2.4 into a stochastic framework.

As in the deterministic case the battery charging problem involves the optimal cost minimizing policy depending on state variables which can be influenced by the decision-maker, such as the battery's current storage level. Further, there are stochastic variables evolving independently from the decision. In our case, the stochastic variables are realized electricity price level, PV-output and actual electricity load throughout day-time hours. If we want to optimize over the time horizon T, we deal with a T-stage, 4-dimensional model.

The state space includes every possible charge, PV-production, price and consumption level. The action space is the amount of electricity that can be charged or discharged and this space is bounded by the capacity constraints introduced in Section 2.4.

To start with, we define a probability space $(\Omega, \Sigma, \mathbb{P})$, with Σ representing some filtration and $\mathbb{P}$ a probability measure.

We refer to the stochastic processes in the following way: For each sample path $\omega \in \Omega$, $p(t, \omega)$ denotes the electricity price at time t, $pv(t, \omega)$ the PV-production at time t and $q(t, \omega)$ the electricity demand at time t. The filtration $\Sigma = \Sigma(t)_{t \in T}$ is generated by the random vector $(p(t), pv(t), q(t))$, i.e. $\Sigma(t) = \sigma((p(t), pv(t), q(t)), t \in \{0, 1, \ldots, T\})$. The decision $\{C(t), D(t)\}$ related to time t has to be $\Sigma(t)$-measurable.

In order to shorten the notation, we will not mention the dependence of random variables on states ω from now on. It is important to keep in mind that every equations has to hold almost surely in a stochastic setting. Furthermore, the optimal value of $\mathcal{J}$ has to be $\Sigma(T)$ measurable random variable at the end of the planning horizon.

Taking over the notation introduced in chapter 2, the stochastic version of the problem can be written as

$$\begin{aligned} min_{C,D} \quad &= \mathbb{E}(C_B + \sum_{t=0}^{T} (Q(t) \cdot p(t)) \\ s.t. \quad & 2.15 \end{aligned} \tag{3.6}$$

where every equation from 2.15 has to hold w.p. 1. The above problem is linear at stage T, since all constraints as well as the objective function are linear. Furthermore, the problem is a convex optimization problem.

The stochastic programming models considered in this section illustrate the general form of a stochastic program. There are certain characteristics which can be used to typify them, and in the course of time many special forms have developed. Another very important way of precising the formulation of 3.3 is through the corresponding *dynamic programming* equations. We will introduce this approach in the next section.

3.3 Stochastic-Dynamic Programming

In general, the core idea in dynamic programming (DP) is to decompose a multistage, dynamic decision problem into a sequence of simpler, single-stage problems. Thereby, the most essential part is to discretize the problem's state space.

Afterwards, backwards recursion can be used to derive the optimal value and the optimal policy at every state. The state variable, which describes the problem at a certain point in time, is defined in such ways that it completely describes the process. The state of the process at the beginning of a stage is known and a decision is made at every stage allowing that the process is transformed to the ending state, once the planning horizon is reached. In general, the objective is to maximize or minimize the expected objective over all states.

The concept of system states is central to DP and one needs a mathematical model representing its evolution over time, as a function of decisions and additional external inputs. In the context of a stochastic setting, those external inputs simply correspond to random risk factors. To be able to apply DP to a problem, a specific structure of the system model is required. Following Brandimarte (2021), the state of the system at time $t+1$ should depend on the state observed at time t, the decision made at time t after observing the state, and the realization of external inputs during the subsequent time interval.

As has already been mentioned, we now want to specify problem 3.3 through the corresponding dynamic programming equations. Therefore we follow Shapiro et al. (2009) and consider the last-stage problem

$$min_{x_T \in \mathcal{X}_T(x_{T-1},\xi_T)} f_T(x_T, \xi_T).$$

We denote the optimal value of this problem as $Q_T(x_{T-1}, \xi_T)$, which is dependent on the decision vector x_{T-1} and data ξ_T. At stage $t = 2, \ldots, T-1$ we formulate the problem

$$\begin{aligned} & min_{x_t} f_t(x_t, \xi_t) + \mathbb{E}\{Q_{t+1}(x_t, \xi_{[t+1]})|\xi_{[t]}\} \\ & s.t. \quad x_t \in \mathcal{X}_t(x_{t-1}, \xi_t), \end{aligned} \tag{3.7}$$

where $\mathbb{E}[\cdot|\xi_{[t]}]$ denotes the conditional expectation. The problem's optimal value depends on the decision x_{t-1} at the previous stage and the realization of the data process $\xi_{[t]}$, which we will denote by $Q_t(x_{t-1}, \xi_{[t]})$. Now, we want to calculate the *cost-to-go* functions $Q_t(x_{t-1}, \xi_{[t]})$ recursively. At the fist stage, we therefore need to solve

$$min_{x_1 \in \mathcal{X}_1} f_1(x_1) + \mathbb{E}[Q_2(x_1, \xi_2)].$$

The corresponding dynamic programming equations are

$$Q_t(x_{t-1}, \xi_{[t]}) = inf_{x_t \in \mathcal{X}_t(x_{t-1}, \xi_t)} \{f_t(x_t, \xi_t) + \mathcal{Q}_{t+1}(x_t, \xi_{[t]})\}, \tag{3.8}$$

where

$$\mathcal{Q}_{t+1}(x_t, \xi_{[t]}) := \mathbb{E}\{Q_{t+1}(x_t, \xi_{[t+1]}) | \xi_{[t]}\}.$$

Definition. (Optimal Policy)
An implementable policy $\bar{x}_t(\xi_{[t]})$ is *optimal* iff for $t = 1, \ldots, T$,

$$\bar{x}_t(\xi_{[t]}) \in arg \min_{x_t \in \mathcal{X}_t(\bar{x}_{t-1}(\xi_{[t-1]}), \xi_t)} \{f_t(x_t, \xi_t) + \mathcal{Q}_{t+1}(x_t, \xi_{[t]})\}, \; w.p. \; 1, \tag{3.9}$$

where for $t = T$ the term $\mathcal{Q}_{T+1}$ is omitted and for $t = 1$ and the set X_1 depends only on ξ_1.

In the dynamic programming formulation, the problem is reduced to solving a family of finite dimensional problems, indexed by t and $\xi_{[T]}$. A main drawback of most of the dynamic programming algorithms, which are developed until today, is that there are often limitations in their application due to the so-called "curse of dimensionality". This phenomenon refers to the rapid expansion of the size of the model with a growing number of states. For the problem to stay tractable, the dimension of the state space must stay small (according to Mokrian and Stephen (2006) no more than 3 or 4 dimensions) and the individual discretization of each dimension must be relatively coarse. This limits the ability of a DP to properly address continuous state and action spaces without great computational expenses.

Another problem which can occur using DP methods is due to the necessary discretization of the environmental-, resource- and the action space. Mokrian and Stephen (2006) mention that the discretization can be problematic, if storage and conversion inefficiencies are included in a battery model, because it is often necessary to round to the nearest state. The problem therein is that the algorithm which is used to solve the optimization model optimizes over this rounding process. In extreme cases, situations in which the profit of the dynamic program exceeds the profit of the deterministic problem can occur. This is clearly problematic because, as we will see in Section 3.6, the deterministic problem should provide an upper bound for the stochastic solution.

Nonetheless, solving a DP can be quite useful. The solution provides an optimal policy for every realizable state of the system. Another advantage is that many decision stages can be considered, as the problem grows linearly in the number of stages.

The stochastic-dynamic decision problem is to select a policy such that the sequence of rewards is optimized. In the following, we take a closer look at a special type of stochastic-dynamic problems, the so called *Markov Decision Problems* (MDPs). Unless the evolution of the stochastic process depends on the complete sequence of previous states and actions, many stochastic-dynamic decision problems arising in real world scenarios can be modeled as MDPs.

3.3.1 Markov Decision Problems

The term "Markov", which we already introduced in Section 3.2.2, is used to highlight the characteristic of a stochastic process that transition probability and reward functions depend on the past, only through current state of the system and action selected by the decision maker in that state.

Markov decision theory, like all other types of optimization problems, is concerned about finding an optimal policy with the largest (or smallest) expected total reward (or loss).

A main characteristic of Markov decision problems is that they can be formulated as recursive functions, which relate the value of currently being in a state at the beginning of a decision epoch to the value of the states that might occur during subsequent periods. Under this property, problem 3.7 simplifies to

$$\begin{aligned} min_{x_t} f_t(x_t, \xi_t) + \mathbb{E}\{Q_{t+1}(x_t, \xi_{[t+1]})|\xi_t\} \\ s.t. \quad x_t \in \mathcal{X}_t(x_{t-1}, \xi_t). \end{aligned} \tag{3.10}$$

3.3.2 The Stochastic-Dynamic Battery Model

The goal in this section is to model the decision process of the battery storage problem as a finite horizon Markov decision process. Therefore, we define a discrete stochastic process on a probability space $(\Omega, \Sigma, \mathbb{P})$, including the variables PV-production, electricity price and consumption and denote it by $(S_t)_{t=1}^T$. We refer to $S_t \in \mathcal{S}_t$ as environmental state, where $\mathcal{S}_t$ is the set of environmental states in t. Furthermore, assume that S_t is adapted to the filtration Σ_1.

The objective of the decision maker is again to minimize the expected costs for fulfilling the electricity demand for a given environmental state S_t and an initial storage state $B_{t-1} \in \mathcal{B}$ at stage $t \in \{1, \dots, T\}$. $\mathcal{B}$ denotes the set of all possible storage states. Let $\mathbb{P}(S_{t+1}|S_t)$ be the state transition probability of the Markov process, x_t for $t = 1, \dots, T$ a decision policy, $f_t(S_t, B_{t-1}, x_t)$ denotes the random profit. Additionally, we introduce a discount factor γ. By introducing a fixed storage state B_0 and V_{T+1}, the value of being in state S_t with initial reservoir states B_{t-1} is given by the following optimality equations:

$$V_t(S_t, B_{t-1}) = \min_{x_t}\{\mathbb{E}[f_t(S_t, B_{t-1}, x_t) + \gamma \sum_{S_{t+1} \in \mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|S_t) V_{t+1}(S_{t+1}, B_t(x_t))]\}, \quad (3.11)$$

for $S_t \in \mathcal{S}_t$, $B_{t-1} \in \mathcal{B}$ and $t = 1, \dots T$. Note that x_t assigns a decision to every possible realization of the environmental state. This results in random reservoir states $B_t(x_t)$. Equation 3.11 can be interpreted in the following way: An optimal decision policy minimizes the sum of expected immediate and expected future costs. Those future costs are dependent on the state transitions from S_t to S_{t+1} and the final storage state $B_T = B_t(x_T)$. In other words: At each stage of the planning horizon the system generates an immediate income or loss and the storage level gets updated. If we follow the stochastic-dynamic programming approach to calculate the expected profit of each action, one has to consider the sum of the immediate profit or loss and the expected value of the next state.

After introducing the stochastic and the stochastic-dynamic version of the battery charging model, we take the model a little bit further. In the real world, most people do not behave in a totally risk neutral way. There are people who like to speculate in stock exchange and those, who take an umbrella to their trip to the Sahara desert. Mathematicians try to capture this aspect of the human behavior by dealing with risk measures.

3.4 Measures for Evaluating Solutions

The mathematical programming models introduced in sections 3.2 and 3.3 maximize the expected value of the objective function and are therefore called the *expected value models*. Expected value models are the mostly used models in approaches to minimize expected costs, maximize the expected value, etc. In general, especially when considering financial decisions, one has to consider two dimensions. The *value dimension* which in most cases measures the expected return and the *risk dimension* which is measured by a risk parameter.

As further extension to the existing literature on battery modeling, we will include risk preferences of the battery operator which we model by coherent acceptability functionals. Therefore, we introduce different types of decision makers.

Hence, by following Fichtinger (2010) we give the following definition:

Definition. (Risk)
Let Π be a random variable, where the utility $u(\Pi)$ is monotone in the realization of Π. We define the term 'risk' as a measure $\rho(\Pi)$ on Π of one-sided deviations from an arbitrarily chosen value m.

Furthermore, we say a decision maker is

(i). *risk-neutral*, if he or she does not take risk into account,

(ii). *risk-averse*, if his or her objective is the minimization of undesirable deviations from m, and

(iii). *risk-seeking*, if his or her objective is the maximization of desirable deviations from m.

3.4.1 Risk functionals

For the subsequent observations we follow Pflug and Romisch (2007) and consider a probability space $(\Omega, \Sigma, \mathbb{P})$. Additionally, defined on this space is a linear space of real-valued random variables, namely either the space $\mathcal{Y}$ of all real functions on (Ω, Σ) or a subset of integrable functions as the L_p-spaces.

Definition. (Deviation risk functional)
A real-valued mapping $\mathcal{D}$ defined on a linear space $\mathcal{Y}$ of random variables on $(\Omega, \Sigma, \mathbb{P})$ is called deviation risk functional, if it exhibits the following properties (D1)-(D3) for all $Y \in \mathcal{Y}$:

(D1) **Translation invariance**

$$\mathcal{D}(Y + c) = \mathcal{D}(\mathcal{Y})$$

for constant c.

(D2) **Convexity**

$$\mathcal{D}(\lambda Y^{(1)} + (1 - \lambda)Y^{(2)}) \leq \lambda\mathcal{D}(Y^{(1)}) + (1 - \lambda)\mathcal{D}(Y^{(2)})$$

for $0 \leq \lambda \leq 1$.

(D3) **Monotonicity**

$$X \leq Y \quad \textit{implies that } \mathbb{E}(X) - \mathcal{D}(X) \leq \mathbb{E}(Y) - \mathcal{D}.$$

The following definition of *coherent risk functionals* was first introduced by Artzner et al. (1999):

Definition. (Coherent measure of risk)
A mapping ρ defined on some linear space $\mathcal{Y}$ of real valued random varibales on $(\Omega, \Sigma, \mathbb{P})$ is called *coherent risk functional*, if the following properties (R1)-(R4) are satisfied for all $Y \in \mathcal{Y}$.

(R1) **Translation antivariance**

$$\rho(Y + c) = \rho(Y) - c$$

for constant c.

(R2) **Convexity**

$$\rho(\lambda Y^{(1)} + (1-\lambda)Y^{(2)}) \leq \lambda\rho(Y^{(1)}) + (1-\lambda)\rho(Y^{(2)})$$

for $0 \leq \lambda \leq 1$.

(R3) **Positive homogenity**

$$\rho(\lambda Y) = \lambda\rho(Y)$$

for $\lambda \geq 0$.

(R4) **Pointwise antimonotonicity**

$$Y^{(1)} \leq Y^{(2)} \Rightarrow \rho(Y^{(1)} \geq \rho(Y^{(2)})$$

In the following, we describe the use of particular risk measures such as the *Value-at-Risk* and the *conditional Value-at-Risk*.

Value-at-Risk (VaR)

The Value-at-Risk, VaR in short, is a very popular risk measure which has even achieved the status of being written into industry regulations (see for example Jorion (2000)). It is used to consider one-sided deviations from the expectation.

Originally, the VaR was developed by JP Morgan (1994) to answer the following questions:

- How much one can expect to lose in one day, week, year, ... with a given probability?
- What is the percentage of the value of the investment that is at risk?

We took the following definition from Rockafellar and Uryasev (2000) and adapted it to loss distributions.

Definition. (Value-at-Risk)
The $VaR_{1-\alpha}$ associated with a given random variable Π is

$$VaR_{1-\alpha}(\Pi) = \inf_{u}\{u : \mathbb{P}(\Pi \leq u)\} \geq 1 - \alpha \tag{3.12}$$

where α is typically chosen to be some small probability, e.g. 0.05.

Statistically, $VaR_{1-\alpha}$ is the left-side $1 - \alpha$-percentile of the random variables loss distribution. In other words, $VaR_{1-\alpha}$ is simply the minimum outcome of a random variable within a certain confidence interval α. Thereby, the width of the confidence interval reflects the level of risk aversion, meaning that a larger confidence interval - implying a smaller α - refers to higher levels of risk aversion.

Following Rockafellar and Uryasev (2000), a main drawback is that the measure is unstable and difficult to work with numerically, when losses are not normally distributed. Moreover, VaR fails to be coherent. Another major point is the definition of $VaR_{1-\alpha}$ as a single point in the value distribution. In general, the definition as the worst outcome of a certain confidence interval sounds helpful, but it implies a necessity to consider the best possible outcome for describing the risk associated with the worst cases of a distribution as well. This means that there is no account for all other losses, regardless of how serious they are. Rockafellar and Uryasev (2000) state that "[i]t is incapable of distinguishing between situations where losses that are worse may be deemed only a little bit worse, and those where they could well be overwhelming.".

Especially for value distribution with fat (left) tails this characteristic is very problematic.

A natural extension of the VaR including the property of being able to quantify for losses that might occur in the tail is the *conditional value-at-risk* (CVaR).

Conditional Value-at-Risk (CVaR)

In contrast to the $VaR_{1-\alpha}$, the $CVaR_{1-\alpha}$ represents not only the $1-\alpha$-quantile itself, but rather describes the expected value of the realizations of a quantity at risk that is above the $1-\alpha$-quantile. Thus, the $CVaR_{1-\alpha}$ corresponds to the average loss triggered by exceeding the $VaR_{1-\alpha}$. While the VaR represents the maximum loss that will not be exceeded with a certainty of $1-\alpha$, the $CVaR_{1-\alpha}$ implies the average loss outside the safety level, i.e. in all other $\alpha \cdot 100\%$ bad cases. Figure 6 shows the connection between $VaR_{1-\alpha}$ and $CVaR_{1-\alpha}$.

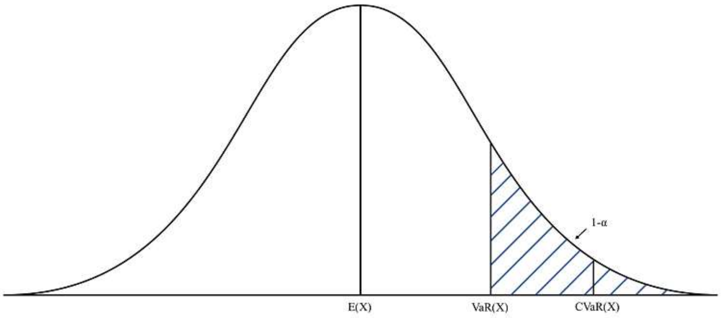


Figure 6: Connection between VaR and CVaR

First introduced by Rockafellar and Uryasev (2000), one can interpret the $CVaR_{1-\alpha}$ as response to the conceptual problems of $VaR_{1-\alpha}$. Following Fichtinger (2010) the conditional Value-at-Risk has several advantages over the Value-at-Risk:

- It is coherent in the sense of the Definition in Section 3.4.1 (see Pflug (2001) for proof).
- It takes events below $F_{\Pi}^{-1}(\alpha)$ into account.
- It can be formulated as a maximization problem and allows for incorporation into optimization problems on decision y affecting the random variable of outcome $\Pi(y)$.

Definition. (Conditional Value-at-Risk)
The conditional Value-at-Risk for a given confidence level α is defined as

$$CVaR_{1-\alpha}(\Pi) = \inf_{u}\{u + \alpha^{-1}\mathbb{E}[(\Pi - u)_+])\}. \tag{3.13}$$

where $(\cdot)_+)$ means $\max\{0, \cdot\}$.

remark. The use of $CVaR_{1-\alpha}$ as an objective function implies sensitivity towards lower values of profit. If we consider $\alpha = 0$, the problem is identical to the risk-neutral problem.

However, for any $0 < \alpha < 1$ only a lower fraction of the sample space is considered. For example, if we consider $\alpha = 5\%$, this implies that the decision maker only takes the 5% worst outcomes of the sample space and bases his or her decisions on these events.

remark. If we assume that the random variable has a continuous, strictly monotone increasing cdf, the following equation holds true:

$$CVaR_{1-\alpha}(\Pi) = \mathbb{E}(\Pi|\Pi \leq VaR_{1-\alpha}(\Pi)).$$

Corollar 3.1. *(Convexity of CVaR)*
If $\Pi(y)$ is convex w.r.t. y, then $CVaR_{1-\alpha}(\Pi(y))$ is convex w.r.t. y as well. Indeed, in this case $\Gamma(y, \psi)$ is jointly concave in (y, ψ).

Proof. See Rockafellar and Uryasev (2002) for a proof. □

3.5 The Risk Formulation of the Battery Problem

In contrast to sections 3.2 and 3.3 we now want to define the objective value as a mixture of expectation and conditional value at risk. Therefore, we have

$$\begin{aligned} \min_{C,D} \quad & \lambda \cdot \mathbb{E}(C_B + \sum_{t=0}^{T} Q(t) \cdot p(t)) + (1-\lambda) \cdot CVaR_{1-\alpha}(C_B + \sum_{t=0}^{T} Q(t) \cdot p(t)) && (3.14) \\ s.t. \quad & 2.15 \\ & C(t), D(t) \lhd \Sigma && (3.15) \end{aligned}$$

where $0 \leq \lambda \leq 1$ is the corresponding weighting factor. The constraint 3.15 means that the decision process has to be adapted to the underlying filtration Σ, i.e. that decisions at time t are only based on information available up to time t.

3.6 The Value of the Stochastic Solution

Stochastic programs have the reputation of being computationally difficult to solve, so people often try to solve simplified versions. In many cases, people ignore the stochastic part of the problem and instead, they solve the deterministic problem obtained by replacing all random variables with their expected values. The question arising is whether this approach can be accurate or not. Trying to answer this question, we follow Pflug and Romisch (2007) and Birge (1982) to introduce the concept of the *expected value of perfect information* (EVPI) that has generally been used to determine the importance of uncertainties in mathematical models and the *value of the stochastic solution* (VSS), which is used to estimate how well a deterministic model performs relatively to solutions from more complicated stochastic programs.

Therefore, for our observations we recall the stochastic linear program introduced in 3.3.

If the decision maker has perfect information, i.e. if he or she is able to look into the future and foresee the realizations of the random variable, he or she would choose optimal first stage decision for each realization of ξ. The decision maker's decisions may depend on ξ, i.e. may be $\Sigma_1 - measurable$, with Σ_1 being the σ-algebra generated by ξ. The expected value of this solution is known as *wait-and-see* solution or *clairvoyant* solution:

Definition. (Wait-and-See Solution)

$$WS = \mathbb{E}_\xi[\min_{x_t} f_t(x_t, \xi)] \tag{3.16}$$

In general, it is assumed in a multi-stage stochastic optimization program that the decision x_t must be taken before the realization of the random variable ξ. This means that when deciding, the decision maker does not know the actual outcome of the random variable, he or she just knows the distribution of ξ. We refer to such decisions as *here-and-now decisions*.

Definition. (Here-and-Now Solution)
The here-and-now solution corresponding to the recourse problem defined in 3.3 can be written as

$$RP = min_{x_t}\mathbb{E}_\xi[f_t(x_t, \xi)] \tag{3.17}$$

Due to the complexity and large size of the recourse problem, in practice people almost always try to avoid it. Instead an approximation of the real problem is solved:

Definition. (Expected Value Problem)
The expected value problem is defined as

$$EV = \min_{x_t} f_t(x_t, \bar{\xi})), \tag{3.18}$$

where $\bar{\xi} = \mathbb{E}(\xi)$. In this problem, every random variable is replaced with its expected value.

The solution of problem 3.18 is called the *expected value solution* and we denote it by $\bar{x_t}(\bar{\xi})$. Then the expected result of using the EV solution is

$$EEV = \mathbb{E}_\xi[f_t(\bar{x_t}(\bar{\xi}), \xi)]. \tag{3.19}$$

In words, EEV measures how $\bar{x}_t(\bar{\xi})$ performs.

The following relation between the defined values was established by Madansky (1960).

Proposition 3.2.

$$WS \leq RP \leq EEV$$

Proof. This result follows directly from Jensen's inequality and the convexity of $f_t(x_t, \xi)$. □

An important indicator for the effect of uncertainty on stochastic programs is the *expected value of perfect information*:

Definition. (Expected Value of Perfect Information)

$$EVPI = RP - WS$$

The EVPI measures the maximum amount a decision maker would pay in return for complete information about the future.

The loss in the objective value between the result of using an expected value solution and the solution of the recourse problem is called the *true value of the stochastic solution.*

Definition. (Value of the Stochastic Solution)

$$VSS = RP - EEV$$

This quantity represents the cost of ignoring uncertainty in choosing a decision.

Proposition 3.3. *(Relations between EVPI and VSS)*
For any stochastic program

$$0 \leq EVPI \tag{3.20}$$

$$0 \leq VSS \tag{3.21}$$

and for stochastic programs with fixed recourse matrix and fixed objective coefficients,

$$EVPI \geq EEV - EV \tag{3.22}$$

$$VSS \geq EEV - EV \tag{3.23}$$

Proof. Proof can be found in Birge and Louveaux (2011). □

remark. 3.3 indicates that EVPI and VSS are not negative and bounded from above by the same quantity. It clearly follows that if $EVV = EV$, both terms vanish. Although this result is sufficient for $VSS = EVPI = 0$, the conditions for $VSS = 0$ and $EVPI = 0$ are not the same.

After understanding the basic concepts of stochastic optimization, the next step is to get familiar with different methods of how stochastic variables can be included in decision problems.

4 Modeling Uncertainty

A widespread approach of including uncertainty in a model is by representing the stochastic data through stochastic data processes, which are used as input parameters in the optimization model. Assumptions must be made about distributions as well as the decision-making structure. A common strategy to represent randomness is to construct a scenario tree which represents the uncertainty. Generally, stochastic optimization problems can only be treated numerically in most cases, if the involved stochastic process can be approximated by a finite number of scenarios. Only under those circumstances, it is possible to represent the scenarios by a scenario tree. This means that the underlying probability distribution of the stochastic process is replaced with a discrete distribution with a finite number of atoms.

One has to face a few challenges if constructing scenario trees. The main problem is that the complexity of the optimization scales with the number of scenarios. Therefore, a lot of work is dedicated to techniques reducing the number of scenarios while retaining the quality of the wanted stochastic solution.However, before reducing, one has to generate different scenarios. The goal hereby is to obtain a set of scenarios a priori which minimizes the approximation error.

In the following, we introduce the theory behind scenario trees and the less-known scenario lattices. Afterwards, we summarize what can be considered state-of-the-art in generating scenarios of multivariate random variables.

4.1 Scenario Trees

To begin with, we give a short introduction on the theory behind scenario trees.

Definition. (Scenario)
A *scenario* at stage t is an outcome of the random process $\xi^t = (\xi_1, \dots, \xi_t)$. The set of all scenarios can be described as $\mathcal{S} = \{\xi^1, \dots, \xi^S\} = \{\xi | \xi_t \in \mathcal{S}_t(\xi^{t-1}),\ \forall\ t > 1\}$, where $\mathcal{S}_t(\xi_t)$ is the support of the conditional probability distribution of ξ_t conditioned on all prior realizations.

Under the assumption that the probability distribution $\mathbb{P}$ of the stochastic process ξ is discrete and has finite support, a *scenario tree* is an explicit representation of the branching process induced by the gradual observation of $\xi_1, \dots, \xi_T$. We call the node associated with the first decision state *root node*. The root node is connected to so called *child nodes* associated to stage 2, one child node for each possible outcome of the random variable ξ_1. Each child node of stage 2 has again its own child nodes, associated with stage 3, one for each outcome of ξ_2, given the observation of ξ_1. This construction process continues until a terminal node is reached. This last nodes are called *leafs*. The unique path from the root node to a leaf defines a scenario.

Furthermore, probability distributions of the random variables are taken into account. Probability masses are associated with the nodes of the scenario tree. Clearly, the root node has probability 1, whereas children nodes are weighted according to the probability to which two nodes are connected. Multiplying the individual probabilities of the nodes of a path gives the total probability of a scenario.

Definition. (Arc probabilities, Path probabilities)
Arc probabilities are the conditional probabilities of ξ_t conditioned on ξ_{t-1},i.e. $\mathbb{P}(\xi_t|\xi^{t-1})$. Path probabilities are defined as the products of arc probabilities up to time $t-1$, i.e. $\mathbb{P}(\xi^{t-1}) = \mathbb{P}(\xi_1)\prod_{i=2}^{t-1}\mathbb{P}(\xi_i|\xi^{i-1})$.

Keeping all those definitions in mind, we are now able to give a concrete definition of a *scenario tree*:

Definition. (Scenario Tree)
A scenario tree is an oriented graph consisting of edges and nodes, with every node having a unique predecessor. Each node represents a possible state and each edge represents a transition between states. The set of nodes V can be disjointly partitioned into $V = \dot{\bigcup}\,\mathcal{N}_t$ with $\mathcal{N}_t$ being the nodes belonging to stage t. Moreover, a probability p_i is assigned to every edge $e_i \in E$, representing the arc probability as defined above.

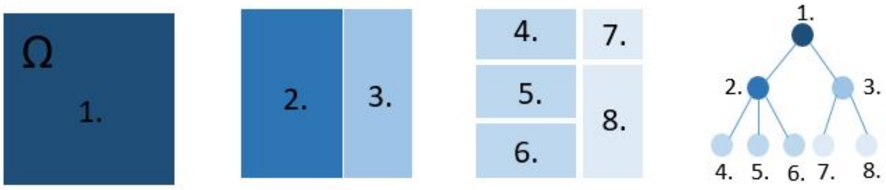


Figure 7: available information

It is important to note the following special features of stochastic scenario trees; In contrast to deterministic scenario trees, one has to consider that, although the decision maker may contemplate as many hypothetical scenarios as desired, the decisions cannot depend on observations that are not yet available when the decision is made. Figure 7 is based on a figure in Defourny et al. (2012) and shows how information on the realization of the random variables becomes gradually available:

At each decision stage, refinements of the partitions of the event space Ω appear gradually in correspondence with possible realizations of new observations. Each subregion arising from this division of the event space can be associated a constant recourse decision. In Figure 7, the area of each subregion could also represent probabilities, for example, the initial square would be allocated to an unit surface. This dynamic process can be captured by a scenario tree: the nodes are corresponding to the subregions of the event space and the edges connect a parent region to its refined subregions obtained by one step of the recursive partitioning process.

4.2 Scenario Lattices

If the stochastic process one wants to include is Markovian the conditional distribution of future states in $t+1$ at stage t does not depend on the entire history of the process but only on the values in t. If we wanted to discretize such a process to a scenario tree, many branches of the tree would have identical sub-trees. The idea of *scenario lattices* is to take advantage of this characteristic and combine the identical trees without loosing any information.

Concretely speaking, a lattice is a graph organized in a finite number of layers. Each discrete point in time is represented by one of those layers and contains a finite number of nodes. Two nodes in successive layers are connected by arcs. Similar to scenario trees, a node represents a possible state of the stochastic process and an arc stands for the probability of transiting from one node into another in a successive layer. Furthermore, each arc is associated with a probability weight, and the weights of all outgoing arcs of one layer sum up to one. The difference to scenario trees is that we do not require for each node to have only one predecessor.

The advantage of this relaxation will be clear after the following construction:
Let N_t be the number of nodes in t and $\bar{F}_{tn}, n \in [N_t]$ the state of the stochastic process in node n at time t. Further, denote $\bar{F}_t = \{\bar{F}_{tn} : n \in [N_t]\}$ as the set of all possible states in the lattice layer corresponding to time t. The set of possible scenarios on the lattice is then given by

$$\bar{F}_1 \times \bar{F}_2 \times \cdots \bar{F}_{T-1} \times \bar{F}_T.$$

As one can see, if more stages are added, the additional nodes needed to expand the lattice are those of the new added stage. In contrast, the number of nodes in a scenario tree would grow exponentially. An illustrative comparison can be found in Figure 8.

The question now arising is how to construct scenario trees and lattices. The key are different scenario generation methods. Over time, the field of scenario generation has grown and today there are numerous approaches on how to generate scenarios. In the following section, we will give an introduction of a few selected methods.

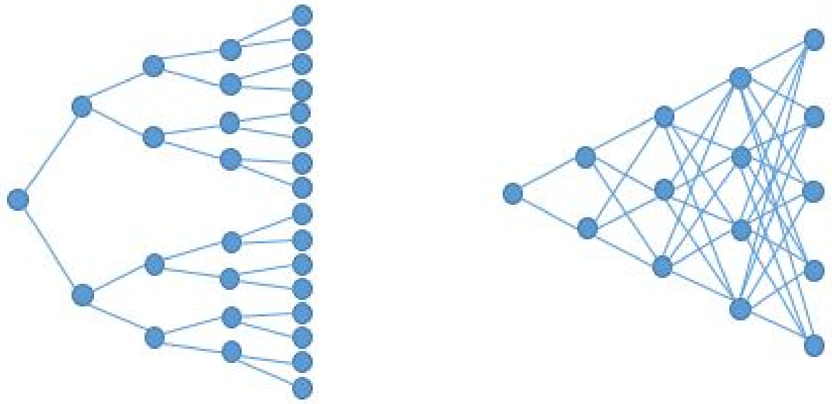


Figure 8: A tree with 31 nodes representing 16 scenarios compared to a scenario lattice with 15 nodes representing 120 scenarios

4.3 Scenario Generation Methods

One of the most commonly used technique to solve real world stochastic optimization problems via scenario tree construction is *sample average approximation*(SAA). The main idea of SAA was explained by Birge and Louveaux (2011) or Shapiro et al. (2009) and is to approximate the probability distribution by a set of discrete scenarios.

As mentioned before, the complexity of a stochastic problem grows with the number of stages, so a great effort is made in finding techniques which reduce the number of scenarios while retaining the quality of the stochastic solution. The aim is to find a set of scenarios which minimizes the approximation errors.

A well researched field are scenario reducing techniques for underlying univariate distributions, typically referred to as *variance reduction techniques* (see for example Shapiro (2003)).

There exist different approaches that extends the methods for univariate distributions to the more general multivariate cases. In the course of this section we will give an overview of what can be considered as state-of-the art to generate scenarios of multivariate random variables for sample average approximation. The presented methods include Monte Carlo methods, methods based on probability metrics, optimal quantization, moment matching and voronoi cell sampling.

The outline of this section is as follows: We start with giving an overview of the general concept of SAA, followed by a description of the existing approaches to generate scenarios for stochastic optimization.

4.3.1 Sample Average Approximation

For an outline of the SAA approach we consider the stochastic optimization problem

$$\min_{x \in X} \{F(x) = \int f(x, z) dG(z)\} \tag{4.1}$$

where x is as usual a decision variable defined on a feasible set $\mathcal{X} \subseteq \mathbb{R}$ and z is a k dimensional random realization of the stochastic variable Z. The random variable is defined by the distribution function $G : \mathbb{R}^k \to [0, 1]$. $f(x, z)$ denotes the cost function we want to minimize by choosing an optimal action $x^* \in \operatorname{argmin}_{x \in \mathcal{X}} F(x)$.

A common way to calculate numerical solutions is to either draw a sample from G or approximate G by a discrete distribution $\hat{G}$. Using $\hat{G}$ instead of G leads to the following problem:

$$\hat{F}(x) = \sum_{\hat{z} \in \hat{G}} \hat{p}(z) f(x, \hat{z}),$$

where $\hat{p}(z)$ denotes the probability of the mass points of $\hat{G}$.

As mentioned before, it would be desirable for a scenario tree to cover all possible outcomes of a random process. However, if the distribution of the random variables is not discrete or the number of stages is not finite, a scenario tree can not exactly represent the random process with a finite number of nodes. The question of how to find good scenario tree approximations in order to extract good decision policies is widely addressed in the literature.

In the following, we will give an overview of commonly applied approaches to generate scenarios of multivariate random variables with sample average approximation, along with their advantages and drawbacks.

4.3.2 Monte Carlo Methods

Monte Carlo Methods are a widespread approach to solve stochastic optimization problems using SAA. A popular class of methods for Monte Carlo sampling, especially in higher dimensions, are quasi-Monte Carlo methods. Originating from number theory, these methods rely on low-discrepancy sequences covering the unit hypercube as uniformly as possible. After an adequate transformation, these sequences can be treated like pseudo-random numbers.

To use Monte Carlo sampling for SAA, we follow the notation of Löhndorf (2016) and create a set of M uniformly distributed, pseudo-random realizations $u_1, \dots, u_m$ with $u_i \in [0,1]^n$. As proposed by Glasserman (2013) we construct a sample from the original distribution function by an appropriate transformation $U \to Z$. If we assume that $z_1, \dots z_M$ is a sample of random realizations from the same distribution as G, we obtain the SAA by evaluating

$$\hat{F}_{MC}(x) = \frac{1}{M} \sum_{i=1}^{M} f(x, z_i). \tag{4.2}$$

The variance is given by

$$\hat{\sigma}^2_{MC} = \frac{1}{M} \mathbb{V}ar(f(x, Z)). \tag{4.3}$$

Shapiro (2003) shows that the Monte Carlo estimate $\hat{F}_{MC}(x)$ converges to $F(x)$ with probability 1, as $M \to \infty$. Nevertheless, to accelerate the speed of convergence of the error bounds it would be desirable to reduce the variance of the estimate more quickly. Shapiro (2003) for example proposes a scenario reduction technique for Monte Carlo sampling.

4.3.3 Methods based on Probability Metrics

The idea behind using probability metrics for scenario reduction is to evaluate the closest approximation of a probability distribution to a discrete distribution with smaller support. Here, a probability metric is used as the objective criterion and can be related to the error occurring from implementing the optimal solution of a stochastic problem using SAA. Following Pflug (2001), another approach is to address the approximation error directly. The approximation error $e(F, \hat{F})$ is defined as the price (measured in the objective function F) one has to pay if optimizing $\hat{F}$ instead of the true F, i.e.

$$e(F, \hat{F}) = F(\operatorname*{argmin}_x \hat{F}(x)) - F(\operatorname*{argmin}_x F(x)). \tag{4.4}$$

In most cases, the error defined in 4.4 is difficult to evaluate. Therefore, one must be satisfied by determining an upper bound of the error by using the following lemma:

Lemma 4.1.

$$e(F, \hat{F}) \leq 2 \sup_x |F(x) - \hat{F}(x)|$$

Proof. The proof of the lemma is given in Pflug (2001). □

To construct another upper bound, which is suggested by Pflug (2001), we need to define a metric according to which the scenario construction takes place:

Definition. (The Lipschitz-constant of order p of h)
The Lipschitz-constant of order p of h is calculated as follows:

$$L_p(f(x)) = \inf\{L : |f(x, z_1) - f(x, z_2)| \leq L|z_1 - z_2| \max(1, |z_1|^{p-1}, |z_2|^{p-1}) \ \forall z_1, z_2\} \leq \bar{L}_p \tag{4.5}$$

where $\bar{L}_p$ is an upper bound. This is an indicator for how fast the cost $f(\cdot, z)$ changes in z.

As next step, we have to introduce the *Fortet-Mourier distance* between F and $\hat{F}$.

Definition. (Fortet-Mourier distance)

$$d_p(F, \hat{F}) = \int \max(1, |z|^p) |G(z) - \hat{G}(z)| dz. \tag{4.6}$$

If $p = 1$ this metric equals the widely known *Wasserstein-distance.*

Lemma 4.2. *The approximation error is then bounded by*

$$e(F, \hat{F}) \leq 2\bar{L}_p d_p(F, \hat{F}). \tag{4.7}$$

This means that if the cost function $f(\cdot, z)$ is continuous with Lipschitz-constant of order p our original problem can be approximated by the problem of minimizing the distance $d_p(F, \hat{F})$. So, if we want to minimize the approximation error, we have to find the discrete distribution $\hat{F}$ closest to F. This means that we are looking for a tree with a distribution $\mathbb{P}_{tree}$ such that

$$d_p(\mathbb{P}, \mathbb{P}_{tree}) \leq \epsilon \tag{4.8}$$

with $\epsilon > 0$ given. The actual tree construction is done based on a scenario fan (a scenario fan is a special case of a scenario tree, where each scenario is independent of other sceanrios) of simulated scenarios, together with stage-by-stage approximations.

Again, an essential point is the reduction of scenarios. This is handled by summarizing distinct scenarios in a single scenario, again with respect to a probability metric. Therefore, a certain precision ϵ is defined and the reduction takes place either via forward selection or backward reduction.

4.3.4 Optimal Quantization

Similar to the methods based on probability metrics, *optimal quantization* also aims at minimizing the average distance between the sample distribution and the discrete distribution. A limitation hereby is that the approach is only tractable for scenario trees with only two stages.

Following Heitsch and Römisch (2003), let $z_1, \dots, z_N$ be a set of equally likely sample vectors, drawn from a continuous distribution. Then, the problem we have to solve is

$$\min_{\hat{z}_1 \dots, \hat{z}_M} \{(\min_{y_{i,j} \in (0,1), q_j \geq 0} \{\sum_{i=1}^{N} \sum_{j=1}^{M} y_{i,j} \|z_i - \hat{z}_j\|^r | \sum_{j=1}^{M} y_{i,j} = 1, \sum_{i=1}^{N} y_{i,j} = q_j N\})^{\frac{1}{r}}\}. \tag{4.9}$$

Solving this equation yields optimal mass points $(\hat{z}_1^*, q_1^*), \dots, (\hat{z}_M^*, q_M^*)$, but finding a concrete solution is $\mathcal{NP}$-hard. For this reason, one has to be satisfied with finding local optimizers. Therefore, a common approach is to start with an initial guess of $\hat{z}$ to then try to find optimal $q_1^*, \dots q_M^*$ by solving

$$\min_{y_{i,j} \in (0,1), q_j \geq 0} \{\sum_{i=1}^{N} \sum_{j=1}^{M} y_{i,j} \|z_i - \hat{z}_j\|^r | \sum_{j=1}^{M} y_{i,j} = 1, \sum_{i=1}^{N} y_{i,j} = q_j N\} \tag{4.10}$$

for $\hat{z}_j$ fixed. The next step is to find optimal $\hat{z}_1^*, \dots \hat{z}_M^*$ by solving the following equation for $y_{i,j}$ fixed:

$$\min_{\hat{z}_1, \dots, \hat{z}_M} \{\sum_{i=1}^{N} \sum_{j=1}^{M} y_{i,j} \|z_i - \hat{z}_j\|^r\}. \tag{4.11}$$

A well-known special case results from taking the squared Euclidian distance, the so called *k-means clustering algorithm*, which was first proposed by MacQueen (1967). If one uses quantizers as scenarios the corresponding optimal objective value is a lower bound:

Proposition 4.3. *If* $\hat{F}(x) = \sum_{j=1}^{M} q_j f(x, \hat{z}_j)$ *and* $f(x, z)$ *convex in* z*, then*

$$\min_x \hat{F}(x) \leq \min_x F(x).$$

Proof. A proof is given by Löhndorf (2016). □

4.3.5 Voronoi Cell Sampling

A main drawback of optimal quantization, as criticized by Hochreiter and Pflug (2007), is that even if one found appropriate minimizers of 4.9 the variance of the set of scenarios could be smaller than that of the sample distribution. Generally, the variance of the marginal distributions tends to decrease in the dimensionality of the joint distribution.

As an alternative scenario generation method which is able to mitigate the contraction of scenarios towards the mean, *Voronoi cell sampling* was developed by Löhndorf (2016). The author discovered the method as a by-product of his work about scenario generation methods. Similar to the optimal quantization method, the discretization of the stochastic process is done by separating the continuous data process into a finite number of disjoint partitions. This method integrates stratified sampling with probability metrics and it turned out to perform in a favorable way, meaning that the approximation error which occurs is very small compared to other methods we have already introduced.

Löhndorf and Wozabal (2017) use this method to approximate the continuous process of natural gas prices by a discrete scenario lattice.

For example, stratified sampling is used to reduce the variance of a Monte Carlo simulation by Corlay and Pagès (2015). This is done by partitioning the outcome space into a number of disjoint regions.

The purpose of including stratified sampling is to reduce the variance of the Monte Carlo estimate by partitioning the outcome space into a number of non- intersecting regions. Fortunately, the optimal quantizers provide such a partitioning as by-product. These special partitions are known as *Voronoi cells.*

To take a closer look at the performance of the Voronoi sampling approach in terms of variance reduction we have to consider p_j, which refers to the proportion of samples allocated to partition Ω_j. Using stratified sampling, the variance of the Monte Carlo estimate is given by

$$\hat{\sigma}^2_{SS} = \frac{1}{M} \sum_{j=1}^{M} \frac{p_j}{q_j} q_j \mathbb{V}ar(f(x,z)|z \in \Omega_j). \tag{4.12}$$

The first step of our journey in reducing the variance is to allocate samples to partitions, i.e. $p_j = q_j$ proportionally. Then we use the law of total variance to get

$$\begin{aligned} \frac{1}{M}\mathbb{V}ar_z(f(x,z)) &= \frac{1}{M}(\mathbb{E}_z(f(x,z)^2) - \mathbb{E}_z(f(x,z))^2) \\ &= \frac{1}{M}(\mathbb{E}_z(\mathbb{E}_z(f(x,z)^2|z \in \Omega_j)) - \mathbb{E}_z(\mathbb{E}_z(f(x,z)|z \in \Omega_j))^2) \\ &\geq \frac{1}{M}\mathbb{E}_z(\mathbb{V}ar_z(f(x,z)|z \in \Omega_j)) \\ &= \frac{1}{M}\sum_{j=1}^{M} q_j \mathbb{V}ar_z(f(x,z)|z \in \Omega_j). \end{aligned} \tag{4.13}$$

Following Löhndorf (2016), we can further reduce variance by using optimal quantizers.

Proposition 4.4. *Voronoi cell sampling provides optimal variance reduction under stratified sampling with proportional allocation.*

Proof. The proof is given in Löhndorf (2016). □

Except all weights being equal, we need fewer partitions than scenarios for proportional allocation, to be able to make multiple draws from partitions with large weights. Instead of drawing from fewer partitions multiple times, we could alternatively draw once, but from more partitions, since 4.13 assures that further stratification of a partition would result in further variance reduction.

Following Löhndorf (2016), we therefore assume that Voronoi cell sampling draws just one random realization from each partition and assigns the resulting scenario to the probability of the nearest quantizer. We define another sequence of candidate scenarios by $(\hat{z}'n, \ldots, \hat{z}'^{n}_{M})^{N}_{n=1}$ with equal starting points then the original scenario, i.e. $\hat{z}'^{0}_{j} = \hat{z}^{0}_{j}, j = 1, \ldots, M$. Then, scenarios can be obtained through

$$\hat{z}'^{n}_{j} = \begin{cases} z_n & \text{if } j = argmin_k\{\|z_n - \hat{z}^{n-1}_{k}\|^2\} \\ \hat{z}'^{n-1}_{j} & \text{otherwise} \end{cases} \tag{4.14}$$

for $j = 1, \ldots, M, n = 1 \ldots, N$.

Proposition 4.5. *The sampling method in 4.14 reduces the variance of the cost function estimate for sufficiently large M.*

Proof. The proof is given in Löhndorf (2016). □

4.3.6 Moment Matching

In moment matching the idea is to generate a set of scenarios where the first four moments of random variables have to coincide with the first four moments of the original model. The method was first proposed by Fleishman (1978) and later adapted to the multivariate setting by Høyland et al. (2003).

The procedure works as follows:

(i). Generate $z_1, \ldots, z_M$ discrete univariate random variables using Monte Carlo sampling, each satisfying a specification for the first four moments.

(ii). Transform the random variables to meet a given correlation structure by premultiplying the vectors with a lower triangular matrix L of covariance matrix Σ:

$$z_j' = Lz_j, \; \Sigma = LL', \; j = 1, \dots, M$$

(iii). Apply the cubic transformation proposed by Fleishman (1978) to obtain sequences that satisfy the specification for the first four moments.

The second point, the transformation of the variables, will distort the marginal moments higher than second order. Therefore, the clue is to start with a different set of higher moments. Following Høyland et al. (2003), the proposed procedure leads to exact desired values for the moments and the correlation, if the generated univariate random variables from step 1 are independent. This is only true in the case of all scenarios being equally probable and the number of outcomes going to infinity. In most cases, those assumptions are not fulfilled and one has to deal with moments and correlations that are not matching the desired ones. In order to address this purpose and to ensure that the moments deviate from their target moments at most by a given ϵ, Høyland et al. (2003) developed an iterative algorithm.

A disadvantage of moment matching is that there exist indefinite numbers of distributions with the same first four moments. So it is unknows if the distribution one has chosen really fits the input data.

4.4 Lattice Reformulation of the Battery Problem

In general, it is possible to reformulate multistage stochastic optimization problems on a scenario tree. An approach was given by Pflug and Romisch (2007).

Pflug and Romisch (2007) start by considering a finite probability space $\Omega = (\omega_1, \dots, \omega_K)$, representing K scenario paths. This setting allows them to represent the stochastic processes by a finite tree with node set $\mathcal{N} = \{1, \dots, N\}$. Here, the levels of the tree correspond to the decision stages. By $\mathcal{N}_t$ they denote the set of nodes at level t for $t = 0, \dots, T$. Using this notation, the set $\mathcal{N}_T$ contains the K leave nodes of the tree. This tree structure represents the filtration of the process. The predecessor node for each node n is denoted by n_- and in the same manner n_+ denotes the child nodes of each node. They assume, that there is an unique root node and as it represents the present we refer to it as 0. As final element of a scenario tree one needs to add probabilities $Q_n \geq 0$ to each node n, where $\sum_{j \in \mathbb{N}_t} Q_j = 1$ has to hold for all points in t.

After this procedure, one can represent the different processes and decision variables in a model w.r.t. the nodes n of the constructed tree.

The problems constraints which have to hold w.p.1 are obtained by formulating them for all nodes of stage $\mathcal{N}_t$ and the objective function is based on the probabilities related to the nodes in the following ways: In the case where one just wants to minimize the expectation, it is possible to calculate the objective value directly by weighting the values of each leaf node with the respective probability.

If we now try to extend this tree-approach to a lattice version and apply it to our battery charging problem the following dilemma arises:

Due to the fact that in a lattice every node can have multiple predecessor nodes, the lattice formulation can not be applied for the charging state of the battery, because the transition from state $B(t) \to B(t+1)$ is not unique.

As a way out, we formulate the problem as a classical stochastic dynamic optimization problem (see 3.3.2), considering the state of charge as the implicitly defined system state variable. Separately, we consider the stochastic environmental variables price, PV production and consumption and model them as Markov Processes on a lattice. This workaround of formulating the problem as DP enables us to keep the advantages of a lattice, compared to a tree and we are still able to handle state variables.

remark. Note that in our problem the only state variable is the charging state of the battery. The state variable is not modeled on the lattice.

remark. In a DP the value function is characterized by the Bellman equation, which has the state variable as arguments. The lattice is only used to calculate the corresponding expectations.

In the next section, we will became familiar with different algorithms that are able to solve problems of such kind.

5 Solution Methods

With a steady increase of computing power, optimization including uncertainty has become a very popular research field. Nowadays, multiple stochastic programming methods allow motivated users to get their models one step closer to reality by incorporating randomness. Unfortunately, as free lunch is nowhere to be found, analytical closed-form solutions to stochastic optimization problems exist only in some rare cases. Especially when dealing with continuous distributions of random variables, some approximation must be made. Most approximation schemes are based on discretization of the underlying random space through scenario trees. In section 4.3, we have already become familiar with a few of methods of how to model uncertainty.

Once the discretization of the underlying probability space has been managed, finite dimensional optimization algorithms can be applied without difficulty. Nevertheless, the challenges of obtaining high precision of the optimal value and of finding an optimal solution remain. As it is usually the case in optimization, the results are strongly dependent on the selected basis functions.

A very popular practical utilization of stochastic programming in the energy sector is the day ahead bidding problem. Typically, it is modeled as a two-stage stochastic program. This approach is quite obvious, as one simply takes bidding decisions at first stage and price realizations as well as operational decisions take place at second stage.

However, what this two-stage approach can definitively not capture, are future storage states and decisions. To take longer planning horizons into account, it is necessary to solve a multi-stage stochastic programming problem. In order to accomplish the challenge of solving such problems, a few basic solution strategies have emerged in the literature:

A method which avoids discretizing the underlying probability space was developed by Bertsekas and Tsitsiklis (1995); The goal is to project the control function or the cost-to-go function into a finite dimensional functional space. By doing so, the dimension of the problem reduces to the cardinality of the functional space. Then the problem could be solved numerically, based on Monte-Carlo methods.

Another strategy, as proposed by Heitsch and Römisch (2003), is to represent the uncertainty by a scenario tree and to then solve the problem as one large mathematical program. The limitation of this approach clearly lies within the comparatively small number of stages it can handle.

Pereira and Pinto (1991) developed a method called *stochastic dual dynamic programming* (SDDP). They proposed their method to solve certain specific problems where randomness only appears at the right-hand-side of constraints and is independent from one stage to another. The procedure is as follows: First, the problem is formulated as dynamic program and then Bender's decomposition algorithm is applied. Recursively, a value function is constructed at each stage around a set of sample decision. The crucial part for the construction is to remember the requirement of stage-wise independence of the random process, because only in this case the respective cost-to-go value functions of the DP equations are independent of the data process. As long as the optimization problem is convex at each stage and the stochastic process is stage-wise independent, SDDP can handle problems with a large number of stages without any difficulties. The convergence of this method was studied in detail by Philpott and Guan (2008).

However, in most real world cases, complete independence of the randomness is not given. To relax this necessary assumption, certain approaches have been suggested dealing with Markovian type dependence structures.

One possibility pointed out by Pereira (1989) and Shapiro et al. (2013) is to model the data process as an autoregressive process and to then add time series transition equations as additional equality constraints to the optimization problem. As a result, realizations of the data process are treated as decision variables. This way of proceeding enables reformulating the problem in terms of stage-wise independent errors of the considered time series. Disadvantages are that the number of stage variables must be increased and that the approach is restricted to linear cases and right-hand side uncertainty.

Another approach to solve problems, with Markovian type dependence structures, in particular discrete-time, continuous-state, risk averse MDP's, was developed by Powell (2007). They proposed a method to approximate the optimal policy of the continuous-state problem which is equivalent to approximating the value function of the MDP. In other words, the so-called *approximate dynamic programming* (ADP) algorithm simulates the state transition process of a MDP and then uses the sampled information to approximate the high dimensional value function through a function of much lower complexity.

Bonnans et al. (2012) and Löhndorf et al. (2013) developed a method for solving a general class of discrete-time, continuous-state, risk-averse MDPs integrating scenario lattices with the SDDP framework. Their approach called *approximate dual dynamic programming* (ADDP) allows modeling any parameter as Markovian data process and, therefore, a much broader range of stochastic models can be used. Similar to SDDP, the decision problem is solved iteratively in ADDP, using forward simulation to sample candidate decisions and backward recursion to construct a polyhedral approximation of the post-decision value function. The clear advantage of this method is that, in contrast to regular SDDP approaches, no stage-wise independence of the stochastic process is required. Moreover, it is rather assumed that randomness follows a Markov process. The disadvantage of the ADDP approach is that no convergence guarantee can be made with respect to the true process.

Our goal is to solve the stochastic dynamic optimization problem of optimally charging a battery, where the stochastic variables follow a Markovian type dependence structure and can be modeled on a lattice (see 4.4). The most promising solution algorithm concerning our setting is ADDP. Therefore, the following sections give an introduction of the underlying concepts. We start with an outline of ADP, followed by an overview of SDDP. Then, we use the gained knowledge to understand the concepts of ADDP.

5.1 Approximate Dynamic Programming

As already discussed in chapter 3.3, in dynamic programming, one always has to deal with the curse of dimensionality. ADP, also known as *neuro-dynamic programming* or *reinforcement learning*, provides an opportunity to break this burden. Avoiding the complete enumeration of the state space is done by using Monte Carlo simulation to sample the Markov decision process. Generally, there are two principal techniques: policy and value iteration.

When developing an ADP algorithm, there are several factors one should keep in mind. It would be desirable that a policy using an approximate value function was able to imitate the behavior of the optimal policy. Moreover, the sampling process has to be designed in such a way that the states which can be reached by the optimal policy are sampled adequately often. Furthermore, the mechanism that updates the value function has to take into account noise occurring in the sampled observations.

To get an overview of the key concepts of ADP we follow Löhndorf (2011). As more detailed literature on the subject, we recommend Bertsekas and Tsitsiklis (1995), Bertsekas (2012), or Powell (2007).

For our purpose it will be enough to consider the method for finite horizon problems, but the concepts can easily be extended to infinite horizon problems (see for example Löhndorf (2011)).

The first remarkable aspect of ADP is that one does not need access to an explicit model of the state transition matrix. It is enough to require a simulation model of the state transition process. For our considerations, we have to introduce the *expectational form* of the optimality equations introduced in equation 3.11:

$$V_t(S_t) = \min_{x_t \in \mathcal{X}(S_t)} \{f_t(S_t, x_t) + \gamma \mathbb{E}[V_{t+1}(S_{t+1})|S_t, x_t]\}, \ \forall\ S_t \in \mathcal{S}_t,\ 1 \leq t \leq T-1. \tag{5.1}$$

Since we now assume that we have no knowledge of the state transition process, we can't compute the expectation explicitly. Therefore, we try the next best method and compute a sample average using Monte Carlo simulation. To do so, we replace the value function with a function of an estimate of the state-action value, $Q_t(S_t, x_t)$, called *Q-factor* of the pair (S_t, x_t). Let us assume that $Q_T(S_T, x_T)$ is given. To compute the Q-factors we simulate N state transitions for each $Q_t(S_t, x_t)$ and solve the following equations using backward recursion:

$$Q_t(S_t, x_t) = f_t(S_t, x_t) + \frac{\gamma}{N} \sum_{n=1}^{N} \min_{x_{t+1}^n \in \mathcal{X}(S_{t+1}^n)} \{Q_t(S_{t+1}^n, x_{t+1}^n)\}\ \forall\ S_t \in \mathcal{S}_t, x_t \in \mathcal{X}(S_t), \tag{5.2}$$

from $t = T-1, \ldots, t$. This approach only works if the action space is small enough.

In addition, one has the problem of looping over the entire action space repeatedly. To bypass this effort, we introduce the concept of expressing the value function in terms of the *post-decision state*. As the name implies, in contrast to Q-factors, the post-decision value function returns the value not until the end of a decision epoch. For this purpose, we reformulate 5.1 as

$$V_t(S_t) = \min_{x_t \in \mathcal{X}_t(S_t)} \{f_t(S_t, x_t) + \gamma \bar{V}_t(S_t, x_t)\}\ \forall\ S_t \in \mathcal{S}_t \tag{5.3}$$

where $\bar{V}_t(S_t, x_t)$ denotes the expectation of the value right after a decision has been made. Using this new formulation, we can rewrite the optimality equation in terms of the post-decision value function as

$$\bar{V}_t(S_{t-1}, x_{t-1}) = \mathbb{E}[\min_{x_t \in \mathcal{X}(S_t)} \{f_t(S_t, x_t) + \gamma \bar{V}_t(S_t, x_t)\}|S_{t-1}, x_{t-1}]\ \forall\ S_t \in \mathcal{S}_t, x_t \in \mathcal{X}(S_t). \tag{5.4}$$

Note that up to now we follow the approach of computing separate values for each state-action pair, which still leaves us with the curse of dimensionality. Hence, let us try to solve this issue by focusing on approximating the value function. The *approximate value function* $\hat{\bar{V}}(\cdot; \omega_t)$ is dependent on a vector $\omega_t \in \mathbb{R}^K$. Then, the objective of an ADP algorithm is

$$\min_{\omega_t} \{\|\hat{\bar{V}}_t(S_t, x_t; \omega_t) - \bar{V}_t(S_t, x_t)\|\},\ \forall\ 1 \leq t \leq T-1 \tag{5.5}$$

which means that one tries to find a w_t that minimizes the approximation error. The resulting function is the wanted approximation of the expectation in 5.4, i.e.

$$\hat{\bar{V}}_t(S_t, x_t; \omega_t) \approx \bar{V}_t(S_t, x_t), \ \forall\ 1 \leq t \leq T-1 \tag{5.6}$$

To find the optimal parameter vector it is necessary to do both, solve the recursion and construct an approximation simultaneously. Two widespread methods for doing so are *approximate value iteration* and *approximate policy iteration*. On continuation we follow Löhndorf (2011) and give a short overview of both approaches.

5.1.1 Approximate Value Iteration

The first step in the approximate value iteration algorithm is to initialize the algorithm with an approximate value function $\hat{\bar{V}}_t(\cdot; \omega_t^0)$ and a set of (possibly identical) initial states S_1^t (Step (1)). Then, over N iterations, the algorithm alternates between a forward pass (Step 2.1) and a backward pass (Step 2.2 and Step 2.3). At each iteration of the forward pass, the algorithm chooses the action that maximizes the sum of the immediate reward and the current estimate of the discounted post-decision value (Step 2.1.1). Then, by calling the simulation model S^M, the next state given the current state and action is generated (Step 2.2.2).

The backward pass proceeds as follows: First, at each iteration, the algorithm recursively computes the discounted value of the state-action pair sampled during forward pass by evaluating

$$v_t^n = f_t(S_t^n, x_t^n) + \gamma v_{t+1}^n$$

(Step 2.2.1). This value is passed on to the updating function U^V (Step 2.2.3) to update the estimate of the approximate value function of the previous state-action pair (Step 2.2.2). After termination, one receives the final approximation of the post-decision value function (Step 3). In the following, an outline of the algorithm is given:

```
(1) Input arguments: approximate value function $\hat{\bar{V}}_t(\cdot, \omega_t^0)$, $t = 1, \ldots, T$; initial states
    $(S_1^n)_{n=1}^N$
(2) Do for $n = 1, 2, \ldots, N$
    (2.1) Do for $t = 1, 2, \ldots, T-1$
        (2.1.1) Solve $x_t^n = argmin_{x \in \mathcal{X}(S_t^n)} \{f_t(S_t^n, x) + \gamma \hat{\bar{V}}_t(S_t^n, x; \omega_t^{n-1})\}$
        (2.1.2) Simulate $S_{t+1}^n = S^M(S_t^n, x_t^n)$
    (2.2) Solve $v_T^n = \min_{x \in \mathcal{X}(S_T^n)}\{f_T(S_T^n, x) + \gamma\hat{\bar{V}}_T(S_T^n, x; \omega_T)\}$
    (2.3) Do for $t = T-1, T-2, \dots, 2$
        (2.3.1) Compute $v_t^n = f_t(S_t^n, x_t^n) + \gamma v_{t+1}^n$
        (2.3.2) Update $\omega_{t-1}^n = U^V(\hat{\bar{V}}_{t-1}, \omega_{t-1}^{n-1}, S_{t-1}^n, x_{t-1}^n, v_t^n)$
(3) Return approximate value function $\hat{\bar{V}}_t(\cdot; \omega_t^N),\ t = 1, \dots, T-1$
```

Note that there also are approximate value iteration approaches which update the value function during the forward pass.

5.1.2 Approximate Policy Iteration

Contrary to what the name may suggest, when performing approximate policy iteration, we are still updating the value function. Due to the misleading name in reinforcement learning communities approximate policy iteration methods are also referred to as *batch methods*. The basic concept is to store a batch of observations of (S_t, x_t, v_{t+1}) tuples and to update the value function only after a complete batch has been collected. Despite the additionally added outer loop, the algorithm works similar to the above discussed approximate value iteration.

In the following, an outline of the algorithm is given:

```
(1) Input arguments: approximate value function $\hat{\bar{V}}_t(\cdot, \omega_t^0),\ t = 1, \dots, T$; initial states
    $(S_1^n)_{n=1}^N$
(2) Do for $m = 1, 2, \dots, M$
    (2.1) Do for $n = 1, 2, \dots, N$
        (2.1.1) Do for $t = 1, 2, \dots, T-1$
            (2.1.1.1) Solve $x_t^{n,m} = argmin_{x \in \mathcal{X}(S_t^{n,m})}\{f_t(S_t^{n,m}, x) + \gamma\hat{\bar{V}}_t(S_t^{n,m}, x_t; \omega_t^{m-1})\}$
            (2.1.1.2) Simulate $S_{t+1}^{n,m} = S^M(S_t^{n,m}, x_t^{n,m})$
        (2.1.2) Solve $v_T^{n,m} = \min_{x \in \mathcal{X}(S_T^{n,m})}\{f_T(S_T^{n,m}, x) + \gamma\hat{\bar{V}}_T(S_T^{n,m}, x, \omega_T)\}$
        (2.1.3) Do for $t = T-1, T-2, \dots, 2$
            (2.1.3.1) Compute $v_t^{n,m} = f_t(S_t^{n,m}, x_t^{n,m}) + \gamma v_{t+1}^{n,m}$
    (2.4) Update $\omega_t^m = U^P(\hat{\bar{V}}_t, \omega_t^{m-1}, (S_t^{n,m}, x_t^{n,m}, v_z^{n,m})_{n=1}^N),\ t = 1, \dots, T-1$
(3) Return approximate value function $\hat{\bar{V}}_t(\cdot, \omega_t^M),\ t = 1, \dots, T-1$
```

Compared to approximate value iteration, the special feature of the policy iteration approach is the additional outer loop. During each iteration of this loop, the algorithm performs N forward and backward passes, similar to approximate value iteration, but instead of updating the approximate value function during backward pass, the algorithm stores the entire information about state, action and value collected during all N sample paths. After N iterations of the second loop, the algorithm passes the collected information on to the updating function U^P in order to obtain a new estimate of the approximate value function.

remark. Updating the value function only every N steps might not be desirable in every application.

5.2 Stochastic Dynamic Dual Programming

As already mentioned, the usual procedure of solving a multi-stage stochastic program is by constructing a scenario tree.

Furthermore, what we also have already discussed is that even with a small number of outcomes per stage, the size of a scenario tree grows exponentially with increasing stages. For two-stage problems with a large number of scenarios, SAA provides a solution approach which allows large-scale problems to be solved. However, due to the curse of dimensionality, multi-stage problems are intractable.

Because of Philpott and Guan (2008), one field of application where multi-stage programming models are widely used is the long-term scheduling of water resources. Those problems normally involve determining a policy of releasing water from water reservoirs for generation of hydro-electricity. The necessity of modeling multiple reservoirs led to the development of various multi-stage stochastic linear programming models using scenario trees. One of those methods is *stochastic dual dynamic programming*, which was developed as an algorithm able to deal with a rapidly growing scenario tree. The general idea is to approximate the future cost function of dynamic programming using a piecewise linear outer approximation. This approximation is defined via cutting planes, short *cuts*, computed by solving linear programs. By using this method, the curse of dimensionality arising from discretizing the scenario tree can be avoided. Another important aspect is that stage-wise independent uncertainty must be assumed. This allows for shared cuts between different states, leading to an efficient reduction of the scenario tree.

The standard implementations of SDDP are risk neutral, but many authors extended the algorithm to implementations accounting for risk. For example, Philpott and de Matos (2012) considered the incorporation of a time-consistent coherent risk measure into a multi-stage stochastic programming model which is solved by SDDP.

Hence, we introduce the risk neutral SDDP method introduced by Pereira and Pinto (1991). Here, we will follow the notation of Philpott and de Matos (2012) who give a good outline of the algorithm.

Let us reconsider the linear multistage stochastic problem, as introduced in 3.3. The problem we consider has T stages, denoted $t = 1, 2, \ldots, T$. Furthermore, we assume that at each stage a random right-hand-side vector $b_t(\xi_t) \in \mathbb{R}^m$ has a finite number of realizations defined by $\xi_t \in \Omega_t$. Another important assumption we have to request is the stage-wise independence of the outcomes ξ_t. Ω_1 consists of just one element. Under these assumptions, the first-stage problem can be written as

$$\begin{aligned} z = \min \quad & c_1^T x_1 + \mathbb{E}[Q_2(x_1, \xi_2)] \\ s.t. \quad & A_1 x_1 = b_1, && (5.7) \\ & x_1 \geq 0 && (5.8) \end{aligned}$$

where $x_1 \in \mathbb{R}^n$ is the first stage decision, $c_1 \in \mathbb{R}^n$ a cost vector, A_1 is a $m \times n$ matrix and $b_1 \in \mathbb{R}^m$. $Q_2(x_1, \xi_2)$ refers to second stage costs associated with decision x_1 and realization $\xi_2 \in \Omega_2$.

Given decision x_{t-1} and realization ξ_t, the problem we have to solve in later stages t is

$$\begin{aligned} Q_t(x_{t-1}, \xi_t) = \min \quad & c_t^T x_t + \mathbb{E}[Q_{t+1}(x_t, \xi_{t+1})] \\ s.t. \quad & A_t x_t = b_t(\xi_t) - E_t x_{t-1}, \quad [\pi_t(\xi_t)] && (5.9) \\ & x_t \geq 0, \end{aligned}$$

In this problem, $x_t \in \mathbb{R}^n$ is the decision in stage t, c_t again the cost, A_t and E_t denote $m \times n$ matrices. By $\pi_t(\xi_t)$, we denote the dual variables of the constraints. Concerning the last state, we assume that $\mathbb{E}[Q_{T+1}(x_T, \xi_{T+1})] = 0$. For all instances of 5.9 we assume relatively complete recourse, whereby 5.9 at stage t has a feasible solution for all values of x_{t-1} that are feasible for the instance of 5.9 at stage $t - 1$. This assumption is not a limiting restriction, as relatively complete recourse can be ensured by simply introducing artificial variables with penalty terms in the objective.

In order to build an approximately optimal policy, the SDDP algorithm performs a sequence of iterations. Each iteration consists of a *forward pass* and a *backward pass*. During the forward pass, a set of N scenarios is sampled from the scenario tree. Then, starting from the first stage, decisions are made at each stage of the N scenarios. Denoting a specific scenario s, all observed values $\bar{x}_t(s)$ of the decision variable x_t and the cost of each stage in all scenarios s are stored.

The determined policy at stage t is defined by a polyhedral outer approximation of $\mathbb{E}[Q_{t+1}(x_t, \xi_{t+1})]$, the specific form of the approximation resulting from the use of *Benders cuts*. Mathematically speaking, in a problem at stage t, $\mathbb{E}[Q_{t+1}(x_t, \xi_{t+1})]$ is replaced with the variable θ_{t+1}. This variable is bounded by

$$\theta_{t+1} - \bar{g}_{t+1,k,s}^T x_t \geq \bar{h}_{t+1,k,s}, \quad k = 1, 2, \ldots, K, \quad s = 1, 2 \ldots, N, \tag{5.10}$$

where K equals the number of backward passes that are already completed.

Using this approximation we can rewrite the first-stage problem from 5.7 as

$$\begin{aligned} z = \min \quad & c_1^T x_1 + \theta_2 \\ s.t. \quad & A_1 x_1 = b_1, \\ & \theta_2 - \bar{g}_{2,k,s}^T x_1 \geq \bar{h}_{2,k,s}, \quad k = 1, 2, \ldots, K, \\ & s = 1, 2, \ldots, N, \\ & x_1 \geq 0 \end{aligned} \tag{5.11}$$

and the problem at stage t becomes

$$\begin{aligned} \tilde{Q}_t(x_{t-1}, \xi_t) = \min \quad & c_t^T x_t + \theta_{t+1} \\ s.t. \quad & A_t x_t = b_t(\xi_t) - E_t x_{t-1}, \quad [\pi_t(\xi_t)], \\ & \theta_{t+1} - \bar{g}_{t+1,k,s}^T x_t \geq \bar{h}_{t+1,k,s}, \quad k = 1, 2, \ldots, K \\ & s = 1, 2, \ldots, N, \\ & x_t \geq 0 \end{aligned} \tag{5.12}$$

Here, the set of cuts is interpreted as empty, when $K = 0$.

For the algorithm to terminate, a convergence criterion is tested at the end of the forward pass. If the termination criterion is satisfied, the iterations stop, otherwise a backward pass is conducted. The convergence check is satisfied when the lower bound of the expected cost at first stage is statistically close to an estimate of the expected total operation cost obtained by averaging the cost of the policy defined by the cuts when applied to the N sampled scenarios. In some cases, the convergence test is simply replaced with a maximal number of iterations after which the algorithm terminates.

In case of the convergence criterion not being satisfied, a backward pass is executed. By doing so, N cuts are added to each stage problem, starting at the last but one stage. Coefficients for the cuts are obtained by solving the next stage problems for all possible realizations (Ω_{t+1}) at each stage t and for each scenario s. The cut for 5.12 is computed after its solution $\bar{x}_t^k(s)$ has been obtained in the forward pass, immediately preceding backward pass k. Solving the $t+1$th (approximate) stage problem for every $\xi_{t+1} \in \Omega_{t+1}$ results in dual variables $\bar{\pi}_{t+1,k,s} = \mathbb{E}[\pi_{t+1}(\xi_{t+1})]$. Then, the cut gradient can be defined as

$$\bar{g}_{t+1,k,s} = -\bar{\pi}_{t+1,k,s}^T E_{t+1} \tag{5.13}$$

and its intercept

$$\bar{h}_{t+1,k,s} = \mathbb{E}[Q_{t+1}(\bar{x}_t^k(s), \xi_{t+1})] + \bar{\pi}_{t+1,k,s}^T E_{t+1} \bar{x}_t^k(s). \tag{5.14}$$

The algorithm is initialized by setting $\theta_t = -\infty,\ t = 2, \ldots, T,\ K = 0,\ k = 1$. In the following, a summary of the algorithm is given:

---

(1) Forward pass

(1.1) Do for $t = 1$,

(1.1.1) Solve 5.11 and save $\bar{x}_1^k(s) = x_1,\ s = 1, \ldots, N$ and $\bar{z}^k = z$;

(1.2) Do for $t = 2, \ldots, T$ and $s = 1, \ldots, N$,

(1.2.1) Solve 5.12 setting $x_{t-1} = \bar{x}_{t-1}^k(s)$, and save $\bar{x}_t^k(s)$ and $\tilde{Q}_t(\bar{x}_{t-1}^k(s), \xi_t)$.

(2) Standard convergence Test (at 100(1-α)% confidence level)

(2.1) Calculate the Upper bound:
$z_u = \frac{1}{N}\sum_{s=1}^N \sum_{t=1}^T c_t^T \bar{x}_t^k(s)$,
$\sigma_u = \sqrt{\frac{1}{N}\sum_{s=1}^N (\sum_{t=1}^T c_t^T \bar{x}_t^k(s))^2 - z_u^2}$

(2.2) Calculate the Lower bound:

```
        (2.2.1) $z_l = \bar{z}^k$
        (2.2.2) Stop if $z_l > z_u - \frac{Z_{\frac{\alpha}{2}}}{\sqrt{N}}\sigma_u$,
                where $Z_\alpha$ is the $(1-\alpha)$ quantile of the standard normal distribution;
        (2.2.3) otherwise go to backward pass.

(3) Backward pass

    (3.1) Do for $t = T, \dots, 2$ and $s = 1, \dots, N$,
        (3.1.1) Do for $\xi_t \in \Omega_t$, solve 5.12 using $\bar{x}^k_{t-1}(s)$ and save $\bar{\pi}_{t,k,s} = \mathbb{E}[\pi_t(\xi_t)]$ and
                $\tilde{Q}_t(\bar{x}^k_{t-1}(s), \xi_t)$;
        (3.1.2) Calculate the $k$th cut for $s$ in stage $t-1$ using 5.13 and 5.14.
    (3.2) Set $K = K + 1,\ k = k + 1$.
```

Figures 9, 10 and 11 give a graphical outline of the main steps of the algorithm.

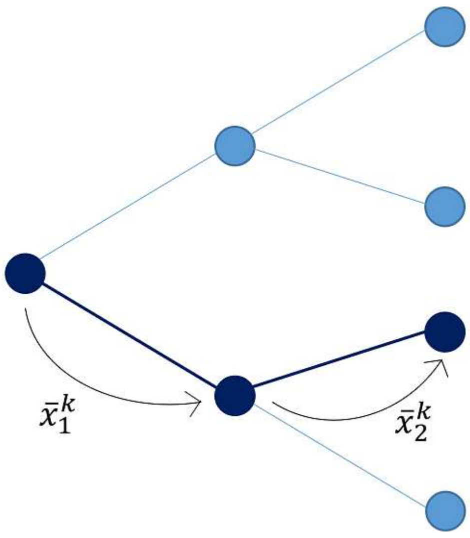


Figure 9: SDDP: Forward pass

5.2.1 SDDP with Markov Processes

So far, we have assumed that the stochastic variables are stage-wise independent in our description of the SDDP algorithm. However, in the battery problem we want to solve, the random variables have a distribution which is depending on an underlying state which is following a Markov process. Therefore, we need to expand the standard SDDP algorithm. Again, we base our approach on the work of Philpott and de Matos (2012).

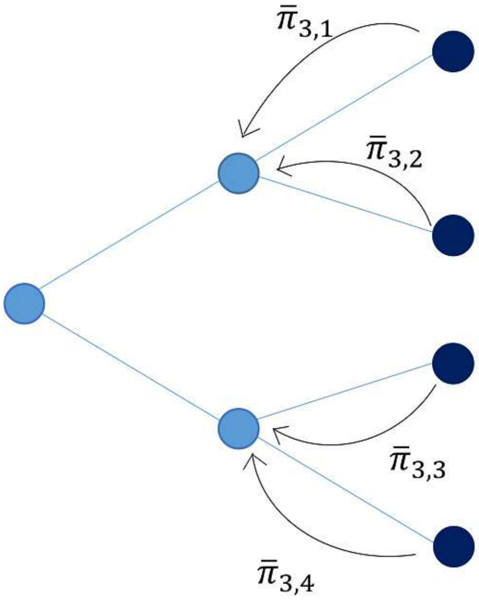


Figure 10: SDDP: Backward pass 1

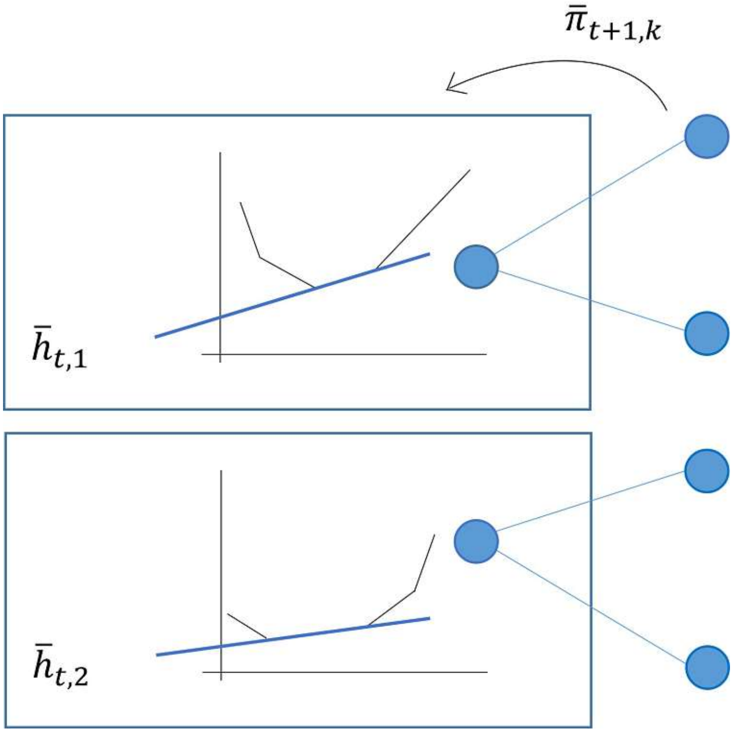


Figure 11: SDDP: Backward pass 2

The main drawback of using an underlying Markov process is the necessity to evaluate a future cost function for each value that the state may take, which results in an increase of the dimension of the dynamic program that has to be solved.

To start with, we assume that the process $W_t,\ t = 1, \dots, T$ is a Markov chain with transition matrices $P^{(t)}$. By $i = 1, 2, \dots, S$, we denote the realizations of W_t. Each Markov state realization i at stage t corresponds to a set Ω_{ti} of outcomes ξ_{ti}. Those particular outcomes are conditioned by the realization of W_t.

For simplification, we assume that $N = 1$, i.e. that the forward pass contains only one scenario.

Coming back to the first-stage problem, we assume that the system is in known state s_1 and that Ω_{1s_1} is a singleton. Therefore, we have

$$\begin{aligned} z = \min \quad & c_1^T x_1 + \sum_{j=1}^{S} P_{s_1 j}^{(1)} \mathbb{E}[Q_{2j}(x_1, \xi_{2j}) | W_2 = j] \qquad (5.15) \\ s.t. \quad & A_1 x_1 = b_1, \\ & x_1 \geq 0 \end{aligned}$$

where $Q_2(x_1, \xi_{2j})$ represents the second stage costs associated with decision x_1 and realization $\xi_{2j} \in \Omega_{2j}$. The problem that has to be solved at stage t then is

$$\begin{aligned} Q_{ti}(x_{t-1}, \xi_{ti}) = \min \quad & c_t^T x_t + \sum_{j=1}^{S} P_{ij}^{(t)} \mathbb{E}[Q_{t+1,j}(x_t, \xi_{t+1,j}) | W_{t+1} = j] \qquad (5.16) \\ s.t. \quad & A_t x_t = b_t(\xi_{ti}) - E_t x_{t-1}, \quad [\pi_t(\xi_{ti})] \qquad (5.17) \\ & x_t \geq 0, \end{aligned}$$

Now, the forward pass consists of a sequence of alternately sampled Markov state realizations and conditional outcomes. At every step we have to solve the problem given the Markov state realization and observed stochastic variables at this stage, using a cutting-plane approximation of future costs. Solving the problems results in a sequence of Markov states and values for decision variables $\bar{x}_t,\ t = 1, \dots, T-1$, which optimize each of the approximate stage problems.

Again, the backward pass is used to compute cuts at stage t at the point $\bar{x}_{t-1}$. Now, a different cut is included for all distinct states of the Markov process, i.e. at each backward pass we have to solve $\sum_{t=1}^{S} |\Omega_{ti}|$ linear programs.

What is left is the definition of the future cost function. This can be obtained in several different ways using *cuts*.

Single-cut version

In the single-cut version, the following minimization problem has to be solved for each Markov state i in order to obtain an outer approximation of the problem at point $\bar{x}_{t-1}$:

$$\begin{aligned}\tilde{Q}_t(\bar{x}_{t-1}, \xi_{ti}) = \min \quad & c_t^T x_t + \theta_{t+1,i} \\ s.t. \quad & A_t x_t = b_t(\xi_{ti}) - E_t \bar{x}_{t-1}, \quad [\pi_t(\xi_{ti})], \\ & \theta_{t+1,i} + \sum_{j=1}^{S} P_{ij}^{S} \bar{\pi}_{t+1,j,k} E_{t+1} x_t \geq \sum_{j=1}^{S} P_{ij}^{(t)} \bar{h}_{t+1,j,k}, \\ & k = 1, 2, \dots, K, \\ & x_t \geq 0\end{aligned}$$

where for each $k = 1, \dots, K$

$$\bar{P}i_{t+1,j,k} = \mathbb{E}[\pi_{t+1}(\xi_{t+1,k}) | W_{t+1} = j] \tag{5.18}$$

evaluated at iterate $\bar{x}_t^k$, and

$$\bar{h}_{t+1,j,k} = \mathbb{E}[\tilde{Q}_{t+1,j}(\bar{x}_t^k, \xi_{t+1,j}) | W_{t+1} = j] + \bar{\pi}_{t+1,j,k}^T E_{t+1} \bar{x}_t^k \tag{5.19}$$

Here $-E_{t+1}^T \pi_{t+1}(\xi_{t+1,j})$ is the subgradient of the optimal value function $\tilde{Q}_{t+1}(x, \xi_{t+1,j})$ for the subproblem solved in state j, which is a convex polyhedral function of x. Furthermore, the conditional expectation of the optimal value function at x given state j is

$$\mathbb{E}[\tilde{Q}_{t+1,j}(x, \xi_{t+1,j}) | W_t = j].$$

This expectation is convex with subgradient

$$\mathbb{E}[-E_{t+1}^T \pi_{t+1}(\xi_{t+1,j}) | W_t = j] = -E_{t+1}^T \bar{\pi}_{t+1,j}$$

at $x = \bar{x}_t$.

The last term to be specified is the future cost function $\theta_{t+1,i}(x)$ evaluated at x in state i with outcome ξ_{ti} at stage t. This is nothing else than the expectation of the optimal value function in each Markov state that might occur at the next stage. Mathematically, this can be expressed as

$$\theta_{t+1,j}(x) = \sum_{j=1}^{S} P_{ij}^{(t)} \mathbb{E}[\tilde{Q}_{t+1,j}(x, \xi_{t+1,j}) | W_t = j],$$

which is convex with subgradient $-E_{t+1}^T \sum_{j=1}^S P_{ij}^{(t)} \bar{\pi}_{t+1,j}$ at $x = \bar{x}_t$.

Note that

$$\theta_{t+1,i}(x) \geq \theta_{t+1,i}(\bar{x}_t) - \sum_{j=1}^{S} P_{ij}^{(t)} \bar{\pi}_{t+1,j,k}^T E_{t+1}(x - \bar{x}_t)$$

which shows that

$$\theta_{t+1,i} + \sum_{j=1}^{S} P_{ij}^{(t)} \bar{\pi}_{t+1,j,k} E_{t+1} x \geq \sum_{j=1}^{S} P_{ij}^{(t)} \bar{h}_{t+1,j,k}$$

is a valid cut for the approximate future cost.

Multi-cut version

The special feature of the multi-cut version is that future costs are represented by cuts for each of the possible Markov states at the next stage. Thus, at stage t we have to compute

$$\begin{aligned} \tilde{Q}_{ti}(\bar{x}_{t-1}, \xi_{ti}) = \min \quad & c_t^T x_t + \sum_{j=1}^{S} P_{(ij)}^{(t)} \theta_{t+1,j} \\ s.t. \quad & A_t x_t = b_t(\xi_{ti}) - E_t x_{t-1}, \quad [\pi_t(\xi_{ti})], \\ & \theta_{t+1,i} + \bar{\pi}_{t+1,j,k}^T E_{t+1} x_t \geq \bar{h}_{t+1,j,k}, \quad j = 1, \ldots, S, \; k = 1, 2, \ldots, K, \\ & x_t \geq 0 \end{aligned}$$

where $\bar{\pi}_{t+1,j,k}^T$ and $\bar{h}_{t+1,j,k}$ are defined by 5.18 and 5.19. Similar to the single cut version, it can be shown that solving the above system defines a valid outer approximation to the future cost function using cutting planes.

In both versions of the algorithm, it is necessary to maintain s sets of cuts at each stage. The difference is that each of the s subproblems uses one set of cuts in the multi-cut version which leads to larger stage problems, whereas each node will use only one set of cut sin the single cut version. The advantage of the multi-cut version is that, even tough the size of each stage problem grows faster, it is expected to require fewer iterations until convergence.

5.2.2 SDDP with Risk Measures

As a next step, we want to extend the SDDP approach to optimize a coherent risk measure in the sense that large losses are penalized, without compromising the expected costs too much. Therefore, we follow Philpott and de Matos (2012) and define a risk measure

$$\rho(Z) = \beta \mathbb{E}(Z) + \gamma CVaR_{1-\alpha}(Z) \tag{5.20}$$

where β and γ are nonnegative. It is easy to show that this risk measure is coherent. In general, the measure is defined as a single period measure. Kovacevic (2012) extended it to a dynamic risk measure $\rho_{t,T}$ over $t = 1, \dots, T$ in the following way: Given a probability space $(\Omega, \Sigma, \mathbb{P})$, a dynamic risk measure applies to situations where we have a random sequence of costs $(Z_1, \dots, Z_T)$ which is adapted to some filtration $\{0, \Omega\} = \Sigma_1 \subset \dots \subset \Sigma_T \subset \Sigma$ of σ-fields and where Z_1 is assumed to be deterministic. Furthermore, we define a dynamic risk measure in terms of a sequence of conditional risk measures $\{\rho_{t,T}\},\ t = 1, \dots, T$.

Corollar 5.1. *Given a dynamic risk measure, one is able to derive a corresponding single-period risk measure by*

$$\rho_t(T_{t+1}) = \rho_{t,T}(0, Z_{t+1}, 0, \dots, 0).$$

Theorem 5.2. *Any time-consistent dynamic risk measure can be constructed in terms of single-period risk measures ρ_t by*

$$\rho_{t,T}(Z_t, Z_{t+1}, \dots, Z_T) = Z_t + \rho_t(Z_{t+1} + \rho_{t+1}(Z_{t+2} + \dots + \rho_{T-2}(Z_{T-1} + \rho_{T-1}(Z_T)) \dots))).$$

Proof. A proof is given by Ruszczyński (2010). □

Now we want to describe such a construction in a specific case of 5.20, namely

$$\rho_t(Z) = (1 - \lambda_{t+1})\mathbb{E}[Z|\Sigma_t] + \lambda_{t+1} CVaR_{1-\alpha}(Z|\Sigma_t)$$

with λ_{t+1} measurable with respect to Σ_t.

The basic description of the SDDP algorithm is exactly the same as the standard approach presented earlier in this chapter except for the problems to be solved and the cut calculation.

To obtain an easier understanding of the influence of the risk measure, we provide an outline of a linear two-stage problem aiming to minimize the first-stage cost as well as the risk measure applied to the second-stage costs. This approach can be generalized to T-stage problems, as it was done by Philpott and de Matos (2012) for example.

As always, we assume that the first stage is deterministic and that the second-stage random variable has finite support Ω_2. For this particular problem, we assume that randomness only enters at the right hand side constrains. We want to solve a problem of the following form:

$$\begin{aligned} SP: \ \min \quad & c_1^T x_1 + (1-\lambda)\mathbb{E}[c_2^T x_2] + \lambda u_2 + \lambda\alpha^{-1}\mathbb{E}[(c_2^T x_2 - u_2)_+] \\ s.t. \quad & A_1 x_1 = b_1, \\ & A_2 x_2(\xi) + E_2 x_1 = b_2(\xi), \ \ for \ all \ \xi \in \Omega_2, \\ & x_1 \geq 0, \ \ x_2(\xi) \geq 0, \ \ for \ all \ \xi \in \Omega_2. \end{aligned} \tag{5.21}$$

Note that one has to take two decisions at first stage, x_1 and u_2.

Given choices $x_1 = \bar{x}_1$ as well as $u_2 = \bar{u}_2$ and replacing $(c_2^T x_2 - u_2)_+$ with $v_2(\xi)$, where

$$v_2(\xi) \geq c_2^T x_2(\xi) - u_2, \ \ for \ all \ \xi \in \Omega_2$$

$$v_2(\xi) \geq 0, \ \ for \ all \ \xi \in \Omega_2,$$

the second-stage problem becomes

$$\begin{aligned} SP(\bar{x}_1, \bar{u}_2): \ \min \quad & (1-\lambda)\mathbb{E}[c_2^T x_2] + \lambda\alpha^{-1}\mathbb{E}[v_2] \\ s.t. \quad & A_2 x_2(\xi) = b_2(\xi) - E_2 \bar{x}_1, \ \ for \ all \ \xi \in \Omega_2, \\ & v_2(\xi) - c_2^T x_2(\xi) \geq -\bar{u}_2, \ \ for \ all \ \xi \in \Omega_2, \\ & x_2(\xi) \geq 0, \ \ v_2(\xi) \geq 0, \ \ for \ all \ \xi \in \Omega_2. \end{aligned} \tag{5.22}$$

Decoupled from scenarios we get

$$\begin{aligned} Q(\bar{x}_1, \bar{u}_2, \xi) = \min \quad & (1-\lambda)c_2^T x_2 + \lambda\alpha^{-1} v_2 \\ s.t. \quad & A_2 x_2 = b_2(\xi) - E_2 \bar{x}_1, \quad [\pi_2(\xi)] \\ & v_2 - c_2^T x_2 \geq -\bar{u}_2, \quad [\phi_2(\xi)] \\ & x_2 \geq 0, \ \ v_2 \geq 0, \end{aligned} \tag{5.23}$$

where $[\pi_2(\xi)]$ and $[\phi_2(\xi)]$ are the dual multipliers. Due to strong duality, the optimal dual solution satisfies

$$Q(\bar{x}_1, \bar{u}_2, \xi) = \pi_2(\xi)^T (b_2 - E_2 \bar{x}_1) - \phi_2(\xi)\bar{u}_2.$$

Then, the problem SP can be written as

$$\begin{aligned} SP: \ \min \quad & c_1^T x_1 + \lambda u_2 + \mathbb{E}[Q(x_1, u_2, \xi)] \\ s.t. \quad & A_1 x_1 = b_1, \\ & x_1 \geq 0. \end{aligned} \tag{5.24}$$

Problem 5.24 can be solved by applying Benders decomposition, solving

$$\begin{aligned} MP: \ \min \quad & c_1^T x_1 + \lambda u_2 + \theta_2 \\ s.t. \quad & A_1 x_1 = b_1, \\ & \theta_{2k} + \bar{\pi}_{2k}^T E_2 x_1 + \bar{\phi}_{2k} u_2 \geq \bar{h}_{2k}, \ \ k = 1, \dots, K \\ & x_1 \geq 0. \end{aligned} \tag{5.25}$$

Here, k counts the cuts that are added to the master problem,

$$\bar{\pi}_{2k} = \mathbb{E}[\pi_{2k}(\xi)],$$

$$\bar{\phi}_{2k} = \mathbb{E}[\phi_{2k}(\xi)],$$

$$\bar{h}_{2k} = \mathbb{E}[Q_2(\bar{x}_{1k}, \bar{u}_{2k}, \xi)] + \bar{\pi}_{2k} E_2 \bar{x}_{1k} + \bar{\phi}_{2k} \bar{u}_{2k}$$

and $\bar{x}_1$ and $\bar{u}_{2k}$ denote the values of the first-stage variables where cut k is evaluated.

5.2.3 SDDP with Risk Aversion and Markov-chain Uncertainty

Our next step is to combine the Markov stage model with the risk measure and to integrate both of them into the SDDP algorithm. Fortunately, we can proceed in the exact same way as we have done when implementing the Markov chain model, whereby each cut is an affine function of u and x.

Starting from the risk -neutral problem 5.16, we now define

$$P_{ij}^{(t)}(\xi) = P_{ij}^{(t)} \mathbb{P}(\xi_{t+1,j} = \xi | W_{t+1} = j).$$

The single-period coherent risk measure we use at stage t is

$$\rho_{t|i}(Z_{t+1}) = (1-\lambda_{t+1}(i)) \sum_{j=1}^{S} \sum_{\xi} P_{ij}^{(t)}(\xi) Z_{t+1,j}(\xi) + \lambda_{t+1}(i) \times \inf_{u \in \mathbb{R}} \{u + \frac{1}{\alpha} \sum_{j=1}^{S} \sum_{\xi} P_{ij}^{(t)}(\xi)(Z_{t+1,j}(\xi) - u)_+\}.$$

Although we use the parameter $\lambda_{t+1}(i)$ to compute the risk of outcomes at stage $t+1$, it is measurable with respect to Σ_t. This is due to the fact that the observed Markov state i, which determines our choice of λ, is measurable with respect to Σ_t.

For every Markov state i at stage t we now require S cuts, one for each possible value of the Markov state at the previous stage. Therefore, we use parameters $\lambda +_t (s)$ and solve

$$
\begin{aligned}
\tilde{Q}_{tsi}(x_{t-1}, u_t, \xi_{ti}) = \min \quad & (1 - \lambda_t(s))(c_t^T x_t + \lambda_{t+1}(i) u_{t+1} + \theta_{t+1,i}) + \lambda_t(s)\alpha^{-1} v_t \\
s.t. \quad & A_t x_t = b_t(\xi_{ti}) - E_t x_{t-1}, \quad [\pi_{ts}(\xi_{ti})], \\
& v_t - (c_t^T x_t + \lambda_{t+1}(i) u_{t+1} + \theta_{t+1}) \geq -u_t, \quad [\phi_{ts}(\xi_{ts})] \\
& \theta_{t+1} + \sum_{j=1}^{S} P_{ij}^{(t)} \bar{\pi}_{t+1,i,j,k}^T E_{t+1} x_t + \\
& \sum_{j=1}^{S} P_{ij}^{(t)} \bar{\phi}_{t+1,i,j,k} u_{t+1} \geq \sum_{j=1}^{S} P_{ij}^{(t)} \bar{h}_{t+1,i,j,k}, \quad k = 1, 2, \ldots, K_{t+1}, \\
& x_t \geq 0, \; v_t \geq 0.
\end{aligned}
$$

At the kth iteration

$$\bar{\pi}_{t+1,i,j,k} = \mathbb{E}[\pi_{t+1,i}(\xi_{t+1,j}) | W_t = i, \; W_{t+1} = j],$$

$$\bar{\phi}_{t+1,i,j,k} = \mathbb{E}[\phi_{t+1,i}(\xi_{t+1,j}) | W_t = i, \; W_{t+1} = j],$$

$$\bar{h}_{t+1,i,j,k} = \mathbb{E}[Q_{t+1,i,j,k}(\bar{x}_t^k, \bar{u}_{t+1}^k, \xi_{t+1,j}) | W_t = i, \; W_{t+1} = j] + \bar{\pi}_{t+1,i,j,k} E_{t+1} \bar{x}_t^k + \bar{\phi}_{t+1,i,k} \bar{u}_{t+1}^k$$

$\bar{x}_t^k$ and $\bar{u}_{t+1}^k$ are the values of x_t and u_{t+1} obtained at the k th forward pass of SDDP, assuming $N = 1$.

The cuts identified by this approach in Markov state i at stage t that correspond to the previous state s are only valid for realizations of the Markov chain that visit s at stage $t-1$. One option would be to compute the cuts alongside the $S-1$ cuts for the remaining states whenever the Markov chain visits state i at stage t. Another approach would be to add the cut corresponding to s one at a time to those cuts stored for Markov state i whenever a forward pass of the algorithm visits state s immediately before i.

A more general approach where the authors are using Markov states to represent stage-wise dependence in the uncertain parameters using *Markov risk measures* can be found in the work of Ruszczyński (2010).

5.3 Approximate Dynamic Dual Programming

Again, the goal is to solve a Markov Decision Problem, but now we want to take advantage of the concept of scenario lattices, that we introduced in Section 4. The algorithm that allows this modification is called *approximate dynamic dual programming* and was first proposed in Löhndorf (2011) and Löhndorf et al. (2013). One of the first problems solved by implementing an ADDP algorithm was the day-ahead bidding market. Löhndorf (2011) divided the multi-stage problem into an intrastage and an interstage problem. The intrastage problem included day-ahead bidding decisions as well as hourly reservoir operations where prices were assumed to be random. On the interstage step of the model, reservoir management from day to day, modeled as a Markov decision process, were handled. To solve the problem efficiently, the authors proposed to integrate SDDP with ideas from ADP.

The following procedure was used: A relaxed version of the intrastage problem was used to approximate the value function of the interstage problem. Then, as in SDDP, the interstage problem was solved by using forward simulation to sample candidate decisions and backward recursion to construct an approximation of the value function. The next step was to construct a polyhedral approximation by removing candidate decisions that did not improve the approximation quality by a given epsilon to accelerate the sampling process. The obtained polyhedral value function approximation of the interstage problem was then used within the original intrastage problem to find near-optimal bidding and operational decision.

Following Löhndorf et al. (2013), we will explain the ADDP method in more detail. Before, we have to make a few assumptions concerning the structure of the problem we want to solve:

- The optimization problem is linear/ convex,
- The environmental state of the problem can be modeled as a Markov process and discretized using a scenario-lattice,
- The problem has relatively complete recourse,
- Uncertainty enters the problem only on the right hand side.

5.3.1 Approximating the Value Function

To refresh our memory, we start with a short overview of the variables we have to deal with.

The objective of the battery operator is to minimize the expected costs for a given environmental state $S_t \in \mathcal{S}_t$ and initial storage states $B_{t-1} \in \mathcal{B}$ at stage $t \in 1, \dots, T$. Here, the variable $\mathcal{B}$ indicates the set of all possible storage states. $x = x_1, \dots, x_T$ again denotes the decision policy. Note that the random profits are now referred to as $f_t(S_t, B_{t-1}, x_t)$. γ again is the discount factor. If we have a given start storage state B_0 and a given salvage value, we can calculate the value of being in state S_t with initial resource state B_{t-1} through

$$V_t(S_t, B_{t-1}) = \max_{x_t}\{\mathbb{E}[f_t(S_t, B_{t-1}, x_t) + \gamma \sum_{S_{t+1} \in \mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|S_t) V_{t+1}(S_{t+1}, B_t(x_t))]\} \quad (5.26)$$

for $S_t \in \mathcal{S}_t,\ B_{t-1} \in \mathcal{B}$ and $t = 1, \dots, T$ (See 3.11,5.1).

As in 5.1, we will need the post decision value for fixed $\bar{V}_T$:

$$\bar{V}_t(S_t, B_t) = \sum_{S_{t+1} \in \mathcal{S}_{t+1}} \mathbb{P}(S_{t+1}|S_t) V_{t+1}(S_{t+1}, B_t) \quad (5.27)$$

for $S_t \in \mathcal{S}_t, B_t \in \mathcal{B}$ and $t = 1, \dots, T$. In the course of the ADDP algorithm we will recursively build an approximation of the post-decision value. To do so, we need to model the post-decision value $\mathcal{V}_s$ as a concave, piecewise-linear function of the final storage state B_{sjT}. This approach of handling the post-decision value function was first proposed by Pereira and Pinto (1991). For a given state S_t, the post-decision value function can be defined as the minimal of a set of hyperplanes $\mathcal{N} = \{1, \dots, N\}$ with intercepts $a_n(S_t)$ and slopes $b_{nj}(S_t)$ so that the future value of storage is given by

$$\mathcal{V}_s = \min_{n \in \mathcal{N}}\{a_n(S_t) + \sum_{j \in \mathcal{J}} b_{nj}(S_t) B_{sjT}\} \qquad \forall s \in \mathcal{S}. \quad (5.28)$$

If we now denote the optimal value of the original problem relaxed to a linear problem by $V'_t(S_t, \cdot)$, the post decision value function can be described as a concave, piecewise-linear function. To construct an approximation $\hat{\bar{V}}'_{t-1}(S_{t-1}, B)$ of the post-decision value function, we first have to sample a set of storage states $\{\hat{B}_1, \dots, \hat{B}_N\}$ with $\hat{B}_n \in \mathcal{B}$. Then, we have to calculate the corresponding hyperplanes at points

$$(\hat{B}_{11}, \dots, \hat{B}_{1J}, V'_t(S_t, \hat{B}_1)), \dots, (\hat{B}_{N1}, \dots, \hat{B}_{NJ}, V'_t(S_t, \hat{B}_N)) \quad \forall S_t \in \mathcal{S}_t.$$

Let $\partial_B V_t(S_t, B_t)$ be the set of super-gradients of the function $B_t \to V_t'(S_t, B_t)$. Next, we select one element, $b(S_t)$ from this set of super-gradients, which is the slope of the supporting hyperplane of $V_t'(S_t, \cdot)$ at $(\hat{B}_1, \dots, \hat{B}_J, V_t'(S_t, \hat{B}))$. Then the hyperplane is given by

$$H(S_t, B; \hat{B}_i) = a(S_t) + b(S_t)^T B, \quad a(S_t) = V_t'(S_t, \hat{B}) - \sum_{j \in \mathcal{J}} b_j(S_t) \hat{B}_{ij}, \tag{5.29}$$

with $a(S_t) \in \mathbb{R}$ and $b(S_t) \in \mathbb{R}^J$. Due to the fact that we are considering linear programs, the slopes $b(S_t)$ can be obtained from the dual variables λ associated with the constraints of the original optimization problem.

To conclude, the obtained approximate post-decision value function is given by

$$\hat{\bar{V}}_{t-1}'(S_{t-1}, B) = \min\{ \sum_{S_t \in \mathcal{S}_t} \mathbb{P}(S_t | S_{t-1})(a_n(S_t) + b_n(S_t)^T (B - \hat{B}_n)),\ n = 1, \dots, N\}. \tag{5.30}$$

Here the hyperplane at point $\hat{B}_n$ is the weighted sum of all hyperplanes $H(S_t, B; \hat{B}_n)$ over all successor states.

Proposition 5.3. *(Value function approximation)*
The value function approximations in any iteration are upper bounds for the real value functions.

Proof. A proof is given by Löhndorf et al. (2013) for example. □

5.3.2 Approximate Dual Dynamic Programming

As it would be to much effort to calculate every single one of the hyperplanes, as introduced in the last section, similar to SDDP, the ADDP algorithm uses Monte Carlo simulation to define a set of sample resource states. The algorithm is initialized with an environmental state S_1, a resource state B_0, initial value functions $\hat{\bar{V}}_t'$ and the set $\mathcal{M}_t = \emptyset,\ t = 1, \dots, T$. In the following, an outline of the algorithm is given:

```
    Input arguments: initial value function $(\hat{\bar{V}}_t')_{t=1}^T$, initial states $S_1$ and $B_0$
Do for $n = 1, 2, \dots, N$
  (1) Forward pass
      Do for $t = 1, 2, \dots, T-1$
        (1.1) Sample a scenario $s$ from the lattice
        (1.1) Solve $\hat{B}_{nt} \leftarrow \text{argmin}_{x_t} \{C(S_t, B_{t-1}, x_t) + \gamma \hat{\bar{V}}_t'(S_t, B_t(x_t))\}$ for the single scenario
              $s$
        (1.1) Sample $S_{t+1} \leftarrow S^M(S_t)$
    (2) Backward pass
        Do for $t = T, T-1, \dots, 2$
        (2.1) Do for all $S_t \in \mathcal{S}_t$
            (2.1.1) Do for $m \in \mathcal{M}_t \cup \{n\}$
                (2.1.1.1) Get hyperplane $(a_m(S_t), b_m(S_t)) \leftarrow H_{mt}(S_t, B; \hat{B}_{mt-1}) \in \partial_B V_t'(S_t, \hat{B}_{mt-1})$
        (2.2) If $\exists S_t \in \mathcal{S}_t : |\hat{V}_t'(S_t, \hat{B}_{nt-1}) - V_t'(S_t, \hat{B}_{nt-1})| > \epsilon$ then $\mathcal{M}_t \leftarrow \mathcal{M}_t \cup \{n\}$
        (2.3) Do for all $S_{t-1} \in \mathcal{S}_{t-1}$
            (2.3.1) $\hat{\hat{V}}_{t-1}'(S_{t-1}, B) \leftarrow \min\{\sum_{S_t \in \mathcal{S}_t} \mathbb{P}(S_t|S_{t-1})(a_m(S_t) + b_m(S_t)^T(B - \hat{B}_{mt})), m \in \mathcal{M}_t\}$
    (3) Return post decision value function $\hat{\hat{V}}_t'$ $(t = 1, \dots, T-1)$
```

For more details on the algorithm we refer to (Löhndorf et al., 2013).

Proposition 5.4. *(Convergence of ADDP)*
Denote x^e as the policy obtained by ADDP for $\epsilon > 0$ and x^* as the optimal policy of the relaxed problem. For a given initial resource state level B_0, the policies obtained by ADDP for $\epsilon = 0$ converge to the optimal policy in a finite number of steps. The values obtained from following x^e are at most $\epsilon(T-1)$ worse than the optimal values.

Proof. See (Löhndorf et al., 2013). □

Concluding this section on solution methods, we have gained all the theoretical know-how needed to construct a real life application. Therefore, we will put our stochastic dynamic battery charging problem to test in an application example in the next section.

6 Numerical results

This section aims to numerically analyze an existing battery storage system. To do that, we use the theory developed in previous sections. The system consists of an office building demanding electricity, several PV plants and a 230 kWh Li-Ion battery. We assume that there is just one proprietor who owns the building, the PV plants and the battery. The goal is to operate the battery optimally, aiming at expenditures as low as possible.

For lattice construction and ADDP, we use the QUASAR optimization library, provided by (Löhndorf, 2017). We implement the library in Python, but implementations in Java, Matlab or R would also be possible.

Section 6.1 explains how the lattice used for calculations is constructed. Section 6.2 introduces the two different scenarios with their associated parameters which we are investigating and discusses the special conditions that are used for optimizing by the software. The last Section of this chapter presents the numerical results obtained in our case study.

6.1 Lattice construction

The first step in our analysis is the construction of the corresponding scenario lattice. The data this lattice must contain is composed of electricity prices,PV-generation and electricity consumption.

One reason why we chose QUASAR is that the integrated dynamic optimizer will converge to an optimal solution of the stochastic-dynamic decision problem if the randomness can be represented by a scenario lattice. A big advantage of QUASAR hereby is that the software itself is able to construct lattices. Therefore, three different approaches can be used

- QUASAR is able to construct lattices automatically by discretizing a (possibly continuous) Markov process.
- One can build a lattice manually, starting with an empty lattice.
- QUASAR can fit a lattice statistically from real data.

In our case study, we use the third strategy.

Our goal is to optimize the charging behavior of the battery over a one year time horizon. Due to numerical tractability we have to assume decision making once every four hours. This results in a stochastic optimization problem with 2190 stages. Using the provided data as input, QUASAR builds a 2190 stage scenario lattice, with 100 nodes per stage and forward estimation as base case. This means that state transition probabilities are generated depending on the partition around each node. As distance measure for optimal quantization, we take the squared Euclidean distance.

In the following, we give an overview of the data we use for our setting and of its modifications.

6.1.1 Electricity Prices

To model the random variable 'price', we use data provided at the platform Neon (2016). For our concerns, we take the day-ahead market data from Germany from the years 2012 to 2018 and adapt it in the following way: First, because it is easier to handle, we modify each year so that it has exactly 52 weeks and starts with a Monday each. Next, we have to take into account that seven years of data is far too less to build a descriptive lattice. To solve this issue, we simply shift the data week-wise, five times forwards and five times backwards, resulting in 77 years of data.

As mentioned before, due to numerical tractability we have to change the time resolution of the data from hourly to four-hourly by just taking the mean in four hour steps. As initial state of the lattice, we take the value 0.15€ as the price component. Figure 12 shows the original price data from 2012 in comparison to the simulated price lattice.

6.1.2 PV Generation

We also model the uncertainty concerning 'PV-production' by accessing the data platform Neon (2016) We adapt the data from 2012 to 2018 for German solar generation to our needs. Again, we define a year by exactly 52 weeks, each year starting on a Monday and shift the generation day-wise, five times forwards and five times backwards, to obtain 77 years of PV production data. Then, we change the time resolution to a four-hourly period. The last thing we have to do regarding PV is to normalize the German data to a particular location in Styria (Austria). We assume an installed PV capacity of 240 kWp, which leads to an annual production quantity of 264 000 kWh, taking into account local conditions such as special angle of solar radiation. In Figures 14 and 15, a comparison of the actual PV production input data from 2012 and the generated scenario lattice for PV production

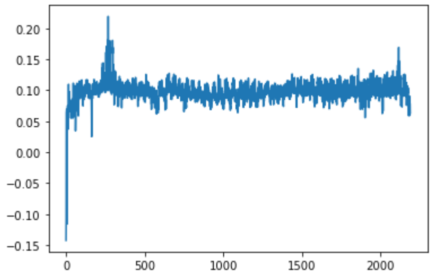


Figure 12: Original price data from 2012

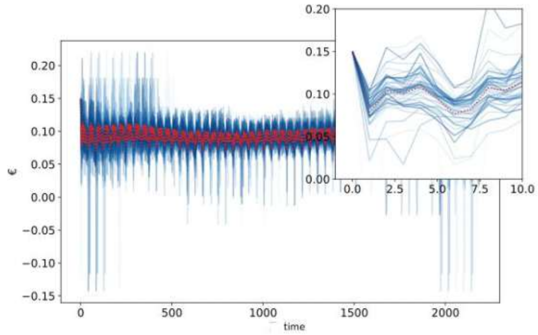


Figure 13: 100 different price scenarios simulated on the lattice. The dotted red line represents the mean of the different scenarios. In the small picture the development of the trajectories, starting at the same initial state is shown.

is given. Figure 16 shows the scenario development on the lattice in greater detail.

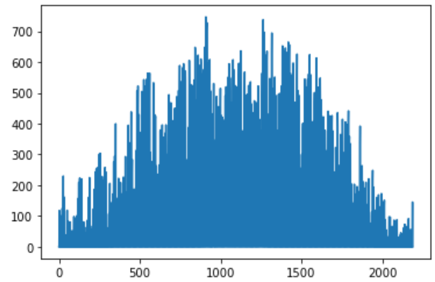


Figure 14: Original PV generation data 2012

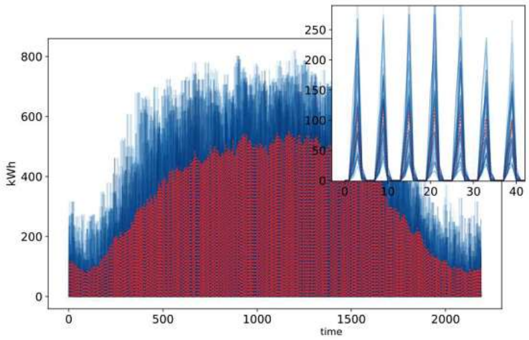


Figure 15: Simulated PV generation scenarios on the lattice. The dotted red line represents the mean of the different scenarios. In the small picture the development of the trajectories for the first week in January is shown.

6.1.3 Electricity Consumption

The hardest part concerning data collection was to find appropriate electricity consumption data for office buildings. Due to the lack of better alternatives, we use data from DOE (2012). One can find hourly load profile data for 16 commercial building types including office buildings at different locations in the United States. Because there is just one

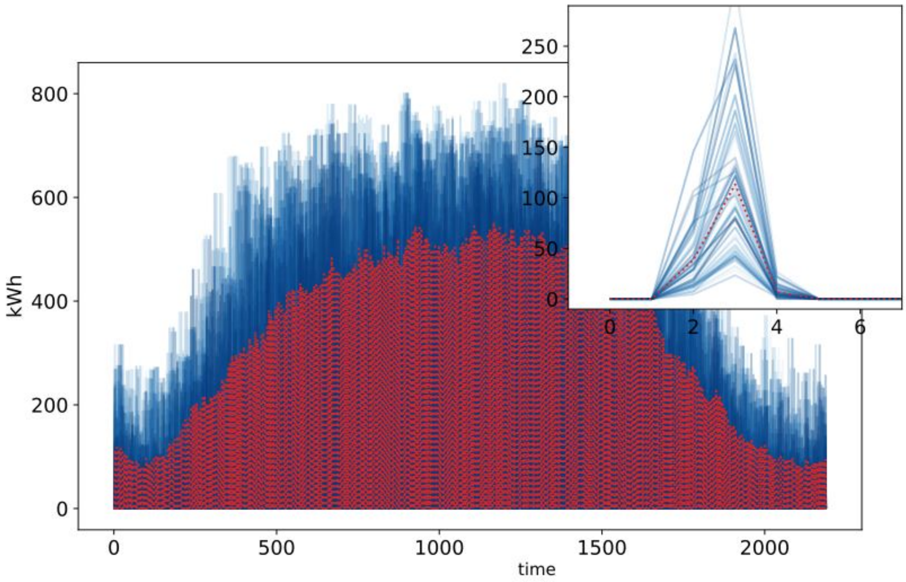


Figure 16: pv scenario development on the lattice. The dotted red line represents the mean of the different scenarios. In the small picture daily fluctuations of pv production are illustrated.

representative year of data available for every building , we take the yearly load profiles of six different office buildings. We adapt the data to the climatic conditions in Austria remove American holidays from the dataset. Then, we exactly proceed as in previous cases of prices and PV production, in particular taking means every four hours and shifting the data week-wise five times forwards and five times backwards, resulting in 77 years of consumption data. We assume the annual electricity demand to be around 800 000 kWh. So as a last step, we have to normalize the yearly consumption. As initial state of the lattice for consumption we take 120 kWh.

Figure 17 shows the input data load profile from 2012 compared to the electricity consumption lattice generated by QUASAR.

Note that, if looking at the data in precise terms, one generally has to pay attention to winter and summer time (sometimes referred to as daylight saving time) conventions for the geographical area in question. Nowadays, some places around the world use these conventions, others don't. The application of daylight saving time includes advancing the clock from 2:00 to 03:00 in spring resulting in a day with only 23 hours. In autumn, the clock is set back from 03:00 to 02:00 implying a 25 hours day. This behavior influences the electricity demand in affected areas, while PV-production is not influenced. As we are conducting an analysis of yearly charging and discharging behavior, the influence of one

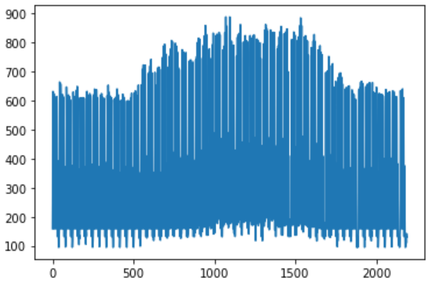


Figure 17: Original consumption data 2012

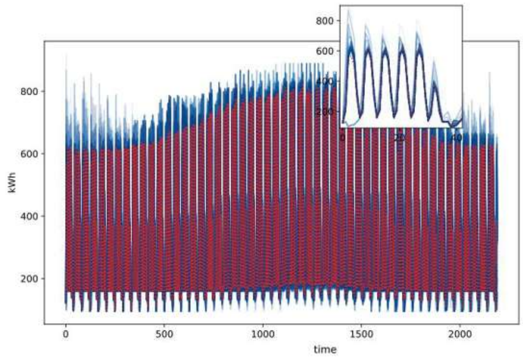


Figure 18: Electricity consumption scenarios simulated on the lattice. The dotted red line represents the mean of the different scenarios. In the small picture the development of the trajectories in the first week of January is shown. The low electricity consumption is representative for the demand profile of an office building

hour more or less is insignificantly small, so we simply ignore it completely.

6.1.4 Comparison of Data and Lattice

The figures in the last paragraph have already shown that there are no striking inconsistencies between the true data and the constructed lattices at first sight. Taking a look at the descriptive statistics in Table 1, we can actually see that important properties such as mean, variance or skewness as well as serial dependence remain unchanged.

Table 1: Descriptive statics data vs lattice

data	**mean**	**std**	**min**	**0.5%**	**2.5%**	**50%**	**97.5%**	**99.5%**	**max**
pv	120.485	176.111	0.000	0.000	0.000	23.451	587.734	689.449	818.954
cons	365.685	221.074	94.537	94.537	114.734	272.874	796.394	835.800	919.311
price	0.094	0.016	-0.142	0.048	0.065	0.093	0.125	0.144	0.219

lattice	**mean**	**std**	**min**	**0.5%**	**2.5%**	**50%**	**97.5%**	**99.5%**	**max**
pv	120.458	175.037	0.000	0.000	0.000	24.891	588.492	687.109	800.302
cons	365.813	220.931	94.537	97.332	116.858	273.256	797.672	826.965	888.876
price	0.092	0.011	-0.142	0.060	0.070	0.093	0.113	0.120	0.181

6.2 Implementation

In the course of our case study, we examine two different scenarios which merely differ in the assumed costs of the battery. In the following, the taken assumptions are summarized.

6.2.1 Parameters and Modeling- Base Scenario

Having constructed the lattice, the next step is to model the decision problem with QUASAR. Therefore, the parameters summarized in Table 2 are used. Furthermore, we make the following assumptions:

Table 2: Parameters used for the base scenario, representing actual battery prices

Parameter	**Value**
capacity of the battery (kWh)	230
efficiency of the battery (%)	98
fixed costs of the battery (€/kWh)	920
battery costs for (dis)charging (€/kWh)	0.076

Assumption 6.1. *On account of the short planning horizon of one year, we assume that there are no maintenance costs.*

Assumption 6.2. *Due to the fact that it is possible to fully charge the battery in four hours, we take the maximal capacity of the battery as upper bounds on storage capacity as well as on charge and discharge rate.*

Assumption 6.3. *Because of the low level (about 3% per month) of self-discharge of a Li-Io battery we ignore this aspect in the simulation completely.*

Assumption 6.4. *We assume that the battery is empty at the begin of the planning horizon.*

6.2.2 Parameters and Modeling- Reduced Battery Costs

Due to the rapid decline in battery prices in the past decades, we decided to examine the effects of a cheaper battery on the simulation outcomes.

The parameters used for modeling the battery system are summarized in Table 3. Furthermore, we still assume that assumptions 6.1 to 6.4 hold true.

Table 3: Parameters used in the scenario with reduced battery costs

Parameter	**Value**
capacity of the battery (kWh)	230
efficiency of the battery (%)	98
fixed costs of the battery (€/kWh)	460
battery costs for (dis)charging (€/kWh)	0.038

6.2.3 Optimizing with QUASAR

After implementing the model with the related parameters from 6.2.1 and 6.2.2 we are ready to hand over the constructed lattice and the model to the optimizer. To solve the stochastic optimization problem, we run ADDP for 2465 iterations. Hereby, the algorithm rejects supporting hyperplanes that improve the current approximation by less than $\epsilon = 1.e - 6$. Dominated cuts are removed from the value function automatically. The accuracy of the convergence check, defined as the ratio of the half width of the confidence interval and the sample mean, is set as 0.05. The width of the confidence level of the convergence check is set as 0.95. After performing 5 iterations, the solver checks a termination criterion. The linear solver used by the program is CLP's dual simplex algorithm.

6.3 Results

Now we turn to studying the actual performance of the ADDP approach. Therefore, we evaluate policies for five different solution strategies for both scenarios, i.e. the base scenario and the scenario with lower battery costs.

First, we compute the solution of the deterministic problem, meaning that we ignore the stochastic part of it and solve the dynamic part of the optimization problem by simply using the expectation of the stochastic process. We refer to this scenario as 'static' in the sense that, once we come up with a plan to solve the problem, we shall not revise it over time on the basis of contingencies. Next, we evaluate the solution of the problem under perfect foresight, the clairvoyant solution. For the third scenario, we use the ADDP approach and therefore assume a risk-neutral decision maker where the battery operator minimizes the expected costs. The fourth problem assumes a strong risk averse decision maker who maximizes a weighted combination of expected value and CVaR. In the last scenario, the optimal behavior of a weak risk averse battery operator is investigated.

Table 4 summarizes the considered solution strategies.

For both scenarios we start with comparing the losses of the five different solution strategies before taking a closer look at each of them.

Table 4: Summary of the considered solution strategies

Name	**Description**
static	deterministic problem, using the expectations of the stochastic process
clairvoyant	solution under perfect foresight
stochastic	stochastic solution with risk neutral decision maker
strong risk aversion	stochastic solution with strong risk aversion of the decision maker
weak risk aversion	stochastic solution with weak risk aversion of decision maker

To inspect the performances of the policies determined for the different solution strategies, we perform forward simulations of the decision problem, using the respective policy to make decisions. Therefore, we generate 100 samples with the input data and analyze the performance of every solution strategy based on the sampled scenarios.

6.3.1 Base Scenario

For the Base Scenario, we consider the parameters introduced in section 6.2.1. Under those conditions, Table 5 summarizes the mean expenditures from following the optimized policies and its standard deviation, whereby positive numbers imply income generated by selling electricity to the grid, while negative numbers account for the amount of money the operator has to pay for electricity. We observe that the values of mean annual electricity costs are negative in every scenario, implying high expenditures for electricity. This can be explained by the relatively small share of PV generation compared to the consumption level. The relation of high consumption level to low PV generation goes hand in hand with a loss of the possibility to sell PV power or stored electricity to the grid, because the whole energy would be needed to fulfill the demand.

Nonetheless, the results are in line with the underlying theory: The clairvoyant solution provides an upper bound for the performance of the stochastic solution, whereas the static

Table 5: Comparison of the loss distributions following the optimal policies

	mean	**std**	**min**	**25%**	**50%**	**75%**	**max**
static	-52 894.818	3 142.911	-57 352.592	-56 181.705	-52 408.751	-50 259.186	-47 471.765
clairvoyant	-52 888.107	3 141.656	-57 352.592	-56 174.060	-52 391.745	-50 252.015	-47 469.163
stochastic	-52 889.180	3 142.041	-57 352.592	-56 174.089	-52 395.597	-50 252.015	-47 469.163
strong risk aversion	-52 888.338	3 141.528	-57 352.592	-56 174.060	-52 391.745	-50 252.015	-47 469.163
weak risk aversion	-52 889.239	3 138.041	-57 317.331	-56 116.587	-52 341.237	-50 198.696	-47 448.011

solution provides a lower bound. The electricity costs resulting from following a risk averse solution strategy are higher than following a risk neutral approach. Figure 19 outlines these findings by comparing the mean annual expenditures for electricity for the different scenarios ignoring risk. Figure 20 shows the loss distribution of the static case compared

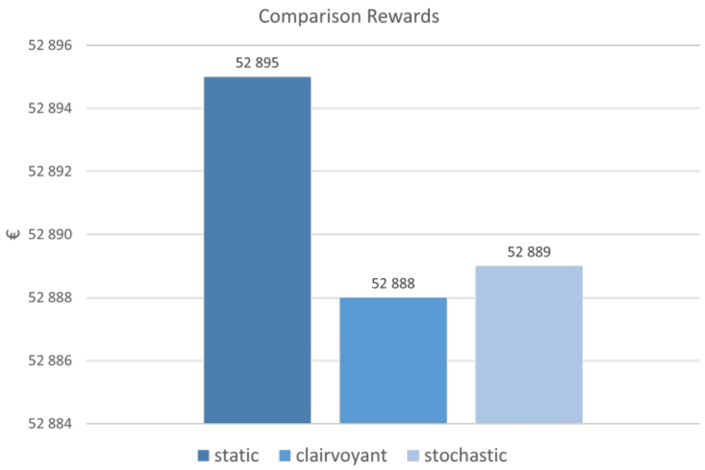


Figure 19: Comparison of the mean annual expenditures under the assumption of high battery costs, considering a static decision maker, a clairvoyant decision maker and a stochastic decision maker.

to the stochastic scenario. We can observe that the mean total costs are higher if we follow the solution determined by the deterministic approach than the results we get following the policy obtained by the ADDP algorithm.

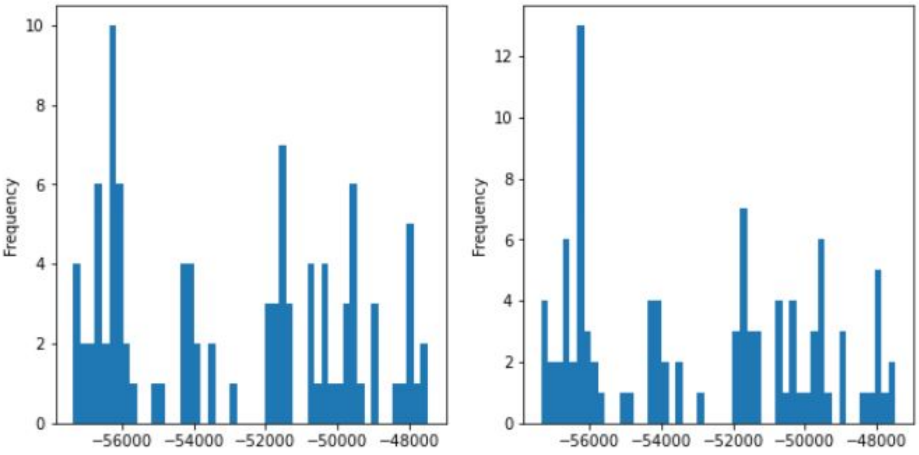
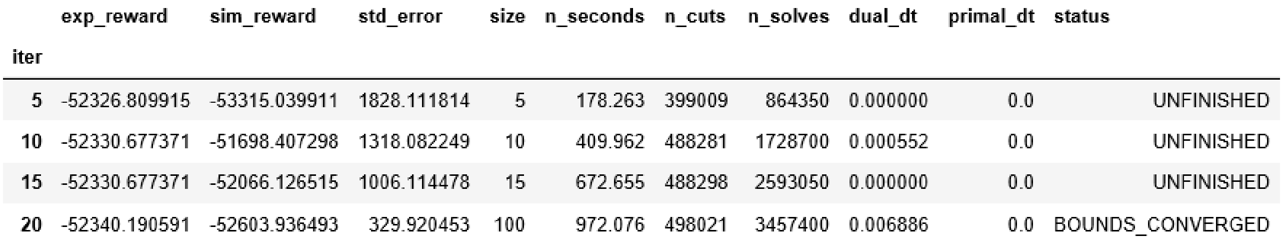


Figure 20: Comparison of the loss distributions of the solution approaches 'stochastic' (left) and 'static' (right)

Stochastic Solution Strategy

Now we take a closer look at the determined stochastic solution policy. First, the convergence behavior of the stochastic solution is analyzed. Therefore, a summary of the algorithm info is given in Table 6, including the expected expenditures of every step, the related simulation losses, the standard deviation, the amount of forward simulations the algorithm run, the amount of time the calculation took, the amount of cuts the algorithm created, the amount of solves it took and the primal and dual values of the problem. We can observe that the solver has already verified statistical convergence after 20 steps. For the convergence check the solver ran 100 forward simulations and thereby tested the difference between expected loss (dual) and simulated loss (primal).

Table 6: Summary of algorithm information for the stochastic scenario



iter	exp_reward	sim_reward	std_error	size	n_seconds	n_cuts	n_solves	dual_dt	primal_dt	status
5	-52326.809915	-53315.039911	1828.111814	5	178.263	399009	864350	0.000000	0.0	UNFINISHED
10	-52330.677371	-51698.407298	1318.082249	10	409.962	488281	1728700	0.000552	0.0	UNFINISHED
15	-52330.677371	-52066.126515	1006.114478	15	672.655	488298	2593050	0.000000	0.0	UNFINISHED
20	-52340.190591	-52603.936493	329.920453	100	972.076	498021	3457400	0.006886	0.0	BOUNDS_CONVERGED

Figure 21 gives an overview of the optimal charging and discharging behavior of the battery. The dark blue dotted line marks the mean over the 100 considered scenarios. One can see that in the base case the battery is nearly never used. This is due to the high expenditures in conjunction with the high battery costs.

Strong Risk Aversion

The objective of the strong risk averse decision maker is to minimize the expected shortfall by penalizing large losses, instead of maximizing the expected profit. The resulting objective function is:

$$\mathcal{A}_{strong}(X) = 0.2\ \mathbb{E}(X) + 0.8\ \mathrm{CVaR}_{0.95}(X) \tag{6.1}$$

It is important to note that in QUASAR no automatic convergence check exists, if we use the algorithm with a risk functional. For that reason we define the iterations until termination as 2500. Table 7 shows a summary of algorithm information using the above defined risk averse objective function. Notice, that the calculation time compared to the risk neutral (stochastic) approach is now about 390 times higher. This is due to the

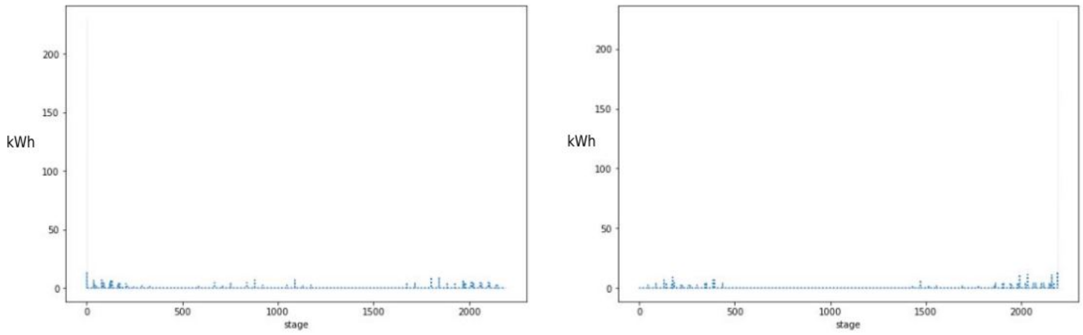
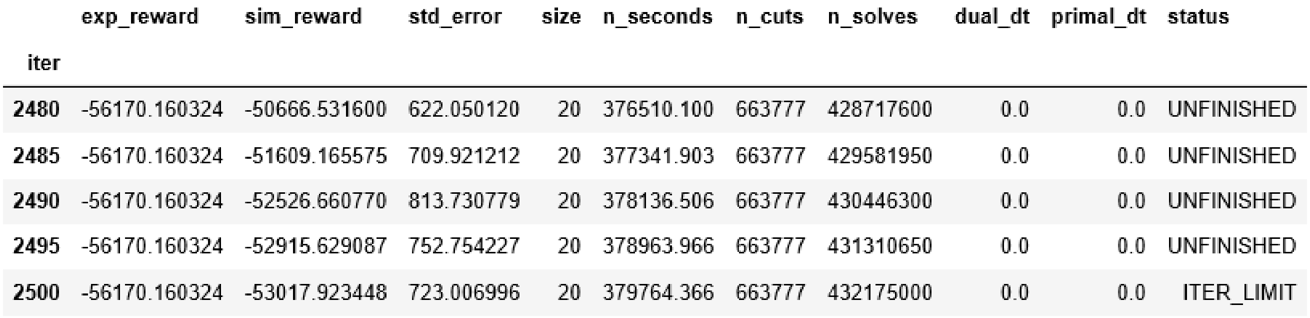


Figure 21: Comparison of the optimal charging and discharging behavior. The blue dotted line represents the mean value of the amount of kWh which are charged and discharged over the considered 100 scenarios. Actually, the battery is nearly never used.

relatively high number of iterations we set. Table 5 shows that the losses of a strong weak averse decision maker are a little bit lower on average, than in the risk neutral case, even if the worst case and best case scenarios are the same.

Weak Risk Aversion

Table 7: Summary of algorithm information using a strong risk averse objective function

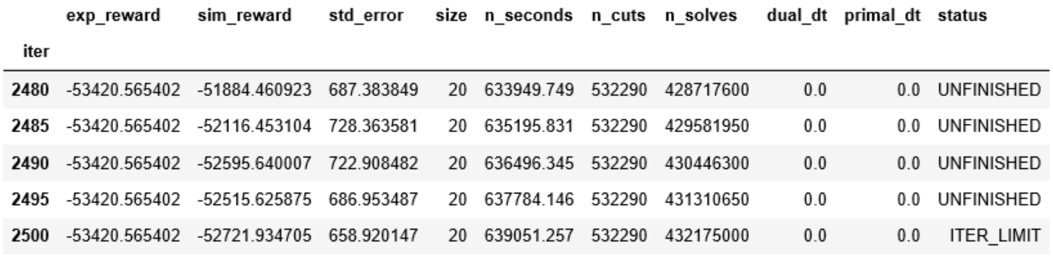


	exp_reward	sim_reward	std_error	size	n_seconds	n_cuts	n_solves	dual_dt	primal_dt	status
iter										
2480	-56170.160324	-50666.531600	622.050120	20	376510.100	663777	428717600	0.0	0.0	UNFINISHED
2485	-56170.160324	-51609.165575	709.921212	20	377341.903	663777	429581950	0.0	0.0	UNFINISHED
2490	-56170.160324	-52526.660770	813.730779	20	378136.506	663777	430446300	0.0	0.0	UNFINISHED
2495	-56170.160324	-52915.629087	752.754227	20	378963.966	663777	431310650	0.0	0.0	UNFINISHED
2500	-56170.160324	-53017.923448	723.006996	20	379764.366	663777	432175000	0.0	0.0	ITER_LIMIT

For the weak risk averse decision maker we assume the following objective function:

$$\mathcal{A}_{weak}(X) = 0.8\ \mathbb{E}(X) + 0.2\ \mathrm{CVaR}_{0.95}(X) \tag{6.2}$$

Table 8 presents the summary of the algorithm information. Again, compared to the strong risk averse solution strategy, the calculation time nearly doubled. As we can see in Figure 22, the distribution of the resulting optimal expenditures is identical to the one in the stochastic case, where just the expectation is maximized, and the distribution function for the strong risk averse decision maker. An explanation for this surprising result



iter	exp_reward	sim_reward	std_error	size	n_seconds	n_cuts	n_solves	dual_dt	primal_dt	status
2480	-53420.565402	-51884.460923	687.383849	20	633949.749	532290	428717600	0.0	0.0	UNFINISHED
2485	-53420.565402	-52116.453104	728.363581	20	635195.831	532290	429581950	0.0	0.0	UNFINISHED
2490	-53420.565402	-52595.640007	722.908482	20	636496.345	532290	430446300	0.0	0.0	UNFINISHED
2495	-53420.565402	-52515.625875	686.953487	20	637784.146	532290	431310650	0.0	0.0	UNFINISHED
2500	-53420.565402	-52721.934705	658.920147	20	639051.257	532290	432175000	0.0	0.0	ITER_LIMIT

Table 8: Summary of the algorithm information for the weak risk averse solution approach

could be, that in our example the results of the different scenarios are almost the same. In combination with the relative high standard deviation error this could lead to almost the same distributions. This result should be verified in further research, maybe using different software or visualization tools.

Alternative initial state

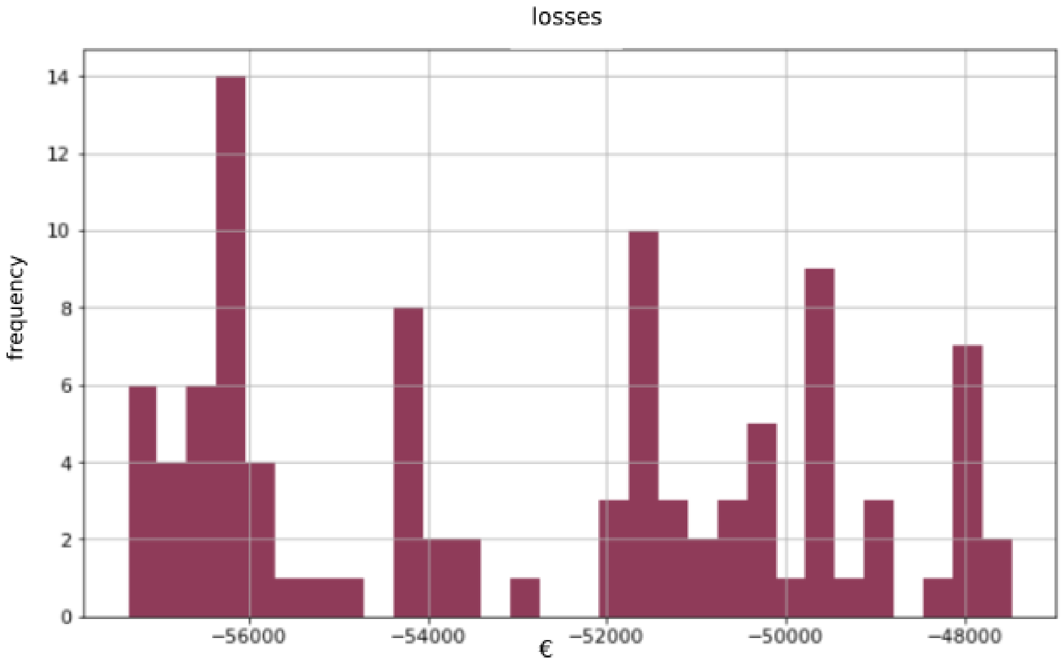


Figure 22: Loss distribution of the strong risk averse solution strategy, the weak risk averse solution strategy and the risk neutral solution strategy. All three of them are identical.

Let us now replace assumption 6.4 with the following assumption:

Assumption 6.5. *We assume that the battery is fully charged at the beginning of the planning horizon.*

The first thing we observe is an extreme change in the decision variable 'state of charge' of the battery. In Figures 23 and 24, a comparison of the development of the charging level of the two different initial states is given. It can be observed that in the case of a fully charged battery at $t = 0$, the battery charge level decreases slowly over the whole time horizon on average.

If we now take a look at the loss distributions of the strong risk averse solution strategy compared to the risk neutral approach again we finally can see the expected shift in the distributions. In this case, the loss of the strong risk averse decision maker is concentrated around its mean with low downside risk but also limited upside potential. This finding is shown in Figure 25. The distribution of the weak risk averse case compared to the risk neutral case is shown in Figure 26. Although, we can observe that the potential of high losses of the weak risk averse battery operator are higher than the losses of the risk neutral operator, the mean losses of the weak risk averse solution strategy are lower than in the risk neutral approach.

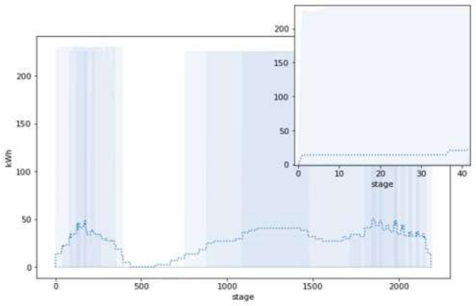


Figure 23: Development of the state of charge of the battery with initial state 0

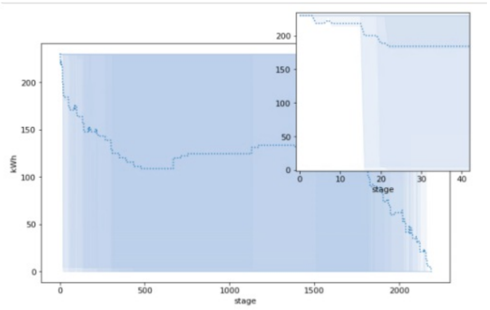


Figure 24: Development of the state of charge of the battery with initial state 230

6.3.2 Reduced Battery Costs

Under the assumption of reduced battery costs the different solution strategies lead to the outputs summarized in Table 9. Again, in line with the underlying theory, the static and clairvoyant solution approaches are the bounds of the stochastic solution strategy. If the decision maker accounts for risk, the maximum expenditures are a bit lower than in the risk neutral approaches, whereas the mean expenditures are a bit higher.

Stochastic Solution Strategy

The first thing we can observe is that in the scenario with reduced battery costs is a

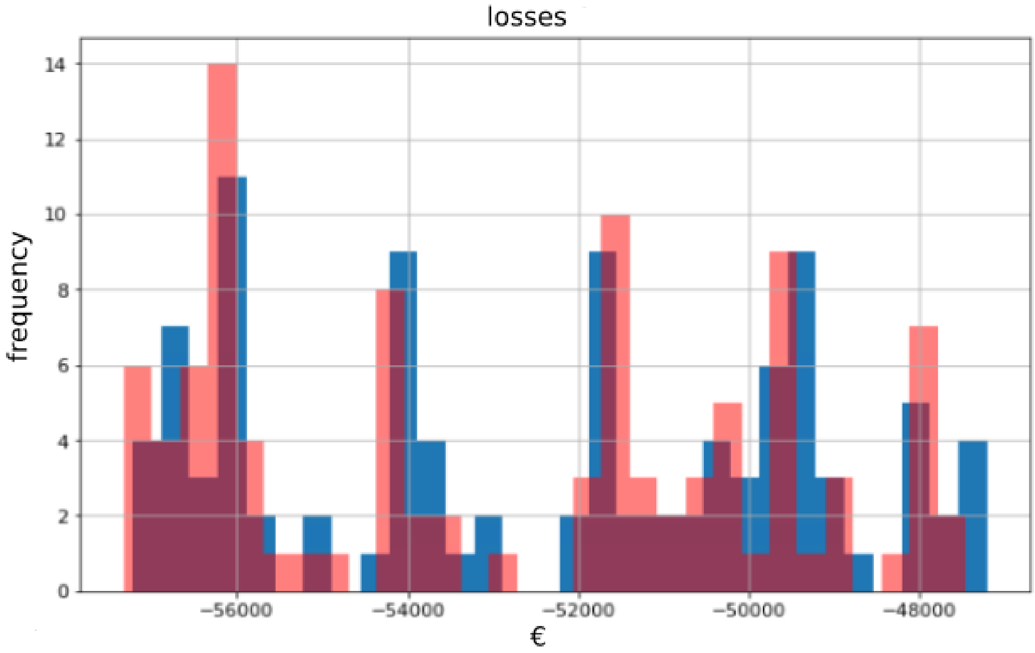


Figure 25: Loss distribution of the strong risk averse solution strategy (red) and the risk neutral solution strategy (blue)

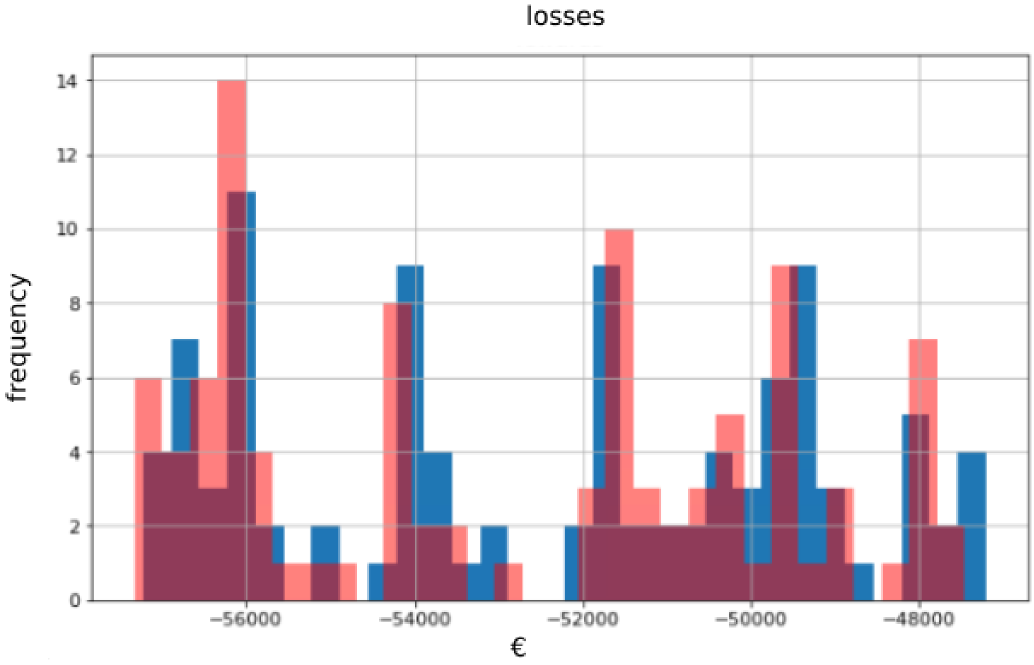


Figure 26: Comparison of the loss distribution of the weak risk averse solution strategy (red) and the risk neutral solution strategy (blue).

Table 9: Comparison of the loss distributions under the assumption of reduced battery costs, following the different solution strategies

	mean	**std**	**min**	**25%**	**50%**	**75%**	**max**
static	-52 881.145	3 138.313	-57 336.640	-56 162.195	-52 385.201	-50 244.159	-47 473.617
clairvoyant	-52 849.176	3 138.136	-57 317.331	-56 116.587	-52 341.237	-50 198.611	-47 446.599
stochastic	-52 849.239	3 138.041	-57 317.331	-56 116.587	-52 341.237	-50 198.696	-47 448.011
strong risk aversion	-52 849.442	3 138.030	-57 317.331	-56 116.587	-52 342.937	-50 198.696	-47 448.011
weak risk aversion	-52 849.239	3 138.041	-57 317.331	-56 116.587	-52 341.237	-50 198.696	-47 448.011

lot more charging and discharging activity than in the base case. Although it would be possible for the model to charge and discharge the battery simultaneously, as predicted, the optimization never finds it optimal to charge and discharge the storage at the same time step (see Figures 27 and 28). Another interesting fact is, that it is nearly never optimal to not fully charge or discharge the battery. This is visualized in Figure 29.

Note that, although the battery is used more often, the overall expenditures are lower than in the scenario with higher battery costs. This is a clear indication of the possibility to reduce electricity expenditures by using a battery.

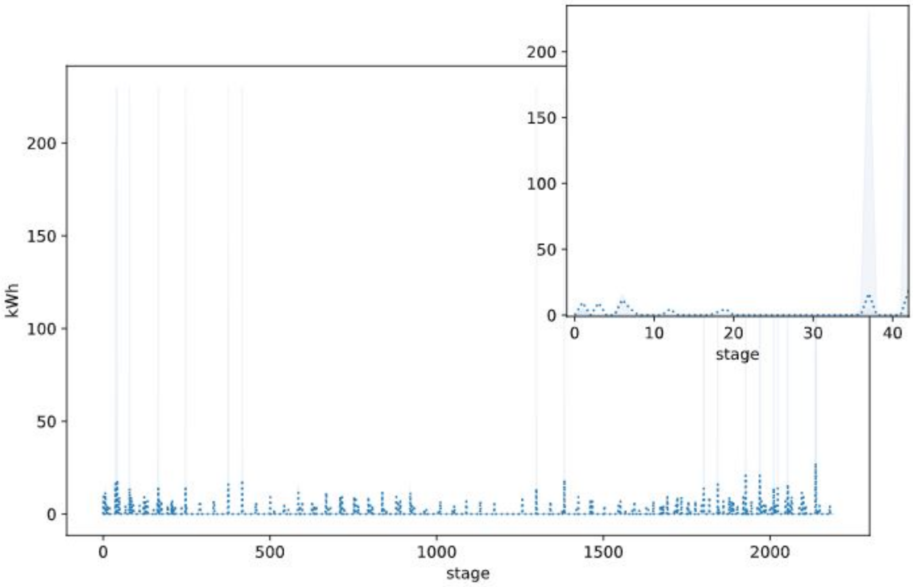


Figure 27: Optimal charging behavior of a battery under the assumption of reduced battery costs. The dotted blue line represents the mean of the 100 considered scenarios. Compared to the Base Scenario with higher battery costs the battery is charged much more frequently.

Figure 30 shows the loss distribution of the static case compared to the stochastic scenario. We can observe that the distributions are in line with the data presented in Table 9. The results determined for the total expenditures are lower on average, if the solution was conducted by stochastic, rather than deterministic optimization.

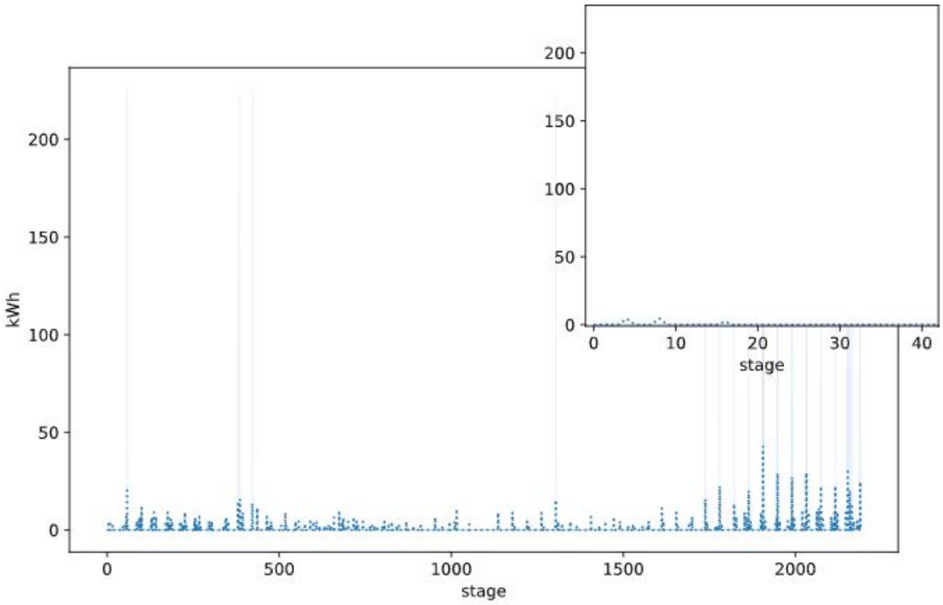


Figure 28: Optimal discharging behavior under the assumption of reduced battery costs. The dotted blue line represents the mean of the 100 considered scenarios. Compared to the Base Scenario with higher battery costs, now, the decision maker makes much more use of the battery.

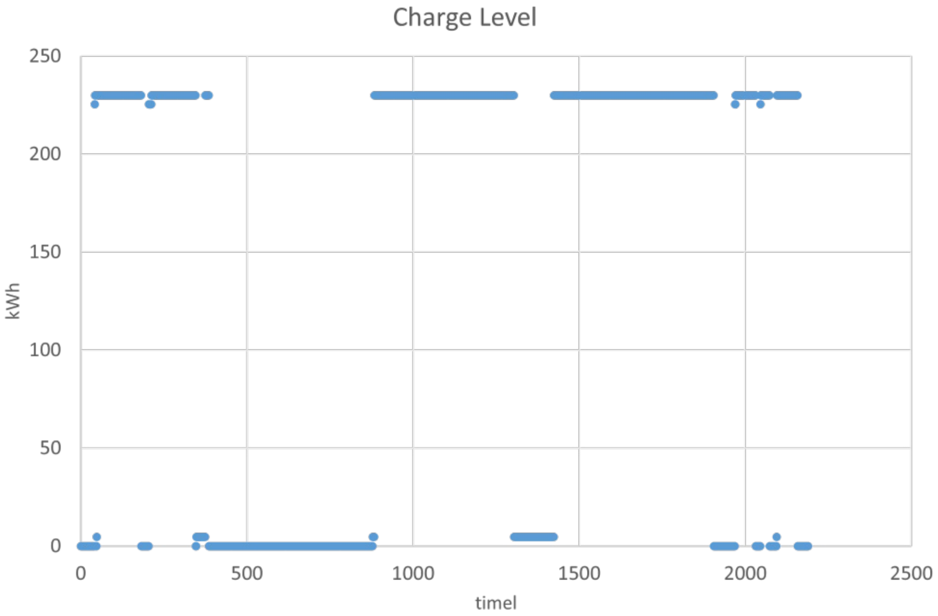


Figure 29: Evolution of the charge level (SoC) of the battery over one year under the assumption of lower battery costs, shown for one representative scenario.

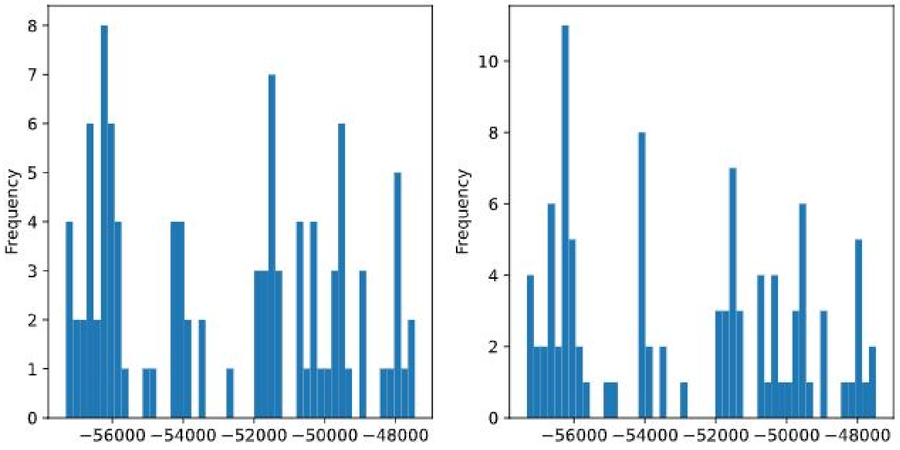


Figure 30: Loss distribution of the stochastic solution approach (left) compared to the loss distribution obtained from the deterministic solution approach (right) under the assumption of reduced battery costs

Figure 31 shows the development of the state of charge of the battery over time.

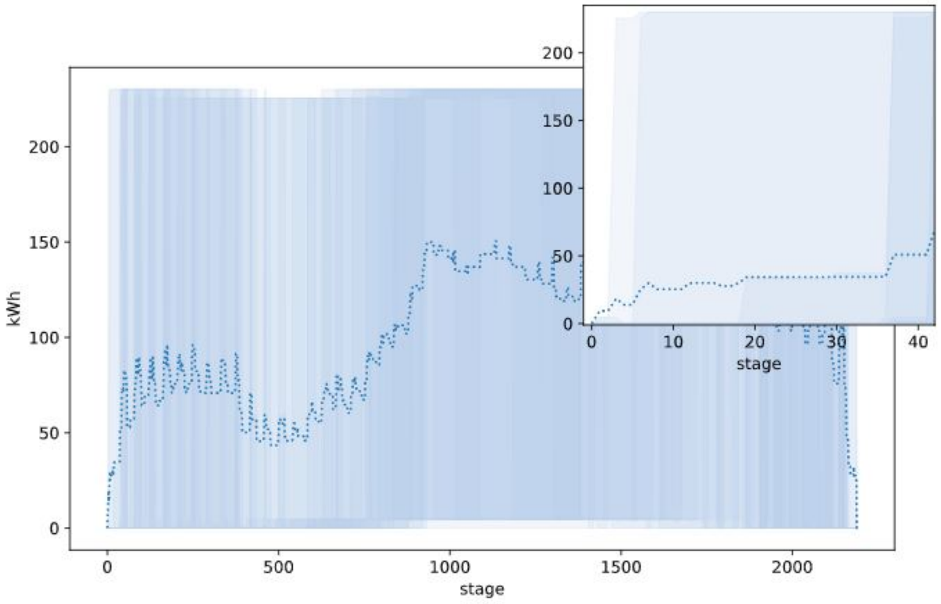


Figure 31: Development of the state of charge of the battery over the simulated year, assuming reduced battery costs. The dotted blue line represents the mean of the 100 considered scenarios.

7 Conclusion

We formulated a multistage stochastic optimization problem for optimal battery system operation, facing stochastic electricity prices, stochastic electricity demand and stochastic PV generation. Furthermore, we modeled different risk preferences of the decision maker, including risk neutral, risk averse and risk loving.

We introduced ADDP as a generic framework for solving discrete-time MDPs and compared the performance of the stochastic solution to a deterministic optimization approach and found that the performance of the stochastic solution is quite promising.

In general, energy storage systems combined with PV panels are an efficient way of increasing the level of self-use of the generated electricity. Batteries also offer the possibility of storing energy at off-peak hours (usually at night), when electricity prices are low, in order to sell or use the stored electricity at a later point in time. But even if we take those profit generating opportunities into account, our case study shows that the operation of a battery is currently not economical under the taken assumptions. However, the combination of PV systems and batteries is increasingly used due to its contribution to other areas, such as emergency protection (blackout protection) or flexibility coverage, often of course due to various funding incentives. Therefore, I propose that economic efficiency should go beyond the classic sense in further considerations and that these aspects should also be taken into account.

The stochastic optimization approach for the integration of renewable energies is very promising. It enables more advantageous decision making compared to conventional optimization. If the battery is operated according to the schedule developed with stochastic optimization, several hundreds of euros can be saved every year. The disadvantage of stochastic optimization is that a lot more input data is required to generate a larger lattice, i.e. more possible nodes per time step, if no underlying probability distribution is used. The rule of thumb herein is:

$$\text{number of nodes per time step} = \sqrt{\text{years of data that is available}}$$

In order to map a realistic number of possible states without storing a probability distribution as a transition function, it would take 10.000 years of available data. Thus, the easier approach would be to construct a stochastic process representing the input data.

In general, it would be important to have good data as a basis for further models. In particular, load profiles for office buildings including a period of several years are currently very difficult to find.

A further disadvantage of stochastic optimization is the currently high computing time. Further procedures and algorithms should be developed in order to make the process suitable for other applications.

Another point that should be examined in more detail are the calculation results concerning the risk averse decision maker. The values should be verified using another software.

The performance price should be integrated as a decision variable in future studies, which could also be a way to reduce electricity costs, especially for bulk purchasers. As a further step, various PV feed-in mechanisms and their effects on the solution should be considered in the calculations. In general, there also is a need to consider the problem in light of the fact that different buildings using the battery may have different owners. Different ways of dividing the profit should be examined and various possible framework conditions for using the battery should be developed.

References

Potential for citizen-produced electricity in the EU. 2016. URL https://www.greenpeace.org/static/planet4-eu-unit-stateless/2018/08/ea9146d5-ea9146d5-160926-potential-of-energy-citizens-media-briefing.pdf.

Hunt Allcott. Rethinking real-time electricity pricing. *Resource and Energy Economics*, 33 (4):820–842, November 2011. ISSN 0928-7655. doi: 10.1016/j.reseneeco.2011.06.003. URL https://www.sciencedirect.com/science/article/pii/S092876551100042X.

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent Measures of Risk. *Mathematical Finance*, 9(3):203–228, 1999. ISSN 1467-9965. doi: 10.1111/1467-9965.00068. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9965.00068. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9965.00068.

Italo Atzeni, Luis G. Ord´o~nez, Gesualdo Scutari, Daniel P. Palomar, and Javier R. Fonollosa. Day-Ahead Bidding Strategies for Demand-Side Expected Cost Minimization. *IEEE*, 2012.

aWATTar GmbH. aWATTar, 2021. URL https://www.awattar.at/.

Kyri Baker, Gabriela Hug, and Xin Li. Energy Storage Sizing Taking Into Account Forecast Uncertainties and Receding Horizon Operation. VOL. 8(1), 2017.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2012. ISBN 978-1-886529-43-4. Google-Books-ID: REp7swEACAAJ.

Dimitri P. Bertsekas and John N. Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE conference on decision and control*, volume 1, pages 560–564. IEEE, 1995.

Arnab Bhattacharya, Jeffrey P. Kharoufeh, and Bo Zeng. Managing Energy Storage in Microgrids: A Multistage Stochastic Programming Approach. (1, January 2018), 2018.

H. Bindner, T. Cronin, P. Lundsager, and James F. Manwell. Lifetime modelling of lead acid batteries. April 2005. URL https://www.osti.gov/etdeweb/biblio/20607163.

John Birge. The Value of the Stochastic Solution in Stochastic Linear Programs With Fixed Recourse. *Mathematical Programming*, 24, December 1982. doi: 10.1007/BF01585113.

John R. Birge and Francois Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer, New York Dordrecht Heidelberg London, 2.auflage edition, 2011. ISBN 978-1-4614-0236-7.

J. Frédéric Bonnans, Zhihao Cen, and Thibault Christel. Energy contracts management by stochastic programming techniques. *Annals of Operations Research*, 200(1):199–222, 2012. Publisher: Springer.

Severin Borenstein. The Long-Run Efficiency of Real-Time Electricity Pricing. 2005a.

Severin Borenstein. The Long-Run Efficiency of Real-Time Electricity Pricing. *The Energy Journal*, 26:93–116, July 2005b. doi: 10.2307/41319500.

Paolo Brandimarte. *From Shortest Paths to Reinforcement Learning: A MATLAB-Based Tutorial on Dynamic Programming*. Springer Nature, 2021.

Paul Breeze. *Power System Energy Storage Technologies - 1st Edition*. AP, 1 edition, 2018. ISBN 978-0-12-812902-9. URL `https://www.elsevier.com/books/power-system-energy-storage-technologies/breeze/978-0-12-812902-9`.

Energie Mobilität Innovation und Technologie Bundesministerium für Klimaschutz, Umwelt. Das Übereinkommen von Paris, 2015. URL `https://www.oesterreich.gv.at/themen/bauen_wohnen_und_umwelt/klimaschutz/1/Seite.1000325.html`.

Berk Celik, Robin Roche, Siddharth Suryanarayanan, David Bouquain, and Abdellatif Miraoui. Electric energy management in residential areas through coordination of multiple smart homes. 2017.

Zhi Chen, Lei Wu, and Yong Fu. Real-Time Price-Based Demand Response Management for Residential Appliances via Stochastic Optimization and Robust Optimization. (4), 2012.

G. Coppez, S. Chowdhury, and S.P. Chowdhury. Impacts of energy storage in distributed Power Generation: A review. In *2010 International Conference on Power System Technology*, pages 1–7, October 2010. doi: 10.1109/POWERCON.2010.5666075.

Sylvain Corlay and Gilles Pagès. Functional quantization-based stratified sampling methods. *Monte Carlo Methods and Applications*, 21(1):1–32, March 2015. ISSN 1569-3961. doi: 10.1515/mcma-2014-0010. URL `https://www.degruyter.com/document/doi/10.1515/mcma-2014-0010/html`. Publisher: De Gruyter Section: Monte Carlo Methods and Applications.

Boris Defourny, Damien Ernst, and Louis Wehenkel. Multistage stochastic programming: A scenario tree based approach to planning under uncertainty. In *Decision theory models for applications in artificial intelligence: concepts and solutions*, pages 97–143. IGI Global, 2012.

Zhang Delong, Li Jianlin, Liu Xueqin, Guo Jianbo, and Xu Shaohua. A Stochastic Optimization Method for Energy Storage Sizing Based on an Expected Value Model. 2019.

European Comission Directorate-General for Energy. Energy : roadmap 2050. 2012. ISSN 978-92-79-21798-2. doi: 10.2833/10759. URL `https://data.europa.eu/doi/10.2833/10759`.

Open Data DOE. Commercial and Residential Hourly Load Profiles for all TMY3 Locations in the United States, 2012. URL `https://data.openei.org`.

B. J. Donnellan, D. J. Vowles, and W. L. Soong. A review of energy storage and its application in power systems. In *2015 Australasian Universities Power Engineering Conference (AUPEC)*, pages 1–6. IEEE, 2015.

European-Comission. The roadmap for transforming the EU into a competitive, low-carbon economy by 2050. 2011. URL `https://ec.europa.eu/clima/sites/clima/files/2050_roadmap_en.pdf`.

Ali Faghih, Mardavij Roozbehani, and Munther A. Dahleh. Optimal utilization of storage and the induced price elasticity of demand in the presence of ramp constraints. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 842–847, December 2011. doi: 10.1109/CDC.2011.6161519. ISSN: 0743-1546.

Johannes Fichtinger. The single-period inventory model with spectral risk measures. In Peter Lang, editor, *Forschungsergebnisse der Wirtschaftsuniversität Wien*, volume 49, page 132. Wien, 1 edition, June 2010. ISBN 978-3-631-61573-7.

Allen I. Fleishman. A method for simulating non-normal distributions. *Psychometrika*, 43 (4):521–532, 1978. Publisher: Springer.

Mahmoud Ghofrani, Amirsaman Arabali, Mehdi Etezadi-Amoli, and Mohammed Sami Fadali. A Framework for Optimal Placement of Energy Storage Units Within a Power System With High Wind Penetration. (2), 2013.

Paul Glasserman. *Monte Carlo Methods in Financial Engineering*. Springer Science & Business Media, March 2013. ISBN 978-0-387-21617-1. Google-Books-ID: aeAlBQAAQBAJ.

Vadim Gorski. Scenario trees models vs. lattice models in stochastic optimization. 2017.

C. Gouveia, J. Moreira, C.L. Moreira, and J.A. Peças Lopes. Coordinating Storage and Demand Response for Microgrid Emergency Operation. *IEEE*, Vol 4(4), 2013.

Abdulelah H. Habib, Vahid R. Disfani, Jan Kleissl, and Raymond A. de Callafon. Optimal Energy Storage Sizing and Residential Load Scheduling to Improve Reliability in Islanded Operation of Distribution Grids. Seattle, 2017.

Josef Haunschmied, Raimund M. Kovacevic, Vladimir Veliov, and Nhat-Vinh Vo. Business models for smart grids including batteries. Technical report, 2016.

Holger Heitsch and Werner Römisch. Scenario Reduction Algorithms in Stochastic Programming. *Computational Optimization and Applications*, 24(2):187–206, February 2003. ISSN 1573-2894. doi: 10.1023/A:1021805924152. URL `https://doi.org/10.1023/A:1021805924152`.

Ronald Hochreiter and Georg Ch. Pflug. Financial scenario generation for stochastic multistage decision processes as facility location problems. *Annals of Operations Research*, 152 (1):257–272, March 2007. ISSN 0254-5330, 1572-9338. doi: 10.1007/s10479-006-0140-6. URL `http://link.springer.com/10.1007/s10479-006-0140-6`.

Kjetil Høyland, Michal Kaut, and Stein W. Wallace. A heuristic for moment-matching scenario generation. *Computational optimization and applications*, 24(2):169–185, 2003. Publisher: Springer.

Mehdi Jafari, Khalid Khan, and Lucia Gauchia. Deterministic models of Li-ion battery aging: It is a matter of scale. *Journal of Energy Storage*, 20, 2018.

Philippe Jorion. Value at Risk: The New Benchmark for Controlling Market Risk. January 2000.

Chanaka Keerthisinghe, Gregor Verbič, and Archie C. Chapman. Addressing the stochastic nature of energy management in smart homes. In *2014 Power Systems Computation Conference*, pages 1–7, August 2014. doi: 10.1109/PSCC.2014.7038377.

Arne Klein, Eric Merkel, Benjamin Pfluger, Anne Held, and Mario Ragwitz. Evaluation of different feed-in tariff design options – Best practice paper for the International Feed-In Cooperation. *Fraunhofer, Energy Economics Group*, 2010.

Iordanis Koutsopoulos, Vassiliki Hatzi, and Leandros Tassiulas. Optimal Energy Storage Control Policies for the Smart Power Grid. *(IEEE SmartGridComm*, 2011.

Raimund Kovacevic. Conditional risk and acceptability mappings as Banach-lattice valued mappings. *Statistics and Risk Modeling*, 29, March 2012. doi: 10.1524/strm.2012.1041.

Raimund M. Kovacevic, Nhat-Vinh Vo, and Jodef Haunschmied. Bilevel Approaches for Distributed DSM Using Internal Individualized Prices. *IEEE*, 2017.

Nils Löhndorf. *Optimization of stochastic-dynamic decision problems with applications in energy and production systems.* phd, uniwien, wien, 2011. URL `https://othes.univie.ac.at/17370/`.

Nils Löhndorf. An empirical analysis of scenario generation methods for stochastic optimization. *European Journal of Operational Research*, 255(1):121–132, November 2016. ISSN 0377-2217. doi: 10.1016/j.ejor.2016.05.021. URL `https://www.sciencedirect.com/science/article/pii/S0377221716303411`.

Nils Löhndorf. QUASAR Optimization Software, 2017. URL `http://quantego.com`.

Nils Löhndorf and D. Wozabal. Indifference pricing of natural gas storage contracts. *undefined*, 2017. URL `https://www.semanticscholar.org/paper/Indifference-pricing-of-natural-gas-storage-L%C3%B6hndorf-Wozabal/38b371a6c04140e4c3f0c85eb7072ea5d9c33abb`.

Nils Löhndorf, David Wozabal, and Stefan Minner. Optimizing Trading Decisions for Hydro Storage Systems Using Approximate Dual Dynamic Programming. *Operations Research*, 61(4):810–823, August 2013. ISSN 0030-364X. doi: 10.1287/opre.2013.1182. URL `https://pubsonline.informs.org/doi/abs/10.1287/opre.2013.1182`. Publisher: INFORMS.

J. MacQueen. Some methods for classification and analysis of mulitvariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, page 690. University of California Press, 1967.

Albert Madansky. Inequalities for stochastic linear programming problems. *Management science*, 6(2):197–204, 1960. Publisher: INFORMS.

Pedram Mokrian and Moff Stephen. A Stochastic Programming Framework for the Valuation of Electricity Storage. In *In: 26th USAEE/IAEE North American Conference. International Association for Energy Economics, Ann Arbor, MI (2006*, January 2006.

Neue Energieönomik Neon. Open Power System Data – A platform for open data of the European power system., 2016. URL https://open-power-system-data.org/.

Dr Jürgen Neubarth. Energiegemeinschaften im zukünftigen österreichischen Strommarkt. page 10, 2020.

A. Oudalov, D. Chartouni, C. Ohler, and G. Linhofer. Value Analysis of Battery Energy Storage Applications in Power Systems. In *2006 IEEE PES Power Systems Conference and Exposition*, pages 2206–2211, October 2006. doi: 10.1109/PSCE.2006.296284.

M. A. Pedrasa, E. D. Spooner, and I. F. MacGill. Robust scheduling of residential distributed energy resources using a novel energy service decision-support tool. In *ISGT 2011*, pages 1–8, January 2011. doi: 10.1109/ISGT.2011.5759152.

Mario VF Pereira. Optimal stochastic operations scheduling of large hydroelectric systems. *International Journal of Electrical Power & Energy Systems*, 11(3):161–169, 1989. Publisher: Elsevier.

Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1):359–375, 1991. Publisher: Springer.

G.Ch. Pflug. Scenario tree generation for multiperiod financial optimization by optimal discretization. *Mathematical Programming*, 89(2):251–271, January 2001. ISSN 1436-4646. doi: 10.1007/PL00011398. URL https://doi.org/10.1007/PL00011398.

Georg Ch Pflug and Werner Romisch. *Modeling, Measuring and Managing Risk*. World Scientific, 2007. ISBN 978-981-270-872-4. Google-Books-ID: ZBZhDQAAQBAJ.

A. B. Philpott and V. L. de Matos. Dynamic sampling algorithms for multi-stage stochastic programs with risk aversion. *European Journal of Operational Research*, 218(2):470–483, April 2012. ISSN 0377-2217. doi: 10.1016/j.ejor.2011.10.056. URL https://www.sciencedirect.com/science/article/pii/S0377221711010332.

Andrew B. Philpott and Ziming Guan. On the convergence of stochastic dual dynamic programming and related methods. *Operations Research Letters*, 36(4):450–455, 2008.

Warren B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons, October 2007. ISBN 978-0-470-18295-6. Google-Books-ID: WWWDkd65TdYC.

Farrokh Rahimi and Ali Ipakchi. Demand Response as a Market Resource Under the Smart Grid Paradigm. *IEEE Transactions on Smart Grid*, 1(1):82–88, June 2010. ISSN 1949-3061. doi: 10.1109/TSG.2010.2045906. Conference Name: IEEE Transactions on Smart Grid.

Mehdi Rahmani-andebili and Haiying Shen. Cooperative distributed energy scheduling for smart homes applying stochastic model predictive control. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2017. doi: 10.1109/ICC.2017.7996420. ISSN: 1938-1883.

Bradford P. Roberts and Chet Sandberg. The Role of Energy Storage in Development of Smart Grids. *Proceedings of the IEEE*, 99(6):1139–1144, June 2011. ISSN 1558-2256. doi: 10.1109/JPROC.2011.2116752. Conference Name: Proceedings of the IEEE.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of risk*, 2:21–42, 2000.

R. Tyrrell Rockafellar and Stanislav Uryasev. Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471, July 2002. ISSN 0378-4266. doi: 10.1016/S0378-4266(02)00271-6. URL `https://www.sciencedirect.com/science/article/pii/S0378426602002716`.

Andrzej Ruszczyński. Risk-averse dynamic programming for Markov decision processes. *Mathematical Programming*, 125(2):235–261, October 2010. ISSN 0025-5610, 1436-4646. doi: 10.1007/s10107-010-0393-3. URL `http://link.springer.com/10.1007/s10107-010-0393-3`.

R.B. Schainker. Executive overview: energy storage options for a sustainable energy future. In *IEEE Power Engineering Society General Meeting, 2004.*, pages 2309–2314 Vol.2, June 2004. doi: 10.1109/PES.2004.1373298.

Alexander Shapiro. Monte Carlo Sampling Methods. In *Handbooks in Operations Research and Management Science*, volume 10 of *Stochastic Programming*, pages 353–425. Elsevier, January 2003. doi: 10.1016/S0927-0507(03)10006-0. URL `https://www.sciencedirect.com/science/article/pii/S0927050703100060`.

Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. *Lectures on Stochastic Programming-Modeling and Theory*. Society for Industrial and Applied Mathematics and the mathematical Programming Society, 2009. ISBN 978-0-0898716-87-0.

Alexander Shapiro, Wajdi Tekaya, Joari Paulo da Costa, and Murilo Pereira Soares. Risk neutral and risk averse stochastic dual dynamic programming method. *European journal of operational research*, 224(2):375–391, 2013. Publisher: Elsevier.

Derek Stenclik, Paul Denholm, and Babu R. Chalamala. The Role of Energy Storage for Renewable Integration, 2017.

John Sun. Car Battery Efficiencies, 2010. URL `http://large.stanford.edu/courses/2010/ph240/sun1/`.

Xiaoqi Tan, Yuan Wu, and Danny H.K. Tsang. Economic analysis of lifetime-constrained battery storage under dynamic pricing. In *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pages 726–731, October 2013. doi: 10.1109/SmartGridComm.2013.6688045.

Stathis Tselepis, I Nickoletatos, Dirk Sauer, and Heinz Wenzl. Literature Survey about Lifetime Expectancy of Components, in particular the Energy Storage System in Existing PV Applications. June 2004.

Peter M. van de Ven, Nidhi Hegde, Laurent Massoulié, and Theodoros Salonidis. Optimal Control of End-User Energy Storage. *IEEE Transactions on Smart Grid*, 4(2):789–797, June 2013. ISSN 1949-3061. doi: 10.1109/TSG.2012.2232943. Conference Name: IEEE Transactions on Smart Grid.