Day-Ahead Optimization of Production Schedules for Saving Electrical Energy Costs

Thomas Stueber
Chair of Communication Networks, University of Tuebingen
Tuebingen, Germany
thomas.stueber@uni-tuebingen.de

Florian Heimgaertner
Chair of Communication Networks, University of Tuebingen
Tuebingen, Germany
florian.heimgaertner@uni-tuebingen.de

Michael Menth
Chair of Communication Networks, University of Tuebingen
Tuebingen, Germany
menth@uni-tuebingen.de

ABSTRACT

The integration of weather-dependent renewable energy sources leads to an increased volatility of electrical energy supply. As a result, considerable intra-day price spreads can be observed at the spot markets for electrical energy. To benefit from variable energy prices, enterprises can use price forecasts for cost-optimized load scheduling. Thereby energy costs can be reduced by shifting energy-intensive processes to times with lower energy prices.

In this work, we propose a method to model an industrial unit including devices, storage units, dependencies, restrictions, and production targets as a mixed integer linear program (MILP). Along with a time series of energy prices, the MILP is used to compute optimal run times for the devices while complying with the specified restrictions.

We use the model of a cement plant as an example. We show potential savings compared to default schedules over individual day, weeks, or over the year 2018. We propose optimization with look-ahead, point out its benefits compared to optimization without look-ahead, and show the influence of storages sizes and price variance on the savings potential.

CCS CONCEPTS

• **Theory of computation** → **Linear programming**; • **Hardware** → **Smart grid**; • **Applied computing** → *Industry and manufacturing*.

ACM Reference Format:
Thomas Stueber, Florian Heimgaertner, and Michael Menth. 2019. Day-Ahead Optimization of Production Schedules for Saving Electrical Energy Costs. In *Proceedings of the Tenth ACM International Conference on Future Energy Systems (e-Energy '19), June 25–28, 2019, Phoenix, AZ, USA*. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3307772.3328302

1 INTRODUCTION

The large-scale integration of renewable energy sources leads to new challenges for electrical power grids and the energy market. Especially the roll-out of weather-dependent energy sources like wind turbines or photovoltaic systems leads to an increased volatility of electrical energy supply. As a result, considerable differences in energy prices can be observed at the spot markets for electrical energy within a day.

In industrial production processes, some devices can be operated at different production rates. Additionally, storage units can decouple the run times of subsequent devices within a production chain. Storage and variable production rates constitute a certain flexibility, i.e., an enterprise can reach the same production targets with the same set of devices while using different schedules. With high-quality price forecasts for the day-ahead markets, industrial enterprises can leverage flexibilities in their production processes to benefit from the variability of energy prices using cost-optimized load scheduling. Based on the forecasts, energy costs can be reduced by shifting energy-intensive processes to times with lower energy prices [4].

In this work, we propose a method for the computation of production schedules that minimize energy costs. The main contribution of our work is a comprehensive framework for modeling an industrial plant including devices, storage units, dependencies, restrictions, and production targets as a mixed integer linear program (MILP). With a time series of energy prices, the MILP computes optimized run times for the devices with the given production rates, storage parameters and restrictions.

Cement production is a prominent example for energy intensive industry, accounting for approximately 12–15% of the total industrial energy consumption [8]. As shown in Section 2.2 cement production is also widely used as a reference use case for scheduling of energy intensive processes. For our study we use the model of a cement plant described by Bazan et. al. [1] as an example to show potential savings compared to default schedules. We quantify the savings for individual days, weeks, and for the year 2018. As another contribution, we propose optimization with look-ahead for this problem and demonstrate its benefits compared to optimization without look-ahead. In addition, we show the influence of storage sizes and price variance on the savings potential. Finally, we report the computation time of our optimization approach.

This paper is structured as follows. Section 2 discusses related work. In Section 3 we describe the proposed approach for modeling of industrial production processes as MILP. Section 4 introduces the evaluation scenario and discusses optimization results. Section 5 draws conclusions and gives an outlook on further work.

e-Energy '19, June 25–28, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6671-7/19/06.
https://doi.org/10.1145/3307772.3328302

2 RELATED WORK

In this section we give an overview about related work. We first discuss general approaches for scheduling in the area of continuous production. Then, we discuss work in the area of energy-aware scheduling, with a focus on scheduling approaches specifically addressing the use case of cement production.

2.1 Continuous Scheduling

Different scheduling strategies for continous production processes are considered in literature. They can be classified by algorithmic techniques, time model and objective. Some of the approaches also support semi-continous or batch processes.

One of the first such models for short-term scheduling of the production of fast moving consumer goods was presented by Ierapetritou and Floudas [5]. The optimization is based on MILP. This model was extended for storages and used to examine properties of the optimization with different storage limitations by Mendez and Cerda [9] and later by Shaik and Floudas [13]. Neumann and Schwindt [11] propose a branch-and-bound algorithm for models with continous and semi-continous processes.

However, these models were designed for optimizing makespans and cost of production processes and do not consider energy cost or usage.

2.2 Energy-Aware Scheduling

Castro et. al. [2] explore different scheduling approaches for continuous production. They use both discrete-time and continuous-time models. Energy consumption is considered in the optimization, but variable energy prices are not used. Shrouf et. al. [14] present an energy-aware scheduling mechanism using linear programming (LP) and a discrete-time model. The optimization objective is reduced energy costs with day-ahead energy prices. However, the scheduling only considers a single device.

Kondili et. al. [7] present an optimization of schedules for whole continous production plants with varying energy prices using MILP. A cement plant is used as an example for real world applications of such models. Based on [2], Mitra et. al. [10] present a model for a cement plant which also addresses variable energy prices. More recent developments in energy market models like day-ahead markets give new objecives in scheduling such processes. Bazan et. al. [1] present a hybrid simulation approach for scheduling of energy demand in a cement plant with a wind turbine and a battery storage. They optimize energy costs using LP and a discrete time model.

Gahm et. al. [3] provide a wide overview of the field of energy-aware scheduling in manufacturing companies.

In contrast to the works mentioned above, this work focuses on the optimization potential which arises from variable energy prices in day-ahead markets. We give a modelling approach for optimizing production schedules of complex production processes instead of single machines. The approach is used to gain first insights for the potential of saving energy costs by taking advantage of flexibilities in production combined with variability of energy prices.

3 MODELING FRAMEWORK

Industrial production processes are defined by devices, storage units, energy and material flows, and technical or organizational constraints. In this paper, we propose a comprehensive framework to model production processes as a MILP that can be used for process scheduling with minimized energy costs. In this section, we first describe the main components of our abstract model for an industrial plant. Then, we explain how they are represented in the mathematical model.

3.1 Model Components

We consider production processes in industrial facilities with continuous production and develop a simple abstract model which is powerful enough to describe many relevant degrees of freedom and restrictions for scheduling. The model consists of a set of devices $\mathcal{D}$, a set of storages $\mathcal{S}$, and a set of fixed consumers $\mathcal{F}$ that are connected by material and energy flows like a directed graph. Additional constraints for scheduling are specified by a set of global restrictions $\mathcal{G}$. In the following, we describe the model components in detail.

3.1.1 Devices $\mathcal{D}$. Devices consume and produce goods and power. The input and output volumes of goods and power depend on the operation mode of the device. The run times and the operation modes constitute the degrees of freedom of our scheduling problem. Various limitations can restrict the set of possible schedules of a device, e.g., prohibited or mandatory run times, preparation and waiting times before and after runs, maximum number of runs, or minimum and maximum production within a run or during the optimization interval.

3.1.2 Storages $\mathcal{S}$. Storages store goods or energy before and after devices in the production process. They cause time dependencies in the model, increasing the complexity of the scheduling problem. However, sufficiently large storages between devices decouple their operation in time and generate scheduling flexibility. Like devices, storages are subject to a set of restrictions, e.g., minimum and maximum level, optional production targets at different points in time, and starting levels.

3.1.3 Fixed Consumers $\mathcal{F}$. Fixed consumers are unscheduled parts of the production process. They can describe constant energy demands and supply of goods needed for production. They can be active only at a specific point in time or during longer time intervals. Fixed energy demands increase the energy costs only by a constant addend, but can be important for compliance with global restrictions.

3.1.4 Global Restrictions $\mathcal{G}$. Global restrictions are constraints that cannot be specified as a property of a single device, e.g., restrictions that apply to multiple devices at the same time. The model currently supports mutual exclusion of arbitrary subsets of devices and global energy and power limits.

3.2 The Mathematical Model

The general problem of computing a schedule for a given production process and a time series of energy prices with minimum energy cost is NP-hard. We provide a proof for that in Appendix A. This

Table 1: Global parameters.

Parameter	Definition
$\mathcal{T}, l$	Index set of all time slots (numbers 1 to $\lvert T \rvert$) and length of a time slot
$\mathcal{D}, \mathcal{S}, \mathcal{F}, \mathcal{G}$	Sets of devices, storages, fixed consumers, global restrictions
c_t	Energy price in time slot t

Table 2: Variables.

Variable	Type	Definition
$x^d_{t,m}$	binary	Does device d run in time slot t with mode m?
s^d_t	binary	Does a run of device d start at the beginning of time slot t?
e^d_t	binary	Does a run of device d end at the end of time slot t?
$k^d_{t,s}$	real	Cumulative production of the current run of device d after time slot t for output storage s.
f^s_t	real	Fill level of storage s at the end of time slot t.
$r^d_{t,m}$	real	Production rate of device d in continuous mode m during time slot t.

property makes the problem very unlikely to be solved by algorithms with polynomial runtime. Therefore, we use MILP to solve the problem although MILP solving algorithms have exponential runtime in general.

In the following, we present global parameters, variables, restrictions, and the objective function of our MILP and discuss its design.

3.2.1 *Global Parameters.*

The MILP computes an optimized schedule for an optimization interval. Like in other MILP models for similar problems [1, 2, 14], the optimization interval is divided into a set of time slots $\mathcal{T}$ and all time slots $t \in \mathcal{T}$ have the same duration l. However, the latter can be easily relaxed. Energy price forecasts take a fixed value during time slots and are given by c_t. Table 1 summarizes all global parameters of the MILP.

3.2.2 *Variables.*

Every device d is modeled with three binary variables and one continuous variable per time slot t. The binary variable $x^d_{t,m}$ indicates whether a device d runs during time slot t. Moreover, the continuous variable $r^d_{t,m}$ indicates the rate of device d when it works in operation mode m in time slot t.

The binary variable s^d_t indicates whether a run of a device d begins at the start of the time slot t. The binary variable e^d_t indicates whether a run of a device d ends at the end of time slot t. The variable $r^d_{t,m}$ indicates the operation mode m of a device d in time slot t. The continuous variable $k^d_{t,s}$ captures the cumulative production of device d for its connected storage s from the beginning of its run until the end of the current time slot t. The continuous variable f^s_t captures the fill state of storage s at the end of time slot t. Table 2 gives a summary of the used variables.

3.2.3 *Restrictions.*

We first discuss implicit restrictions of our model and then explicit restrictions for devices, storages, fixed consumers, and global restrictions, which were all mentioned in Section 3.1 that are enforced by additional inequalities.

Implicit Restrictions. An essential restriction of our model is that devices run in the same operation mode during a time slot. This limitation facilitates modeling of storages. Their fill states at the end of a time slot can be computed from the level at the beginning of the same time slot and the sum of all devices which charge or discharge the storage during that slot. Furthermore, it facilitates a simple restriction for minimum and maximum fill states. As devices run for entire time slots with constant rates, storage levels are increased or decreased linearly during a time slot. Therefore, ensuring minimum and maximum fill levels at the ends of all time slots is sufficient to comply with restrictions also within time slots.

Technical Restrictions. Some inequalities are needed to enforce the semantics of the variables mentioned in Section 3.2.2. They are presented in Appendix B as they are of technical nature and are not used to model features.

Device Restrictions. Devices may be connected to several storages from which they receive input or to which they deliver output. We denote the set of all storages of a device d, to which it delivers output, by O^d. The restrictions for a device d require parameters given in Table 3 and are expressed as follows:

$$\sum_{t \in T} s^d_t \leq n^d_{start} \tag{1}$$

$$\forall o \in O^d : w^d_{o,min} \leq \sum_{t \in \mathcal{T}} l \cdot \left(\sum_{m \in \mathcal{M}_s} x^d_{t,m} \cdot m(o) + \sum_{m \in \mathcal{M}_c} r^d_{t,m} \cdot \text{eff}^d_{o,m} \right) \leq w^d_{o,max} \tag{2}$$

$$\forall s \in O^d \forall t \in \mathcal{T} : k^d_{t,s} \leq v^d_{s,max} \tag{3}$$

$$\forall s \in O^d \forall t \in \mathcal{T} : \lvert \mathcal{T} \rvert \cdot l \cdot Max^d \cdot (e^d_t - 1) \leq k^d_{t,s} - v^d_{s,min} \tag{4}$$

$$\forall t \in \mathcal{T}_{must} : \sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{t,m} = 1 \tag{5}$$

$$\forall t \in \mathcal{T}_{forb} : \sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{t,m} = 0 \tag{6}$$

Inequation (1) guarantees that the number of starts, which is also the number of runs, cannot exceed n^d_{start}. While Inequation (2) holds, the global production for every output fullfills the demanded minimum and maximum production. Inequation (3) ensures that the cumulative production of a run stays below the maximum allowed production. Inequation (4) implies that the cumulative production of the run is larger than the minimum production per run if a run finishes in the respective time slot. Mandatory and prohibited times can be encoded with (5) and (6) by forcing the respective run variables to be 0 or 1.

Table 3: Device parameters.

Parameter	Definition
n^d_{start}, c^d_{start}	Maximum number of runs and startup cost of device d
$w^d_{s,max}, w^d_{s,min}$	Maximum/minimum global production for output storage s of device d
$v^d_{s,max}, v^d_{s,min}$	Maximum/minimum production per run for output storage s of device d
$\mathcal{T}^d_{must}, \mathcal{T}^d_{forb}$	Set of time slots in which device d must/must not run
t^d_{lead}, t^d_{over}	Number of time slots device d has to wait before/after a run
$\mathcal{M}^d_s$	Set of semi-continuous modes of device d. A semi-continuous mode is a mapping of inputs/outputs of the device to production rates.
$\mathcal{M}^d_c$	Set of continuous modes of device d. A continuous mode m is a 2-tuple with m_{min}, m_{max} being the minimal/maximal production rate in this mode.
$\text{eff}^d_{s,m}$	The factor of production rate of continuous mode m to procution input/output of storage s
O^d	Set of output storages of device d
Max^d	A number which is bigger than the production to all outputs if the machines runs the whole planning horizon
P^d_m	Power input of device d in semi-continuous mode m or energy efficiency for continuous mode m

Table 4: Storage parameters.

Parameter	Definition
f^s_{min}, f^s_{max}	Minimum/maximum fill level of storage s
f^s_0	Initial fill level of storage s
f^s_{prod}	Target fill level of storage s at the end of the planning horizon
$\mathcal{I}^s, O^s$	Set of charging/discharging devices of storage s

Lead time before the run of a device can be modeled by the inequality:

$$s^d_t \leq 1 - \sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{t',m} \tag{7}$$

which must hold for all $t \in \mathcal{T}$ and all t' with $t - t^d_{lead} \leq t' \leq t - 1$. Overrun after a run can be modeled analogously.

Restrictions for Storages. Storages are charged and discharged by devices or fixed consumers. The set of all devices charging a storage is its set of input devices $\mathcal{I}^s$ and the set of all devices discharging it is its set of output devices O^s. Fixed consumers may be used to model constant charging or discharging of a storage.

Table 5: Parameters for fixed consumers.

Parameter	Definition
$\mathcal{T}^F_{con}$	Set of time slots, in which fixed consumer F is active
R^F_S	Consumption rate of fixed consumer F from storage S
P^F	Power input of fixed consumer F

Table 6: Parameters for global restrictions.

Parameter	Definition
$\mathcal{T}^R_{lim}$	Set of time slots, in which global restriction R must hold
$E^R_{max}, P_{max}(R)$	Maximum amount of energy or peak power for global restriction R
$\mathcal{D}^R$	Set of devices, of which at most can run at the same time by global restriction R

The new fill level of a storage s at the end of a time slot t is given by

$$\begin{aligned}\forall t \in \mathcal{T} : f^s_t &= f^s_{t-1} \\ &+ \sum_{i \in \mathcal{I}^s} \left(\sum_{m \in \mathcal{M}^i_s} l \cdot m(s) \cdot x^i_{t,m} + \sum_{m \in \mathcal{M}^i_c} l \cdot r^i_{t,m} \cdot \text{eff}^i_{s,m} \right) \\ &- \sum_{o \in O^s} \left(\sum_{m \in \mathcal{M}^o_s} l \cdot m(s) \cdot x^o_{t,m} + \sum_{m \in \mathcal{M}^o_c} l \cdot r^o_{t,m} \cdot \text{eff}^o_{s,m} \right) \\ &- \sum_{F \in \mathcal{F} : t \in \mathcal{T}^F_{con}} R^F_s .\end{aligned} \tag{8}$$

The equation considers the charging of a storage by all its input devices, the discharging by all its output devices, and the charging or discharging by all fixed consumers that are active in time slot t. If an active fixed consumer F does not charge or discharge a storage s, this is expressed by $R^F_s = 0$.

Storage level and production targets can be expressed by the following inequality:

$$f^s_{|T|} \geq f^s_{prod} \tag{9}$$

$$\forall t \in \mathcal{T} : f^s_{min} \leq f_t \leq f^s_{max} \tag{10}$$

Global Restrictions. A global restriction R enforcing a maximum of used energy (11), maximum peak power (12), or mutual exclusion of at most k devices of a set of devices (13) is implemented, depending on the respective type, with the following inequations:

$$\sum_{t\in\mathcal{T}^R_{lim}} \sum_{d\in\mathcal{D}} \left(\sum_{m\in\mathcal{M}^d_s} x^d_{t,m} \cdot P^d_m \cdot l + \sum_{m\in\mathcal{M}^d_r} r^d_{t,m} \cdot P^d_m \cdot l \right) \le E^R_{max} \tag{11}$$

$$t \in \mathcal{T}^R_{lim} : \sum_{d\in\mathcal{D}} \left(\sum_{m\in\mathcal{M}^d_s} x^d_{t,m} \cdot P^d_m + \sum_{m\in\mathcal{M}^d_r} r^d_{t,m} \cdot P^d_m \right) \le P^R_{max} \tag{12}$$

$$t \in \mathcal{T}^R_{lim} : \sum_{d\in\mathcal{D}^R} \sum_{m\in\mathcal{M}^d_s\cup\mathcal{M}^d_r} x^d_{t,m} \le k \tag{13}$$

3.2.4 Objective Function. The objective is to minimize the cost function, while complying with all the restrictions of the model, which is given by

$$\sum_{F\in\mathcal{F}} \sum_{t\in\mathcal{T}^F_{con}} P^F \cdot c_t \cdot l + \sum_{t\in\mathcal{T}} \sum_{d\in\mathcal{D}} s^d_t \cdot c^d_{start} + \sum_{m\in\mathcal{M}^d_s} x^d_{t,m} \cdot c_t \cdot P^d_m \cdot l + \sum_{m\in\mathcal{M}^d_c} r^d_{t,m} \cdot c_t \cdot P^d_m \cdot l \tag{14}$$

3.2.5 Discussion. In contrast to most other works on this field, we specified the model using only binary and real valued variables. Current MILP solvers benefit from this simplification due to the effectiveness of cut generation and the benefits of binary variables in branch-and-bound algorithms. Therefore, our MILP formulation can be solved rather efficiently by appropriate solvers.

The rates of a device d are preferentially modeled with a single mode and a continuous rate $r^d_{t,m}$ for each time slot t or with multiple modes and a fixed rate for each mode.

4 EVALUATION

In this section, we evaluate the effect of our proposed optimization method. To that end, we first consider a model of a cement plant from literature and real day-ahead energy prices. For comparison purposes, we compute a default schedule which is optimized for constant energy prices. We illustrate the nature of a schedule and its impact on storages. We study schedules optimized for variable energy prices, show that they reduce energy costs compared to default schedules, in particular if sufficient scheduling flexibility is available, and that they exhibit more variable storage levels. Then, we introduce optimization with look-ahead and demonstrate that it can further reduce energy costs. We also show that the savings potential depends on storage sizes and energy price variability. Finally, we report on the runtime performance of the optimization programs.

4.1 Model Description

We describe the model of the production process and the energy prices used for the optimization case study.

4.1.1 Cement Plant Model. Our evaluation is based on the cement plant model described in [1]. Its complexity is rather low. Therefore, only a subset of the features of our modeling framework needs to be leveraged. However, cement production is an energy-intensive process, for which scheduling optimization may be very helpful and effective to save energy costs.

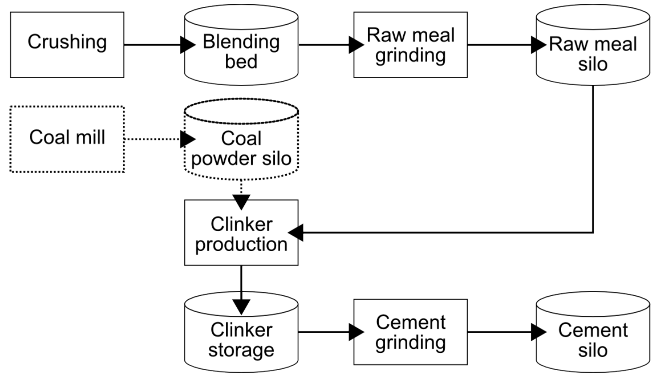


Figure 1: Process of a cement plant [1].

Table 7: Device parameters [1].

Device	Electrical demand	Material efficiency	Maximum Rate
Crusher	0.0016 MWh/t	1	200 t/h
Raw mill	0.01 MWh/t	0.8	200 t/h
Clinker production	0.017 MWh/t	1.52	95 t/h
Grinder	0.033 MWh/t	0.95	200 t/h

Table 8: Storage parameters [1].

Storage	Min. level	Max. level
Blending bed	200 t	1 800 t
Raw meal silo	200 t	1 800 t
Clinker storage	2 000 t	18 000 t
Cement silo	2 000 t	18 000 t

An overview of the cement plant model is depicted in Figure 1. The raw material is crushed and stored in a blending bed. From there it is ground by a raw mill and filled into the raw meal silo from where it is taken for clinker production. In the last step, the cement is ground from the material in the clinker silo and stored in a cement silo where it can be picked up on demand.

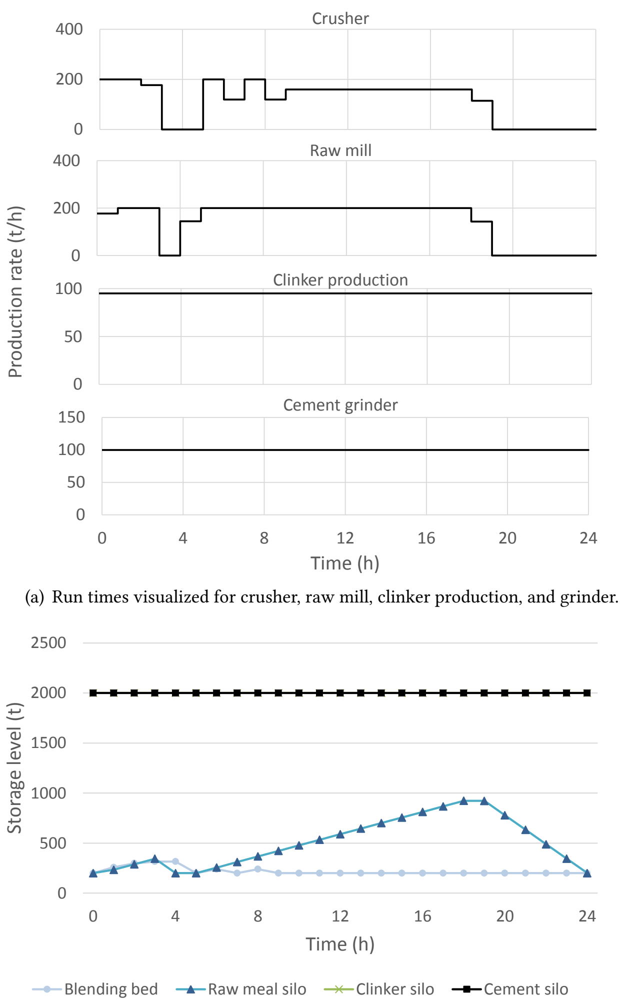
The parameters of these devices and storages are compiled in Tables 7 and 8. Table 7 shows the energy needed by a device to produce one ton of its output material. Crusher, raw mill, and grinder have variable, continuous production rates with a maximum of 200 t/h. The grinder is the most energy-intensive process followed by the clinker production. However, the latter must continuously run at its maximum rate so that it does not provide any scheduling flexibility. The raw mill and especially the crusher require substantially less energy. The table also shows the material efficiencies of the devices, i.e., the factor of how much input material is needed to produce a specific amount of output material. Instead of a production target for the entire factory, a constant cement demand of

100 t/h is specified. Detailed technical information about the coal mill and the coal powder silo are not provided in [1]. Therefore, we omit that part of the process in the evaluation.

We model the cement plant using time slots with a duration of 1 hour because price forecasts are available on a hourly base. This granularity is sufficient to compute the minimum energy costs as rates are assumed to be continuous in our model. Shorter or variable-length time slots would not increase the optimization potential.

4.2 Energy Prices

Instead of price forecasts, our evaluation is based on historical day-ahead prices. We use spot market prices of the year 2018 provided by Nord Pool [12] for the day-ahead market in western Denmark (DK1).



(a) Run times visualized for crusher, raw mill, clinker production, and grinder.

(b) Time-dependent fill states of blending bed, raw meal silo, clinker silo, and cement silo.

Figure 2: Default schedule obtained from optimization for constant energy prices.

4.3 Default Schedule

We derive a default schedule which is optimized for constant energy prices. To that end, we choose a constant energy price of 30 €/MWh. For times between 7 pm and 7 am we add a penalty of 10 €/h and running device to respect increased operation costs due to shift work at night. This is only needed to obtain a schedule which preferentially runs between 7 am and 7 pm and has no impact on real energy costs. Based on this input, an optimized default schedule is computed. The real energy cost of the default schedule are computed by taking its run times with desired energy prices into account instead of constant energy prices.

The resulting run times of the devices are illustrated in Figure 2(a). The time slots are indicated on the x-axis and the height of the bars represents the production rate in the respective time slot. The crusher and the raw mill run preferentially between 7 am and 9 pm plus in the early morning hours to get the plant working. The clinker production runs permanently as this is a model-inherent requirement. The cement grinder basically works in lock step with the clinker production because the produced clinker is just enough for the demanded cement output of the factory and there is not enough material in the clinker storage so that the grinder cannot do any advance work.

Figure 2(b) shows the corresponding fill states of the storage units in the cement plant model. All minimum storage levels are met, i.e., 200 t for blending bed and raw meal silo, and 2 000 t for clinker storage and cement silo. The blending bed is mostly filled to its minimum as crusher and raw mill almost work in lock step. The raw meal silo is filled between 7 am and 7 pm and drained afterwards. As mentioned above, clinker production and cement silo are filled only to their minimum.

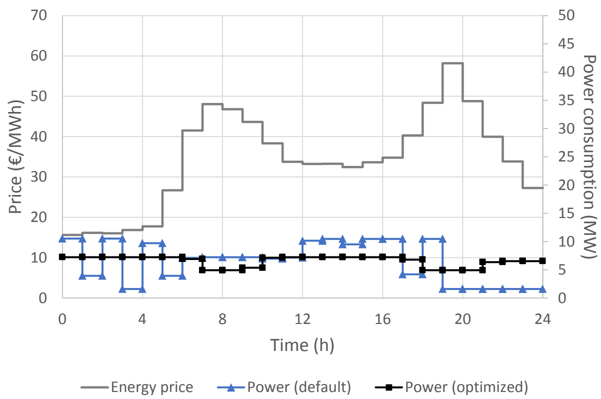
As the schedule in Figure 2(a) and the storage fill levels in Figure 2(b) are rather complex and difficult to interpret, we focus in the remainder of the paper on the fill state of the cement silo to discuss effects of different schedules. The reason for that choice is that the cement silo is filled by the grinder which is the most energy-intensive device.

4.4 Single-Day Optimization

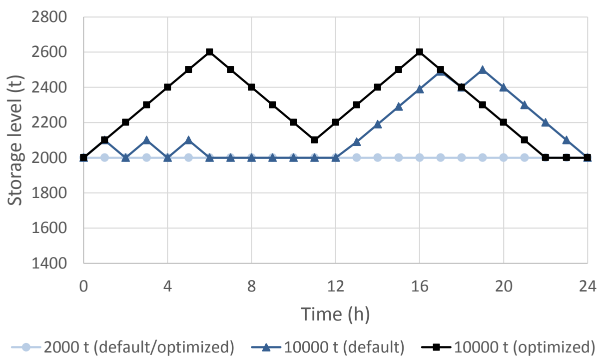
We first evaluate our optimization model for the day 2018-05-07 whose energy prices are compiled in Figure 3(a).

We presume that the plant starts with the minimum allowed fill states for all storages as indicated in the model description. We compute an optimized schedule and illustrate its effect by the fill state of the cement silo in Figure 3(b). It is exactly the same curve as the one for the default schedule. The reason for that is that the cement plant started with the minimum fill state for both the clinker storage and cement silo and that the clinker storage is constantly filled with just as much clinker as needed for the constant cement output. Therefore, both storages remain at their minimum level and the grinder cannot do any advance work but needs to work in lock step with the clinker production. As a result, there is hardly any flexibility for the schedule of the grinder. Therefore, schedule optimization is only little effective. Table 9 shows that only 3.11% of the energy costs can be saved by the optimized schedule.

It is very unfortunate that the most cost-intensive device of the production process cannot be moved in time. To avoid that



(a) Energy prices and power consumption for a start fill state of the clinker storage of 10 000 t.



(b) Storage fill states of the cement silo; the start fill state of the clinker storage is indicated; optimization is without look-ahead.

Figure 3: Investigation of various default and optimized schedules for 2018-05-07 with different start fill states for the clinker storage.

Table 9: Energy costs and savings for two different fill states of the clinker storage at start; optimization is without look-ahead.

Fill state	Schedule	Total cost	Abs. sav.	Rel. sav.
2 000 t	Default	5 365.22 €	–	–
2 000 t	Optimized	5 198.30 €	166.92 €	3.11%
10 000 t	Default	5 232.29 €	–	–
10 000 t	Optimized	4 494.37 €	737.92 €	14.10%

phenomenon, we now consider the model with a start fill state of 10 000 t for the clinker storage and compute a new default and optimized schedule. Figure 3(b) shows that the default schedule now fills the cement silo before 7 pm just as much that the grinder does not need to run anymore until the end of the day. This also reduces the energy cost for the default schedule by 2.48%. However, this improvement is rather by chance as the grinder accidentally runs at cheaper times, which may be different on another day.

Table 10: Energy costs and savings for the week from 2018-03-03 until 2018-03-09.

Schedule	Total cost	Abs. sav.	Rel. sav.
Default	46 833.14 €	–	–
Opt. w/o look-ahead	42 717.43 €	4 115.71 €	8.79%
Opt. w/ 1 day look-ahead	41 452.72 €	5 380.42 €	11.49%
Opt. w/ 2 days look-ahead	41 077.39 €	5 755.75 €	12.29%
Opt. w/ 6 days look-ahead	41 000.44 €	5 832.70 €	12.45%

In contrast, the optimized schedule intentionally leverages cheap times in the morning and in the afternoon to run the grinder filling the cement silo. Again, the fill state of the cement silo returns to its allowed minimum at the end of the day. Table 9 shows that the optimized schedule now saves 14.10% compared to the new and cheaper default schedule.

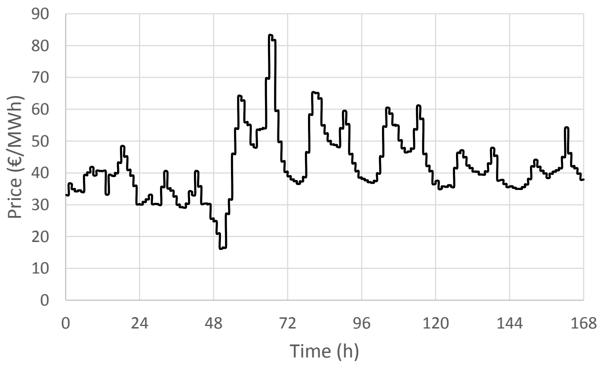
Figure 3(a) visualizes the power consumption for a start fill state of the clinker storage of 10 000 t for the default schedule and the optimized schedule. Based on this information, energy is bought at the day-ahead market. The time-dependent power consumption shows the effect of the schedule on required energy. The optimized schedule reduces power consumption when energy prices are highest, which leads to lowest total energy cost according to Table 9. However, optimization can reduce power consumption only to a certain extent during times of high energy prices as conditions like production goals and bounds for storage levels at the end of the day must be met.

4.5 Optimization with Look-Ahead

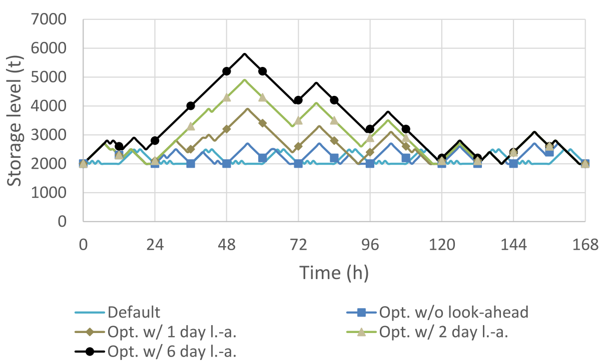
We observe that the single-day optimization leads to minimum storage fill states at the end of the day. When planning consecutive days, this implies that work cannot be done a day in advance even if price forecasts indicate more expensive prices the day after. To get rid of this artificial restriction, we introduce optimization with look-ahead. That means, when we plan the schedule for the next day, we consider n additional days for the optimization, i. e., the optimization looks ahead into the future. However, only the schedule for the next day is taken as a result and the fill states at the end of that day are used as start states for optimizing the day after using the same method. Thus, day-ahead optimization with look-ahead requires energy price forecasts for $n + 1$ days as well as predicted storage fill levels for the end of the current day.

In the following, we evaluate the benefit of optimization with look-ahead based on an interval of one week and one year, respectively. We apply the method as follows. If we apply optimization with a look-ahead of n days and there are only $m < n + 1$ days left in the interval, then we reduce the look-ahead to $m - 1$ days. This is needed to avoid unnecessarily large storage levels and energy prices at the end of the considered interval.

4.5.1 Evaluation over One Week. We evaluate optimization with look-ahead based on the energy prices of the week from 2018-03-03 until 2018-03-09 which are depicted in Figure 4(a). Figure 4(b) illustrates the fill states of the cement silo for the default schedule and for schedules optimized with a look-ahead of 0, 1, 2, and 6 days.



(a) Energy prices.



(b) Fill states of the cement silo for schedules optimized with look-ahead.

Figure 4: Investigation of the week from 2018-03-03 until 2018-03-09.

The default schedule results in periodic fill states. The schedule optimized without look-ahead also exhibits a daily pattern but with different peaks. In contrast, schedules optimized with 1 or more days look-ahead can fill storages to a larger extent and keep them filled for longer time than a single day. The maximum fill level increases with the size of the look-ahead. In all cases, the cement silo is drained to its minimum fill state at the end of the week. This is a required feature as any larger fill states would imply more energy consumed by the grinder.

Table 10 compiles the total energy costs for the week from 2018-03-03 until 2018-03-09. It shows them for the default schedule and for schedules optimized with a different numbers days look-ahead. We observe that optimization without look-ahead reduces the energy costs compared to the default schedule significantly (8.8%). Optimization with 1 or more days look-ahead reduces energy costs even further (11.5%–12.5%). However, energy price forecasts for more than 2 days are less precise, which would degrade the quality of the planning result in practice. Thus, taking only the near future into account makes the method less susceptible to forecast errors. Therefore, we consider only 1 day look-ahead from Section 4.6 on.

4.5.2 Evaluation over One Year. We now apply optimization with look-ahead to the energy prices of the entire year 2018 to assess the benefit of the method in the long run. Table 11 compiles the total energy costs for that year. We observe an energy cost reduction of about 8.0% for single-day optimization and about 10.5%–11.9% for optimization with look-ahead.

Table 11: Energy costs and savings through schedule optimization for the year 2018.

Schedule	Total cost	Abs. sav.	Rel. sav.
Default	2 524 375.50 €	–	–
Opt. w/o look-ahead	2 323 047.60 €	201 327.89 €	7.98%
Opt. w/ 1 day look-ahead	2 258 921.07 €	265 454.43 €	10.52%
Opt. w/ 2 days look-ahead	2 242 216.13 €	282 159.38 €	11.18%
Opt. w/ 6 days look-ahead	2 224 976.50 €	299 399.00 €	11.86%

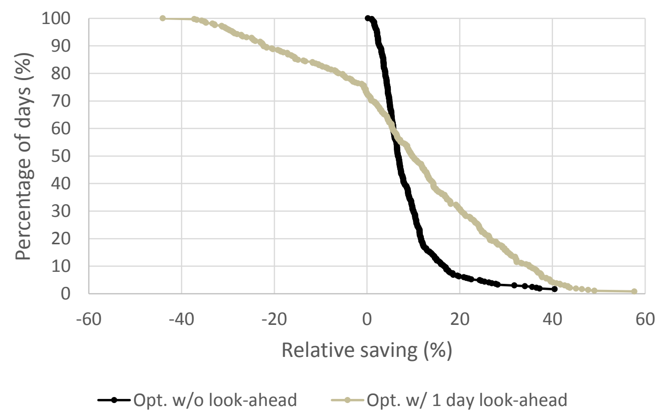


Figure 5: Daily energy savings in 2018 for schedules optimized with and without look-ahead relative to the energy costs of default schedules.

To better understand the effect of schedules optimized with and without look-ahead, we compare their energy costs to the one of default schedules on individual days. To that end, we study relative energy savings per day. Figure 5 quantifies how they are distributed for the days of the year 2018. The figure indicates the percentage of days with relative energy savings larger than a given value. For optimization without look-ahead, energy is saved on any day of the year. In 30% of the days, energy savings are larger than 10%. In contrast, for optimization with 1 day look-ahead, 28% of the days exhibit slightly larger energy costs than those of the default schedule, but 49% of the days have energy costs that are 10% cheaper than those of the default schedule. The reason for increased energy costs on some days is that work is carried out in advance due to cheap energy prices, i.e., electricity demand is shifted across day boundaries. We omit the energy savings curves for 2 and 6 days look-ahead as they are very similar to the one for 1 day look-ahead.

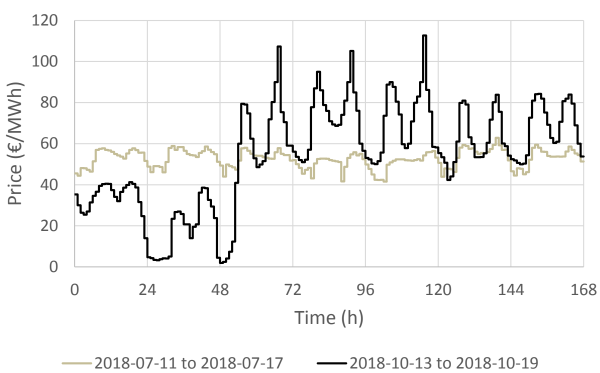
4.6 Impact of Storage Sizes

Optimization potential depends on how much energy consumption can be shifted over time. It depends on scheduling flexibility which is limited through storage sizes. To underline this proposition, we reduce the maximum fill level for all storages to the maximum values that were taken under the default schedule. Those are 320 t for the blending bed, 922 t for the raw meal silo, 10 000 t for the clinker storage, and 2 500 t for the cement silo.

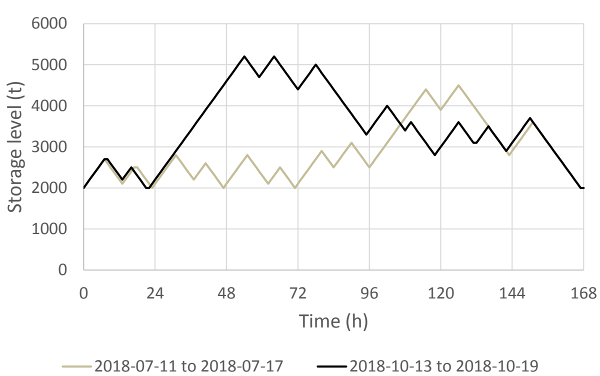
Table 12: Energy costs and savings through schedule optimization for the year 2018 depending on the storage sizes; optimization leverages a look-ahead of 1 day.

Storage	Schedule	Total cost	Abs. sav.	Rel. sav.
Standard	Default	2 524 375.50 €	—	—
Standard	Optimized	2 258 921.07 €	265 454.43 €	10.52%
Reduced	Optimized	2 333 514.02 €	190 861.48 €	7.56%

Table 12 compiles the total energy costs for 2018, for different storage sizes, and for different schedules. By construction of the experiment, reducing the storage sizes does not change the energy costs of the default schedule. For schedules optimized with 1 day look-ahead, the energy costs can be now reduced only by 7.56% for the smaller storage limits instead of 10.52% for the normal storage limits compared to the default schedule. Thus, storage sizes can significantly limit the optimization potential for energy costs.



(a) Energy prices for the weeks with the smallest and largest standard deviation in 2018: week from 2018-07-11 until 2018-07-17 (std. dev. 4.41) and week from 2018-10-13 until 2018-10-19 (std. dev 24.98).



(b) Fill states of the cement silo for both weeks for schedules optimized with 1 day look-ahead.

Figure 6: Impact of energy price variability on optimal schedules.

Table 13: Energy costs and relative savings through optimized schedules for the weeks from 2018-07-11 until 2018-07-17 (low variability, std. dev. 4.41) and from 2018-10-13 until 2018-10-19 (high variability, std. dev. 24.98); optimization leverages a look-ahead of 1 day.

Week	Schedule	Total cost	Abs. sav.	Rel. sav.
Low variability	Default	57 698.34 €	—	—
	Optimized	55 520.07 €	2 178.27 €	3.76%
High variability	Default	58 534.67 €	—	—
	Optimized	46 994.13 €	11 540.54 €	19.72%

Table 14: Computation times for different optimization intervals.

Optimization interval	Variables	Avg. computation time
1 day	652	313 ms
2 days	1 300	606 ms
3 days	1 948	1 009 ms
7 days	4 540	5 729 ms

4.7 Impact of Energy Price Variability

It is obvious that constant energy prices do not offer any potential for energy cost reduction when optimizing schedules based on energy prices. The potential for energy cost reduction obviously depends on energy price variability. To obtain an impression of the savings potential in the presence of realistic energy prices with low and high variability, we choose the weeks from 2018 with least and most variable energy prices. Their energy costs are given in Figure 6(a). The week from 2018-07-11 until 2018-07-17 has energy prices with a standard deviation of 4.41 € while the week from 2018-10-13 until 2018-10-19 has energy prices with a standard deviation of 24.98 €. We optimize schedules using 1 day look-ahead. Figure 6(b) visualizes them by the fill states of the cement silo. Both curves fluctuate as storages are filled during times of low energy cost, no matter how strong the price variability is. However, looking at Table 13, we recognize that energy costs of the default schedule can be reduced only by 3.76% for the week with little variability while they can be reduced by 19.72% for the week with high variability. Thus, if future energy prices will be more variable than today, we can expect from schedule optimization larger benefits over the year than evaluated for 2018.

4.8 Performance Considerations

The presented case study was executed on an Intel Core i5-8250U CPU @ 1.60 GHz, using 8 virtual cores and 16 GB memory. The software for constructing the models and invoking the mathematical programming solver was implemented in Java 8. We used IBM CPLEX 12.8.0.0 to solve the MILPs. The solving process needed approximately 1 GB memory for all case studies. The computation time for constructing a model and solving it using CPLEX increased with the duration of the optimization interval as the number of variables increased. This is illustrated in Table 14. Longer look-ahead requires longer optimization intervals and more variables. However, we recommend to utilize only 1 day look-ahead as this is good tradeoff between savings improvements and precise energy price forecasts.

We point out that the presented computation times are specific to the considered case study. They may be significantly larger when the model is more complex to optimize.

5 CONCLUSION

In this work, we proposed a comprehensive framework to model an industrial plant including devices, storage units, dependencies, restrictions, and production targets for the purpose of energy cost reduction. We formulated the optimization program as a MILP. For a time series of day-ahead energy prices, the MILP computes optimal run times for the devices to minimize energy costs. To demonstrate the applicability of the proposed framework, we modeled a cement plant from the literature [1] and computed optimal schedules based on real day-ahead energy prices.

The results showed that this method works, that storages need to have appropriate fill states, and that 8% of the energy costs could be saved in 2018. We proposed optimization with look-ahead to cope with the problem that empty storages at the end of the next day may be counterproductive for the planning of the day after. It essentially extends the optimization interval but leverages only the planning for the next day which then may have non-empty storages at its end. We showed that this approach can utilize large storages to a larger extent and over a longer duration than optimization without look-ahead. It allowed improved energy cost reduction of 10.5%–11.9% in 2018, depending on the duration of the look-ahead. In addition, we showed that the optimization potential depends on storage sizes and energy price variability. The run time for the MILP was rather short, mostly below 1 second although a large number of variables were required.

From these results, we conclude that energy-intensive enterprises can save considerable energy costs using the proposed schedule optimization when purchasing energy from day-ahead markets with highly variable energy prices.

Future work encompasses the modeling and optimization of more complex plants. In particular, we will extend our model to account for time- or mode-dependent operating costs for devices which may reflect, e.g., shift work at night, and other additional costs. Additional costs can influence optimal schedules as they should lead to least overall costs. In our case study, additional costs were not taken into account to lack of information in the model from literature. Criteria to predict scheduling flexibility and optimization complexity may be helpful for efficient modeling and optimization. Furthermore, storage dimensioning and appropriate start states to leverage flexibilities for energy cost reduction may be an issue. Additionally, the robustness of the proposed solution regarding forecast errors will be investigated.

ACKNOWLEDGMENTS

The research leading to these results received funding from the German Federal Ministry for Economic Affairs and Energy under the ZIM programme (Zentrales Innovationsprogramm Mittelstand), grant no. 16KN039521. The authors alone are responsible for the content of this paper.

The authors thank Bernd Thomas and Uwe Ziegler for valuable feedback as well as Niels Schieber and Dominik Kriese for initial models and heuristic algorithms.

REFERENCES

[1] Peter Bazan, David Steber, and Reinhard German. 2017. Hybrid Simulation and Energy Market Based Optimization of Cement Plants. *Computer Science - Research and Development* 32, 1 (2017), 65–77.

[2] Pedro M. Castro, Iiro Harjunkoski, and Ignacio E. Grossmann. 2011. Optimal Scheduling of Continuous Plants with Energy Constraints. *Computers & Chemical Engineering* 35, 2 (2011), 372–387.

[3] Christian Gahm, Florian Denz, Martin Dirr, and Axel Tuma. 2016. Energy-efficient scheduling in manufacturing companies: A review and research framework. *European Journal of Operational Research* 248, 3 (2016), 744–757.

[4] Florian Heimgaertner, Uwe Ziegler, Bernd Thomas, and Michael Menth. 2018. A Distributed Control Architecture for a Loosely Coupled Virtual Power Plant. In *Proceedings of the ICE/IEEE International Technology Management Conference (ICE/IEEE ITMC)*.

[5] Marianthi G. Ierapetritou and Christodoulos A. Floudas. 1998. Effective Continuous-Time Formulation for Short-Term Scheduling. 2. Continuous and semicontinuous Processes. *Industrial & engineering chemistry research* 37, 11 (1998), 4360–4374.

[6] Richard M. Karp. 1972. Reducibility Among Combinatorial Problems. In *Proceedings of a symposium on the Complexity of Computer Computations*. Yorktown Heights, New York, USA, 85–103.

[7] Emilia Kondili, Nilay Shah, and Constantinos C. Pantelides. 1993. Production Planning for the Rational Use of Energy in Multiproduct Continuous Plants. *Computers & Chemical Engineering* 17 (1993), S123–S128.

[8] Naseer Abboodi Madlool, Rahman Saidur, M. Shouquat Hossain, and Nasrudin Abd Rahim. 2011. A Critical Review on Energy Use and Savings in the Cement Industries. *Renewable and Sustainable Energy Reviews* 15, 4 (2011), 2042–2060.

[9] Carlos Alberto Mendez and Jaime Cerda. 2002. An Efficient MILP Continuous-Time Formulation for Short-Term Scheduling of Multiproduct Continuous Facilities. *Computers & Chemical Engineering* 26, 4-5 (2002), 687–695.

[10] Sumit Mitra, Ignacio E. Grossmann, Jose M. Pinto, and Nikhil Arora. 2012. Optimal Production Planning under Time-Sensitive Electricity Prices for Continuous Power-Intensive Processes. *Computers & Chemical Engineering* 38 (2012), 171–184.

[11] Klaus Neumann, Christoph Schwindt, and Norbert Trautmann. 2005. Scheduling of Continuous and Discontinuous Material Flows with Intermediate Storage Restrictions. *European Journal of Operational Research* 165, 2 (2005), 495–509.

[12] Nord Pool AS. 2018. Historical Market Data. https://www.nordpoolgroup.com/historical-market-data/.

[13] Munawar A Shaik and Christodoulos A Floudas. 2007. Improved Unit-Specific Event-Based Continuous-Time Model for Short-Term Scheduling of Continuous Processes: Rigorous Treatment of Storage Requirements. *Industrial & engineering chemistry research* 46, 6 (2007), 1764–1779.

[14] Fadi Shrouf, Joaquin Ordieres-Meré, Alvaro García-Sánchez, and Miguel Ortega-Mier. 2014. Optimizing the Production Scheduling of a Single Machine to Minimize Total Energy Consumption Costs. *Journal of Cleaner Production* 67 (2014), 197–207.

APPENDIX

A PROOF OF NP-HARDNESS

To proof the NP-hardness of the considered scheduling problem, it must first be formulated as a decision problem.

Definition 1 (Energy-Scheduling).
Given: *A system of devices, storages, fixed consumers, and restrictions, defined as in Section 3, given by the respective parameters, energy prices for the entire planning horizon, and maximal cost* $C \in \mathbb{Z}$.
Question: *Is there a schedule which fullfils all restrictions imposed by the given model while inducing costs of at most* C*?*

The following problem is also needed for our proof.

Definition 2 (Knapsack).
Given: *Objects* $(w_1, v_1), ..., (w_n, v_n)$*, consisting of weight and value,*

a maximum weight $W \in \mathbb{N}$ and a minimum value $N \in \mathbb{N}$
Question: *Is it possible to choose a subset of objects such that the sum of the weights of its elements does not exceed W while the sum of the values of its elements is at least N?*

Knapsack is a well known NP-complete problem, a fact which was first proven by Karp [6].

Theorem A.1. *Energy-Scheduling is NP-hard.*

Proof. By giving a polynomial-time reduction from Knapsack to Energy-Scheduling, NP-completeness of Energy-Scheduling can be proven. Let $(w_1, v_1), \ldots, (w_n, v_n)$ be an instance of Knapsack. Construct one device per object (w_i, v_i). This device has only one continuous mode. The maximum rate of this mode is normalized such that the device will produce v_i units of end products if the device runs for the entire optimization interval. The energy consumption of the mode is also normalized such that w_i units of energy are consumed if the device runs for the entire optimization interval. The minimal production per run is set to v_i. All these devices are connected to one common storage. The production target of this storage is set to N. Energy prices are set to one unit during the entire optimization interval. The maximum cost C is set to W.

If there is a subset of objects which fulfills the requirements of Knapsack, there is a schedule for the constructed model with costs of at most W. Such a schedule can be constructed by letting work the respective devices of the objects contained in the subset for the entire optimization interval. All other devices do not work at all. Through the normalization of production rates corresponding to the values of the respective objects, the production target of the common storage is fulfilled. By the same argument, the cost induced by this schedule is at most W.

For the contrary, suppose there is no subset of objects with the needed requirement, but there is a schedule for the constructed model which fulfills the production target and maximum cost restriction. It is implied by the minimal production per run that a device can only work for the whole planning horizon or not at all. By taking the corresponding objects of the running devices in this schedule, one gets due to the normalization of rates and energy demands a subset of objects with a sum of weights of at most W and a sum of values of at least N. This contradicts the assumption that there is no such subset, so there cannot be such a schedule for the constructed model. To see the polynomial run time of this construction, observe that only one device is constructed per object with one additional common storage and C is just a copy of W. So the construction is indeed a polynomial-time reduction from Knapsack to Energy-Scheduling, which completes the proof of NP-hardness. □

If there is an algorithm which computes the optimal schedule in polynomial time, it could be used to decide Energy-Scheduling, which would imply the commonly as unlikely seen statement of $P = NP$.

B MILP-REPRESENTATION OF ADDITIONAL CONSTRAINTS

Auxiliary parts of the MILP are presented in this section. They enforce the intended semantics of the variables presented in Table 2.

In every valid assignment of variables, for a device d must hold that start- and end-of-run variables, which are set to 1, must alternate. Additionally, the first of these variables, which is set to 1, must be a start-of-run variable while the last one has to be an end-of-run variable. The following inequalities implement these restrictions.

$$\forall t \in \mathcal{T} : 0 \leq \sum_{t' \in \mathcal{T},\ t' \leq t} s^d_{t'} - \sum_{t' \in \mathcal{T},\ t' < t} e^d_{t'} \leq 1 \tag{15}$$

$$\forall t \in \mathcal{T} : \sum_{t' \in \mathcal{T},\ t' \leq t} e^d_{t'} \leq \sum_{t' \in \mathcal{T},\ t' \leq t} s^d_{t'} \tag{16}$$

$$\forall t \in \mathcal{T} : 1 + \sum_{t' \in \mathcal{T},\ t' \leq t} e^d_{t'} \geq \sum_{t' \in \mathcal{T},\ t' \leq t} s^d_{t'} \tag{17}$$

$$\sum_{t' \in \mathcal{T}} e^d_{t'} = \sum_{t' \in \mathcal{T}} s^d_{t'} \tag{18}$$

That a device can only run in at most one mode in every slot is modeled by

$$\forall t \in \mathcal{T} : \sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{t,m} \leq 1. \tag{19}$$

The semantics of the run variables for every slot demand that they are only set to 1 if and only if there is a start-of-run variable set to 1 in an earlier slot and no end-of-run variable in a slot between. Because at most one of the run variables of a single device in a given slot can be set to 1, the sum of these variables can be understood as a single binary variable itself.

$$\forall t \in \mathcal{T} : s^d_t \leq \sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{t,m} \tag{20}$$

$$\forall t \in \mathcal{T} \setminus \{1\} : -e_{t-1} + \sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{t-1,m} \leq \sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{t,m} \tag{21}$$

$$\forall t \in \mathcal{T} \setminus \{1\} : -s_t - \sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{t-1,m} \leq \sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{t,m} \tag{22}$$

$$\forall t \in \mathcal{T} \setminus \{1\} : e_{t-1} - s_t \leq 1 - \sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{t,m} \tag{23}$$

$$\sum_{m \in \mathcal{M}_s \cup \mathcal{M}_c} x^d_{1,m} \leq s^d_1 \tag{24}$$

At last, the semantics of the cumulative-production variables need to grow over a run dependent on the production rate in every time slot and should be set to 0 when a run ends. After the first slot in the optimization interval, the cumulative variable of every device should be initialized with the production of the respective device in the first slot. Let $s \in O(A)$ be an output of the device d.

$$\forall t \in \mathcal{T} \setminus \{1\} : k^d_{t,s} - l \cdot \left(\sum_{m \in \mathcal{M}_s} m(s) \cdot x^d_{t,m} + \sum_{m \in \mathcal{M}_s} r^d_{t,m} \right) \leq n \cdot l \cdot Max^d \cdot (1 - e^d_{t-1}) \tag{25}$$

$$\forall t \in \mathcal{T} \setminus \{1\} : l \cdot \left(\sum_{m \in \mathcal{M}_s} m(s) \cdot x^d_{t,m} + \sum_{m \in \mathcal{M}_c} r^d_{t,m} \right) - k^d_{t,s} \leq n \cdot l \cdot Max^d \cdot (1 - e^d_{t-1}) \tag{26}$$

$$\forall t \in \mathcal{T} \setminus \{1\} : \left(k^d_{t-1,s} + l \cdot \left(\sum_{m \in \mathcal{M}_s} m(s) \cdot x^d_{t,m} + \sum_{m \in \mathcal{M}_c} r^d_{t,m} \right) \right) - k^d_{t,s} \leq n \cdot l \cdot Max^d \cdot e^d_{t-1} \tag{27}$$

$$\forall t \in \mathcal{T} \setminus \{1\} : k^d_{t,s} - \left(k^d_{t-1,s} + l \cdot \left(\sum_{m \in \mathcal{M}_s} m(s) \cdot x^d_{t,m} + \sum_{m \in \mathcal{M}_c} r^d_{t,m} \right) \right) \leq n \cdot l \cdot Max^d \cdot e^d_{t-1} \tag{28}$$

$$\sum_{m \in \mathcal{M}_s} l \cdot m(s) \cdot x^d_{1,m} + \sum_{m \in \mathcal{M}_c} l \cdot r^d_{1,m} = k^d_{1,s} \tag{29}$$

For continuous modes, it must be enforced that the continuous rate variable is 0 if and only if the run variable is set to 0 for the respective mode in all time slots. Additionally, the rate must be within the respective bounds of the mode.

$$\forall t \in \mathcal{T} \forall m \in \mathcal{M}^d_c : r^d_{t,m} \leq m_{max} \tag{30}$$

$$\forall t \in \mathcal{T} \forall m \in \mathcal{M}^d_c : -r^d_{t,m} + m_{min} \leq m_{max} - (m_{max} \cdot x^d_{t,m}) \tag{31}$$

$$\forall t \in \mathcal{T} \forall m \in \mathcal{M}^d_c : 0 \leq r^d_{t,m} \leq m_{max} \tag{32}$$