

Integrated day-ahead and intraday self-schedule bidding for energy storage systems using approximate dynamic programming

Benedikt Finnah^a, Jochen Gönsch^{a,*}, Florian Ziel^b,

^a*Mercator School of Management, University of Duisburg-Essen, Lotharstraße 65, 47057 Duisburg, Germany*

^b*Faculty of Business Administration and Economics, University of Duisburg-Essen, Berliner Platz 6-8, 45127 Essen, Germany*

Abstract

Most modern energy markets trade electricity in advance for technical reasons. Thus, market participants must commit to delivering or consuming a certain amount of energy before the actual delivery. In Germany, two markets with daily auctions coexist. In the day-ahead auction market, the energy is traded in 60-minute time slots, which are further partitioned into 15-minute time slots for the intraday auction market. Because of the slow ramp-ups of nuclear and fossil power plants, these price-makers trade mostly in the day-ahead market. Only the residual energy is traded in the intraday market, where the market prices fluctuate substantially more. These fluctuations as well as the expected price difference between these markets can be exploited by fast ramping energy storage systems.

We address the decision problem of an owner of an energy storage who trades on both markets, taking ramping times into account. Because the state variable of our dynamic programming formulation includes all features of our high-dimensional electricity price forecast, this problem cannot be solved to optimality. Instead, we use approximate dynamic programming. In a numerical study based on real-world data, we benchmark the algorithm against an adapted state-of-the-art approach from literature and an expectation model with a receding horizon. Furthermore, we investigate the influence of the price forecast on expected profit and demonstrate that it is essential for the dynamic program to capture the high dimensionality of the price forecast to compete with the expectation model, which does not suffer from the curses of dimensionality.

Keywords: OR in energy, trading for energy storage, multiple sequential markets, electricity price forecast, backwards approximate dynamic programming

*Corresponding author

Email addresses: benedikt.finnah@uni-due.de (Benedikt Finnah), jochen.goensch@uni-due.de (Jochen Gönsch), florian.ziel@uni-due.de (Florian Ziel)

1. Introduction

The energy sector is a large producer of CO₂ and therefore contributes significantly to climate change. Therefore, many countries try to reduce the CO₂ emissions by encouraging power producers to increase the share of renewable energies. This is encouraged by subsidies for produced green energy or by CO₂ prices which have to be paid by conventional CO₂ producing power plants. As the penetration of renewable energy resources increases, an increasing share of the energy production is not controllable. However, the energy production has to always match the fluctuating energy demand. In liberalized energy markets, electricity prices reflect the balance of production and demand. Accordingly, energy storage systems which buy energy at low prices and sell it later at higher prices help to match production and demand, and thus improve grid stability. In most energy markets, market participants must commit to delivering or consuming a certain amount of energy before the actual delivery. In addition, in most countries multiple energy markets coexist. The market rules including the pricing and the temporal sequence differ between countries.

In this paper, we focus on profit maximizing self-schedule bidding of a fast ramping energy storage in the German day-ahead and intraday auction markets. We formulate the decision problem as a dynamic program and solve it heuristically using backwards approximate dynamic programming. Our approach is largely independent of the storage technology, but we focus on a pumped hydro storage (PHS). In Section 1.1 we introduce the focused decision problem. The assumptions needed are stated in Section 1.2. Section 1.3 describes our contribution to the literature and outlines the remainder of the paper.

1.1. Problem statement

We focus on profit maximizing self-schedule bidding for a PHS in the German day-ahead and intraday auction market. In the following, we describe the considered key characteristics of the PHS and the two markets.

The main components of a PHS are the lower and upper water reservoir, the pump, and the turbine. While storing energy, the pump uses electricity to pump water from the lower reservoir to the upper reservoir. This converts the electric energy into potential energy. While discharging the PHS, the water flows from the upper reservoir through the turbine into the lower one. Meanwhile, the turbine converts the potential energy into electric energy. In this paper, we consider the following properties of the PHS: capacity, working power range, efficiency, start-up cost, and ramping-time. The capacity is the maximum amount of energy which can be stored in the PHS. Due to technical restrictions, the pump and the turbine cannot operate at arbitrary low power. They are either switched off completely or operate in the working power range defined by the maximum and minimum power of the turbine and the pump. While the pump or the turbine converts the electric energy into potential energy or vice versa, some energy is lost for the PHS. The share of energy which is not lost is the efficiency of the pump/turbine. When the pump or

the turbine starts to work, they suffer from increased abrasion. The corresponding increase of the maintenance costs is captured by the start-up cost. The ramping-time describes how long the pump and the turbine need to change from one power level to a new, increased/decreased power level. During ramping, the storage owner cannot meet the commitments on the energy markets exactly. The difference is balanced by the transmission system operator (TSO) and the storage owner is paid/billed by the TSO for the under/overproduction. Based on the TSO contracts, intentional under/overproduction is forbidden.

In Germany, two auction markets with daily auction coexist; the day-ahead and the intraday auction market. The key characteristic of the German markets is that the intraday auction market differs in terms of the time length (15 min) of the traded products from the day-ahead market (60 min). In the day-ahead market, market participants can submit multiple price-volume combinations, called bids, for each traded product, until noon. The traded products are commitments to generate or consume a certain amount of energy for each hour of the following day. Forty minutes later, the market clearing prices (MCP) are published and the market participants must fulfill their accepted bids to the MCP. A buying bid is accepted if the price component of the bid is above the MCP. A selling bid is accepted if the price component is below the MCP. To remove the uncertainty regarding acceptance of bids, market participants can use price-accepting bids (also called self-schedule bidding). The latter term is derived from the fact that the production/consumption schedule of the market participant only depends on the self-chosen volumes and not on the market's outcome. The market participant's actual profit/cost, of course, still depends on the realized market prices. The German intraday auction market works similar to the day-ahead market, but with quarter-hourly products. After the intraday auction market closure at 15:00, the market results are published. The main purpose of this market is to capture load ramps with its finer temporal resolution (15 min instead of 1 h at the day-ahead markets). Primarily because of these ramps, the prices in the intraday auction market fluctuate around the day-ahead prices. This is visualized in Figure 1 for an exemplary week. The fluctuation can be exploited by fast ramping energy storage systems. For a detailed overview of the German energy markets see [Viehmann \(2017\)](#). While market participants consume energy they have to pay the obligatory grid fees.

1.2. Assumptions

In this section, we state the assumptions underlying this work in detail.

Assumption 1. *The PHS only trades in the day-ahead and intraday auction market.*

Even if it is desirable to consider all possible trading opportunities (see [Viehmann \(2017\)](#) for an overview), we restrict the trading to the two auction markets. Assumption 1 reduces the complexity and computational burden of the decision problem. The restriction of the considered markets even down to a single market setting is common in the literature (e.g. [Gönsch & Hassler \(2016\)](#), [Zhou et al. \(2016\)](#), [Salas & Powell \(2018\)](#), [Scott \(2019\)](#), [Durante et al. \(2020\)](#), [Ghavidel et al. \(2020\)](#),

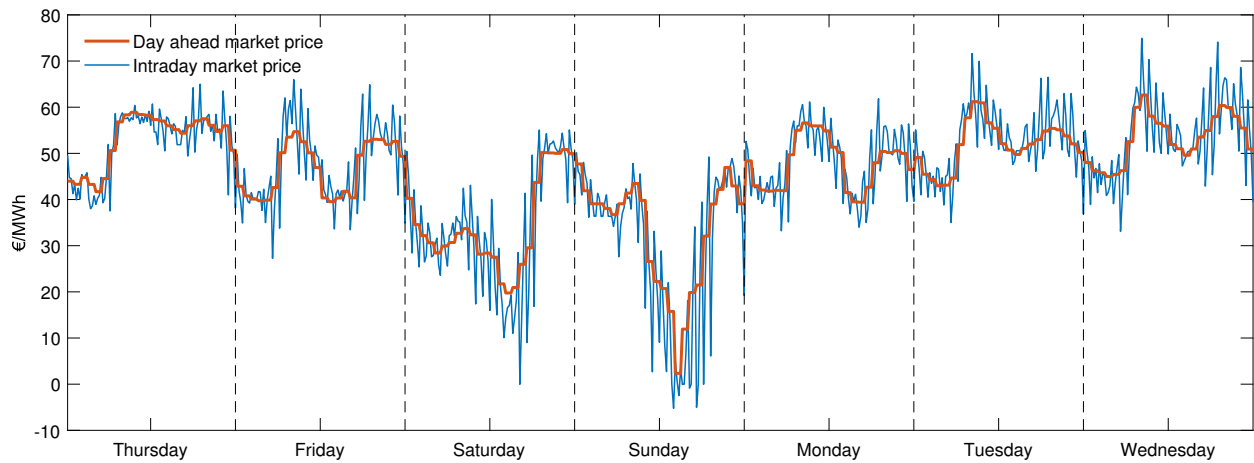


Figure 1: Market prices of an exemplary week (04 July 2018 to 10 July 2018)

[Finnah & Gönsch \(2021\)](#)). In Section 7 we discuss the limitations of our research regarding further trading opportunities (derivative markets, balancing markets, continuous intraday markets, over-the-counter markets) and how our research can be beneficial for market participants which trade in more markets.

Assumption 2. *The PHS uses price-accepting bids/self-schedule bidding.*

Assumption 2 further reduces the complexity of the decision problem. Self-schedule bidding is common for energy storage systems (e.g. [Löhdorf & Minner \(2010\)](#), [Heredia et al. \(2018\)](#), [Khatami et al. \(2020\)](#), [Vahedipour-Dahraie et al. \(2020\)](#), [Xu et al. \(2020\)](#)), as energy storage systems are profitable if the difference between the buying price and the selling price is large enough. The price difference cannot be reflected by price-volume bids. In [Appendix A](#), we numerically investigate the benefits of price-volume bids in a simplified setting. In particular, we show that price-volume bids are not needed for energy storage systems with realistic correlation of the noise of the price process. Price-volume bids can only outperform self-schedule bids with artificially reduced correlation. This may be different for hydropower plants with huge reservoirs (discharge time of days to weeks) and a natural water inflow. For these power plants to be profitable, it is essential to consume the resource at high price hours (e.g. [Fleten & Kristoffersen \(2008\)](#), [Boomsma et al. \(2014\)](#)).

Assumption 3. *The PHS ramps linearly with identical rates for ramping up and down.*

Identical ramping rates reduce the notation needed but do not influence the computational burden of the decision problem. However, linearity reduces complexity, as this allow us to model the daily decision problem as a mixed integer linear program (MILP).

Assumption 4. *Balancing prices are linear in intraday market prices.*

During ramping, the storage owner cannot meet the commitments on the day-ahead and intraday market exactly. The difference is balanced by the transmission system operator (TSO) and

the storage owner is paid/billed by the TSO for the under/overproduction. Based on the TSO contracts, intentional under/overproduction is forbidden. To reflect this, the PHS must integrate the balancing prices into the decision problem. Assumption 4 is in line with the literature (e.g. Löhndorf & Minner (2010), Kim & Powell (2011), Moghaddam et al. (2013), Kuznetsova et al. (2015), Gönsch & Hassler (2016), Vahedipour-Dahraie et al. (2020), Finnah & Gönsch (2021)). The modeling approach with balancing prices allows to make the deviation from the commitment financially unattractive. A special case is to set the balancing prices equal to the intraday market prices. This reduces the needed notation but does hardly decrease the computational burden. Additionally, this contradicts the TSO contracts, as deviations may now be financially attractive and, thus, occur intentionally when following an optimal policy.

Assumption 5. *The PHS pays grid fees for the stored energy quantity.*

While the PHS stores energy the energy storage acts like a consumer and has to pay the obligatory grid fees. In Germany, there are several types of grid fees, based on the consumed energy (MWh) and the yearly peak demand (MW). For an overview of the different grid fees see Sterner & Stadler (2017, Ch. 15.4). Based on Assumption 5, we do not take the grid fees based on the peak demand into account. Since it is reasonable to assume that the PHS uses the pump's maximum power at least once a year, the peak demand is out of focus.

Assumption 6. *The PHS owner's price forecasts depend only on the last week's market prices.*

Assumption 6 restricts the storage owner's price forecasts, but this kind of price processes is well researched and accepted in the price forecasting literature (e.g., Ziel (2016), Marcjasz et al. (2019), Narajewski & Ziel (2020b)). Furthermore, price processes which follow Assumption 6 are considerably more sophisticated than price processes usually handled by dynamic programs in the energy storage literature. Their inclusion is challenging as dynamic programs have to extend the state variable by all information which is necessary for the considered price processes. Because of the curses of dimensionality, dynamic programs with large state spaces cannot be solved using standard discrete lookup tables in an appropriate time and can rarely be solved analytically. Therefore, dynamic programs have to use simplified price processes (one may argue that price processes are always approximations and, thus, no exact model/solution with regard to reality exists) and/or have to be solved approximately by approximate dynamic programming (ADP) algorithms. Both can lead to a poor solution quality, so it is crucial to find a reasonable trade-off between complexity of the price process and solution quality together with a suitable solution approach. Especially algorithms that approximate the value function can struggle with high dimensional forecasts (Powell, 2019). To keep the dynamic program simple, nearly all authors use price processes which only depend on a single electricity price (e.g., current, previous, or average of past prices) and state-independent features such as the day of the week (e.g., Gönsch & Hassler (2016), Zhou et al. (2016), Zéphyr & Anderson (2017), Salas & Powell (2018), Scott (2019)). To

capture the characteristics of the electricity price like price spikes, some authors add jump-diffusion or regime-switching models (e.g. [Jiang et al. \(2014\)](#), [Zhou et al. \(2016\)](#), [Salas & Powell \(2018\)](#), [Scott \(2019\)](#), [Durante et al. \(2020\)](#)). For a detailed overview of energy price forecasting see [Weron \(2014\)](#).

1.3. Contribution and outline

We make the following contributions to the existing literature: (1) We model the energy storage problem with high-dimensional, competitive, Markovian price processes for sequential auction markets as a dynamic program and solve it using a backwards approximate dynamic programming algorithm. (2) We compare this algorithm with the state-of-the-art approximate dual dynamic programming approach ([Löndorf et al. \(2013\)](#), [Löndorf & Shapiro \(2019\)](#), [Wozabal & Rameseder \(2020\)](#)) and an expectation model with a receding horizon. We demonstrate that dynamic programs have to handle a high dimensional state variable to capture the features of the underlying price processes to compete with a simple benchmark which does not suffer from the curses of dimensionality. (3) In our numerical study, we focus on a pumped hydro storage in both German auction markets. We show that the integrated trading in both markets outperforms the sequential trading. For this we evaluate the trading strategies with real world data.

The remainder of the paper is organized as follows. Section 2 is the literature review. Section 3 models the storage owner's decision problem as a dynamic program. In Section 4, we state a simplified version of the base model and explain how we apply the solutions of this simplified model. In other words, how we derive a policy for the base model from the simplified model's solution. Section 5 describes the approximate dynamic programming algorithm to solve the aforementioned simplified problem. Section 6 presents the numerical study. Section 7 discusses limitations with a focus on the markets considered before Section 8 concludes the paper. The appendix contains further numerical investigations and summarizes the notation.

2. Literature review

The existing literature on the management of energy storage systems is already discussed in multiple literature reviews (e.g., [Fathima & Palanisamy \(2015\)](#), [Rahman et al. \(2015\)](#), [Weitzel & Glock \(2018\)](#)). Therefore, we focus the following review on the key topics that distinguish our paper from the existing literature: modeling/solution approaches, price processes used in optimization, and integrated trading in multiple markets.

The most commonly used modeling approaches for the energy storage management which capture the stochastics and dynamics of the problem are multi-stage stochastic programming (MSSP) and dynamic programming. To solve MSSPs, the model is usually replaced with the sample average approximation (SAA). Because the computational burden of MSSP increases exponentially in the optimization horizon, it is only suitable for short-term management (a few days). For longer

horizons, the problem can be solved with a receding horizon. A main advantage is MSSPs' independence of the underlying price processes. By contrast, dynamic programs have to extend the state variable by all information which is necessary for the considered price processes. Therefore, competitive price processes are rarely used in dynamic programs.

[Löhndorf et al. \(2013\)](#) formulate the decision problem of an owner of a hydro storage with stochastic inflow on the German day-ahead and continuous intraday market as a dynamic program and use approximate dual dynamic programming (ADDP) to solve it. The energy price process is based on a few fundamentals like mean temperature, wind and solar power generation, and the gas price. The ADDP algorithm contains a forward pass and a backward pass to approximate the value function of the post-decision state with hyperplanes. During the forward pass, the ADDP algorithm explores new states. During the backward pass, the algorithm revisits the states from the forward pass. For each state, it iterates over all states in the state space that share the same endogenous parts, i.e. that only differ in the exogenous information part to derive new hyperplanes. Therefore, this approach is only suitable for discrete or discretized exogenous information. [Boomsma et al. \(2014\)](#) focus a profit maximizing power producer in the Nordic day-ahead market and a balancing market. The problem is formulated as a MSSP and takes price/volume bids into account. Based on the assumption that the scenarios in the SAA are the true distribution of the market prices, the solution is exact. In addition, the authors derive an upper bound for the gains from coordinated bidding in both markets. The price processes are modeled as a SARIMA process, but the noise is assumed to be independent and identically distributed. In fact, the forecast error of day-ahead market prices can be strongly correlated. [Ding et al. \(2015\)](#) use stochastic programming to (re-)optimize the management of a wind turbine combined with an energy storage on the Spanish day-ahead, intraday, and real-time market. Here, they employ four different timescales down to one minute to capture the dynamics of the real-time market. Instead of simultaneously optimizing all considered markets, they use separate stochastic programs in a receding horizon manner. Moreover, the authors assume that the next market's price is deterministic. This dramatically decreases complexity. [Crespo-Vazquez et al. \(2018\)](#) focus on self-schedule bidding of a wind power producer with an energy storage on two Spanish auction markets and an implicitly modeled balancing market with hourly products. The decision problem is formulated as a stochastic program with an optimization horizon of 24 hours. Again, the price component of the bids is not modeled. To generate market price scenarios, [Crespo-Vazquez et al. \(2018\)](#) cluster historic data. [Heredia et al. \(2018\)](#) use MSSP to optimize the management of a wind turbine combined with an energy storage. The considered power producer maximizes the daily profits on two Spanish auction markets and a reserve power market. Therefore, the Spanish energy markets are not modeled in full complexity. Again, the used scenarios are based on clustered historic data. Only [Brijs et al. \(2019\)](#) considers both German auction markets in the energy storage problem. In addition, the authors include the balancing market. But they completely ignore the stochastics and assume perfect foresight. This dramatically simplifies the problem and overestimates the value of coordinated trading. Electricity

prices are known to have hardly predictable price spikes, which can be exploited with perfect foresight. However, they demonstrate that under perfect foresight coordinated trading in multiple markets outperforms sequential trading, giving an upper bound on profits without perfect foresight. [Khatami et al. \(2020\)](#) optimize a compressed air energy storage in a day-ahead and real-time market and take start-up cost as well as ramps into account: The decision problem is modeled as an MILP and re-optimized in a receding horizon manner. [Vahedipour-Dahraie et al. \(2020\)](#) consider a virtual power plant with stochastic demand and supply in a day-ahead and a reserve market. Again, the problem is formulated as a stochastic program. The market prices follow a normal distribution. [Wozabal & Rameseder \(2020\)](#) model integrated trading in the Spanish auction markets as a dynamic program. As no energy storage is modeled, each day is optimized independently. For the market prices, the authors use an autoregressive model which depends on the last 24 market prices. The problem is solved with ADDP. For this, the stochastic processes are discretized by clustering. [Finnah \(2021\)](#) extends the model proposed in [Wozabal & Rameseder \(2020\)](#) and uses a more complex price process. Again, no energy storage is modeled and each day is optimized independently. Since the modeled dynamic program has a simple structure, the optimal policy is derived analytically via backward recursion.

To the best of our knowledge, no paper addresses the stochastic storage problem in both German auction markets. The key characteristic is that the German intraday auction market differs in terms of the time length of the traded products from the German day-ahead market. Since the German intraday market is the only one with 15-minute products, models for other markets have to be reworked to be applicable. While this does not have to be challenging (depending on the model), this may structurally change the numerical results. In addition, we are the first to extend the state variable of a dynamic program by all Markovian features of a high-dimensional price forecast (several hundred dimensions/ explanatory variables). We handle the full complexity of the state variable while solving the problem.

3. Dynamic program

In this section, we model the storage owner's decision problem for T days as a dynamic program. More precisely, the exogenous information, the state variables, and the action variables are defined in Section 3.1 to Section 3.3. The contribution function is described in Section 3.4, while the transition function is defined in Section 3.5. These are used in Section 3.6 for the value function. The notation used is summarized in Table D.5 in [Appendix D](#).

The PHS owner trades in the two German auction markets: the day-ahead and the intraday market. In the German day-ahead auction market, each day t until noon, the PHS owner submits a self-schedule bid (volume) $x_{t,h}^{DA}$ for each hour h of the following day ($t+1$). Forty minutes later, the market clearing prices (MCPs) $P_{t+1,h}^{DA}$ for each hour are published, and the volumes come into effect at the MCPs. Subsequently, until 15:00, the PHS owner makes an offer $x_{t,h,q}^{ID}$ for each quarter

hour q of each hour h of the following day ($t+1$) in the intraday auction market. Again, the MCPs $P_{t+1,h,q}^{ID}$ are published after market closure. This sequence of events is illustrated in Figure 2.

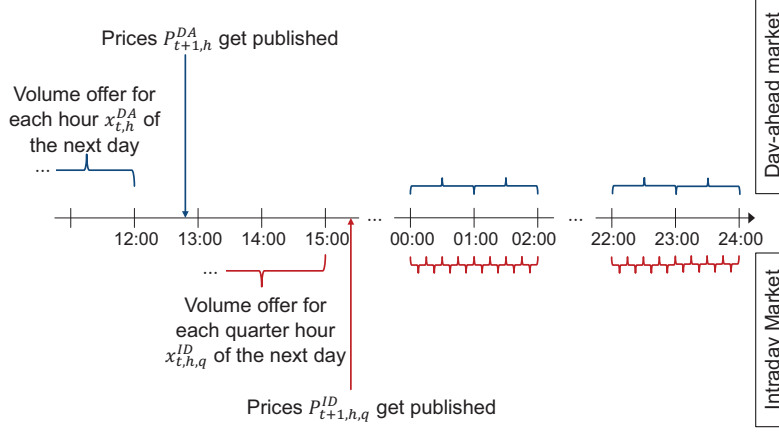


Figure 2: Sequence of events

In the following, we formulate the problem as a dynamic program. Since each day $t \in \{0, \dots, T-1\}$ consists of two decision stages (day-ahead market and intraday market) we model the problem with alternating stages. The day-ahead market stages are labeled with the superscript DA and the intraday market stages with ID .

3.1. Exogenous information

For the 24-dimensional vector of day-ahead market prices $P_{t+1}^{DA} = (P_{t+1,0}^{DA}, \dots, P_{t+1,23}^{DA})$, we use the autoregressive price forecast proposed by Ziel (2016), and similarly used by e.g. Uniejewski & Weron (2018); Marcjasz (2020); Jędrzejewski et al. (2021); Narajewski & Ziel (2021). It linearly depends on the previous 168 prices (7 days) and the day of the week, in line with Assumption 6.

$$P_{t+1,h}^{DA} = \beta_{0,h}^{DA} + \sum_{j=1}^7 \beta_{j,h}^{DA} DoW_{t+1}^j + \sum_{j=1}^7 \sum_{k=0}^{23} \beta_{j,k,h}^{DA} P_{t+1-j,k}^{DA} + \epsilon_{t+1,h}^{DA} \forall h \in \{0, \dots, 23\} \quad (1)$$

$$\epsilon_{t+1}^{DA} = (\epsilon_{t+1,0}^{DA}, \dots, \epsilon_{t+1,23}^{DA}) \sim N(0, \Sigma^{DA}) \quad (2)$$

with day of the week dummies

$$DoW_{t+1}^1 = \begin{cases} 1 & , t+1 \text{ is Monday} \\ 0 & , \text{otherwise} \end{cases}, \dots, DoW_{t+1}^7 = \begin{cases} 1 & , t+1 \text{ is Sunday} \\ 0 & , \text{otherwise} \end{cases} \quad (3)$$

and correlation matrix Σ^{DA} . The regression parameters are denoted $\beta_{(\cdot)}^{DA}$. Thus, the exogenous information which becomes known right after the day-ahead market decision stage is $W_{t+1}^{DA} = P_{t+1}^{DA}$.

We assume that the 96-dimensional intraday price vector $P_{t+1}^{ID} = (P_{t+1,0,0}^{ID}, \dots, P_{t+1,23,3}^{ID})$ depends linearly on the day of the week, the intraday, and day-ahead prices of the last week, including

the known day-ahead prices of the same day P_{t+1}^{DA} .

$$P_{t+1,h,q}^{ID} = \beta_{0,h,q}^{ID} + \sum_{j=1}^7 \beta_{j,h,q}^{ID} DoW_{t+1}^j + \sum_{j=0}^7 \sum_{k=0}^{23} \beta_{j,k,h,q}^{ID} P_{t+1-j,k}^{DA} + \sum_{j=1}^7 \sum_{k=0}^{23} \sum_{m=0}^3 \beta_{j,k,m,h,q}^{ID} P_{t-j,k,m}^{ID} + \epsilon_{t+1,h,q}^{ID} \forall h \in \{0, \dots, 23\}, q \in \{0, \dots, 3\} \quad (4)$$

$$\epsilon_t^{ID} = (\epsilon_{t,0,0}^{ID}, \dots, \epsilon_{t,23,3}^{ID}) \sim N(0, \Sigma^{ID}) \quad (5)$$

with correlation matrix Σ^{ID} and regression parameters $\beta_{(\cdot)}^{ID}$. Thus, the exogenous information which becomes known right after the intraday market decision stage is $W_{t+1}^{ID} = P_{t+1}^{ID}$.

3.2. State variable

The state variables include everything the storage owner's decision depends on and all necessary information to evaluate the transition and contribution functions. In particular, it must include all prices of the last week as they are necessary for the price forecasts. Since the day of the week $DoW_{(\cdot)}^{(\cdot)}$ can be derived based on the period/index t , it does not have to be part of the state even if it is needed for the price forecasts.

The state S_t^{DA} is the system right before the day-ahead market decision x_t^{DA} is made.

$$S_t^{DA} = (R_t^{start}, x_t^{start}, P_t^{history}) \quad (6)$$

with $P_t^{history} = (P_{t-6}^{DA}, \dots, P_t^{DA}, P_{t-6}^{ID}, \dots, P_t^{ID}) \in \mathbb{R}^{840}$.

Obviously, the storage owner's decisions depend on the current storage level. R_t^{start} is the storage level at the start of day t . x_t^{start} is the current power in- or outflow. This is needed for the considered start-up cost and ramping. Altogether, the state S_t^{DA} has 842 continuous variables.

The state S_t^{ID} is the state of the system right before the intraday market decision x_t^{ID} is made and after the day-ahead market prices P_{t+1}^{DA} become known.

$$S_t^{ID} = (S_t^{DA}, x_t^{DA}, P_{t+1}^{DA}) \quad (7)$$

The state S_t^{ID} extends the state S_t^{DA} by the commitments for the day-ahead market x_t^{DA} (which are not physically fulfilled) and the known day-ahead market prices P_{t+1}^{DA} which are needed for the forecast of the intraday market prices P_{t+1}^{ID} .

3.3. Action variable

The day-ahead market decision x_t^{DA} is the 24 commitments for this market's products.

$$x_t^{DA} = (x_{t,0}^{DA}, \dots, x_{t,23}^{DA}) \quad (8)$$

s.t.

$$x_{t,h}^{DA} \in [-x_{max}^{turbine}, x_{max}^{pump}] \forall h \in \{0, \dots, 23\} \quad (9)$$

A positive $x_{t,h}^{DA}$ corresponds to buying energy and a negative one to selling energy. To reduce speculation on the price split between day-ahead and intraday market, we introduce the no-short-selling constraint (9). According to this constraint, the commitments must be in the working range of the energy storage. x_{max}^{pump} is the maximum power of the pump and $x_{max}^{turbine}$ the maximum power of the turbine.

Likewise, the intraday market decision x_t^{ID} is the 96 commitments for this market's products.

$$x_t^{ID} = (x_{t,0,0}^{ID}, \dots, x_{t,23,3}^{ID}) \quad (10)$$

s.t.

$$x_{t,h}^{DA} + x_{t,h,q}^{ID} = x_{t,h,q}^{pump} - x_{t,h,q}^{turbine} \quad (11)$$

$$x_{t,h,q}^{comp} y_{t,h,q}^{comp} \leq x_{t,h,q}^{comp} \leq x_{max}^{comp} y_{t,h,q}^{comp} \quad \forall comp \in \{pump, turbine\} \quad (12)$$

$$y_{t,h,q}^{pump} + y_{t,h,q}^{turbine} \leq 1 \quad (13)$$

$$\Delta x_{t,h,q}^{comp,up} - \Delta x_{t,h,q}^{comp,down} = x_{t,h,q}^{comp} - x_{t,h,q-1}^{comp} \quad \forall comp \in \{pump, turbine\} \quad (14)$$

$$R_{t,h,q} = R_{t,h,q-1} + \eta^{pump} \left(\Delta t^{ID} x_{t,h,q}^{pump} - \frac{\Delta t^{ramp}}{2} \Delta x_{t,h,q}^{pump,up} + \frac{\Delta t^{ramp}}{2} \Delta x_{t,h,q}^{pump,down} \right) - \frac{1}{\eta^{turbine}} \left(\Delta t^{ID} x_{t,h,q}^{turbine} - \frac{\Delta t^{ramp}}{2} \Delta x_{t,h,q}^{turbine,up} + \frac{\Delta t^{ramp}}{2} \Delta x_{t,h,q}^{turbine,down} \right) \quad (15)$$

$$R_{t,h,q} \in [0, R_{max}] \quad (16)$$

$$y_{t,h,q}^{pump}, y_{t,h,q}^{turbine} \in \{0, 1\} \quad (17)$$

$$x_{t,h,q}^{pump}, x_{t,h,q}^{turbine}, \Delta x_{t,h,q}^{pump,up}, \Delta x_{t,h,q}^{pump,down}, \Delta x_{t,h,q}^{turbine,up}, \Delta x_{t,h,q}^{turbine,down} \geq 0 \quad (18)$$

All constraints are defined for all $q \in \{0, 1, 2, 3\}$ and $h \in \{0, \dots, 23\}$. Additionally, $(\cdot)_{t,h,4}$ is defined as $(\cdot)_{t,h+1,0}$ and $R_{t,0,-1} = R_t^{start}$ and $x_{t,0,-1}^{pump} = \max \{x_t^{start}, 0\}$ and $x_{t,0,-1}^{turbine} = \max \{-x_t^{start}, 0\}$. Compared to the day-ahead market, additional constraints and auxiliary variables like $\Delta x_{t,h,q}^{turbine,up}$ are needed because costs like grid fees, TSO balancing or start-up that depend on net commitments in both markets are considered in the profit related to the intraday market (see Section 3.4).

Constraint (11) splits the summed trading volume on both markets into a positive (store energy, $x_{t,h,q}^{pump}$) and a negative part (generate electricity, $x_{t,h,q}^{turbine}$). If a component (pump or turbine) is working ($y_{t,h,q}^{comp} = 1$), constraint (12) limits the power to its feasible range; otherwise, it forces the power to zero. According to the working mode constraint (13), the pump and the turbine cannot work at the same time. Constraint (14) captures how much the components have to ramp up ($\Delta x_{t,h,q}^{comp,up}$) or ramp down ($\Delta x_{t,h,q}^{comp,down}$) to reach the new power level $x_{t,h,q}^{comp}$. The storage level equation is constraint (15). The new storage level is equal to the previous plus the incoming minus the outgoing energy taking the ramps and pump and turbine efficiency ($\eta^{pump}, \eta^{turbine}$) into

account. According to Assumption 3, the energy storage ramps-up/down linearly with constant ramping time. We denote the ramping time as Δt^{ramp} . While ramping to reach the new power level, $\frac{\Delta t^{ramp}}{2} \Delta x_{t,h,q}^{(\cdot)}$ less/more energy is consumed/produced. Δt^{ID} is the temporal resolution of the intraday auction market. Constraint (16) keeps the storage level non-negative and within the storage capacity R_{max} . Constraint (17) defines the only binary variables, and constraint (18) ensures non-negativity for the remainder of variables except $x_{t,h,q}^{ID}$.

3.4. Contribution function

The contribution function C_t^{DA} is the storage owner's one-stage profit at the day-ahead market.

$$C_t^{DA} (S_t^{DA}, x_t^{DA}, W_{t+1}^{DA}) = - \sum_{h=0}^{23} \Delta t^{DA} P_{t+1,h} x_{t,h}^{DA} \quad (19)$$

The one-stage profit at the day-ahead market is the sum of the volumes times the market clearing prices times the duration of the day-ahead market products Δt^{DA} . Since a positive $x_{t,h}^{DA}$ corresponds to buying energy, the sum has a negative sign.

The storage owner's one-stage profit at the intraday market is denoted by C_t^{ID} .

$$\begin{aligned} C_t^{ID} (S_t^{ID}, x_t^{ID}, W_{t+1}^{ID}) &= \sum_{h=0}^{23} \sum_{q=0}^3 -\Delta t^{ID} P_{t+1,h,q}^{ID} x_{t,h,q}^{ID} - \Delta t^{ID} c^{gf} x_{t,h,q}^{pump} \\ &+ \frac{\Delta t^{ramp}}{2} \left(\left(\Delta x_{t,h,q}^{pump,up} + \Delta x_{t,h,q}^{turbine,down} \right) Q_{t+1,h,q}^{over} - \left(\Delta x_{t,h,q}^{pump,down} + \Delta x_{t,h,q}^{turbine,up} \right) Q_{t+1,h,q}^{under} \right) \\ &- c^{pump} z_{t,h,q}^{pump} - c^{turbine} z_{t,h,q}^{turbine} \end{aligned} \quad (20)$$

The first row is the profits based on the commitments on the intraday market times the duration of the intraday market products Δt^{ID} and the grid fees. While the energy storage stores energy, it acts as a consumer and has to pay the obligatory grid fees c^{gf} . The second row corresponds to the profits and costs due to the TSO's balancing. During ramping, the storage owner cannot meet the commitments exactly and the difference is balanced by the TSO. The deviation from the commitment is $\frac{\Delta t^{ramp}}{2} \cdot \Delta x_{t,h,q}^{(\cdot)}$. The storage owner is paid/billed by the TSO based on the balancing prices $Q_{t+1,h,q}^{over}$ and $Q_{t+1,h,q}^{under}$. According to Assumption 4, the balancing price is linear in the intraday market prices. While the pump ramps up or the turbine ramps down, the storage owner consumes less resp. produces more energy than committed. In this case, the storage owner is paid the balancing price for overproduction

$$Q_{t+1,h,q}^{over} = Q_{intercept}^{over} + Q_{slope}^{over} P_{t+1,h,q}^{ID} \quad (21)$$

with intercept $Q_{intercept}^{over}$ and slope Q_{slope}^{over} . While the pump ramps down or the turbine ramp up, the storage owner consumes more resp. produces less energy than committed. In this case, the

storage owner is billed the balancing price for underproduction

$$Q_{t+1,h,q}^{under} = Q_{intercept}^{under} + Q_{slope}^{under} P_{t+1,h,q}^{ID} \quad (22)$$

with intercept $Q_{intercept}^{under}$ and slope Q_{slope}^{under} .

The last row in (20) is the start-up cost of the pump (c^{pump}) and the turbine ($c^{turbine}$). They capture the increased maintenance cost (e.g. abrasion). The variables $z_{t,h,q}^{pump}$ and $z_{t,h,q}^{turbine}$ are binaries that capture if the pump or the turbine has to start-up. They are integrated by the following constraints.

$$y_{t,h,q}^{comp} - y_{t,h,q-1}^{comp} \leq z_{t,h,q}^{comp} \quad \forall comp \in \{pump, turbine\} \quad (23)$$

$$z_{t,h,q}^{pump}, z_{t,h,q}^{turbine} \geq 0 \quad (24)$$

These constraints are defined for all $q \in \{0, 1, 2, 3\}$ and $h \in \{0, \dots, 23\}$. Additionally, $(\cdot)_{t,h,4}$ is defined as $(\cdot)_{t,h+1,0}$ and $y_{t,0,-1}^{pump} = 1_{x_t^{start} > 0}$ and $y_{t,0,-1}^{turbine} = 1_{x_t^{start} < 0}$.

Constraint (23) forces the non-negative start-up variable $z_{t,h,q}^{comp}$ to be at least 1 if the pump/-turbine starts up. Because $z_{t,h,q}^{comp}$ exerts a negative influence on the intraday stage contribution function (20) and does not influence future profits, $z_{t,h,q}^{comp}$ is binary in every optimal solution without being forced to be binary. Constraint (24) restricts the start-up variables to be non-negative.

3.5. Transition functions

The two transition functions capture how the system transitions from one decision stage to the next. The transition function $S^{M,DA}$ simply unions the state, action and exogenous information of the day-ahead market stage.

$$S_t^{ID} = S^{M,DA} \left(S_t^{DA}, x_t^{DA}, W_{t+1}^{DA} \right) = \left(S_t^{DA}, x_t^{DA}, P_{t+1}^{DA} \right) \quad (25)$$

The transition function from the intraday market stage to the day-ahead market stage is denoted $S^{M,ID}$ and only slightly more complex:

$$\begin{aligned} S_{t+1}^{DA} &= S^{M,ID} \left(S_t^{ID}, x_t^{ID}, W_{t+1}^{ID} \right) = \left(R_{t+1}^{start}, x_{t+1}^{start}, P_{t+1}^{history} \right) \\ &= \left(R_{t,23,3}, x_{t,23}^{DA} + x_{t,23,3}^{ID}, P_{t-5}^{DA}, \dots, P_{t+1}^{DA}, P_{t-5}^{ID}, \dots, P_{t+1}^{ID} \right) \end{aligned} \quad (26)$$

where $R_{t,23,3}$ follows the storage level equation (15).

3.6. Value function

Obviously, the decisions x_t^{DA} and x_t^{ID} depend on the current states S_t^{DA} and S_t^{ID} . Thus, we define a policy π as a function that maps every state to a decision: $X_t^{DA,\pi} \left(S_t^{DA} \right)$ and $X_t^{ID,\pi} \left(S_t^{ID} \right)$.

Now, we have all necessary notation and can formally state the optimization problem faced by the storage owner who maximizes the profit over an optimization horizon of length T .

$$\max_{\pi} \mathbb{E} \left[\sum_{t=0}^{T-1} C_t^{DA} \left(S_t^{DA}, X_t^{DA,\pi}(S_t^{DA}), W_{t+1}^{DA} \right) + C_t^{ID} \left(S_t^{ID}, X_t^{ID,\pi}(S_t^{ID}), W_{t+1}^{ID} \right) | S_0^{DA} \right] \quad (27)$$

s.t. (9), (11) to (18), (23) to (26) and initial state S_0^{DA} . The objective is the expectation of the summed contribution over all periods.

An alternative formulation of the optimization problem are the recursive Bellman equations

$$V_t^{DA} \left(S_t^{DA} \right) = \max_{x_t^{DA}} \mathbb{E} \left[C_t^{DA} \left(S_t^{DA}, x_t^{DA}, W_{t+1}^{DA} \right) + V_t^{ID} \left(S_t^{ID} \right) | S_t^{DA} \right] \quad (28)$$

$$V_t^{ID} \left(S_t^{ID} \right) = \max_{x_t^{ID}} \mathbb{E} \left[C_t^{ID} \left(S_t^{ID}, x_t^{ID}, W_{t+1}^{ID} \right) + V_{t+1}^{DA} \left(S_{t+1}^{DA} \right) | S_t^{ID} \right] \quad (29)$$

with the boundary condition

$$V_T^{DA} \left(S_T^{DA} \right) = 0 \quad (30)$$

Obviously, the same constraints have to be respected. One way to find a policy is to compute the value functions V_t^{DA} and V_t^{ID} and derive the decisions based on (28) and (29) with respect to their corresponding constraints. For other ways to derive a policy see [Powell \(2019\)](#).

4. Policy

Since we cannot solve (27) s.t. (9), (11) to (18), (23) to (26) analytically, we define in this section the suboptimal policy functions $X_t^{DA,\pi}(\cdot)$ and $X_t^{ID,\pi}(\cdot)$ to ease the numerical solution. The policy function $X_t^{DA,\pi}(\cdot)$ maps each state S_t^{DA} to a decision $X_t^{DA,\pi}(S_t^{DA})$ for the day-ahead market stage. The policy function $X_t^{ID,\pi}(\cdot)$ does this for the intraday market stage. To define the policy functions, we introduce the following policy model. For this, we replace the alternating decision stages by an expectation model. This is illustrated in Figure 3.

The upper half of Figure 3 illustrates the base model described in Section 3. It visualizes the alternating decision stages for one period t . Squares symbolize decision nodes, whereas circles denote stochastic nodes where new information arrives. The period starts with state S_t^{DA} in the day-ahead market stage, where the day-ahead market decision x_t^{DA} has to be determined. Then, the exogenous information W_{t+1}^{DA} (i.e. the day-ahead market prices) becomes known. This brings the system to state S_t^{ID} in the intraday market stage, where now the intraday market decision x_t^{ID} has to be determined. Then, the exogenous information W_{t+1}^{ID} (intraday market prices) arrives. This completes period t and transitions the system to the first state of period $t+1$, S_{t+1}^{DA} .

The lower half of Figure 3 shows on an intuitive level the policy model for the decision problem. The policy model is a dynamic program where at each decision stage t , the decisions x_t^{DA} and

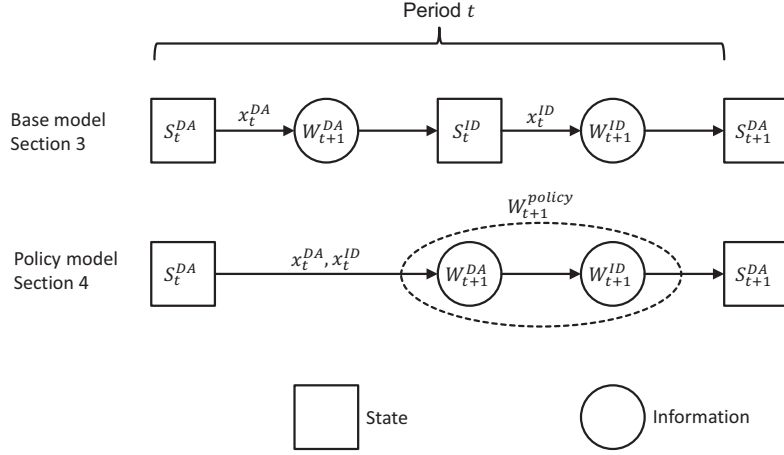


Figure 3: Policy design

x_t^{ID} are made simultaneously before the exogenous information W_{t+1}^{DA} and W_{t+1}^{ID} becomes known. Different from the base model, the intraday market decision x_t^{ID} is already determined in state S_t^{DA} . In the following, we state the policy model and how we use it to derive the policy functions $X_t^{DA,\pi}(\cdot)$ and $X_t^{ID,\pi}(\cdot)$. The notation is summarized in Table D.6 in Appendix D.

The state variable of the policy model is equal to the state of the day-ahead market decision stage (6). The action variable combines the actions of the day-ahead and intraday decision stages.

$$x_t^{policy} = (x_t^{DA}, x_t^{ID}) \quad (31)$$

The exogenous information is the joint exogenous information of the day-ahead and intraday market stage, where the latter one is conditioned on the first one.

$$W_{t+1}^{policy} = (W_{t+1}^{DA}, W_{t+1}^{ID} | W_{t+1}^{DA}) \quad (32)$$

Since we will remove the stochastics in the contribution function of the policy model, W_{t+1}^{policy} is only used for the state's transition.

The transition function $S^{M,policy}$ is the function composition

$$S_{t+1}^{DA} = S^{M,policy}(S_t^{DA}, x_t^{policy}, W_{t+1}^{policy}) = S^{M,ID}(S^{M,DA}(S_t^{DA}, x_t^{DA}, W_{t+1}^{DA}), x_t^{ID}, W_{t+1}^{ID}) \quad (33)$$

We denote the policy model's one-stage contribution by C_t^{policy} . Instead of modeling a stochastic contribution based on the exogenous information W_{t+1}^{policy} and later derive the expectation, we model C_t^{policy} as the expectation itself. For this, we need the expectation of the day-ahead market prices, intraday market prices, and the balancing prices. We denote these expectations as $\bar{P}_{t,t+1,h}^{DA}$, $\bar{P}_{t,t+1,h,q}^{ID}$, $\bar{Q}_{t,t+1,h,q}^{over}$, and $\bar{Q}_{t,t+1,h,q}^{under}$. The subscript t denotes that these expectations use only information that is available at day t . The subscript $t+1$ indicates that these are the expectations of exogenous information which become known right before day $t+1$. Since the exogenous in-

formation is modeled linearly, the expectations' calculation is straightforward. Finally, the policy model's one-stage contribution function is

$$\begin{aligned}
C_t^{policy} \left(S_t^{DA}, x_t^{policy} \right) = & \sum_{h=0}^{23} -\Delta t^{DA} \bar{P}_{t,t+1,h}^{DA} x_{t,h}^{DA} + \sum_{q=0}^3 -\Delta t^{ID} \bar{P}_{t,t+1,h,q}^{ID} x_{t,h,q}^{ID} - \Delta t^{ID} c^{gf} x_{t,h,q}^{pump} \\
& + \frac{\Delta t^{ramp}}{2} \left(\left(\Delta x_{t,h,q}^{pump,up} + \Delta x_{t,h,q}^{turbine,down} \right) \bar{Q}_{t,t+1,h,q}^{over} - \left(\Delta x_{t,h,q}^{pump,down} + \Delta x_{t,h,q}^{turbine,up} \right) \bar{Q}_{t,t+1,h,q}^{under} \right) \\
& - c^{pump} z_{t,h,q}^{pump} - c^{turbine} z_{t,h,q}^{turbine}
\end{aligned} \tag{34}$$

This results in the value function

$$V_t^{policy} \left(S_t^{DA} \right) = \max_{x_t^{policy}} \left\{ C_t^{policy} \left(S_t^{DA}, x_t^{policy} \right) + \mathbb{E} \left[V_{t+1}^{policy} \left(S_{t+1}^{DA} \right) \middle| S_t^{DA} \right] \right\} \tag{35}$$

s.t. $V_T^{policy} \left(S_T^{DA} \right) = 0$, (9), (11) to (18), (23) to (24), (33).

In Section 5, we derive approximations \bar{V}_t^{policy} of the value function V_t^{policy} to yield the policy $X_t^{DA,\pi} \left(S_t^{DA} \right) = \arg \max_{x_t^{DA}} \max_{x_t^{ID}} \left\{ C_t^{policy} \left(S_t^{DA}, \left(x_t^{DA}, x_t^{ID} \right) \right) + \mathbb{E} \left[\bar{V}_{t+1}^{policy} \left(S_{t+1}^{DA} \right) \middle| S_t^{DA} \right] \right\}$ s.t. (9),

(11) to (18), (23) to (24), (33) and

$X_t^{ID,\pi} \left(S_t^{ID} \right) = \arg \max_{x_t^{ID}} \mathbb{E} \left[C_t^{ID} \left(S_t^{ID}, x_t^{ID}, W_{t+1}^{ID} \right) + \bar{V}_{t+1}^{policy} \left(S_{t+1}^{DA} \right) \middle| S_t^{ID} \right]$ s.t. (11) to (18), (23) to (24), (26). Accordingly, we adjust the intraday market decisions based on the new day-ahead market prices. More precisely, the storage owner does not actually use the intraday market decision of the policy model, but re-optimizes the decision according to the policy function $X_t^{ID,\pi} \left(S_t^{ID} \right)$ taking into account the outcome of the day-ahead market.

While the expectation of the linear $C_t^{ID} \left(S_t^{ID}, x_t^{ID}, W_{t+1}^{ID} \right)$ is again straightforward, we explain in the following section how we compute the expectation of the value function. In Appendix B, we demonstrate that the policy defined by $X_t^{DA,\pi} \left(S_t^{DA} \right)$ and $X_t^{ID,\pi} \left(S_t^{ID} \right)$ is as good as the solution of the SAA of the stochastic counterpart on a daily base.

5. Approximate dynamic programming

As the formulated policy model is continuous and high-dimensional, we cannot solve it optimally. To tackle this problem, we propose a BADP algorithm. This algorithm solves a problem with discretized state space in which R_t^{start} and x_t^{start} are discretized and the price history $P_t^{history}$ is replaced by a set of sample paths. The notation is summarized in Table D.7 in Appendix D.

5.1. BADP

It is common practice to discretize the state space, for example by discretizing the endogenous part of the state and replacing the exogenous part with a scenario lattice. The corresponding sample paths are sampled with the stochastic processes based on the initial state. To derive the

expectation of the next value function, transition probabilities $P(P_{t+1}^{history} | P_t^{history})$ are needed. One method for this is to choose the scenario lattice as time depending cluster-centroids and estimate the transition probabilities by counting the transitions from one cluster-centroid to another (e.g. Löhndorf et al. (2013), Löhndorf & Shapiro (2019), Wozabal & Rameseder (2020)). Scenario lattices are also named vectorial quantization tree (e.g. Bonnans et al. (2011)) or recombining tree (e.g. Wozabal & Rameseder (2020)). Hannah & Dunson (2011) used such cluster centroids for a problem with 24 exogenous dimensions and Wozabal & Rameseder (2020) for a problem with 48 exogenous dimensions in the state variable.

In our numerical study in Section 6, we show that this approach has a poor solution quality for the problem considered here. Therefore, we propose in the following an algorithm that does not need transition probabilities nor cluster-centroids. Instead, *BADP* (Algorithm 1) estimates the expected value of being in the next state by a linear combination of the sample paths. This approach is similar to an approach proposed in Broadie et al. (2000) and Glasserman (2003, Ch. 8.5) for pricing options. The authors estimate weights that mimic transition probabilities based on a constrained optimization problem. In contrast, the algorithm presented here does allow negative weights that do not have to sum up to one. In Section 5.2, we enhance the distance measure used in the constrained optimization problem computing the weights. Furthermore, we introduce box constraints which depend on the distance of the sample paths.

Algorithm 1 BADP

```

1: initialize  $P_{0,n}^{history} = P_0^{history}, n \in \{1, \dots, N\}$ 
2: for  $n = 1$  to  $N$  do
3:   for  $t = 0$  to  $T - 1$  do
4:     compute expected price history  $\bar{P}_{t,t+1,n}^{history} := \mathbb{E} [P_{t+1}^{history} | P_t^{history} = P_{t,n}^{history}]$ 
5:     sample  $P_{t+1,n}^{history} | P_t^{history} = P_{t,n}^{history}$ 
6:   end for
7: end for
8: initialize lookup table  $\bar{V}_T^{policy} = 0$ 
9: for  $t = T - 1$  to  $0$  do
10:   for  $n = 1$  to  $N$  do
11:      $(\bar{\rho}_1, \dots, \bar{\rho}_N) := \arg \min_{\rho_1, \dots, \rho_N} \left\| \bar{P}_{t,t+1,n}^{history} - \sum_{k=1}^N \rho_k P_{t+1,k}^{history} \right\|_2$ 
12:     for  $R_t^{start} = 0$  to  $R_{max}$  do
13:       for  $x_t^{start} = -x_{max}^{turbine}$  to  $x_{max}^{pump}$  do
14:          $\bar{V}_t^{policy}(S_t^{DA}) := \max_{x_t^{policy}} \left\{ C_t^{policy}(S_t^{DA}, x_t^{policy}) + \sum_{k=1}^N \bar{\rho}_k \bar{V}_{t+1}^{policy}(S_{t+1}^{DA} | P_{t+1,k}^{history}) \right\}$ 
15:         s.t. (9), (11) to (18), (23) to (24), (33).
16:       end for
17:     end for
18:   end for

```

BADP (Algorithm 1) mimics the classical backwards recursion for discrete dynamic programs. The main purpose of *BADP* is to fill a look-up table which approximates the value of being in one of the considered discrete states (Step 14). Steps 1 to Step 7 sample N scenarios $P_{t,n}^{history}$, $n \in \{1, \dots, N\}$ for the price history $P_t^{history}$ and pre-process data for the further algorithm. In particular, Step 1 initializes the sampled price history in the starting period $P_{0,n}^{history}$ as the price history $P_0^{history} \in \mathbb{R}^{840}$ of the initial state S_0^{DA} . Step 4 computes the expectation of the price history of the next period $\bar{P}_{t,t+1,n}^{history}$, conditioned on the current price history sample $P_{t,n}^{history}$. For this, the transition function (33) and the price processes in Section 3.1 are used. The expected price history is needed later in the algorithm (Step 11). Since our price processes are linear, the expectation is straightforward. The subscript t indicates that only information available at iteration t is used to derive the expectation for stage $t + 1$ (indicated by subscript $t + 1$). Step 5 samples a new price history for the next period. The price history sample $P_{t+1,n}^{history}$ in Step 5 is determined by the transition function (33) and a sampled day-ahead and intraday market price, based on the price processes stated in Section 3.1 and the current sample $P_{t,n}^{history}$. Note, $P_{t+1,n}^{history}$ is a sample of a multivariate normal distribution with mean $\bar{P}_{t,t+1,n}^{history} \in \mathbb{R}^{840}$.

After this, Step 8 initializes the lookup table according to the boundary condition. Step 9 goes backwards in time, while Step 10 iterates over all sample paths. Subsequently, in Step 11, we regress the expected price history computed in Step 4 on the price histories in the set of sample paths sampled in Step 5. We do not enforce properties such as non-negativity for the regression parameters $\bar{\rho}_k$. These parameters are used in Step 14 to estimate/approximate the expectation of the next period's value function approximation. Note, for $N > 840$, no unique solution in Step 11 exists. In our numerical study we choose $N = 50$.

We now iterate over all discretized storage levels (Step 12) and trading volumes (Step 13) to fill the lookup table for the discrete states $S_t^{DA} = (R_t^{start}, x_t^{start}, P_{t,n}^{history})$ in period t . The state consists of the discretized storage level, the discretized trading volume, and the price history $P_{t,n}^{history}$, sampled in Step 5. The states on the right hand side in Step 14 adhere to the transition function $S_{t+1}^{DA} = S^{M,policy}(S_t^{DA}, x_t^{policy}, W_{t+1}^{policy})$ and therefore depend on the decision x_t^{policy} and the discretized state. The exogenous part of the state is one of the sampled price histories.

The expectation of the next period's value function is estimated by a weighted sum based on the regression parameters $\bar{\rho}_k$. These parameters mimic (on an intuitive level) the transition probabilities, but as they can be negative, the parameters are able to extrapolate exogenous states that are not evaluated. This is illustrated in Figure 4 (explained later) for out-of-sample data.

If a decision in Step 14 leads to a storage level R_{t+1}^{start} or trading volume x_{t+1}^{start} that is not part of the lookup table, we linearly interpolate the value function in these. Therefore, the look-up table \bar{V}_{t+1}^{policy} can be interpreted as a piece-wise linear function. Thus, Step 14 can be solved with standard MILP-solvers. This linear interpolation is standard in the literature (e.g., Löhndorf et al. (2013), Wozabal & Rameseder (2020)).

After fixing the value function approximation \bar{V}_t^{policy} , the policy in Section 4 can be applied.

The expectation of the next stage's value function necessary to calculate the policy functions $X_t^{DA,\pi}(\cdot)$ and $X_t^{ID,\pi}(\cdot)$ is approximated as in Algorithm 1. A pseudo code for the execution logic can be found in Appendix C. In contrast to other ADP algorithms, it can be applied without re-optimization. However, this may increase solution quality.

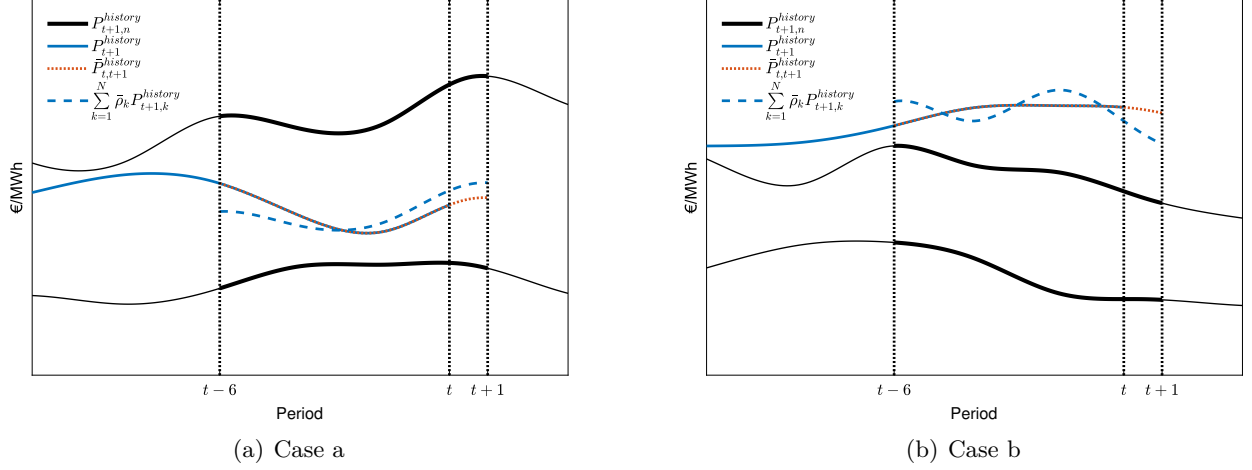


Figure 4: Illustration of the value function approximation technique

The approximation of the value function by the linear combination based on the parameters $\bar{\rho}_k$ is illustrated in Figure 4. Cases a and b each use two sample paths to approximate an out-of-sample realized price history $P_t^{history}$. More precisely, the figure illustrates how we approximate the expectation of the next period's value function approximation for out-of-sample price histories $P_{t+1}^{history}$. Each plot shows two price sample paths (thin black lines). At time t , the two corresponding price histories are $P_{t+1,n}^{history} \in \mathbb{R}^{840}$ and comprise of the last 7 days' prices. Thus, these histories are a part of the two paths and depicted by bold lines overlaying parts of the thin black lines. Now, the blue line is the trajectory of realized prices until period t . Based on this, the expected price history $\bar{P}_{t,t+1}^{history}$ is computed (orange dashed line). It comprises of the realized prices of the last 6 days and the expectations for the next day (day $t+1$).

Now, this expected price history is regressed on the price history samples $P_{t+1,n}^{history}$. The fitted regression $\sum_{k=1}^N \bar{\rho}_k P_{t+1,k}^{history}$ is drawn with a blue dashed line. In Case a, the expected price history can be approximated well by a convex combination of the samples. The corresponding regression parameters can be easily interpreted as transition probabilities. In Case b, the expected price history lies outside of the sampled price histories. Therefore, one needs negative regression parameters and/or they do not sum to one.

5.2. BADP-weighted

We modify the *BADP* algorithm by replacing the regression in Step 11 with a box-constrained, weighted regression. With this, we take into account that the influence of some prices on the price

forecast is more important than the effect of others. In addition, due to the box-constraints, the optimal regression parameters may be unique even for $N > 840$.

$$(\bar{\rho}_1, \dots, \bar{\rho}_N) := \arg \min_{\rho_1, \dots, \rho_N} \left\| \bar{P}_{t,t+1,n}^{history} - \sum_{k=1}^N \rho_k P_{t+1,k}^{history} \right\|_{w_t} \quad (36)$$

s.t.

$$-K_{t,k}^d \leq \rho_k \leq K_{t,k}^d \quad \forall k \in \{1, \dots, N\} \quad (37)$$

For the upper and lower bound we select $K_{t,k}^d$ as

$$K_{t,k}^d := \exp \left(\frac{1}{d^2} \left\| \bar{P}_{t,t+1,n}^{history} - P_{t+1,k}^{history} \right\|_{w_t}^2 \right) \quad (38)$$

which is the Gaussian kernel with the weighted Euclidean norm $\|\cdot\|_{w_t}$. We set the bandwidth d such that

$$\sum_{k=1}^N K_{t,k}^d = \ln(N) \quad (39)$$

as this works well in the preliminary test.

We select the vector of weights $w_t = (w_{t,1}, \dots, w_{t,j}, \dots, w_{t,840})$ for the weighted Euclidean norm $\|\cdot\|_{w_t}$ based on the influence of each element $P_{t+1,k}^{history,j}$ on future prices. In particular, we define the influence as the cumulative absolute change of all future prices if one element of the price history is ceteris paribus changed by 1 €/MWh. Algorithm 2 performs the necessary calculations.

Algorithm 2 Computation of weights

```

1: for  $t = 0$  to  $T - 1$  do
2:   for  $j = 1$  to 840 do
3:      $P_t^{history} := e_j, \bar{P}_{t,t}^{history} := P_t^{history}$ 
4:      $w_{t,j} := 0$ 
5:     for  $\tau = t$  to  $T - 1$  do
6:        $w_{t,j} := w_{t,j} + \left\| \mathbb{E} \left[ W_{\tau+1}^{policy} | \bar{P}_{t,\tau}^{history} \right] \right\|_1$ 
7:        $\bar{P}_{t,\tau+1}^{history} := \mathbb{E} \left[ P_{\tau+1}^{history} | \bar{P}_{t,\tau}^{history} \right]$ 
8:     end for
9:   end for
10: end for

```

Algorithm 2 works as follows. For each period (Step 1) and over the length of the price history (Step 2), we select the price history as the j -th unit vector (Step 3). Additionally, we initialize the expected price history $\bar{P}_{t,t}^{history}$ which is needed in the first iteration of Step 6 and Step 7. We advance forwards (Step 5) in time and update the price history (Step 7), while we add up the absolute values of the expected outcomes (Step 6). We do not normalize the weights $w_{t,j}$. To exclude the effects of the day of the week, we set their parameters to zero in the price processes.

With the weighted regression, we take into account that some prices are substantially more important for the future prices and should be better approximated than others. For instance, the last day-ahead price of the previous day contains the newest information about the cumulative renewable energy production which influences the next day's electricity prices (Ziel, 2016). With the box constraints, we force the regression to approximate the expected price history with the closest scenarios in the set of sample paths, as the value function should be close as well.

6. Numerical study

In Section 6.1, we state the parameters for our numerical study and specify the implemented approaches. The parameter values are summarized in Table D.9 in Appendix D. Based on this, Section 6.2 compares the different implemented algorithms. Additionally, Section 6.3 indicates that the decisions of the best approach are reasonable. For this, we investigate the storage level over time. Section 6.4 and Section 6.5 show the gains from using a competitive price forecast and from the integrated trading in both markets.

6.1. Data for the numerical study

We choose a pumped hydro storage with a storage capacity of $R_{max} = 100$ MWh and pump and turbine working power range of $[x_{min}^{pump}, x_{max}^{pump}] = [x_{min}^{turbine}, x_{max}^{turbine}] = [5 \text{ MW}, 10 \text{ MW}]$ with symmetrical efficiency $\eta^{pump} = \eta^{turbine} = 0.9$. The start-up costs are $c^{pump} = c^{turbine} = 15 \text{ €}$. We assume that both components need $\Delta t^{ramp} = 2 \text{ min}$ to ramp up or down. For the balancing by the TSO, we set the parameters $Q_{intercept}^{over} = -3 \text{ €/MWh}$, $Q_{slope}^{over} = \frac{1}{1.2}$, $Q_{intercept}^{under} = 3 \text{ €/MWh}$, $Q_{slope}^{under} = 1.2$. Thus, a deviation from the commitments is financially unattractive. While the storage owner buys energy, the energy storage acts like a consumer and has to pay the obligatory grid fees (up to 30 €/MWh in Germany). However, because energy storage systems stabilize the grid, they only pay reduced fees. This reduction is a case by case decision Sterner & Stadler (2017, Ch. 15.4). We choose rather low fees of $c^g = 5 \text{ €/MWh}$. Finally, the optimization horizon is $T = 30$ days. Since the storage can be fully charged/discharged in half a day, the start-of-horizon and end-of-horizon effects are quite small. In practice, a storage owner would re-optimize in a rolling horizon fashion before the end-of-horizon. Since all approaches specified in this section suffer from the same start-of-horizon and end-of-horizon effects, we do not control for them and the results are still comparable.

While solving and evaluating the energy trading problem, multiple sources of stochasticity exist. This is illustrated in Figure 5. The arrows indicate the dependence of the three main steps (estimating price processes, solving trading problem, and evaluation) in the numerical study and how they are influenced by the stochastics. Because of the random division of the training data into subsets for the cross validation, the parameters of the estimated price processes are stochastic (explained in more detail during this section). Based on these price processes, we sample the sample

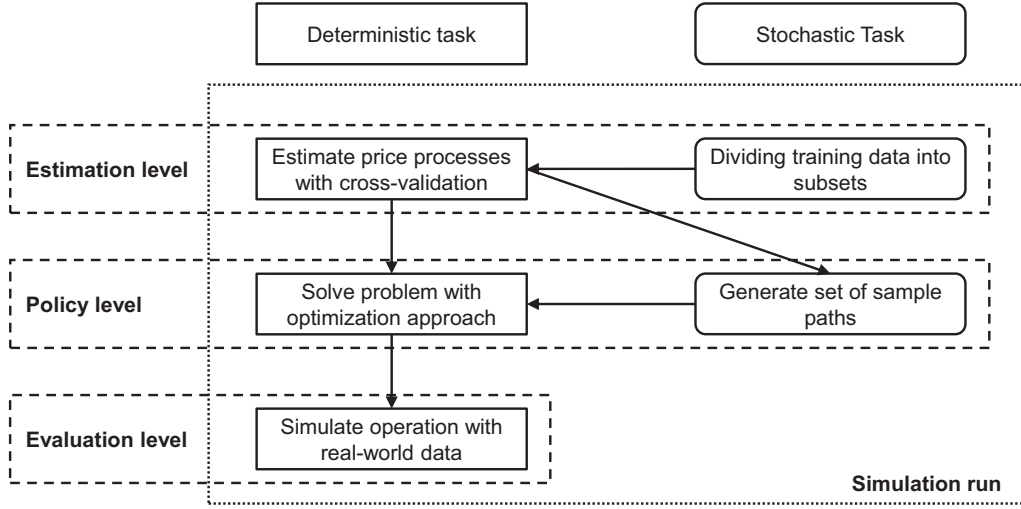


Figure 5: Setup of the numerical study

paths for our optimization approaches (e.g., *BADP*). Additionally, the price processes are used in the optimization (explained in Section 5.1). Therefore, the policy functions $X_t^{DA,\pi}(\cdot)$ and $X_t^{ID,\pi}(\cdot)$ are influenced by this stochasticity. Thus, the actual decisions and the profit on the evaluation level are stochastic, even when we evaluate with real-world data and no further stochastic influence on the evaluation level exists. To compute confidence bands that capture the stochastics present in evaluation, we run the simulation (estimating price processes, solving trading problem, and simulation of operation) 20 times.

Because the German energy markets' prices behave differently over the course of a year (e.g. due to increased solar radiation in summer), we consider four different settings denoted as *Winter*, *Spring*, *Summer* and *Autumn*. The data is summarized in Table 1.

Table 1: Data of the stochastic processes

Season	Training data	Test data
<i>Winter</i>	16 Jan 2019 to 15 Jan 2020	16 Jan 2020 to 14 Feb 2020
<i>Spring</i>	02 May 2019 to 30 Apr 2020	01 May 2020 to 30 May 2020
<i>Summer</i>	01 Aug 2018 to 31 Jul 2019	01 Aug 2019 to 30 Aug. 2019
<i>Autumn</i>	17 Oct 2018 to 15 Oct 2019	16 Oct 2019 to 14 Nov 2019

For each season, the price processes are estimated based on data from [EPEX SPOT SE \(2020\)](https://www.epeexspot.com/) with 365 days in the training set. For this we use the least absolute shrinkage and selection operator (lasso) with ten-fold cross-validation. For the cross-validation, the training data is randomly partitioned into ten subsets. Note that on 10/01/2018, there was a major change in the German auction markets when Austria left the German system. However, to estimate *Summer* with 365 days in the training set, we have to include data before this major change.

Table 2 shows the out-of-sample mean absolute error (MAE) and the root mean squared error

(RMSE) of our price processes. To compute the errors, we forecast the next day's day-ahead and intraday market prices with the estimated price processes based on the real-world data known at this time. Afterwards, we measure the error between the forecasted prices and the next real-world data. This is done for all days in the test data set.

The parameters of the estimated price processes ($\beta_{(\cdot)}^{DA}$ and $\beta_{(\cdot)}^{ID}$) contain some kind of randomness. More precisely, the splitting of the training data set into subsets for the cross-validation is random (illustrated in Figure 5). To tackle this randomness, we estimate the price processes 20 times and report the average MAE and RMSE. For each estimation with ten-fold cross-validation, the training data is newly divided into ten subsets. To ease the interpretation of the errors, we report the average market prices of the test sets.

Therefore, the data in Table 2 captures the randomness on estimation level. In practice, one would average the 20 estimated price processes. This is not done in our numerical study for two reasons. First, the parameters of the price processes hardly differ for the cross-validation runs. Second, this allows us to investigate the influence of the stochastics on expected profit on the estimation level.

Table 2: Forecast errors of the stochastic processes and average prices

Season	Day-ahead market			Intraday market		
	MAE	RMSE	Average prices	MAE	RMSE	Average prices
<i>Winter</i>	7.03	10.29	31.26	4.90	7.53	31.06
<i>Spring</i>	7.26	12.13	18.13	4.27	6.08	18.31
<i>Summer</i>	6.03	9.92	36.95	3.22	4.63	37.04
<i>Autumn</i>	5.07	7.16	39.57	3.45	4.73	38.84

Table 2 shows that the day-ahead market prices are harder to forecast than the intraday market prices. Both MAE and RMSE are much higher for the day-ahead market. Especially hard to predict are the day-ahead-market prices in *Spring*, as it shows the highest error measurements while the mean price is rather low.

Following the standard approach for the evaluation of ADP algorithms, we conduct numerical comparisons with benchmark approaches. Therefore, we implement several approaches using Matlab version R2020a and run simulations on a standard PC with a 3.6 GHz AMD Ryzen 5 and 16 GB of RAM. We evaluate every policy with the data in the test data set. In particular, we evaluate the following BADP variants and benchmark approaches:

- **BADP:** This is the BADP algorithm described in Algorithm 1. We discretize the storage levels R_t^{start} in 11 equidistant points. Preliminary tests indicate that the last trade x_t^{start} exerts less influence on the decision. Therefore, we choose a rougher discretization for it: $x_t^{start} \in \{-x_{max}^{turbine}, 0, x_{max}^{pump}\}$. We set the number of sample paths as $N = 50$. This results in $11 \cdot 3 \cdot 50 = 1,650$ discrete states. The MILP in Step 14 is solved with the Matlab solver `intlinprog`. After fixing the value functions, the induced policy is applied without re-optimization.

- **BADP- w :** For this BADP approach we replace Step 11 in Algorithm 1 with the quadratic problem (36) and (37). The coefficients for the lower and upper bounds are described by (38) and (39). The weights are computed with Algorithm 2. We select the same discretization for the states and the same number of samples as for the unweighted BADP. After fixing the value function approximations, the induced policy is applied without re-optimization.
- **BADP- w (re-optimized):** This approach is largely the same as BADP- w , but the value function approximations are re-optimized once at $t = 15$. In more detail, the value function approximations are derived first with BADP- w . After applying the induced policy for $t \in \{0, \dots, 14\}$ with fixed value function-approximations, these are re-optimized. For this, we use BADP- w with a new set of sample paths which is sampled based on the current state in $t = 15$.
- **ADDP:** We adapt the approximate dual dynamic programming algorithm proposed by Löhndorf et al. (2013). This algorithm was recently used by Wozabal & Rameseder (2020) for a problem with 48 exogenous dimensions in the state variable. This algorithm is a version of stochastic dual dynamic programming (SDDP) but allows the stochastic processes to be Markovian and is not limited to stagewise independent processes (Wozabal & Rameseder (2020)). Originally, this algorithm iterates over K iterations and approximates the value functions of the post-decision states with hyperplanes. Since the value function approximations are piecewise linear and the stochastic processes are linear and the transition function is linear in these, we can derive the expectation in (35) analytically. So, there is no computational advantage to use the post-decision state formulation. Therefore, we approximate the value functions of the pre-decision states. Like BADP, the ADDP approach replaces the exogenous part of the state $P_t^{history}$ with a scenario lattice. But in contrast to BADP and in line with Löhndorf et al. (2013), the scenario lattice is described by N cluster centroids (we use $N = 50$). For generating the scenario lattice, we sample 1,000 price histories for each period t and apply k -means clustering with the city block norm. The necessary transition probabilities are approximated by counting the transitions between the clusters. Each iteration of the ADDP algorithm contains a forwards pass and a backwards pass. During the forwards pass, ADDP generates new endogenous states (R_t^{start} and x_t^{start}). After this, during the backwards pass, we iterate over the endogenous states and all price histories in the scenario lattice to add a new hyperplane for each combination.

Preliminary tests indicate that $K = 10$ iterations are a reasonable trade-off between solution quality and machine time. More precisely, additional iterations hardly improve the profit. To speed the algorithm up, when we add a new hyperplane, we remove all hyperplanes which are dominated by the new one. We do not implement the rejection of hyperplanes with negligible improvement. This could speed the ADDP approach up, but also worsen the results slightly.

After fixing the value function approximation \bar{V}^{policy} , the policy in Section 4 can be applied. If the transition leads to exogenous states which are not part of the scenario lattice, the tran-

sition probabilities have to be updated. This is in contrast to the other approaches which are nevertheless applicable. To speed this up, we follow [Wozabal & Rameseder \(2020\)](#) and base the decisions on the transition probabilities of the closest cluster centroid.

- **EM-7:** This approach is an expectation model for seven days. While the other approaches approximate the value function of the policy model, this benchmark is a direct lookahead which, thus, does not suffer from the curses of dimensionality. The problem (34) s.t. (9), (11) to (18), (23) to (24) is a daily expectation model. For this benchmark approach, we modify the daily expectation model. Rather than an expectation model for one day, we use an expectation model for seven days with a receding horizon. More precisely, at each decision stage (DA and ID) and each day, the expectation model for the next seven days is re-optimized. Only the decisions of the first decision stage are executed. As the modification is straightforward, we do not show the reformulation. Again, we use `intlinprog` to solve this MILP. To derive the expectation of future market prices based on the currently known information with the price processes stated in Section 3.1, we use the law of iterated expectations.
- **Perfect foresight:** This benchmark solves the energy storage problem with deterministic prices. Since all electricity prices are known before trading, the corresponding optimization problem can be modeled as an MILP. Obviously, *Perfect foresight* does not define a policy which is applicable in real world but generates an upper bound.

Overall, we only test different value function approximation algorithms and a direct lookahead approach. For other possible solution methods see, for example, [Powell \(2019\)](#).

The value function approximation approaches (*BADP*, *BADP-w*, *ADDP*) are sample-based approaches. Thus, the induced policy highly depends on the sampled (and clustered) sample paths. Additionally, the sample paths depend on the estimation level. See Figure 5 for an illustration. Therefore, a comparison of the implemented approaches cannot rely on a single run for each approach. Accordingly, we run the approaches 20 times, once for each set of estimated price processes (estimation level). The only stochastics which influences *EM-7* is the stochastics on estimation level. *Perfect foresight* is not influenced by stochastics at all.

6.2. Comparison of the solution approaches

In this section, we compare the solution approaches based on the parameters in the previous section. To capture the randomness on estimation level and policy level, we solve the trading problem 20 times with each approach, once for each set of estimated price processes and evaluate with the test data set (see above). We report the relative mean profit summed over the four seasons of the approaches and the 95% confidence bands. We do not report the profits for the single seasons, as the approaches' behavior does hardly change in the different seasons.

The reported machine time is the average machine time to learn the value functions for 30 stages. Since the machine time hardly differs for the different seasons and estimated price processes, we

do not show the confidence bands. Furthermore, we do not report the machine time for deriving the policies during evaluation because, even with the slowest approach (*EM-7*), the policies are derived in a few seconds per decision stage.

Table 3: Performance of *BADP* variants, *ADDP*, *EM-7*, and *Perfect foresight*

Algorithm	Profit [€]	Relative profit [%]	95% confidence band	Learning time [s]
<i>EM-7</i>	93,480.52	100.00	[99.64, 100.36]	0.00
<i>BADP</i>	93,200.08	99.70	[99.09, 100.30]	1,401.32
<i>BADP-w</i>	94,826.64	101.44	[101.04, 101.84]	1,411.83
<i>BADP-w</i> (re-optimized)	94,854.68	101.47	[101.00, 101.93]	2,139.18
<i>ADDP</i>	77,401.87	82.80	[82.17, 83.44]	5,172.40
<i>Perfect foresight</i>	173,593.30	185.70	—	0.00

Table 3 compares the different implemented approaches. There is no significant difference between the profit of *EM-7* and *BADP*. However, the weighted *BADP-w* approach outperforms both. Increasing the number of scenarios N can slightly increase the profit (not shown here). *ADDP* is dominated in terms of profit and machine time by all other approaches. We acknowledge that the rejection of hyperplanes or better redundancy detection techniques could further reduce the machine time, but we do not think this would allow to considerably increase the solution quality as the performance appears to be quite flat in this regard (not shown here). As *EM-7* and *Perfect foresight* do not have to learn the value functions, the machine time is zero. The large gap between the upper bound *Perfect foresight* and the other approaches is not surprising. Electricity prices are hard to forecast and show price spikes which can be exploited by *Perfect foresight*. This underlines that deterministic approaches with full hindsight information have little value in this context. We also tested a variant of the popular least squares monte carlo approach (with lasso and ten-fold cross-validation, not shown here), which yielded 94.31% of *EM-7*.

As mentioned, the approaches behave similar in all four seasons (not shown here). But further investigations indicate that *BADP-w* benefits more from reliable forecasts than *EM-7*. While there is no significant difference of the profits in *Spring*, where the prices are hard to forecast, the largest difference is in *Autumn*, where the error measurements are lowest (see Table 2).

Re-optimizing with *BADP-w* in the middle of the optimization horizon (at $t = 15$) has almost no benefit. Even if more re-optimization steps may slightly improve the profit, this already indicates that *BADP-w* works well without re-optimization. *BADP* and *ADDP* could benefit from re-optimizing as well. To save space, we do not investigate this in the numerical study, since these approaches cannot compete with *BADP-w*.

The objective function includes only costs that are variable in the considered trading problem. We use the term “profit” for simplicity, although, formally, “contribution margin” would be more appropriate. In reality, additional costs like wages and depreciation of investment have to be taken

into account. Therefore, the 1.44 percentage point increase from *EM-7* to *BADP-w* increases the net profit much more and may even turn a loss into a profit. Because of the high investment and running costs of energy storage systems they usually have a low internal rate of return (IRR) of 0.2% to 9.1% (Steffen (2012), Zhu et al. (2013), Olaszi & Ladanyi (2017)). The actual IRR depends on the specifications and costs of the energy storage and the trading strategy. If one can increase the “contribution margin” with *BADP-w* by 1.44 percentage points this also increases the IRR by nearly 1.44 percentage points. Therefore, the IRR can be dramatically increased by 16 to 840 %.

6.3. Decision making insights

To investigate the decisions of the storage owner and to show that the decisions derived by *BADP-w* are reasonable, we analyze the storage level for *Autumn*. Since we derived 20 different policies with our best approach *BADP-w* we report the average storage level over the 20 policies in Figure 6. Additionally, we report the maximum and minimum storage levels as well as the day-ahead prices. As reference points the Mondays (0:00) of *Autumn* are marked.

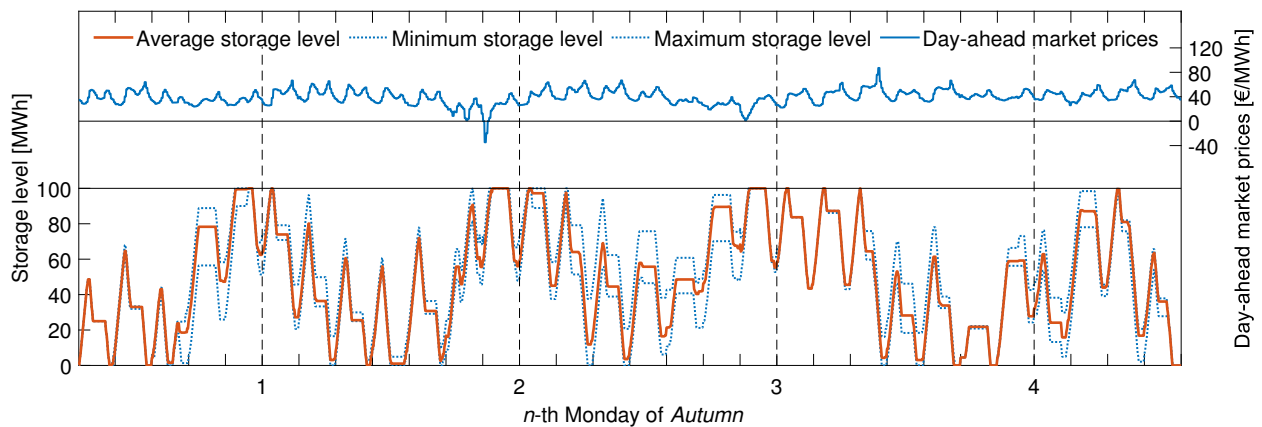


Figure 6: Storage level over time with prices

Overall, we observe strong daily and weekly patterns in the storage level, which follow electricity prices. On a weekly level, the storage is charged on weekends and discharged at the beginning of the week. On a daily level, it is charged right after midnight and discharged during the day. In particular, we see that at the first three Sundays the storage is full, as it was charged during the weekends where the prices are usually low to sell electricity during the week where the prices are higher. Only during the fourth weekend the storage is not fully charged as prices did not drop.

After the storage is fully charged, not all energy is sold directly on Monday. This is only true for the first Monday. After the second Monday, most energy is sold Tuesday and after the third Monday at Wednesday. These were the weekdays where the prices peaked.

In some hours or even days the gap between the policies with maximum storage level and minimum storage level is quite large (e.g. in the week of the second Monday). This happens if multiple nearly equally good decisions exist (e.g. discharging a day later at nearly equal prices).

6.4. Influence of the price forecast

To check if it is worth to handle a high-dimensional price forecast in the dynamic program, we vary the number of days in the price forecast and therefore the dimension of the exogenous part of the state $P_t^{history}$. We denote the number of days in the price history as D . The corresponding price history is $P_t^{history} = (P_{t+1-D}^{DA}, \dots, P_t^{DA}, P_{t+1-D}^{ID}, \dots, P_t^{ID})$. For each $D \in \{1, \dots, 10\}$ we estimate the price forecast using lasso with ten-fold cross validation 20 times and learn the value functions with our best approach $BADP-w$. Again we report the mean profit and the 95 % confidence bands. In addition, we use $EM-7$ with seven days in the price history as a benchmark, since this does not suffer from the curses of dimensionality.

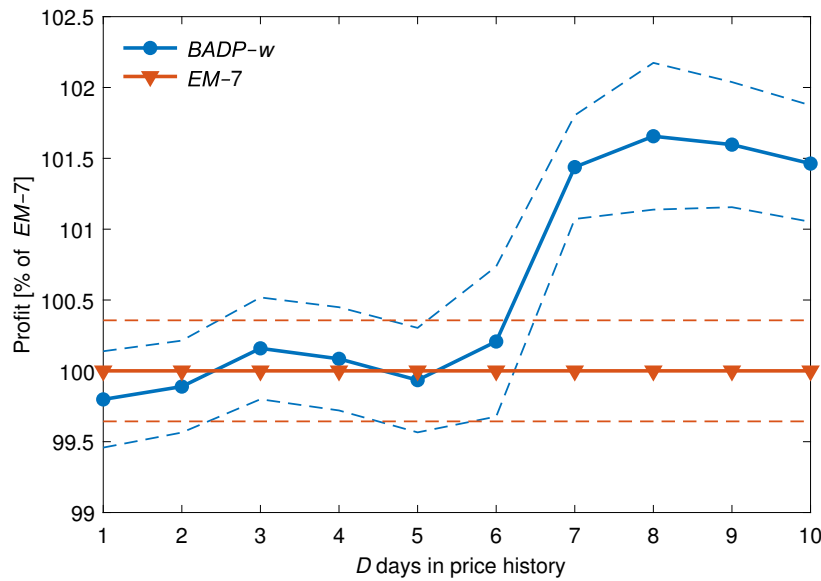


Figure 7: Influence of price forecast used in optimization on expected profits

Figure 7 illustrates the influence of the length of the price history used in the optimization on profit. We estimate ten different price processes for each market with one ($S_t^{DA} \in \mathbb{R}^{122}$) up to ten days ($S_t^{DA} \in \mathbb{R}^{1202}$) in the price history. We fit the value functions with $BADP-w$ and derive the decisions based on this forecast and evaluate these with the data in our test data set. The difference of the mean profits with 1 to 6 days in the price history is not significant. But the seventh day in the price history is important for $BADP-w$ to outperform $EM-7$ which always uses seven days in the price history to calculate the forecast used in the expectation model. This significant increase in the profit follows because the electricity prices have a strong weekly pattern and, thus, the prices of the same day of the last week are good predictors for current prices. Again, the difference of the mean profits with 7 to 10 days in the price history is not significant. Note that the fact that profit increases in the price history is not obvious, because a larger state variable is harder to handle, which could also lead to a decline in profit. Thus, Figure 7 demonstrates that $BADP-w$ can handle a large state variable. More importantly, as there is no reason not to use a high-dimensional price

forecast for the benchmark approach, Figure 7 indicates that the dynamic program needs at least seven days in the price history to outperform the benchmark approach with seven days. Thus, if we are not willing to handle the large state variable in the dynamic program, the much simpler benchmark approach would be the better choice. Moreover, we also tested the influence of the price forecast on $EM-7$ (i.e. with D days in the history used to compute the expectations for the next 7 days); not shown here. The profit of $EM-7$ shows a similar pattern as the profit of $BADP-w$, but $BADP-w$ outperforms $EM-7$ for every D .

Since Figure 7 indicates that the seventh day in the price history is more important than the previous days, this suggests the reduction of the state space via feature selection. One could only include the most relevant prices of the previous week. While the importance of features (dimensions of the price history) is considered in the weights of $BADP-w$, the feature selection is not investigated in full detail. As the modeled price processes must follow the Markov property, the feature selection is not straightforward. It is possible to fit the value functions with $BADP-w$ based on a simplified forecast and therefore simplified model (e.g. $D = 1$) but derive the decisions based on a more complex forecast (e.g. $D = 7$). For this, the higher dimensional S_t^{DA} and therefore $P_t^{history}$ is treated as a latent state variable while deriving the trading decisions based on the fixed value function approximations. Therefore, we compare this approach with the data in Figure 7. For this, we derive the decisions based on our basic forecast ($D = 7$).

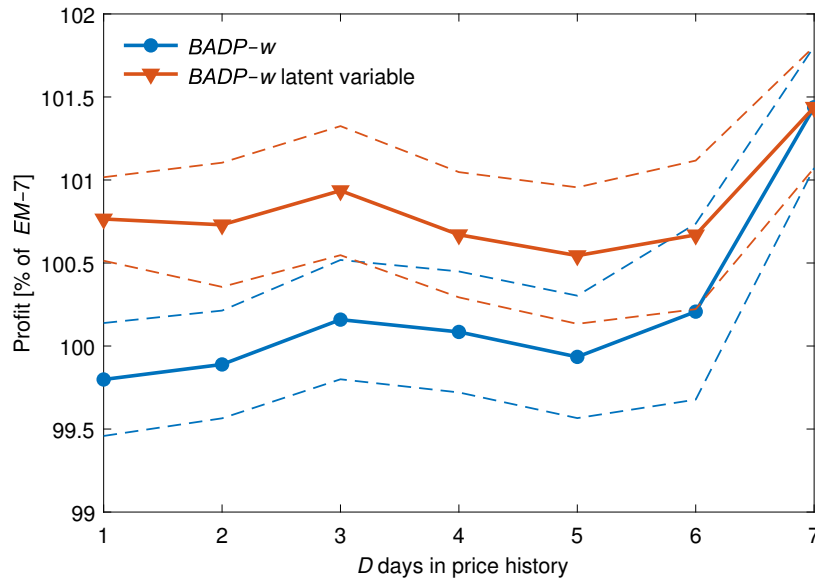


Figure 8: Influence of underlying price process on expected profits

Figure 8 is similar to Figure 7, but is extended by the latent state variable approach where the decisions for $D < 7$ are made based on the $D = 7$ forecast (basic setting). To keep it simple, we do not show the $EM-7$ benchmark with its confidence bands. Figure 8 demonstrates that treating a higher dimensional forecast as a latent variable pays off, but the gain decreases in D . The latent

state variable approach with three days in the price history comes closest to the basic setting $D = 7$, but further tests (not shown here) indicate that the difference is still significant. We do not include $D > 7$, because in this cases the price history would contain dimensions that are not used for the forecast and therefore would be the irrelevant.

In conclusion, it is essential for dynamic programs to use a competitive forecast to outperform a simple benchmark approach.

6.5. Gains from integrated trading

In this section, we compare the integrated day-ahead and intraday trading with restricted trading settings. In particular we consider the following settings:

- **Day-ahead only (DA):** This setting trades only in the day-ahead market and corresponds to our policy functions $X_t^{DA,\pi}(\cdot)$ and $X_t^{ID,\pi}(\cdot)$ with additional constraint $x_{t,h,q}^{ID} = 0 \forall h, q$.
- **Intraday only (ID):** This setting trades only in the intraday market. This can be reflected with our policy functions $X_t^{DA,\pi}(\cdot)$ and $X_t^{ID,\pi}(\cdot)$ by adding the constraint $x_{t,h}^{DA} = 0 \forall h$.
- **DA+ID, sequential:** This setting trades in the day-ahead market without anticipating the intraday market. Afterwards, the commitments are adjusted on the intraday market. To reflect this with our policy functions, we add the constraint $x_{t,h,q}^{ID} = 0 \forall h, q$ to $X_t^{DA,\pi}(\cdot)$ and the constraint $x_{t,h}^{DA} = 0 \forall h$ to $X_t^{ID,\pi}(\cdot)$.
- **DA+ID, integrated:** This is the basic setting considered in the remainder of the paper.
- **DA+ID, integrated, no arbitrage:** Integrated trading in both auction markets can speculate on the price difference between the hourly day-ahead market products and the price of the corresponding quarter-hours on the intraday market. In this setting, we modify the intraday market price process defined in Section 3.1 to enforce that the expected market prices are equal. Therefore, no arbitrage based on the expected price difference is possible. For this setting, we shift the expected intraday market prices to the expected day-ahead market prices and use them in the contribution function of the policy model (34):

$$\bar{P}_{t,t+1,h,q}^{ID'} := \bar{P}_{t,t+1,h,q}^{ID} - \frac{1}{4} \sum_{q'=0}^3 \left(\bar{P}_{t,t+1,h,q'}^{ID} - \bar{P}_{t,t+1,h}^{DA} \right) \forall h \quad (40)$$

All settings are optimized with the *BADP-w* approach with seven days in the price history and evaluated with real-world data. Again, we run the simulation (estimating price processes, solving trading problem, and evaluation) 20 times, to capture the stochastics (see, Figure 5). We report the mean of the summed profits over the four seasons and the 95% confidence bands.

Table 4 compares our basic setting (DA+ID, integrated) with four other settings: only using the day-ahead market (DA), only using the intraday market (ID), using both markets with a

Table 4: Comparison to restricted trading settings

Trading and optimization	Profit €	Relative profit [%]	95% confidence band
Only day-ahead (DA)	37,693.59	39.75	[39.03, 40.47]
Only intraday (ID)	92,465.45	97.51	[97.20, 97.83]
DA+ID, sequential	87,221.54	91.98	[91.74, 92.22]
DA+ID, integrated	94,826.64	100.00	[99.61, 100.39]
DA+ID, integrated, no arbitrage	94,712.84	99.88	[99.39, 100.37]

sequential optimization (DA+ID, sequential), and trading integrated but without exploiting the price differences (DA+ID, integrated, no arbitrage). If only one market is used, the intraday market is much more profitable for the storage owner than the day-ahead market. This is because the intraday market prices fluctuate substantially more than the day-ahead market prices, which can be exploited by the fast ramping energy storage. The immense difference between the day-ahead and intraday profits decreases in the ramping time (not shown here). In the sequential day-ahead and intraday setting, the storage owner first trades on the day-ahead market, as if only the day-ahead market exists, i.e. exactly as in setting (DA). After this, the storage owner adjusts the commitments on the intraday market. The integrated bidding can enhance the profit by 8 percentage points compared to the sequential consideration of the markets and 2.5 percentage points compared to the intraday market setting. In our study, the intraday market setting outperforms the sequential setting. This is because decisions on the day-ahead market cause a bad starting point for trading on the intraday market and, thus, profit is lower. This changes with increasing ramping times (not shown here). If the storage owner does not exploit the expected price difference between the markets, the relative profit only decreases by 0.12 percentage points. The difference between these two settings is not significant. Thus, (almost) no arbitrage is possible, which indicates that the German auction markets are efficient.

7. Discussion

In this section, we discuss possible limitations of our research. We observed that integrated trading in both German auction markets outperforms sequential trading and trading in a single auction market. We did not investigate whether integrated trading in the auction markets outperforms trading in other markets (e.g., derivative markets, continuous intraday market, balancing markets, over-the-counter). Furthermore, we did not investigate if integrated trading with additional markets is beneficial. In the following, we discuss how this limits our research and how practice can still benefit.

Derivative markets: Trading derivatives can be an important instrument to hedge the risk (e.g., variance of returns), see [Hain et al. \(2018\)](#). Since we optimize expected profit, the derivative markets are not beneficial.

Continuous intraday market: As the continuous intraday market works similar to stock markets, a product's current price includes all relevant information and is therefore the best predictor for the future price of the product (see, e.g., [Narajewski & Ziel \(2020a\)](#)). Since the continuous intraday market starts with the intraday auction market clearing prices and both markets have the same temporal resolution, integrated trading in the intraday auction market and the continuous intraday market should not be profitable. Moreover, the continuous intraday market has a rather high market entrance barrier and a 24/7 trading floor is needed.

Nevertheless, sequential trading in the continuous intraday market after integrated trading in the auction markets should outperform pure auction market trading. Since the trading in the auction markets restricts the trading in the continuous intraday market, pure continuous intraday market trading could be even better than a heuristic solution of integrated trading.

Balancing markets: The balancing markets can be an important opportunity for fast ramping energy storage systems. We do not include them in our research for the following reason. The rules of the German balancing markets changed multiple times in recent years (e.g., 2018, 2019, and 2020). Since these changes happen during the time span considered in the numerical study of Section 6, nearly no data for the current regime exists.

Over-the-counter markets: For energy storage systems, the over-the-counter (OTC) markets are hardly relevant, because there, most energy is contracted several days or months ahead. Most energy storage systems are fully charged and discharged within hours, so they benefit most from short-term price fluctuations. Additionally, there is no public information about OTC markets available.

In summary, we are confident that we modeled the two most important German markets, but acknowledge that the integration of additional markets, like balancing markets, is desirable.

Based on our correspondence with practitioners and in line with literature, we see that market participants often trade in more than two markets but in a sequential manner. As seen in Section 6.5 this can decrease profit, even compared to trading in fewer markets. Nevertheless, market participants that trade in a sequential manner can use our model and approach to combine two links in the chain to improve their trades. This is the first step towards a fully anticipating trading strategy that integrates all relevant markets in an optimization problem.

8. Conclusion and insights

In this paper, we modeled the energy trading problem for both German auction markets with a competitive price forecast as a high-dimensional dynamic program. We implemented a suitable backwards ADP solution algorithm and showed that the approach successfully overcomes the resulting curse of dimensionality. Furthermore, the implemented approach outperforms an intuitive

and widely used benchmark using an expectation model in a receding horizon fashion as well as a state-of-the-art approach from literature. However, it is important to use a long enough price history in the optimization's forecast to outperform the expectation model. This is an important observation as the resulting explosion of the DP's state space often forces researchers to use shorter histories, whereas this is much less a problem for expectation models. Furthermore, one has to use an approach that is designed for handling high-dimensional price processes, albeit common methods from literature cannot.

Regarding the integrated trading, we have shown that integrated trading in both German auction markets is beneficial. Further, we have shown that trading in a sequential manner can be even worse than only trading in the intraday auction market. This shows that storage owners that sequentially trade have to rethink this and anticipate later markets. This may also be true for markets not included in our research. To reduce the computational burden of such an integrated optimization it is sufficient to use self-schedule bidding in the auction markets. If one needs to reduce the computational burden even more, an expectation model with a competitive price process is more appropriate than an ADP approach with a simple price process.

Future work could go in several directions. First, from a methodological perspective, infinite-horizon dynamic programs should be considered. Whereas our proposed dynamic program is time-independent (except the day of the week), future work can reformulate the problem as a steady state problem and solve the storage problem with an infinite horizon. This has its own challenges. For example, the set of sample paths can no longer simply be sampled with the price processes because this depends upon the initial price history. Instead, one can use all past price histories of the last few years. Second, another research avenue would be to investigate the influence of ramping time and other properties of the energy storage on expected profit. Especially if investment costs are taken into account one can, for example, investigate the benefit of a small buffer battery for a pumped hydro storage to enable instant ramping.

References

- Bonnans, J. F., Cen, Z., & Christel, T. (2011). Energy contracts management by stochastic programming techniques. *Annals of Operations Research*, 200, 199–222.
- Boomsma, T. K., Juul, N., & Fleten, S.-E. (2014). Bidding in sequential electricity markets: The nordic case. *European Journal of Operational Research*, 238, 797–809.
- Brijs, T., Geth, F., Jonghe, C. D., & Belmans, R. (2019). Quantifying electricity storage arbitrage opportunities in short-term electricity markets in the CWE region. *Journal of Energy Storage*, 25, 100899.
- Broadie, M., Glasserman, P., & Ha, Z. (2000). Pricing american options by simulation using a stochastic mesh with optimized weights. In *Nonconvex Optimization and Its Applications* (pp. 26–44). Springer US.
- Crespo-Vazquez, J. L., Carrillo, C., Diaz-Dorado, E., Martinez-Lorenzo, J. A., & Noor-E-Alam, M. (2018). A machine learning based stochastic optimization framework for a wind and storage power plant participating in energy pool market. *Applied Energy*, 232, 341–357.
- Ding, H., Hu, Z., & Song, Y. (2015). Rolling optimization of wind farm and energy storage system in electricity markets. *IEEE Transactions on Power Systems*, 30, 2676–2684.

- Durante, J., Nascimento, J., & Powell, W. B. (2020). Backward approximate dynamic programming with hidden semi-markov stochastic models in energy storage optimization. Working paper, Princeton University, Princeton.
- EPEX SPOT SE (2020). Market data. <https://www.epexspot.com/en/market-data>. Accessed 21 July 2020.
- Fathima, A. H., & Palanisamy, K. (2015). Optimization in microgrids with hybrid energy systems – a review. *Renewable and Sustainable Energy Reviews*, 45, 431–446.
- Finnah, B. (2021). Optimal bidding functions for renewable energies in sequential electricity markets. *OR Spectrum*, . doi:10.1007/s00291-021-00646-9.
- Finnah, B., & Gönsch, J. (2021). Optimizing trading decisions of wind power plants with hybrid energy storage systems using backwards approximate dynamic programming. *International Journal of Production Economics*, 238, 108–155.
- Fleten, S.-E., & Kristoffersen, T. K. (2008). Short-term hydropower production planning by stochastic programming. *Computers & Operations Research*, 35, 2656–2671.
- Ghavidel, S., Ghadi, M. J., Azizivahed, A., Aghaei, J., Li, L., & Zhang, J. (2020). Risk-constrained bidding strategy for a joint operation of wind power and CAES aggregators. *IEEE Transactions on Sustainable Energy*, 11, 457–466.
- Glasserman, P. (2003). *Monte Carlo Methods in Financial Engineering*. Springer New York.
- Gönsch, J., & Hassler, M. (2016). Sell or store? an ADP approach to marketing renewable energy. *OR Spectrum*, 38, 633–660.
- Hain, M., Schermeyer, H., Uhrig-Homburg, M., & Fichtner, W. (2018). Managing renewable energy production risk. *Journal of Banking & Finance*, 97, 1–19.
- Hannah, L. A., & Dunson, D. B. (2011). Approximate dynamic programming for storage problems. In *Proceedings of the 28th International Conference on International Conference on Machine Learning ICML'11* (pp. 337–344). USA: Omnipress.
- Heredia, F. J., Cuadrado, M. D., & Corchero, C. (2018). On optimal participation in the electricity markets of wind power plants with battery energy storage systems. *Computers & Operations Research*, 96, 316–329.
- Jędrzejewski, A., Marcjasz, G., & Weron, R. (2021). Importance of the long-term seasonal component in day-ahead electricity price forecasting revisited: Parameter-rich models estimated via the LASSO. *Energies*, 14, 3249.
- Jiang, D. R., Pham, T. V., Powell, W. B., Salas, D. F., & Scott, W. R. (2014). A comparison of approximate dynamic programming techniques on benchmark energy storage problems: Does anything work? In *2014 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*. IEEE.
- Khatami, R., Oikonomou, K., & Parvania, M. (2020). Look-ahead optimal participation of compressed air energy storage in day-ahead and real-time markets. *IEEE Transactions on Sustainable Energy*, 11, 682–692.
- Kim, J. H., & Powell, W. B. (2011). Optimal energy commitments with storage and intermittent supply. *Operations Research*, 59, 1347–1360.
- Kuznetsova, E., Ruiz, C., Li, Y.-F., & Zio, E. (2015). Analysis of robust optimization for decentralized microgrid energy management under uncertainty. *International Journal of Electrical Power & Energy Systems*, 64, 815–832.
- Löhndorf, N., & Minner, S. (2010). Optimal day-ahead trading and storage of renewable energies—an approximate dynamic programming approach. *Energy Systems*, 1, 61–77.
- Löhndorf, N., & Shapiro, A. (2019). Modeling time-dependent randomness in stochastic dual dynamic programming. *European Journal of Operational Research*, 273, 650–661.
- Löhndorf, N., Wozabal, D., & Minner, S. (2013). Optimizing trading decisions for hydro storage systems using approximate dual dynamic programming. *Operations Research*, 61, 810–823.
- Marcjasz, G. (2020). Forecasting electricity prices using deep neural networks: a robust hyper-parameter selection scheme. *Energies*, 13, 4605.
- Marcjasz, G., Uniejewski, B., & Weron, R. (2019). On the importance of the long-term seasonal component in day-ahead electricity price forecasting with narx neural networks. *International Journal of Forecasting*, 35, 1520–1532.

- Moghaddam, I. G., Nick, M., Fallahi, F., Sanei, M., & Mortazavi, S. (2013). Risk-averse profit-based optimal operation strategy of a combined wind farm-cascade hydro system in an electricity market. *Renewable Energy*, 55, 252–259.
- Narajewski, M., & Ziel, F. (2020a). Econometric modelling and forecasting of intraday electricity prices. *Commodity Markets*, 19, 100–107.
- Narajewski, M., & Ziel, F. (2020b). Ensemble forecasting for intraday electricity prices: Simulating trajectories. *Applied Energy*, 279, 115801.
- Narajewski, M., & Ziel, F. (2021). Optimal bidding on hourly and quarter-hourly day-ahead electricity price auctions. URL: <https://arxiv.org/pdf/2104.14204.pdf>.
- Olaszi, B. D., & Ladanyi, J. (2017). Comparison of different discharge strategies of grid-connected residential pv systems with energy storage in perspective of optimal battery energy storage system sizing. *Renewable and Sustainable Energy Reviews*, 75, 710–718.
- Powell, W. B. (2019). A unified framework for stochastic optimization. *European Journal of Operational Research*, 275, 795–821.
- Rahman, H. A., Majid, M. S., Jordehi, A. R., Kim, G. C., Hassan, M. Y., & Fadhl, S. O. (2015). Operation and control strategies of integrated distributed energy resources: A review. *Renewable and Sustainable Energy Reviews*, 51, 1412–1420.
- Salas, D. F., & Powell, W. B. (2018). Benchmarking a scalable approximate dynamic programming algorithm for stochastic control of grid-level energy storage. *INFORMS Journal on Computing*, 30, 106–123.
- Scott, P. (2019). Approximate dynamic programming for energy storage with new results on instrumental variables and projected bellman errors. *Submitted to Operations Research*, .
- Steffen, B. (2012). Prospects for pumped-hydro storage in germany. *Energy Policy*, 45, 420–429.
- Sterner, M., & Stadler, I. (2017). *Energiespeicher - Bedarf, Technologien, Integration*. Springer Berlin Heidelberg.
- Uniejewski, B., & Weron, R. (2018). Efficient forecasting of electricity spot prices with expert and lasso models. *Energies*, 11, 2039.
- Vahedipour-Dahraie, M., Rashidizadeh-Kermani, H., Anvari-Moghaddam, A., & Siano, P. (2020). Risk-averse probabilistic framework for scheduling of virtual power plants considering demand response and uncertainties. *International Journal of Electrical Power & Energy Systems*, 121, 106126.
- Viehmann, J. (2017). State of the german short-term power market. *Zeitschrift für Energiewirtschaft*, 41, 87–103.
- Weitzel, T., & Glock, C. H. (2018). Energy management for stationary electric energy storage systems: A systematic literature review. *European Journal of Operational Research*, 264, 582–606.
- Weron, R. (2014). Electricity price forecasting: A review of the state-of-the-art with a look into the future. *International Journal of Forecasting*, 30, 1030–1081.
- Wozabal, D., & Rameseder, G. (2020). Optimal bidding of a virtual power plant on the spanish day-ahead and intraday market for electricity. *European Journal of Operational Research*, 280, 639–655.
- Xu, X., Hu, W., Cao, D., Huang, Q., Liu, Z., Liu, W., Chen, Z., & Blaabjerg, F. (2020). Scheduling of wind-battery hybrid system in the electricity market using distributionally robust optimization. *Renewable Energy*, 156, 47–56.
- Zéphyr, L., & Anderson, C. L. (2017). Stochastic dynamic programming approach to managing power system uncertainty with distributed storage. *Computational Management Science*, 15, 87–110.
- Zhou, Y. H., Scheller-Wolf, A., Secomandi, N., & Smith, S. (2016). Electricity trading and negative prices: Storage vs. disposal. *Management Science*, 62, 880–898.
- Zhu, D., Wang, Y., Yue, S., Xie, Q., Pedram, M., & Chang, N. (2013). Maximizing return on investment of a grid-connected hybrid electrical energy storage system. In *2013 18th Asia and South Pacific Design Automation Conference (ASP-DAC)* (pp. 638–643).
- Ziel, F. (2016). Forecasting electricity spot prices using lasso: On capturing the autoregressive intraday structure. *IEEE Transactions on Power Systems*, 31, 4977–4987.

Appendix A. Self-schedule bidding

To check the self-schedule assumption Assumption 2, we show numerically that the gains from price-volume bids are negligible for electricity prices with highly correlated noise (like in Germany). For this, we solve a simplified energy trading problem numerically with price-volume bids and with self-schedule bids and evaluate these with common out-of-sample scenarios. In this simplified setting, the PHS only trades in the day-ahead market and ramping is ignored. The notation is summarized in Appendix D. To show the effect of the correlation of the noise, we draw the scenarios based on the price process defined in Section 3.1, but artificially reduce the correlation. We denote the artificial correlation matrix as $\Sigma^{DA,\gamma}$. This is defined as

$$\Sigma^{DA,\gamma} = (\sigma_{i,j}^{DA,\gamma})_{i=1,\dots,24;j=1,\dots,24} := \begin{cases} \sigma_{i,j}^{DA} & i = j \\ \gamma \sigma_{i,j}^{DA} & \text{else} \end{cases} \quad (\text{A.1})$$

with $\gamma \in [0, 1]$ and the originally estimated correlation matrix $\Sigma^{DA} = (\sigma_{i,j}^{DA})_{i=1,\dots,24;j=1,\dots,24}$. For $\gamma = 1$, the matrix $\Sigma^{DA,\gamma}$ has the correlation estimated with real-world data. For $\gamma = 0$, the correlation is zero.

We model the simplified decision problem as a two-stage stochastic program. Based on a state $S_t^{DA} = (R_t^{start}, x_t^{start}, P_t^{history})$, the here-and-now decisions are B selling and buying bids for each hour. The wait-and-see decisions are the actual dispatching of the PHS after the market is cleared. The volume components of the bids are denoted $x_{t,h,b}^{buy,bid}$ and $x_{t,h,b}^{sell,bid}$ with hour h and the number of the bid $b \in \{1, \dots, B\}$. The price components are denoted $P_{t,h,b}^{buy,bid}$ and $P_{t,h,b}^{sell,bid}$. While we model the volumes as continuous decision variable, the prices are fixed parameters. Therefore, we map each price parameter $P_{t,h,b}^{buy,bid}$ and $P_{t,h,b}^{sell,bid}$ to volume $x_{h,b}^{buy,bid}$ and $x_{b,h}^{sell,bid}$.

As we replace the stochastic program with the SAA based on S scenarios, we sample the price scenarios $P_{t+1,h,s}^{DA}$ for each hour h based on the price process defined in Section 3.1 with correlation matrix $\Sigma^{DA,\gamma}$ and the price history $P_t^{history}$. Based on these samples, we can pre-process the parameters $a_{t,h,s,b}^{buy}$ and $a_{t,h,s,b}^{sell}$.

$$a_{t,h,s,b}^{buy} = \begin{cases} 1 & , P_{t,h,b}^{buy,bid} \geq P_{t+1,h,s}^{DA} \\ 0 & , \text{otherwise} \end{cases} \quad (\text{A.2})$$

$$a_{t,h,s,b}^{sell} = \begin{cases} 1 & , P_{t,h,b}^{sell,bid} < P_{t+1,h,s}^{DA} \\ 0 & , \text{otherwise} \end{cases} \quad (\text{A.3})$$

If $a_{t,h,s,b}^{buy} = 1$, the buying bid b in hour h and scenario s is accepted (analogously for $a_{t,h,s,b}^{sell}$).

$$\begin{aligned} \max \frac{1}{S} \sum_{s=1}^S \Delta t^{DA} \sum_{h=1}^{24} & - \left(P_{t+1,h,s}^{DA} + c^{gf} \right) x_{t,h,s}^{buy} + P_{t+1,h,s}^{DA} x_{t,h,s}^{sell} + Q_{t+1,h,s}^{over} x_{t,h,s}^{over} - Q_{t+1,h,s}^{under} x_{t,h,s}^{under} \\ & - c^{pump} z_{t,h,s}^{pump} - c^{turbine} z_{t,h,s}^{turbine} \end{aligned} \quad (\text{A.4})$$

s.t.

$$x_{t,h,s}^{buy} = \sum_{b=1}^B a_{t,h,s,b}^{buy} x_{t,h,b}^{buy,bid} \quad (\text{A.5})$$

$$x_{t,h,s}^{sell} = \sum_{b=1}^B a_{t,h,s,b}^{sell} x_{t,h,b}^{sell,bid} \quad (\text{A.6})$$

$$x_{t,h,s}^{pump} + x_{t,h,s}^{over} = x_{t,h,s}^{buy} \quad (\text{A.7})$$

$$x_{t,h,s}^{turbine} + x_{t,h,s}^{under} = x_{t,h,s}^{sell} \quad (\text{A.8})$$

$$R_{t,h,s} = R_{t,h-1,s} + \eta^{pump} \Delta t^{DA} x_{t,h,s}^{pump} - \frac{\Delta t^{DA}}{\eta^{turbine}} x_{t,h,s}^{turbine} \quad (\text{A.9})$$

$$x_{min}^{comp} y_{t,h,s}^{comp} \leq x_{t,h,s}^{comp} \leq x_{max}^{comp} y_{t,h,s}^{comp} \quad \forall comp \in \{pump, turbine\} \quad (\text{A.10})$$

$$z_{t,h,s}^{comp} \geq y_{t,h,s}^{comp} - y_{t,h-1,s}^{comp} \quad \forall comp \in \{pump, turbine\} \quad (\text{A.11})$$

$$R_{t,h,s} \in [0, R_{max}] \quad (\text{A.12})$$

$$x_{t,h,b}^{buy}, x_{t,h,b}^{sell} \geq 0 \quad (\text{A.13})$$

$$x_{t,h,s}^{over}, x_{t,h,s}^{under}, z_{t,h,s}^{pump}, z_{t,h,s}^{turbine} \geq 0 \quad (\text{A.14})$$

$$y_{t,h,s}^{pump}, y_{t,h,s}^{turbine} \in \{0, 1\} \quad (\text{A.15})$$

All constraints are defined for all $s \in \{1, \dots, S\}$ and $h \in \{0, \dots, 23\}$. Only constraint (A.13) is defined for all $b \in \{1, \dots, B\}$ and $h \in \{0, \dots, 23\}$. Additionally, $R_{t,-1,s}$ is defined as R_t^{start} and $y_{t,0,-1}^{pump} = 1_{x_t^{start} > 0}$ and $y_{t,0,-1}^{turbine} = 1_{x_t^{start} < 0}$. The objective (A.4) maximizes the average profit over all samples. While pumping, the PHS must pay the market price $P_{t+1,h,s}^{DA}$ and the grid fees c^{gf} . While generating energy, the PHS receives the market price. If the PHS does not deliver the energy sold (e.g., empty storage), it has to pay the balancing price for under supply $Q_{t+1,h,s}^{under}$. If the PHS stores less energy than bought (e.g., full storage), it receives the balancing price for over supply $Q_{t+1,h,s}^{over}$. The increased maintenance costs during start-up are reflected by c^{pump} and $c^{turbine}$. Constraints (A.5) and (A.6) derive the contracted commitments based on the decided volumes $x_{t,h,b}^{(\cdot,bid)}$ which are mapped to the parameters $P_{t,h,b}^{(\cdot,bid)}$. Therefore, in scenario s and hour h , the PHS buys $x_{t,h,s}^{buy}$ and sells $x_{t,h,s}^{sell}$. Constraints (A.7) and (A.8) derive the difference between the contracted energy and the stored/delivered energy. Constraint (A.9) is the storage balance equation. Constraint (A.10) limits the power of the turbine and the pump to the feasible range. The binary variables $y_{t,h,s}^{pump}$ and $y_{t,h,s}^{turbine}$ are equal to one, if the pump/turbine work in scenario s and hour h . Constraint (A.11)

captures if the pump or the turbine has to start-up. The constraints (A.12) to (A.15) define the domain of the variables.

If not stated otherwise, we use the same parameters as in the numerical study of Section 6. Since this section focuses on daily energy trading, we use $R_t^{start} = 0$ and $x_t^{start} = 0$. Moreover, we define balancing prices for over- and undersupply that are linear in the day-ahead market price.

$$Q_{t+1,h,s}^{over} = Q_{intercept}^{over} + Q_{slope}^{over} P_{t+1,h,s}^{DA} \quad (\text{A.16})$$

$$Q_{t+1,h,s}^{under} = Q_{intercept}^{under} + Q_{slope}^{under} P_{t+1,h,s}^{DA} \quad (\text{A.17})$$

The following parameters differ for the energy trading problem with price-volume bids and the problem with self-schedule bids:

- **Price-volume bids:** Here we use $S = 50$ price scenarios drawn with the estimated price process but artificial correlation matrix $\Sigma^{DA,\gamma}$. As parameters for the price components of the $B = 33$ bids, we use $P_{t,h,b}^{buy,bid}, P_{t,h,b}^{sell,bid} \in \{-\infty, \bar{P}_{t,t+1,h}^{DA} - 30, \bar{P}_{t,t+1,h}^{DA} - 28, \dots, \bar{P}_{t,t+1,h}^{DA} + 30, \infty\} \forall h$. Thus, self-schedule bids within $\pm 30 \text{ €/MWh}$ of the expected market prices are used.
- **Self-schedule bids:** Here we use a single price scenario ($S = 1$) with $P_{t+1,h,s}^{DA} = \bar{P}_{t,t+1,h}^{DA} \forall h$. Therefore, the prices are equal to the expected day ahead market on day $t + 1$ only using data known at day t . To force (A.4) to (A.15) to use self schedule bids, we use as price components of the bids $P_{t,h,b}^{buy,bid} = \infty \forall h$ and $P_{t,h,b}^{sell,bid} = -\infty \forall h$. Thus, only one bid ($B = 1$) results.

Obviously we cannot report the objective value of (A.4) but have to evaluate this with out-of-sample random numbers. In the following pseudocode we state how to apply (A.4) to (A.15) on out-of-sample data:

Algorithm 3 Evaluate (A.4) to (A.15)

- 1: Sample S day-ahead market prices $P_{t+1,h,s}^{DA}$
 - 2: Solve (A.4) to (A.15)
 - 3: Use decisions $x_{t,h,b}^{buy,bid}$ and $x_{t,h,b}^{sell,bid}$ and observe $P_{t+1,h}^{DA}$
 - 4: Solve (A.4) to (A.15) with the single scenario $P_{t+1,h,s}^{DA} = P_{t+1,h}^{DA}$ and fixed $x_{t,h,b}^{buy,bid}$ and $x_{t,h,b}^{sell,bid}$
-

To show the effect of the correlation of the noise, we solve the daily trading problem for each of the 120 days in the test data presented in Section 6 with $\gamma \in \{1, 0.75, 0.5, 0.25, 0\}$. Since we estimated the price processes 20 times with cross validation, we solve the problem 20 times for each day. In contrast to the remainder of the paper, we only use the corresponding price history $P_t^{history}$ and do not evaluate with real-world data. The reason for this is that the real world data follows a price process with $\gamma = 1$. Therefore, we have to evaluate with randomly drawn scenarios based

on the correlation matrix $\Sigma^{DA,\gamma}$. This procedure introduces a new level of randomness (evaluation level) which is not included in Figure 5. To reduce the computational burden, we set a relative MILP solution gap of 1% while deriving the first-stage decisions.

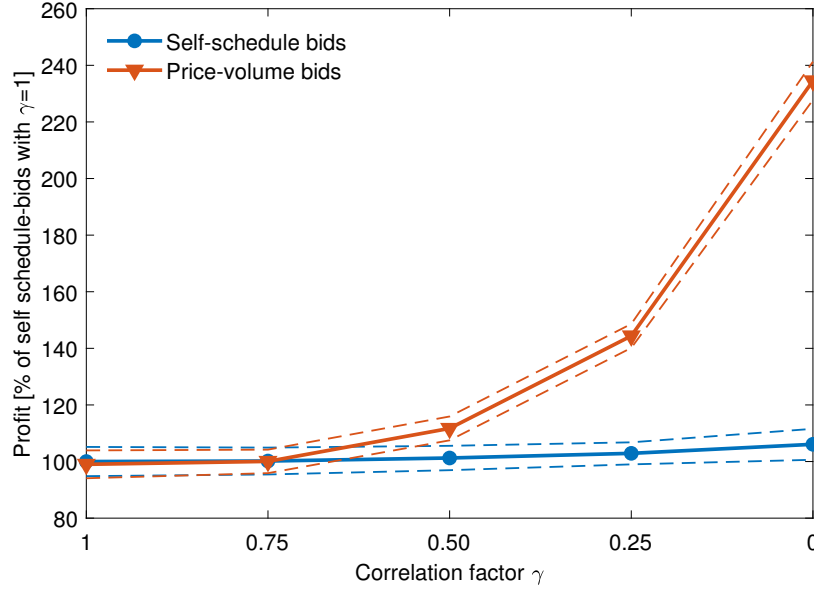


Figure A.9: Influence of correlation on expected profit ($\gamma = 0$: no correlation, $\gamma = 1$: correlation of German market)

Figure A.9 shows the relative average profit summed over the 120 days including the 95% confidence bands based on self-schedule bids and price-volume bids with $\gamma \in \{1, 0.75, 0.5, 0.25, 0\}$. For the correlation based on real-world data ($\gamma = 1$) there is no significant difference. The reason for this is, that energy storage systems are profitable if the price difference between the buying hours prices and selling hours prices is large enough. Since the forecast errors of the 24 day-ahead market prices are very highly correlated (coefficient of correlation up to 0.98 for consecutive hours), forecasts usually over- or underestimate the whole day's prices, which hardly influences the daily price differences. This justifies our self-schedule Assumption 2. Furthermore, the machine time with price-volume bids is over 5,000 times higher than the machine time with self-schedule bids (not shown here). With decreasing γ and therefore decreasing correlation the price-volume bids become increasingly profitable. For uncorrelated noise ($\gamma = 0$), price-volume bids dramatically outperform self-schedule bids by over 130 percentage points. Therefore, in energy markets with lower correlation, energy storage systems must use price-volume bids. For example, the Nordic markets behave very different than the German markets because of the immense penetration of storeable hydropower.

Please note that we only show above that price-volume bids are not needed for the daily energy trading. However, we are confident that this is sound enough for the following reasons: First, from

a theoretical perspective, price-volume bids outperform self-schedule bids if one is able to solve the corresponding problems exactly. Thus, it is reassuring that price-volume bids never perform significantly worse than self-schedule bids. Second, one may ask whether this also holds for the problem with multiple days. We did not investigate this setting here, as the tractability mentioned above becomes more difficult as the problem grows. However, since most energy storage systems are fully charged/discharged within hours, the daily price differences are the most important ones for energy storage systems. In contrast, hydropower plants have huge reservoirs (discharge time of days to weeks) and a natural water inflow. For these power plants it is essential to release the exogenous resource at high price hours to be profitable (e.g. Fleten & Kristoffersen (2008), Boomsma et al. (2014)).

Appendix B. Expectation model versus stochastic program

To demonstrate that the policy designed in Section 4 works well, we compare it with a two-stage stochastic program in a single-day setting. While the here-and-now decisions correspond to the day-ahead market stage, the wait-and-see decisions are the intraday market stage decisions. To solve the two-stage stochastic program we use the sample average approximation with S scenarios.

$$\begin{aligned} \max \quad & \frac{1}{S} \sum_{s=1}^S \sum_{h=0}^{23} -\Delta t^{DA} P_{t+1,h,s}^{DA} x_{t,h}^{DA} + \sum_{q=0}^3 -\Delta t^{ID} \bar{P}_{t+1,h,q,s}^{ID} x_{t,h,q,s}^{ID} - \Delta t^{ID} c^{gf} x_{t,h,q,s}^{pump} \\ & + \frac{\Delta t^{ramp}}{2} \left((\Delta x_{t,h,q,s}^{pump,up} + \Delta x_{t,h,q,s}^{turbine,down}) \bar{Q}_{t+1,h,q,s}^{over} - (\Delta x_{t,h,q,s}^{pump,down} + \Delta x_{t,h,q,s}^{turbine,up}) \bar{Q}_{t+1,h,q,s}^{under} \right) \\ & - c^{pump} z_{t,h,q,s}^{pump} - c^{turbine} z_{t,h,q,s}^{turbine} \end{aligned} \quad (B.1)$$

s.t.

$$x_{t,h}^{DA} \in [-x_{max}^{turbine}, x_{max}^{pump}] \quad (B.2)$$

$$x_{t,h}^{DA} + x_{t,h,q,s}^{ID} = x_{t,h,q,s}^{pump} - x_{t,h,q,s}^{turbine} \quad (B.3)$$

$$x_{min}^{comp} y_{t,h,q,s}^{comp} \leq x_{t,h,q,s}^{comp} \leq x_{max}^{comp} y_{t,h,q,s}^{comp} \quad \forall comp \in \{pump, turbine\} \quad (B.4)$$

$$y_{t,h,q,s}^{pump} + y_{t,h,q,s}^{turbine} \leq 1 \quad (B.5)$$

$$\Delta x_{t,h,q,s}^{comp,up} - \Delta x_{t,h,q,s}^{comp,down} = x_{t,h,q,s}^{comp} - x_{t,h,q-1,s}^{comp} \quad \forall comp \in \{pump, turbine\} \quad (B.6)$$

$$\begin{aligned} R_{t,h,q,s} = R_{t,h,q-1,s} + \eta^{pump} \left(\Delta t^{ID} x_{t,h,q,s}^{pump} - \frac{\Delta t^{ramp}}{2} \Delta x_{t,h,q,s}^{pump,up} + \frac{\Delta t^{ramp}}{2} \Delta x_{t,h,q,s}^{pump,down} \right) \\ - \frac{1}{\eta^{turbine}} \left(\Delta t^{ID} x_{t,h,q,s}^{turbine} - \frac{\Delta t^{ramp}}{2} \Delta x_{t,h,q,s}^{turbine,up} + \frac{\Delta t^{ramp}}{2} \Delta x_{t,h,q,s}^{turbine,down} \right) \end{aligned} \quad (B.7)$$

$$R_{t,h,q,s} \in [0, R_{max}] \quad (B.8)$$

$$y_{t,h,q,s}^{pump}, y_{t,h,q,s}^{turbine} \in \{0, 1\} \quad (B.9)$$

$$x_{t,h,q,s}^{pump}, x_{t,h,q,s}^{turbine}, \Delta x_{t,h,q,s}^{pump,up}, \Delta x_{t,h,q,s}^{pump,down}, \Delta x_{t,h,q,s}^{turbine,up}, \Delta x_{t,h,q,s}^{turbine,down} \geq 0 \quad (B.10)$$

$$y_{t,h,q,s}^{comp} - y_{t,h,q-1,s}^{comp} \leq z_{t,h,q,s}^{comp} \quad \forall comp \in \{pump, turbine\} \quad (B.11)$$

$$z_{t,h,q,s}^{pump}, z_{t,h,q,s}^{turbine} \geq 0 \quad (B.12)$$

All constraints are defined for all $q \in \{0, 1, 2, 3\}$, $h \in \{0, \dots, 23\}$, and $s \in \{1, \dots, S\}$. Additionally, $(\cdot)_{t,h,4,s}$ is defined as $(\cdot)_{t,h+1,0,s}$ and $R_{t,0,-1,s} = R_t^{start}$ and $x_{t,0,-1,s}^{pump} = \max\{x_t^{start}, 0\}$ and $x_{t,0,-1,s}^{turbine} = \max\{-x_t^{start}, 0\}$. Since this section focuses on daily energy trading, we use $R_t^{start} = 0$ and $x_t^{start} = 0$.

The SAA (B.1) to (B.12) is the stochastic counterpart of the policy model. The scenarios for the day-ahead market price $P_{t+1,h,s}^{DA}$ are sampled based on the price process defined in Section 3.1 and the current price history. The samples for the expected intraday market price $\bar{P}_{t+1,h,q,s}^{ID}$ are derived based on the price process defined in Section 3.1 conditioned on the sample $P_{t+1,h,s}^{DA}$. The expected balancing prices are computed based on these the usual way. Obviously, we cannot report the objective value of (B.1) but have to evaluate this with the real-world data in our test data set. In the following pseudocode we state how to apply (B.1) to (B.12) to real-world data:

Algorithm 4 Evaluate (B.1) to (B.12)

- 1: sample S day-ahead market prices $P_{t+1,h,s}^{DA}$ and compute the conditioned expectations $\bar{P}_{t+1,h,q,s}^{ID}$, $\bar{Q}_{t+1,h,q,s}^{over}$, and $\bar{Q}_{t+1,h,q,s}^{under}$
 - 2: solve (B.1) to (B.12)
 - 3: use decisions $x_{t,h}^{DA}$ and observe $P_{t+1,h}^{DA}$
 - 4: compute the expectations $\bar{P}_{t+1,h,q}^{ID}$, $\bar{Q}_{t+1,h,q}^{over}$, and $\bar{Q}_{t+1,h,q}^{under}$ conditioned on the observation $P_{t+1,h}^{DA}$
 - 5: solve (B.1) to (B.12) with the single scenario $\bar{P}_{t+1,h,q}^{ID}$, $\bar{Q}_{t+1,h,q}^{over}$, and $\bar{Q}_{t+1,h,q}^{under}$ and fixed $x_{t,s}^{DA}$.
 - 6: use decisions $x_{t,h,q}^{DA}$ and observe $P_{t+1,h,q}^{ID}$
-

We solve the daily problem for each day in our test data set and evaluate the decisions with the real world data. We generate the samples based on the price processes with seven days in the price history. Since we estimated the price processes 20 times, we solve the daily problems 20 times and report the relative mean profits for $S \in \{10, 20, 30, 40, 50\}$ and the 95% confidence bands. To reduce the computational burden, we set a relative MILP solution gap of 1% while deriving the first-stage decisions (Step 2 in Algorithm 4).

Figure B.10 shows the relative cumulative (over all days) profit and the relative cumulative machine time of the stochastic program with different S . The relative profit slightly increases in the number of scenarios, but even with 50 scenarios, the stochastic program cannot significantly outperform the substantially faster and simpler expectation model. In addition, the expectation model has a much smaller confidence band. The width of the confidence band of the stochastic program decreases in S . Since the machine time dramatically increases in S , we do not test higher values. Even with $S = 50$, the stochastic program is 400 times slower than the expectation model. This shows that the daily expectation model works well. It would be possible to learn value

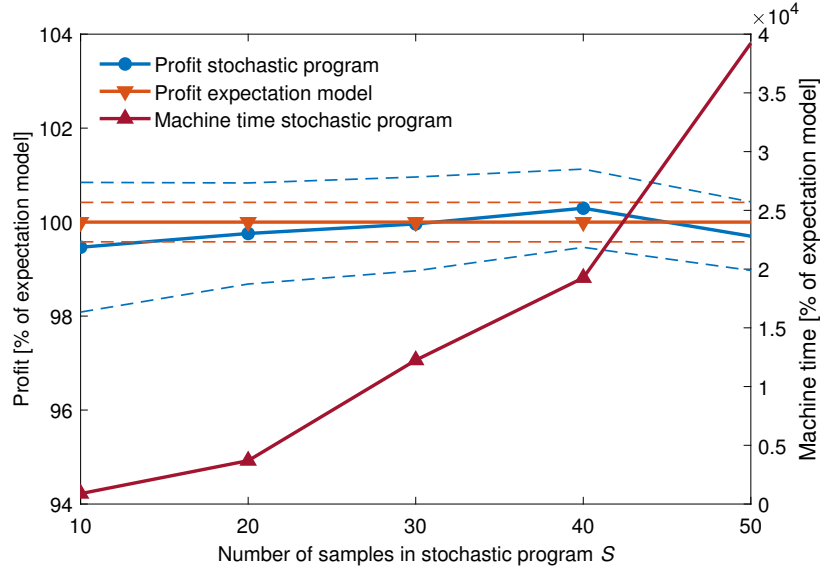


Figure B.10: Influence of the number of scenarios on the expected profit

functions with $BADP-w$ and the policies (expectation model) designed in Section 4 and derive the decisions while evaluating with the stochastic program, but we do not think that the stochastic program would benefit dramatically enough to significantly outperform the expectation model.

Appendix C. Evaluating BADP

Using the value function approximation derived by $BADP$ to evaluate the induced policies works similar to Step 4, Step 11, and Step 14 in Algorithm 1. The execution logic can be found in Algorithm 5.

Algorithm 5 Evaluate BADP

```
1: initialize  $S_0^{DA}$ 
2: for  $t = 0$  to  $T - 1$  do
3:   compute expected price history  $\bar{P}_{t,t+1}^{history} := \mathbb{E} [P_{t+1}^{history} | P_t^{history}]$ 
4:    $(\bar{\rho}_1, \dots, \bar{\rho}_N) := \arg \min_{\rho_1, \dots, \rho_N} \left\| \bar{P}_{t,t+1,n}^{history} - \sum_{k=1}^N \rho_k P_{t+1,k}^{history} \right\|_2$ 
5:   derive  $x_t^{DA} := X_t^{DA,\pi}(S_t^{DA})$  based on  $(\bar{\rho}_1, \dots, \bar{\rho}_N)$ 
6:   use decisions  $x_t^{DA}$  and observe  $P_{t+1}^{DA}$ 
7:   compute expected price history  $\bar{P}_{t,t+1}^{history} := \mathbb{E} [P_{t+1}^{history} | P_t^{history}, P_{t+1}^{DA}]$ 
8:    $(\bar{\rho}_1, \dots, \bar{\rho}_N) := \arg \min_{\rho_1, \dots, \rho_N} \left\| \bar{P}_{t,t+1,n}^{history} - \sum_{k=1}^N \rho_k P_{t+1,k}^{history} \right\|_2$ 
9:   derive  $x_t^{ID} := X_t^{ID,\pi}(S_t^{DA})$  based on  $(\bar{\rho}_1, \dots, \bar{\rho}_N)$ 
10:  use decisions  $x_t^{ID}$  and observe  $P_{t+1}^{ID}$ 
11: end for
```

Algorithm 5 shows how we use the value function approximation derived with *BADP* to derive decisions for the day-ahead and the intraday market. More precisely, how we evaluate the policy functions $X_t^{DA,\pi}(S_t^{DA})$ and $X_t^{ID,\pi}(S_t^{ID})$. Step 1 defines the initial state S_0^{DA} . Step 2 iterates over the optimization horizon. In Step 3, we derive the next day's expected price history conditioned on the current price history. The expected price history is regressed on the N sampled price histories (from Algorithm 1) to derive the regression parameters $(\bar{\rho}_1, \dots, \bar{\rho}_N)$ in Step 4. These are used to compute $X_t^{DA,\pi}(S_t^{DA}) = \arg \max_{x_t^{DA}} \max_{x_t^{ID}} \left\{ C_t^{policy}(S_t^{DA}, (x_t^{DA}, x_t^{ID})) + \sum_{k=1}^N \bar{\rho}_k \bar{V}_{t+1}^{policy}(S_{t+1}^{DA} | P_{t+1,k}^{history}) \right\}$ s.t. (9), (11) to (18), (23) to (24), (33). Afterwards, the decision x_t^{DA} is applied and the MCPs P_{t+1}^{DA} becomes known. With the realized day-ahead market prices, the expected price history $\bar{P}_{t,t+1}^{history}$ is updated (Step 7) and again regressed on the sampled price histories (Step 8). The decision for the intraday market is $X_t^{ID,\pi}(S_t^{ID}) = \arg \max_{x_t^{ID}} \mathbb{E} [C_t^{ID}(S_t^{ID}, x_t^{ID}, W_{t+1}^{ID}) | S_t^{ID}] + \sum_{k=1}^N \bar{\rho}_k \bar{V}_{t+1}^{policy}(S_{t+1}^{DA} | P_{t+1,k}^{history})$ s.t. (11) to (18), (23) to (24), (26). The decision x_t^{ID} is used and the intraday market prices become known.

Appendix D. Notation and parameter values

Table D.5: Notation introduced in Section 3

Indices	
t	Day
h	Hour
q	Quarter-hour
Endogenous variables	
$x_t^{DA} = (x_{t,0}^{DA}, \dots, x_{t,23}^{DA})$	Day-ahead market decision
$x_t^{ID} = (x_{t,0,0}^{ID}, \dots, x_{t,23,3}^{ID})$	Intraday market decision
S_t^{DA}	State variable in the day-ahead market stage
S_t^{ID}	State variable in the intraday market stage
R_t^{start}	Storage level at the start of day t
x_t^{start}	Inflow/outflow volume at the start of day t
$x_{t,h,q}^{turbine}, x_{t,h,q}^{pump}$	Inflow/outflow volume
$y_{t,h,q}^{turbine}, y_{t,h,q}^{pump}$	Binary working mode variable
$z_{t,h,q}^{turbine}, z_{t,h,q}^{pump}$	Start-up variable
$\Delta x_{t,h,q}^{turbine,up}, \Delta x_{t,h,q}^{turbine,down}$	Up/down ramping volume of the turbine
$\Delta x_{t,h,q}^{pump,up}, \Delta x_{t,h,q}^{pump,down}$	Up/down ramping volume of the pump
$R_{t,h,q}$	Storage level
Exogenous information	
$P_t^{DA} = (P_{t,1}^{DA}, \dots, P_{t,24}^{DA})$	Day-ahead market prices
$P_t^{ID} = (P_{t,0,0}^{ID}, \dots, P_{t,23,3}^{ID})$	Intraday market prices
$P_t^{history}$	Past prices used for price forecast
$Q_t^{over} = (Q_{t,0,0}^{over}, \dots, Q_{t,23,3}^{over})$	Payments for overproduction
$Q_t^{under} = (Q_{t,0,0}^{under}, \dots, Q_{t,23,3}^{under})$	Payments for underproduction
W_t^{DA}	Exogenous information which become known after the day ahead market stage
W_t^{ID}	Exogenous information which become known after the intraday market stage

Parameters

T	Number of time periods
$DoW^{(\cdot)}_{(\cdot)}$	Day of the week dummy
$\beta^{(\cdot)}_{(\cdot)}$	Regression parameters in price forecast
$\Sigma^{(\cdot)}$	Correlation matrix
$\epsilon^{(\cdot)}_t$	Noise
$x^{pump}_{min}, x^{pump}_{max}$	Working range of the pump
$x^{turbine}_{min}, x^{turbine}_{max}$	Working range of the turbine
R_{max}	Storage capacity
Δt^{DA}	Temporal resolution of the day-ahead market
Δt^{ID}	Temporal resolution of the intraday market
Δt^{ramp}	Ramping time
$\eta^{pump}, \eta^{turbine}$	Efficiency of the pump/turbine
c^{gf}	Grid fees
c^{pump}, c^{pump}	Start-up cost of the pump/turbine
$Q^{over}_{intercept}, Q^{under}_{intercept}$	Intercept of the linear imbalance cost
$Q^{over}_{slope}, Q^{under}_{slope}$	Slope of the linear imbalance cost
Functions	
C_t^{DA}	One-stage profit of the day-ahead market stage
C_t^{ID}	One-stage profit of the intraday market stage
$S^{M,DA}$	Transition function from the day-ahead market stage to the intraday market stage
$S^{M,ID}$	Transition function from the intraday market stage to the day-ahead market stage
$X_t^{DA,\pi}$	Policy function of the day-ahead market stage
$X_t^{ID,\pi}$	Policy function of the intraday market stage
V_t^{DA}	Value function of the day-ahead market stage
V_t^{ID}	Value function of the intraday market stage

Table D.6: Notation introduced in Section 4

Endogenous variables	
x_t^{policy}	Decision variable of the policy model
Exogenous information	
W_t^{policy}	Exogenous information of the policy model
Parameters	
$\bar{P}_{t,t+1,h}^{DA}$	Expectation of the day-ahead market prices
$\bar{P}_{t,t+1,h,q}^{ID}$	Expectation of the intraday market prices
$\bar{Q}_{t,t+1,h,q}^{over}$	Expectation of the balancing prices for overproduction
$\bar{Q}_{t,t+1,h,q}^{under}$	Expectation of the balancing prices for underproduction
Functions	
C_t^{policy}	One-stage profit of the policy model
$S^{M,policy}$	Transition function of the policy model
V_t^{policy}	Value function of the policy model
\bar{V}^{policy}	Value function approximation of the policy model
P	Transition probabilities

Table D.7: Notation introduced in Section 5

Indices	
n	Sample index
Parameters	
$P_{t,n}^{history} = (P_{t,n}^{history,1}, \dots, P_{t,n}^{history,840})$	Sample of the price history
$\bar{P}_{t,t+1,n}^{history}$	Expectation of the next price history based on the n -th sample
$\bar{P}_{t,t+1}^{history}$	Expectation of the next price history based on the realized price history
$\bar{\rho}_n$	Interpolation/extrapolation parameters
$K_{t,k}^d$	Upper/lower bounds for the Interpolation/extrapolation parameters
w_t	Vector of weights
e_j	j -th Unit vector
d	Bandwidth
N	Number of sample paths

Table D.8: Notation introduced in Section 6

Parameters	
K	Number of iterations of $ADDP$
D	Number of days in the price history

Table D.9: Parameter values for the numerical study

Parameter	Value	Parameter	Value
R_{max}	100 MWh	T	30
x_{min}^{pump}	5 MW	$x_{min}^{turbine}$	5 MW
x_{max}^{pump}	10 MW	$x_{max}^{turbine}$	10 MW
Δt^{DA}	1 h	Δt^{ID}	15 min
Δt^{ramp}	2 min	c^{gf}	5 €/MWh
c^{pump}	15 €	$c^{turbine}$	15 €
η^{pump}	0.9	$\eta^{turbine}$	0.9
$Q_{intercept}^{over}$	-3 €/MWh	Q_{slope}^{over}	$\frac{1}{1.2}$
$Q_{intercept}^{under}$	3 €/MWh	Q_{slope}^{under}	1.2
N	50	c^{gf}	5 €/MWh

Table D.10: Notation introduced in Appendix [Appendix A](#)

Indices	
b	Bid
s	Scenario
Parameters	
B	Number of bids
S	Number of scenarios
γ	Correlation factor
$\Sigma^{DA,\gamma}$	Artificial correlation matrix
$P_{t,h,b}^{buy,bid}$	Price component of the buying bid
$P_{t,h,b}^{sell,bid}$	Price component of the selling bid
$a_{t,h,s,b}^{buy}$	Indicator if buying bid is accepted
$a_{t,h,s,b}^{sell}$	Indicator if selling bid is accepted
Endogenous variables	
$x_{t,h,b}^{buy,bid}$	Volume component of the buying bid
$x_{t,h,b}^{sell,bid}$	Volume component of the selling bid
$x_{t,h,s}^{buy}$	Contracted buying volume
$x_{t,h,s}^{sell}$	Contracted selling volume
$x_{t,h,s}^{over}$	Overproduction
$x_{t,h,s}^{under}$	Underproduction