Wavelet-based grid-adaptation for nonlinear scheduling subject to time-variable electricity prices

Pascal Schäfer[1], Artur M. Schweidtmann[1], Philipp H. A. Lenz[1], Hannah M. C. Markgraf[1], and Alexander Mitsos[2,1,3,*]

[1] RWTH Aachen University
AVT - Aachener Verfahrenstechnik
Process Systems Engineering
52072 Aachen, Germany

[2] JARA-ENERGY
52056 Aachen, Germany

[3] Forschungszentrum Jülich
Energy Systems Engineering (IEK-10)
52425 Jülich, Germany

Abstract:

Using nonlinear process models in discrete-time scheduling typically prohibits long planning horizons with fine temporal discretizations. Therefore, we propose an adaptive grid algorithm tailored for scheduling subject to time-variable electricity prices. The scheduling problem is formulated in a reduced space. In the algorithm, the number of degrees of freedom is reduced by linearly mapping one degree of freedom to multiple intervals with similar electricity prices. The mapping is iteratively refined using a wavelet-based analysis of the previous solution. We apply the algorithm to the scheduling of a compressed air energy storage. We model the efficiency characteristics of the turbo machinery using artificial neural networks. Using our in-house global solver MAiNGO, the algorithm identifies a feasible near-optimal solution with <1% deviation in the objective value within <5% of the computational time compared to a solution considering the full dimensionality.

Keywords

Discrete-time scheduling, reduced-space, global optimization, adaptive refinement, artificial neural networks

Corresponding author: *A. Mitsos
AVT Process Systems Engineering, RWTH Aachen University, 52056 Aachen, Germany
E-mail: amitsos@alum.mit.edu

Nomenclature

Reduced-space scheduling problem

Symbol	Description
d	Continuous degree of freedom
x	Intermediate variable
y	Binary degree of freedom
Φ	Objective function

Wavelet transform

Symbol	Description
$\lambda_{a,b}$	Wavelet coefficient
$\psi_{a,b}$	Wavelet basis function

Case study

Symbol	Description	Unit
m^C	Mass processed by compression train	[kg]
m^E	Mass processed by expansion train	[kg]
m^N	Net mass accumulation in cavern	[kg]
m	Currently stored mass	[kg]
p	Cavern pressure	[Pa]
W^C	Electricity consumption of compression train	[J]
W^E	Electricity production of expansion train	[J]
w^C	Specific electricity consumption of compression train	[J/kg]
w^E	Specific electricity production of expansion train	[J/kg]
y^C	Binary variable for compression train	[-]
y^E	Binary variable for expansion train	[-]
Φ	Objective function (= negative savings)	[EUR]
Π^C	Pressure ratio for compression train	[-]
Π^E	Pressure ratio for expansion train	[-]

Indices

Symbol	Description
i	Set of continuous degrees of freedom
j	Set of binary degrees of freedom
k	Set of intermediate variables
l	Set of inequality constraints
m	Set of equality constraints
s	Iteration index
t	Temporal index for variables
$\hat{t}$	Index for degrees of freedom in reduced dimensionality

1 Introduction

In economies with a high share of electricity generation from intermittent renewable sources, the exploitation of temporal price spreads at electricity markets is considered a promising measure to increase the competitiveness of both energy suppliers and consumers [46]. Consequently, raising these potentials through optimized schedules has become a vibrant research field [66]. Process scheduling has been addressed in literature using discrete-time [30, 14, 40, 64] and continuous-time formulations [50, 53, 24, 25, 41]. Advantages and applications in process systems engineering of both types of time representation have been widely reviewed [17, 39, 21]. For scheduling subject to time-variable input data, such as electricity prices, only few authors propose continuous-time formulations (e.g., [12, 13, 20]), as the input data is in general inherently discrete (e.g., hourly changing electricity prices in day-ahead markets). Consequently, the vast majority of literature relies on discrete-time formulations, mostly by applying the same discretization as the input data and assigning optimal loads and modes of operation to the individual intervals. Therefore, we herein confine to discrete-time formulations. Furthermore, we explicitly focus on scheduling subject to time-variable electricity prices.

Early work from Ierapetritou et al. [26] already addresses discrete-time scheduling problems subject to time-variable electricity prices considering an air separation unit as application. The authors consider multiple stationary operating modes and formulate the optimal production schedule as a mixed-integer linear program (MILP). Later, Karwan and Keblis [27] and Mitra et al. [43, 45] tighten and extend these MILP formulations. Exemplary applications of such MILP scheduling formulations include the capacity planning for air separation units [44] and the economic assessment of a switchable chlorine electrolysis [11]. However, real processes are often not well represented by simplified linear models. Zhang et al. [68] therefore propose a framework to approximate nonconvex feasible sets and nonlinear objective functions with MILP formulations that can be handled efficiently by state-of-the-art algorithms/solvers. This is achieved by approximating the feasible set by a union of polytopes with piecewise linear objective function. The approach is referred to as convex region surrogate (CRS) model and has been succesfully applied to scheduling of air separation units [67, 70, 69]. However, CRS models for multivariate nonlinear behavior typically result in a large number of disjunctions, which complicates the solution of the MILPs. This motivates the consideration of nonlinearities directly in the optimization problem.

Only very few works apply nonlinear process models in scheduling, leading to mixed-integer nonlinear programs (MINLPs) (e.g., [18]). The use of global solution methods is desired for such problems. Moreover, the number of variables and constraints scales linearly with the number of considered intervals in discrete-time scheduling problems, such that prohibitive problem sizes are easily reached. Global solution of MINLP scheduling problems consequently becomes highly challenging for relevant planning horizons with fine discretizations.

Reduction of the problem size can be addressed by clustering methods that represent a time series with

a large number of intervals by an aggregated time series consisting of a smaller number of representative intervals. Methods for time series aggregation are based on ideas dating back decades and have been widely applied in energy systems engineering, cf. [60] and the references therein. Most of them aim at including information about time-variable input data into design optimization [2, 31, 3]. There are systematic approaches for iterative decisions whether to refine the resolution of the aggregated time series, e.g., by monitoring the course of the optimal design variable values (cf., [37]) or the objective function value (cf., [48]) over the iterations. However, there are currently no methods to determine which parts of the time series require a fine resolution and where a coarser one is appropriate. Moreover, a drawback of methods for time series aggregation is that the chronology of time steps is in general not preserved, i.e., they cannot capture if the model outcome in one time step depends on previous time steps. This is however crucial for process systems, where storage is important to raise economic potentials. There are recently proposed aggregation methods that are able to retain the chronology for periods of the original time series [5]. Nevertheless, for short-term scheduling subject to time-variable electricity prices, approaches are desired that guarantee compliance with the chronology for the entire scheduling horizon.

Therefore, we propose an alternative approach to achieve a low temporal dimensionality in the optimization problem. The first key idea of our algorithm is to linearly map a reduced set of new degrees of freedom (DoF) for optimization onto the original DoFs, which represent each time interval of the scheduling horizon individually. In particular, we assign the same DoF to multiple time intervals with similar electricity prices, which has basically a similar effect as aggregating the time series. However, using a linear mapping procedure maintains the chronology of the time series and allows for consideration of constraints for all time intervals in contrast to aggregation methods. This enables the modeling of storage relations and guarantees feasibility of the solution point if it is inserted into the original problem with full temporal dimensionality.

In the algorithm, we furthermore apply the wavelet-based adaptive grid algorithm from Schlegel et al. [56] originally proposed for iteratively refining the control vector parametrization and thus the number of DoFs in dynamic optimization problems (cf., [7]). Therein, a wavelet transform of the solution of the previous iteration with coarser discretization is conducted. An analysis of the coefficients of the wavelet transform is then performed for systematic adjustments of the discretization. This analysis identifies both parts of the considered horizon that need a finer discretization and parts where a coarser one can be used, thus allowing for systematic introduction of additional and deletion of insignificant DoFs from the optimization problem. We apply this grid-adaptation to iteratively refine the assignment of DoFs to time intervals. Thereby, our proposed algorithm generates a sequence of feasible points for the optimization problem with full temporal dimensionality. Furthermore, the algorithm can guarantee nonincreasing objective values between subsequent iterations. By solving the optimization problems with

reduced dimensionality to global optimality, the sequence ultimately converges to the globally optimal schedule. As the iterative increase in problem sizes involves an increasing computational demand, we also adapt heuristic stopping criteria and tuning parameters from Schlegel et al. [56] that balance the trade-off between improving the objective function and limiting the computational burden.

Finally, in order to facilitate the application of the algorithm, we introduce a reduced-space scheduling formulation, which allows for optimization only in the space of the DoFs instead of the full-space comprising all model variables [47], and which has recently been shown advantageous for global optimization of process flowsheets [8, 9] as well as optimizations with artificial neural networks (ANNs) embedded [58, 57]. We highlight that reduced-space formulations of scheduling problems show similarities with sequential solution approaches for dynamic optimizations [52, 32]. This can be seen as a vanishing border between scheduling formulations that classically rely on quasi-stationary models and formulation of dynamic optimization problems. We emphasize that the presented approach is consequently also relevant for the emerging research field of scheduling with dynamic models [4], where several authors have proposed the construction of low-dimensional dynamic models to capture the dynamics of scheduling-relevant variables [15, 29, 61] and applied them to the operational optimization of process systems [49, 62]. Nevertheless, we herein focus on the application of the algorithm to scheduling with quasi-stationary models, as they still appear predominant.

The proposed algorithm is applied to the scheduling of a compressed air energy storage subject to time-variable electricity prices. The scheduling problem is considered a suitable case study for assessing the performance of the algorithm for computationally highly challenging problems, as substantial nonconvexities originating from the nonlinear efficiency characteristics of the turbo machines exist, for which we use ANNs as powerful surrogate models for operational optimization.

The remainder of this article is structured as follows: we first present a generic scheduling problem with time-variable electricity prices in a reduced-space formulation. Afterwards, the algorithm itself is introduced. Furthermore, we discuss its attributes. Then, the case study is presented and the specific MINLP scheduling model formulation is introduced. Computations are conducted for different scheduling horizons.

2 Generic reduced-space scheduling problem

The algorithm introduced in Section 3 targets the solution of the generic scheduling problem (1a)-(1e). This corresponds to a reduced-space scheduling formulation. Note that a significant share of all discrete-time scheduling problems with time-variable electricity prices that have been considered in literature so far can be formulated equivalently. We do not claim that the reduced-space formulation is always beneficial. However, it facilitates the application of the grid-adaptation.

In (1a)-(1e), the schedule for fulfilling a production target is optimized by minimizing costs. Alternative objectives might be possible, such as maximizing profitability, and would require only minor changes. The objective function Φ (1a) is a sum over single interval cost functions ϕ_t for each time interval $t = 1 \dots T$. In most cases, this simply corresponds to the product of the instantaneous electricity price and the current electricity consumption. d_i denote the continuous process inputs (such as mass flows, power consumptions, etc.) and y_j the binary (e.g., operating modes). The values of the process inputs in each time interval t are the DoFs for optimization, referred to as $d_{i,t}$ and $y_{j,t}$ and limited by respective bounds (1d)-(1e). If the DoFs are specified, the entire schedule can be calculated. Moreover, two kinds of constraints are imposed: one that needs to be fulfilled in every interval (1b), e.g., operating ranges, and one for the last interval (1c), which allows for consideration of production targets. Other types of constraints (e.g., targets at intermediate time points, minimum requirements at final time, etc.) are possible as well and could be added in a similar way as (1b)-(1c), but are omitted for readability. The (in)equality signs in (1b)-(1c) are chosen to match the case study, however different ones could be applied as well. Furthermore, we assume that the function $g_{l,t}$ in (1b) depends on all DoFs up to time t, as this is typically the case for real processes. However, the assumption is not necessarily required.

$$\min_{d_{i,t}, y_{j,t}} \quad \Phi = \sum_{t=1}^{T} \phi_t(d_{i,t}, y_{j,t}) \tag{1a}$$

$$\text{s.t.} \quad 0 \leqslant g_{l,t}(d_{i,1}, \dots, d_{i,t}, y_{j,1}, \dots, y_{j,t}), \quad \forall l, t \tag{1b}$$

$$0 = h_m(d_{i,1}, \dots, d_{i,T}, y_{j,1}, \dots, y_{j,T}), \quad \forall m \tag{1c}$$

$$d_i^l \leqslant d_{i,t} \leqslant d_i^u, \quad \forall i, t \tag{1d}$$

$$y_{j,t} \in \{0, 1\}, \quad \forall j, t \tag{1e}$$

We herein also allow for further intermediate variables $x_{k,t}$, that are however not exposed to the optimizer and thus not presented as part of the reduced-space optimization problem (1a)-(1e). Instead, they are calculated successively in an explicit manner from the DoFs and their initial conditions $x_{k,1}$.

$$x_{k,t+1} = f_k(x_{k,t}, d_{i,t}, y_{j,t}), \quad \forall k, t \backslash \{T\} \tag{2}$$

We thereby explicitly allow for temporal dependencies to enable the modeling of storage devices. Constraints on $x_{k,t}$ (e.g., storage limits) can also be accounted for in the reduced-space formulation above by including them into (1b)-(1c).

3 Wavelet-based grid-adaptation

Recall that the proposed algorithm comprises two key features in addition to the reduced-space formulation (i.e., hiding (2) from the optimizer): (i) the linear mapping of a reduced set of new DoFs for optimization onto the original DoFs and (ii) the wavelet-based grid-adaptation from Schlegel et al. [56]. Both are explained in the following. Furthermore, properties of the algorithm are discussed.

3.1 Linear mapping of the degrees of freedom

We represent the original DoFs of the optimization problem $d_{i,t}$ and $y_{j,t}$ by new DoFs $\hat{d}_{i,\hat{t}}$ and $\hat{y}_{j,\hat{t}}$ with reduced dimensionality. Formally, the new DoFs for optimization are mapped to the original DoFs by a linear relation using the matrix $\boldsymbol{A} \in \mathbb{R}^{T \times \hat{T}}$ ((3a)-(3b)). The index for the new DoFs $\hat{t} = 1 \dots \hat{T}$ does not necessarily have a chronological meaning. In fact, we thereby also aggregate disjoint time intervals. A significant reduction of the number of DoFs is achieved by choosing $\hat{T} \ll T$.

$$(d_{i,1}, \dots, d_{i,T})^T = \boldsymbol{A}(\hat{d}_{i,1}, \dots, \hat{d}_{i,\hat{T}})^T, \quad \forall i \tag{3a}$$

$$(y_{j,1}, \dots, y_{j,T})^T = \boldsymbol{A}(\hat{y}_{j,1}, \dots, \hat{y}_{j,\hat{T}})^T, \quad \forall j \tag{3b}$$

The use of a common matrix for all inputs is not mandatory and only selected to simplify the notation. As the algorithm targets the solution of scheduling problems with time-variable electricity prices, we consider a mapping that assigns one degree of freedom to multiple intervals of the scheduling problem that comprise similar electricity prices. Thus, each row of $\boldsymbol{A}$ has $\hat{T} - 1$ zero entries and one entry equal to one.

Inserting the right-hand side of (3a)-(3b) into the optimization problem (1a)-(1e), we obtain a reduced-space formulation with lower dimensionality.

$$\min_{\hat{d}_{i,\hat{t}}, \hat{y}_{j,\hat{t}}} \quad \hat{\Phi} = \sum_{\hat{t}=1}^{\hat{T}} \hat{\phi}_{\hat{t}}(\hat{d}_{i,\hat{t}}, \hat{y}_{j,\hat{t}}) \tag{4a}$$

$$\text{s.t.} \quad 0 \leqslant \hat{g}_{l,t}(\hat{d}_{i,1}, \dots, \hat{d}_{i,\hat{T}}, \hat{y}_{j,1}, \dots, \hat{y}_{j,\hat{T}}), \quad \forall l, t \tag{4b}$$

$$0 = \hat{h}_m(\hat{d}_{i,1}, \dots, \hat{d}_{i,\hat{T}}, \hat{y}_{j,1}, \dots, \hat{y}_{j,\hat{T}}), \quad \forall m \tag{4c}$$

$$d_i^l \leqslant \hat{d}_{i,\hat{t}} \leqslant d_i^u, \quad \forall i, \hat{t} \tag{4d}$$

$$\hat{y}_{j,\hat{t}} \in \{0, 1\}, \quad \forall j, \hat{t} \tag{4e}$$

In comparison to problem (1a)-(1e), there are fewer optimization variables in (4a)-(4e). The objective remains a summation over single functions, but now with index $\hat{t}$, i.e., with fewer summands. The number

of constraints (4b) in contrast remains unchanged, i.e., constraints have to hold for every t. Moreover, internal calculations of intermediate variables (2) also remain unaffected, such that the system (4a)-(4e) still respects the chronology of the horizon. Thus, dependencies between model variables in time steps t and $t+1$ can still be accounted for, allowing for the consideration of storage relations.

Directly analyzing the left-hand side of (3a)-(3b) for iterative adaptation of $\boldsymbol{A}$ would typically involve the analysis of a strongly fluctuating time series, such that the application of the algorithm from Schlegel et al. [56] might show poor performance. Thus, we propose the following decomposition of $\boldsymbol{A}$ introducing matrices $\boldsymbol{P}$ and $\boldsymbol{B}$:

$$\boldsymbol{A} = \boldsymbol{P}^{-1}\boldsymbol{B}.$$

Having electricity prices $\boldsymbol{c} = (c_1, \dots, c_T)^T$, the permutation matrix $\boldsymbol{P} \in \mathbb{R}^{T\times T}$ is defined in a way that $\boldsymbol{Pc}$ furnishes a vector of electricity prices in descending order. As $\boldsymbol{P}$ is constant during the iterations, the remaining matrix $\boldsymbol{B} \in \mathbb{R}^{T\times \hat{T}}$ is in fact to be adapted over the iterations. In $\boldsymbol{B}$, each row also has $\hat{T}-1$ zero entries and one entry equal to one. However, the rows are arranged in a way that blocks of column vectors with all entries equal to one are formed. The blocks are located from top left to bottom right. The size of the blocks thus determines how many intervals with similar prices are represented by the corresponding DoF. Compared to the multiplication with $\boldsymbol{A}$ (i.e., the left-hand side of (3a)-(3b)), the series after multiplication with $\boldsymbol{B}$ will likely show substantially less fluctuations and is thus expected more suitable for the wavelet-based grid-adaptation. The entire mapping procedure is illustrated in Figure 1 for a small example.

3.2 Wavelet-based analysis and grid-adaptation

We apply the algorithm from Schlegel et al. [56] (cf., Figure 2) to iteratively adjust the first mapping in Figure 1, i.e., the matrix $\boldsymbol{B}$. More precisely, this corresponds to deciding how many intervals with similar prices should be represented by the same DoFs. Hence, starting in iteration 0 from an initial assignment matrix $\boldsymbol{B}^0$, the algorithm iteratively generates a sequence $\boldsymbol{B}^0, \boldsymbol{B}^1, \dots, \boldsymbol{B}^s$ by solving (4a)-(4e). The corresponding solutions $\hat{d}^{s*}_{i,\hat{t}}$ and $\hat{y}^{s*}_{j,\hat{t}}$ are analyzed to determine the adapted assignment matrix $\boldsymbol{B}^{s+1}$ for the next iteration until a desired stopping criterion is reached. The adaptations of the matrix $\boldsymbol{B}$ correspond to adding additional DoFs to the optimization problem ("grid point insertion") or to deleting irrelevant ones ("grid point deletion").

The algorithm decides on whether to increase or decrease the number of DoFs by analyzing the differences in the solution from the previous iteration, i.e., by analyzing $|\Delta\hat{d}^{s*}_{i,\hat{t}}| = |\hat{d}^{s*}_{i,\hat{t}} - \hat{d}^{s*}_{i,\hat{t}-1}|$. Easily spoken, if $|\Delta\hat{d}^{s*}_{i,\hat{t}}|$ is small, using individual DoFs $\hat{d}^{s+1}_{i,\hat{t}}$ and $\hat{d}^{s+1}_{i,\hat{t}-1}$ in the next iteration is insignificant. In contrast, if $|\Delta\hat{d}^{s*}_{i,\hat{t}}|$ is large, it is worth investigating benefits from increasing the number of DoFs to

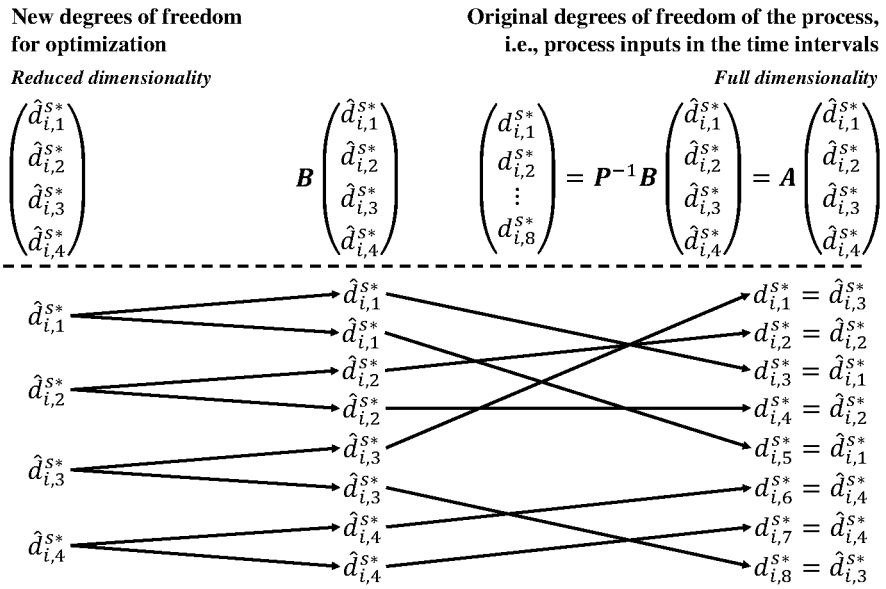

Figure 1: Illustration of the linear mapping procedure with $T = 8$ and $\hat{T} = 4$. The first multiplication with matrix $\boldsymbol{B}$ is a mapping of a small number of DoFs onto a larger number of intervals. The following multiplication with $\boldsymbol{P}^{-1}$ is a rearrangement. The first DoF is used for the intervals with highest and second highest electricity price ($t = 3, 5$). The second DoF is used for the intervals with third and fourth highest electricity price ($t = 2, 4$). Likewise, the third and fourth DoFs are also assigned to two intervals ($t = 1, 8$ and $t = 6, 7$ respectively).

represent this part of the time series.

In order to provide systematic decision criteria for grid point insertion and deletion, a wavelet transform $\mathcal{W}_\psi\{\cdot\}$ is conducted. Considering one of the continuous DoFs $\hat{d}^{s*}_{i,\hat{t}}$, this reads:

$$\mathcal{W}_\psi\left\{\boldsymbol{B}^s(\hat{d}^{s*}_{i,1},\ldots,\hat{d}^{s*}_{i,\hat{T}})^T\right\} = \sum_{a=-1}^{N-1}\sum_{b=0}^{M_a}\lambda^s_{a,b}\cdot\psi_{a,b}. \tag{5}$$

Thus, the time series after the first linear mapping (cf., Figure 1) is represented by a linear combination of wavelet basis functions $\psi_{a,b}$ with coefficients $\lambda^s_{a,b}$. For a detailed treatment of the wavelet representation, we refer to the relevant literature (e.g., [23]). The procedure to obtain $\lambda^s_{a,b}$, called fast wavelet transform (FWT), is given in Appendix A.1. M_{-1} is defined to be 0, while for $a \geqslant 0$, $M_a = 2^a - 1$ holds. Thus, if T time intervals are considered, T basis functions $\psi_{a,b}$ are required for the wavelet representation. The set of basis functions $\psi_{a,b}$ with nonzero coefficients $\lambda^s_{a,b}$ is strongly related to the matrix $\boldsymbol{B}^s$ (cf., Appendix A.3). E.g., assuming pairwise different $\hat{d}^{s*}_{i,\hat{t}}$ in Figure 1, the wavelet representation would in general use four basis functions $\psi_{-1,0}$, $\psi_{0,0}$, $\psi_{1,0}$, and $\psi_{1,1}$ with nonzero coefficients.

The coefficients of the wavelet transform (5) are analyzed for deciding on grid point deletions and insertions as proposed by Schlegel et al. [56]. A detailed treatment can be found in their work. Thus, we only review the key ideas. The required formulas and further information are given in Appendix A.2.

The aforementioned decision whether to increase or decrease the number of DoFs to represent a specific part of the time series can also be based on the absolutes of the wavelet coefficients $|\lambda^s_{a,b}|$

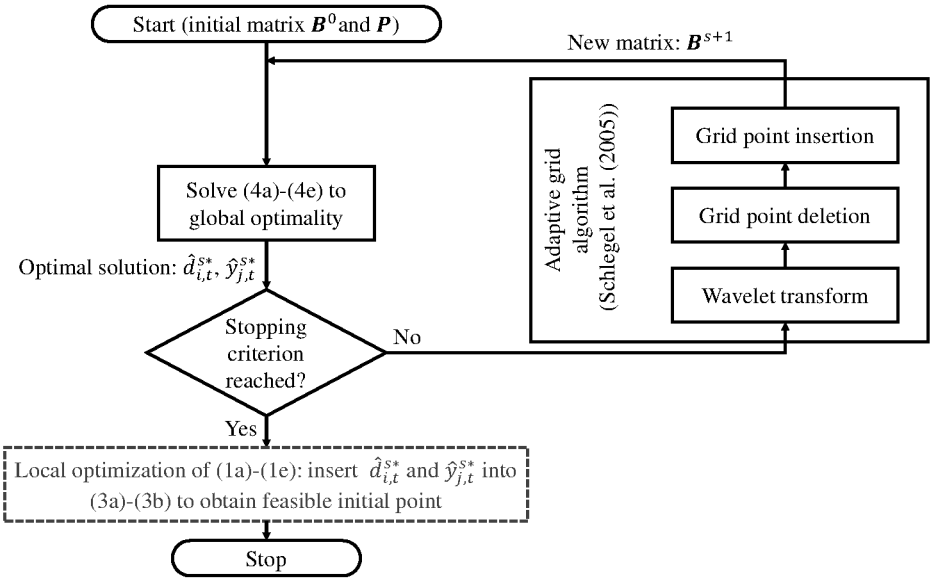

Figure 2: Flowchart of the wavelet-based grid-adaptation. The last gray box corresponds to an optional a posteriori search.

instead of the differences $|\Delta\hat{d}^{s*}_{i,\hat{t}}|$. Comparably low $|\lambda^s_{a,b}|$ correspond to the presence of DoFs with low informational content that can be deleted from the optimization problem with minor influences on the objective value to be expected. In contrast, comparably high $|\lambda^s_{a,b}|$ indicate that using additional DoFs might enable significant benefits. Consequently, threshold values e_d for deletion and e_i for insertion are used and can be tuned to increase the performance. The so-called norm equivalence of the wavelet transform (cf., [38]) allows for proper scaling, such that $0 \leqslant e_d, e_i \leqslant 1$ holds.

By performing grid point insertions and deletions, the relevant set of basis functions $\psi_{a,b}$ for the next iteration is identified. Based on this, the new mapping can be derived using the procedure given in Appendix A.3. Assuming that in the wavelet representation of the example in Figure 1, the coefficient $\lambda_{1,0}$ would fulfill the criterion for insertion and $\lambda_{1,1}$ the criterion for deletion, in the next iteration, there would be individual DoFs for the interval with highest ($t = 3$), with second highest ($t = 5$), with third highest ($t = 2$), and with fourth highest electricity price ($t = 4$). The remaining fifth DoF would be assigned to all other intervals. This would correspond to using basis functions $\psi_{-1,0}$, $\psi_{0,0}$, $\psi_{1,0}$, $\psi_{2,0}$, and $\psi_{2,1}$. Note that, in contrast to dynamic optimization problems, there is a finest resolution for the DoFs that gives exactly a one-to-one assignment in (3a)-(3b) with $T = \hat{T}$. Consequently, inserting grid points that correspond to using basis functions with an assignment of multiple DoFs to the same interval (i.e., basis functions $\psi_{N,b}$, $\psi_{N+1,b}$, etc.) is not allowed.

Further note that the wavelet transform (5) requires $T = 2^N, N = 1, 2, 3, \ldots$, i.e., scheduling horizons of $2, 4, 8, 16, \ldots$ intervals (e.g., hours). In the computational study, we confine to these cases. For different scheduling horizons, e.g., $T = 24$ (one day with hourly discretization), one could decompose the horizon into multiple subhorizons - in this case $T_1 = 16$ and $T_2 = 8$.

3.3 Properties of the algorithm

If the deletion of grid points is prohibited, the objective value in iteration $s+1$ is guaranteed to be as least as good as the objective value in iteration s. Thus, using $\epsilon_d = 0$ gives a nonincreasing sequence of objective values for any ϵ_i. If $\epsilon_i = 1$ is additionally chosen (i.e., selecting all possible grid points for insertion), the algorithm ultimately results in a one-to-one assignment in (3a)-(3b) with $\boldsymbol{B} \in \mathbb{R}^{T\times T}$. Consequently, the solution of (4a)-(4e) then becomes equivalent to the solution (1a)-(1e). If the subproblems in each iteration are solved to global optimality, the sequence of solution points converges to the global minimizer of (1a)-(1e) in this case.

Due to the computational complexity of the original problem, convergence of the optimization problems to solve to (1a)-(1e) is in general not desired. A reasonable selection of ϵ_d and ϵ_i thus should balance between exploration through increasing the number of DoFs for improved objective values and maintenance of low dimensionality in the optimization problem to limit the computational demand. Furthermore, appropriate stopping criteria should be applied. A suitable criterion indicating near-optimal solution points furnished by the grid-adaptation is the relative improvement in the objective function from one iteration to the next:

$$\Delta\Phi^s = \frac{|\Phi^{s-1} - \Phi^s|}{|\Phi^s|}.$$

If $\Delta\Phi^s$ falls below a threshold, the algorithm is terminated. As a reasonable selection for the threshold value, we herein apply the relative optimality tolerance *optcr* for the solution of (4a)-(4e) for this purpose as well, i.e., the algorithm is terminated if $\Delta\Phi^s < optcr$. Moreover, if the insertion step only proposes refinements that would lead to a finer resolution than a one-to-one assignment, the algorithm is also terminated.

In each iteration s, inserting the optimal solution $\hat{d}^{s*}_{i,\hat{t}}$ and $\hat{y}^{s*}_{j,\hat{t}}$ into (3a)-(3b) gives a feasible point for the optimization problem (1a)-(1e). Note that this point will in general not satisfy any optimality conditions. However, as the point is feasible and expected near-optimal if sufficiently many iterations have been performed, it represents a suitable initial solution for a local search considering the full temporal dimensionality of optimization problem (1a)-(1e). The search does not necessarily need to be performed among all optimization variables. For instance, performing the local search only on continuous variables that can be handled very efficiently in state-of-the-art local NLP solvers seems to be an appealing alternative. We acknowledge that there is no guarantee that the local solver will converge to the actual global minimum. However, a feasible point satisfying local optimality conditions and likely further improving the objective value can be furnished requiring comparably low additional computational effort. Alternatively, the feasible point generated through (3a)-(3b) could be used as initial point for a global solution of (1a)-(1e); this would be beneficial for global algorithms whose computational requirements

strongly depend on having good initial guesses.

Various works demonstrate the successful practical application of the grid-adaptation to dynamic optimization problems. In particular, authors have shown that the algorithm can identify high-quality solutions by using problem specific discretizations that are comparable to those when using very fine equidistant discretizations, but require substantially less computational time [56, 51, 22]. In particular, authors have also shown that an adequate balance between maintaining a low number of DoFs and improving the objective function can be achieved by tuning the parameters and that few adaptation steps starting from coarse discretizations commonly suffice for obtaining near-optimal solutions.

Nevertheless, the performance of the algorithm will certainly depend on the actual problem considered. As the algorithm essentially exploits the strong relation between instantaneous electricity prices and optimal values of the DoFs in the corresponding intervals, we expect the performance to be best if this relation is very distinct.

4 Case Study

Storage of electrical energy is currently considered as an important measure to promote the penetration of renewable electricity generation from intermittent sources [6]. Among the variety of opportunities, compressed air energy storage (CAES) belongs to the few technologies that have been commercialized and are readily available at large scale [36]. In a CAES plant (Figure 3), electricity is bought at low prices. The energy is used to pressurize and store air in a cavern. In times of high electricity prices, the cavern is discharged via turbines that power a generator. The concept relies on large price spreads to overcome the efficiency losses of the turbo machinery [34]. An optimal scheduling is thus crucial for the economic performance of the plant [35].

Scheduling of a CAES plant represents a computationally challenging optimization problem, as the efficiency characteristics of turbo machinery commonly show highly nonlinear behavior. Moreover, scheduling of CAES plants involves a strong linking between time intervals, in a sense that the pressure of the cavern in interval t and thus also the necessary work for storing additional gas/the released work by expanding stored gas depends on the cavern pressure of the previous interval $t-1$.

4.1 Scheduling model

In the following, the considered set of equations for modeling a CAES plant with a distinct focus on the relations inside the turbo machines is presented to highlight the capabilities of the presented algorithm to solve nonlinear scheduling problems involving linking between the intervals.

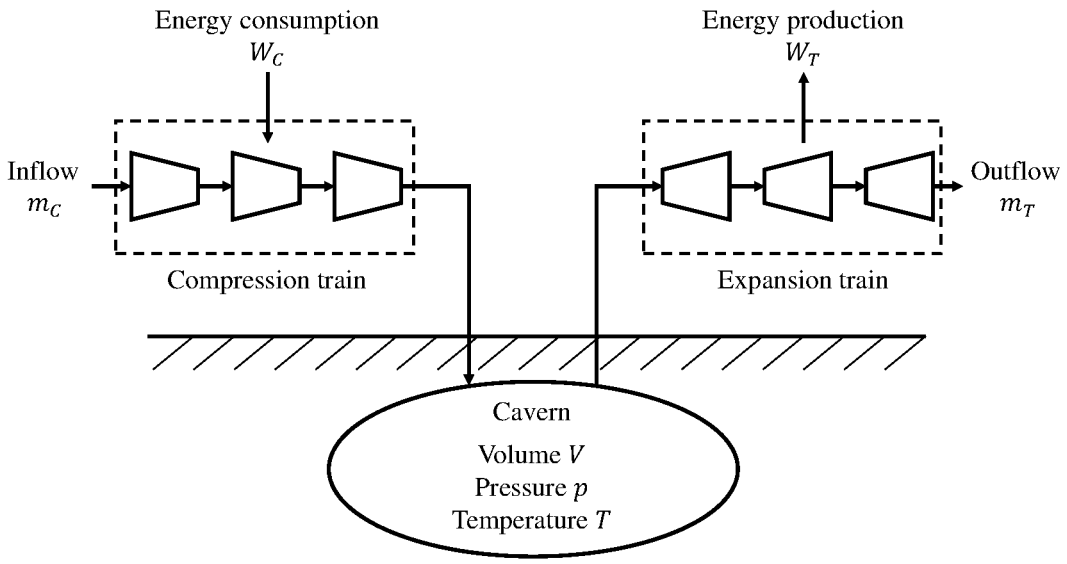

Figure 3: Simplified flowsheet of a compressed air energy storage.

4.1.1 Model equations

We aim at maximizing the profit, which corresponds to the difference between the revenues from electricity disposal and the cost for electricity purchase. The objective Φ to be minimized reads:

$$\Phi = -\sum_{t=1}^{T} c_t (W_t^E - W_t^C),$$

where c_t is the instantaneous spot electricity price in hour t (without distinction of selling vs. purchasing price). We use historic German day-ahead spot electricity prices from end of January 2018, which are retrieved from EPEX SPOT SE (www.epexspot.com). W_t^E and W_t^C denote the electricity production/consumption of the expansion/compression train in the corresponding hour. We use w_t^E and w_t^C to denote specific energies and m_t^E and m_t^C as masses that are processed by the expansion/compression train.

$$W_t^E = m_t^E w_t^E, \quad \forall t$$
$$W_t^C = m_t^C w_t^C, \quad \forall t$$

We further introduce binary variables y_t^E and y_t^C that indicate if the cavern is charged or discharged in the current interval. There is no sense in simultaneously charging and discharging if energy is bought and sold at the same price. However, there is the possibility of neither charging nor discharging during intervals with intermediate electricity prices. We thus require

$$y_t^E + y_t^C \leqslant 1, \quad \forall t.$$

Furthermore, we set limits for the mass flows only in case that the operating mode is active. Otherwise we force them to be zero

$$y_t^E m_{min}^E \leqslant m_t^E \leqslant y_t^E m_{max}^E, \quad \forall t$$
$$y_t^C m_{min}^C \leqslant m_t^C \leqslant y_t^C m_{max}^C, \quad \forall t.$$

We thereby use big-M formulations for the disjunctions, although convex hull formulations generally lead to tighter relaxations [19]. Moreover, a linear balance is used for modeling the accumulation of stored mass within the cavern

$$m_{t+1} = m_t + (m_t^C - m_t^E)$$

with m_t as the stored mass. Assuming air to be an ideal gas, the cavern pressure p_t can be calculated via

$$p_t = \frac{m_t R T}{M V}$$

using $R = 8.314 \ \frac{\text{J}}{\text{mol·K}}$ (ideal gas constant), $M = 0.028949 \ \frac{\text{kg}}{\text{mol}}$ (molar mass of air), $T = 323.15$ K (constant cavern temperature), and $V = 40,000$ m^3 (cavern volume). The cavern pressure is used to determine the pressure ratios of the expansion/compression train Π_t^E and Π_t^C. We account for pressure losses in the pipes through a constant $\Delta p = 50,000$ Pa. $p_a = 100,000$ Pa further corresponds to the ambient pressure.

$$\Pi_t^E = \frac{p_t - \Delta p}{p_a}$$
$$\Pi_t^C = \frac{p_t + \Delta p}{p_a}$$

Having the respective pressure ratios as well as the processed amount of air, the specific energies can be obtained from the efficiency maps of the turbo machines. We use artificial neural networks (ANNs) to represent the efficiency correlations. We use one efficiency map for the expansion and one for compression train, each at optimized operating points. More precisely, the efficiency maps are obtained from individual operational optimizations of the turbine/compressor configuration by optimally setting pressure ratios between the single turbo machines that lead to lowest energy consumption/highest energy

production for given total flow and pressure ratio. This procedure is explained in detail in our earlier work [58]. The ANN functions are denoted by f^E_{ANN} and f^C_{ANN}. In both cases, feed-forward ANNs with hyperbolic tangent activation functions comprising one hidden layer and 25 neurons are used. Training of the ANNs is conducted using the Levenberg-Marquardt algorithm from MATLAB's Deep Learning Toolbox (Mathworks, Inc.).

$$w^E_t = f^E_{\text{ANN}}(m^E_t, \Pi^E_t)$$
$$w^C_t = f^C_{\text{ANN}}(m^C_t, \Pi^C_t)$$

Furthermore, we also represent the operating bounds of the turbo machines, such as surge limits, via ANNs. In total, there are four operating bounds for both the expansion and the compression train. In each case, two functions define maximum pressures (left and upper boundaries of the efficiency maps represented by functions $g^E_{\text{ANN},1}$, $g^E_{\text{ANN},2}$, $g^C_{\text{ANN},1}$ and $g^C_{\text{ANN},2}$) and two functions define minimum pressures (right and lower boundaries of the efficiency maps represented by functions $g^E_{\text{ANN},3}$, $g^E_{\text{ANN},4}$, $g^C_{\text{ANN},3}$ and $g^C_{\text{ANN},4}$). ANN-functions defining the operating bounds comprise one hidden layer with 12 neurons.

$$p_t \leqslant g^E_{\text{ANN},1}(m^E_t) + (1 - y^E_t)M$$
$$p_t \leqslant g^E_{\text{ANN},2}(m^E_t) + (1 - y^E_t)M$$
$$p_t \geqslant g^E_{\text{ANN},3}(m^E_t) - (1 - y^E_t)M$$
$$p_t \geqslant g^E_{\text{ANN},4}(m^E_t) - (1 - y^E_t)M$$
$$p_t \leqslant g^C_{\text{ANN},1}(m^C_t) + (1 - y^C_t)M$$
$$p_t \leqslant g^C_{\text{ANN},2}(m^C_t) + (1 - y^C_t)M$$
$$p_t \geqslant g^C_{\text{ANN},3}(m^C_t) - (1 - y^C_t)M$$
$$p_t \geqslant g^C_{\text{ANN},4}(m^C_t) - (1 - y^C_t)M$$

Choosing a sufficiently large constant M, which needs to be as high as the maximum cavern pressure that can be reached, ensures that the equations are always fulfilled in time intervals, where the operating mode is not active. This again corresponds to big-M formulations with the aforementioned implications. Finally, we prevent an emptying of the cavern over the horizon by demanding:

$$0 = \sum_{t=1}^{T} (m^C_t - m^E_t).$$

4.1.2 Degrees of freedom and reduced-space formulation

Revisiting the model equations in the previous subsection, one recognizes that if values are given for the process inputs (turbine/compressor flow and the operating modes) for all t, all other model equations can be evaluated in an explicit manner. The reduced-space scheduling problem for the CAES thus reads (cf., (1a)-(1e))

$$\min_{m_t^E, m_t^C, y_t^E, y_t^C} \quad \Phi = \sum_{t=1}^{T} \phi_t(m_t^E, m_t^C, y_t^E, y_t^C) \tag{6a}$$

$$\text{s.t.} \quad 0 \leqslant g_{l,t}(\ldots), \quad \forall l, t \tag{6b}$$

$$0 = \sum_{t=1}^{T} (m_t^C - m_t^E), \tag{6c}$$

$$0 \leqslant m_t^E \leqslant m_{max}^E, \quad \forall t \tag{6d}$$

$$0 \leqslant m_t^C \leqslant m_{max}^C, \quad \forall t \tag{6e}$$

$$y_t^E, y_t^C \in \{0, 1\}, \quad \forall t \tag{6f}$$

with m_t^E, m_t^C, y_t^E, y_t^C corresponding to the DoFs for optimization and summarizing all inequality constraints in (6b). A trivial solution is selecting all DoFs to be zero ($m_t^E = m_t^C = y_t^E = y_t^C = 0$), resulting in an initial upper bound $\Phi_0 = 0$.

For the application of the adaptive grid algorithm, we select the same matrix $\boldsymbol{A}^s$ for all process inputs within one iteration. In fact, this means that for $\hat{T}$ DoFs per process input, there are always $4\hat{T}$ optimization variables. Due to the special structure of the problem, it appears to be appropriate to decide on grid point insertion and deletion (i.e., on increasing or decreasing the number of DoFs per process input) based on the analysis of the net flow $m_t^N = m_t^E - m_t^C$. This selection originates from the fact that in each interval either m_t^E or m_t^C is allowed to be nonzero. Furthermore, choosing more DoFs to represent the time series for y_t^E and y_t^C than for m_t^E and m_t^C is pointless. Although a reduction of the number of binary variables might be computationally highly beneficial, we consciously disregard the possibility to represent y_t^E and y_t^C by substantially fewer DoFs than m_t^E and m_t^C, as the refinement procedure originally proposed by Schlegel et al. [56] solely focused on continuous optimization problems. Refinement strategies explicitly targeting an efficient handling of binary variables by accounting for their characteristics (commonly, substantially less frequent fluctuations) is a promising future work.

4.2 Application of the grid-adaptation

The reduced-space scheduling problem (6a)-(6f) is implemented in our in-house software for deterministic global optimization MAiNGO [10], which utilizes the concept of McCormick relaxations [42, 47, 63].

MAiNGO provides very tight relaxations for ANN functions, which enables substantial improvements compared to commercial state-of-the-art solvers [58]. We also tested full-space formulations of the process model in GAMS (GAMS Development Corp.) and tried a solution using BARON [59]. However, the computational performance of the reduced-space formulation applied in MAiNGO was found superior, which is in good agreement with previous work [58].

Default settings are used within MAiNGO for the lower and the upper bounding procedure. Optimization is stopped either if a relative optimality gap of <1% is reached or if a CPU time limit of $5 \cdot 10^5$ s is exceeded. The grid-adaptation is always initialized with four equally-distributed DoFs per process input, i.e., the same DoFs are used for the 25% of intervals with highest electricity prices, etc. The grid-adaptation is either stopped if the relative change in the objective function between subsequent iterations is below 1% or if the finest resolution is reached and no further grid points are to be inserted. The applied threshold values $\epsilon_d = 0.0001$ and $\epsilon_i = 0.7$ have been found by trial and error and provide a good balance between improving the objective function and maintaining a low number of DoFs.

All calculations are conducted on an Intel® Xeon® CPU E5-2630 v3 @ 2.40GHz with 192 GB RAM. Calculations were performed using a single CPU of the server only and thus did not make use of MAiNGO's parallel computing capabilities. We tested three solvers in the upper bounding procedure (SLSQP [33], IPOPT [65], and KNITRO [16]). CPLEX (IBM Corporation) is used as solver in the lower bounding.

4.2.1 Performance of the algorithm for different scheduling horizons

Figure 4 depicts the progress of the adaptive grid algorithm for varied lengths of the scheduling horizon with $N = 4, 5, 6, 7$, which corresponds to $T = 16, 32, 64, 128$ time intervals. Confining to an hourly discretization in accordance with the electricity market, we thus consider scheduling horizons between 16 h and ∼5 d. For $N = 4, 5$, we also show the results of a straight-forward global search considering individual DoFs for each interval of the full time series. As can be seen, a global search considering the full temporal dimensionality becomes computationally prohibitive. In fact, no feasible point with objective value better than Φ_0 can be found within the time limit of $5 \cdot 10^5$ s for $N \geqslant 5$. Results depicted are produced using SLSQP as solver in the upper bounding. Using KNITRO as solver with mixed-integer capabilities gives no improvements. We thus conclude that even for moderately long scheduling horizons, the generation of favorable feasible points is not a simple task.

Moreover, we expect that for $N = 5$ the current lower bound after exceeding the time limit still substantially underestimates the global minimum. Consequently, a straight-forward application of deterministic global solution algorithms to scheduling problems of comparable complexity will not lead to desired results, emphasizing the need for advanced solution algorithms.

As can be seen from Figure 4, the adaptive grid algorithm reaches one of the stopping criteria for all

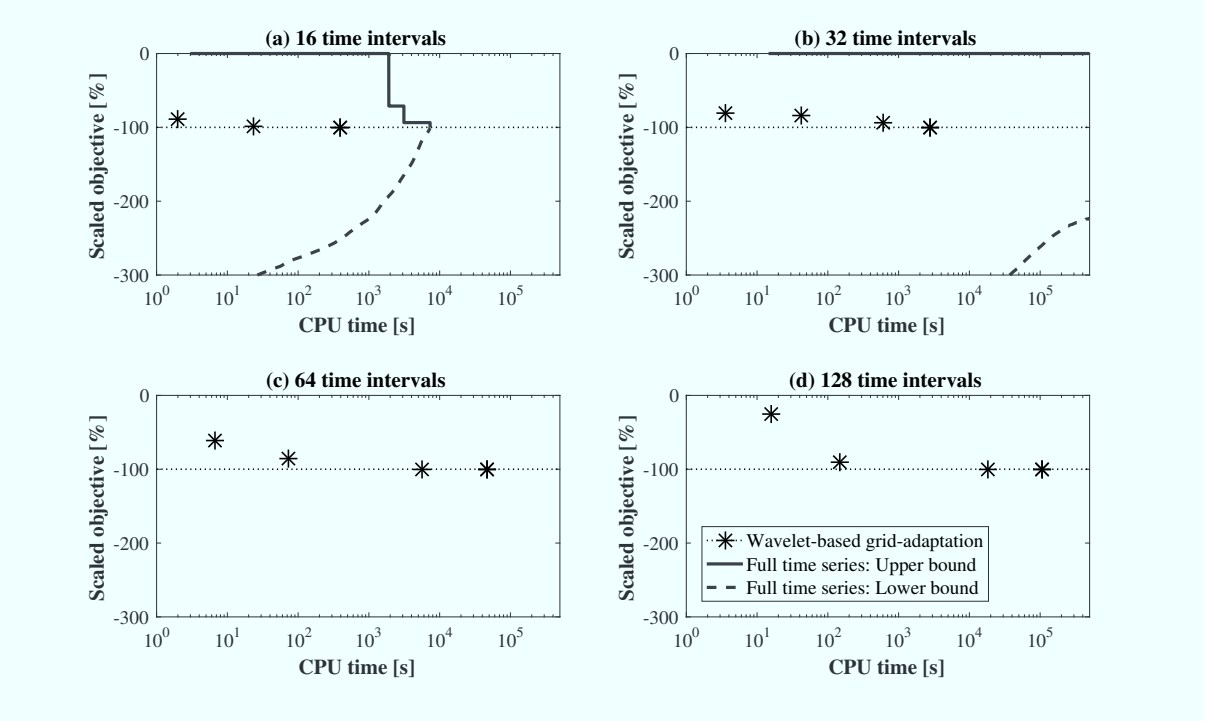

Figure 4: Progress of the grid-adaptation and comparison to a global optimization considering individual DoFs for each interval of the time series. Results are depicted for $T = 16, 32, 64, 128$ ((a) - (d)). Objective function values are relative to the best feasible point furnished by the algorithm (dotted line). Asterisks depict the outcome of the algorithm after each iteration. The solid lines give current upper bounds and dashed lines current lower bounds when solving the problem considering individual DoFs for each interval.

scheduling horizons within at most four iterations. Note that only for $N = 5$, the fourth iteration of the algorithm also results in an improvement of more than 1%; however, no further iterations are performed as a further grid point insertion is only possible in insignificant parts of the time series. The reason for this anomaly lies in a very distinct price peak in the underlying time series of electricity prices. Here, an exceptionally high electricity price occurs for only one hour of the horizon. As in the last iteration of the algorithm, individual DoFs are assigned exclusively to this singular hour, the improvement in the objective function is comparably high between the third and the fourth iteration.

Final solutions depicted in Figure 4 use 12, 14, 16, and 17 DoFs per process input to represent the entire time series. We emphasize that for the longest scheduling horizon of 128 hourly intervals, this corresponds to a reduction of the dimensionality of the DoFs for optimization by almost 90% compared to assigning individual DoFs to each interval of the horizon. In case of $N = 4$, i.e., $T = 16$, the objective value after the last iteration of the grid-adaptation can be compared to the global optimum obtained by a deterministic search under consideration of the full temporal dimensionality. The error between the objective values is substantially less than 1% and thus lower than the remaining optimality gap of the global search. The comparison to the global optimum is not possible for $N \geqslant 5$, as the global solver does not converge within the time limit. However, also in these cases, our results indicate that solution

points close to the global optima have been furnished. In particular, the sequences for $N = 6$ and $N = 7$ seem to converge with the improvement between subsequent iterations consistently decreasing, e.g., for $N = 7$, the improvement between the zeroth (four DoFs per process input) and the first iteration (eight DoFs per process input) is $\Delta\Phi^1 = 72\%$. Further increasing the number of DoFs to 14 in the second iteration gives $\Delta\Phi^2 = 9\%$ and finally $\Delta\Phi^3 < 0.1\%$ using 16 DoFs in the third one. Moreover, comparing the absolute values of the objective function for different N allows for a rough plausibility check: If the fluctuations show repeating patterns with similar amplitudes, savings when doubling the horizon are expected twice as high. As can be seen in Table 1, our results are in a good agreement with this rule of thumb. In fact, when comparing the objective for $N = 5, 6$ with that for $N - 1$, the improvement is larger than a factor two, which can be explained by a larger spread of electricity prices. For $N = 7$, the factor is slightly lower than two; however, in this case, storage capacities become a relevant factor, limiting the achievable savings (cf., Figure 5).

Table 1: Comparison of savings for different horizon lengths and the fluctuations of the corresponding electricity price time series.

Horizon	**Objective value Φ [EUR]**	**Standard deviation of price time series [EUR/MWh]**
$N = 4$ ($T = 16$)	-72.5	22.13
$N = 5$ ($T = 32$)	-156.5	25.14
$N = 6$ ($T = 64$)	-354.3	32.60
$N = 7$ ($T = 128$)	-684.5	32.57

We further tested the two optional a posteriori search strategies (cf., Subsection 3.3) on the original problem with full temporal dimensionality for all scheduling horizons. The local search strategies ((i) local searches considering all variables using mixed-integer solvers and (ii) local searches considering continuous variables only using NLP solvers) both yielded slightly improved solutions within short time. However, improvements are considered insignificant (less than 1% and thereby less than the final optimality gap of the global solution from the last iteration). Global solution strategies starting from the outcome of the adaptive grid algorithm did not result in any further improved feasible points within a CPU time limit of 50,000 s. We thus assume the feasible points that are obtained after stopping of the grid-adaptation to be a close approximation of the global solution of the optimization problem considering the full temporal dimensionality.

Looking at the solution times given in Figure 4 allows for estimating the savings in computational times when applying the grid-adaptation. For $N = 4$, the algorithm furnishes a feasible point with an objective value extremely close to the global minimizer as stated above within less than a CPU time of 400 s by using 12 DoFs per process input to represent the time series. The computational time for the global search considering the full temporal dimensionality, i.e., 16 DoFs per process input, is however $\sim$8,000 s and thus higher by a factor of $\sim$20. In case of $N = 5$, the adaptive grid algorithm terminates

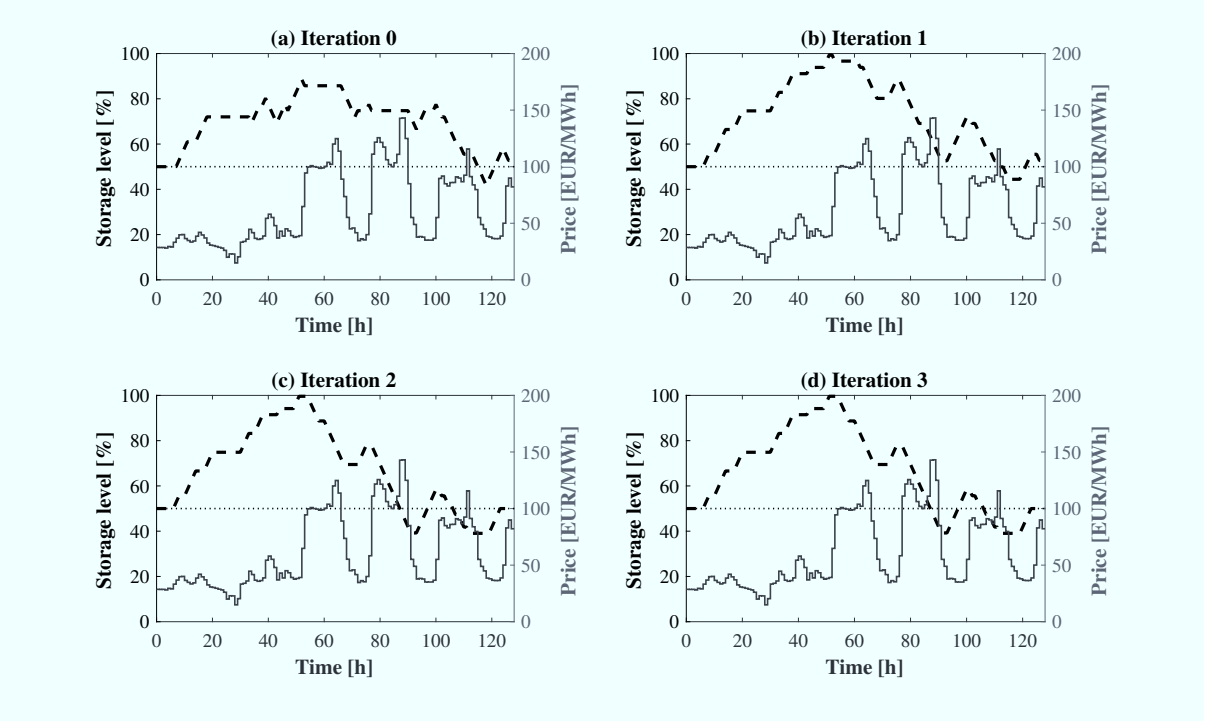

Figure 5: Operation schedule of the CAES plant after each of the four iterations of the grid-adaptation ((a) - (d)) for a scheduling horizon of $T = 128$. Dashed thick black lines depict the storage level (i.e., the cavern pressure), solid thin gray lines correspond to the electricity price profile for the case study. The dotted line depicts the production target.

after a CPU time of less than 3,000 s finally using 14 DoFs per process input to represent the entire time series, compared 32 DoFs per process input when considering the full temporal dimensionality. The global search on the original problem with full temporal dimensionality is stopped after exceeding the CPU time limit of $5 \cdot 10^5$ s with a substantial remaining optimality gap. Time savings enabled by the grid-adaptation thus already correspond to a factor of $\gg$100, i.e., to multiple orders of magnitude. Extrapolating the results presented in Figure 7 and discussed in Subsection 4.2.2 allows for approximating solution times when considering the full temporal dimensionality. E.g., for $N = 5$, solution times in the order of $1 \cdot 10^7 \ldots 1 \cdot 10^8$ s are expected. Solution times for $N \geqslant 5$ thus appear prohibitive even when using multiple cores in parallel.

In Figure 5, the hourly operation schedule of the CAES plant after each iteration of the grid-adaptation is given for a horizon of $T = 128$, i.e., $N = 7$. Note that this corresponds to a planning horizon of more than 5 days, which is a scheduling horizon with relevant length considering the limited predictability of future electricity prices. Recap that the depicted storage level is not a DoF of the optimization problem. In contrast, it integrates the input and output mass flows and thereby the two continuous DoFs (cf., Subsection 4.1). Thus, the slope of the storage level in one interval corresponds to the instantaneous value of the DoFs. Besides, a great advantage of depicting the storage level lies in the fact that it is highly relevant for the limiting constraints in the operational optimization and thereby

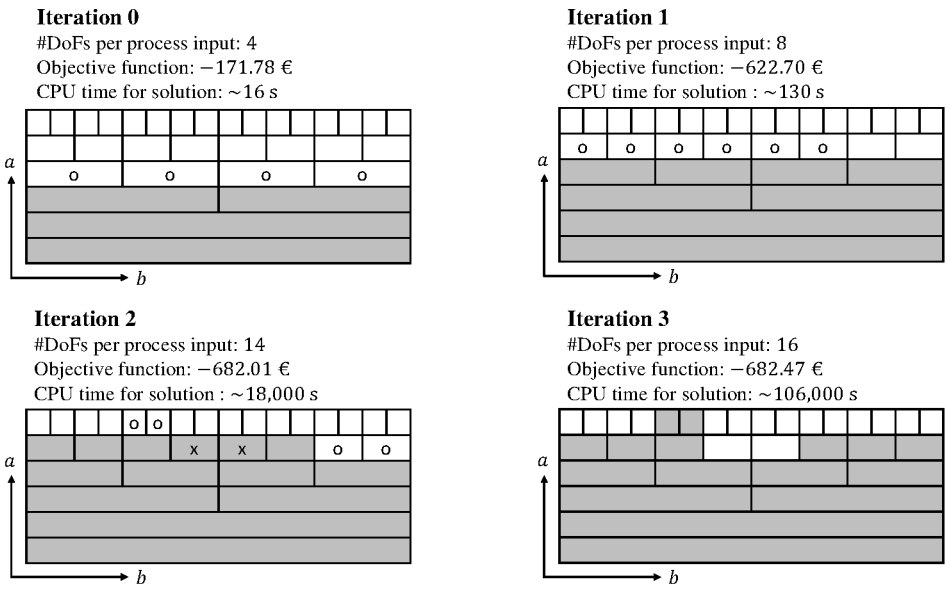

Figure 6: Wavelet representation of the optimal solution during the iterations for a scheduling horizon of $T = 128$. Used (nonzero) wavelet coefficients are in gray. Wavelet coefficients representing grid points for insertion (increasing the number of DoFs) are marked with (o). Wavelet coefficients corresponding to grid points which are deleted are marked with (x) (decreasing the number of DoFs).

facilitates discussions of the schedule as will be explained in the following. Furthermore, a schematic depiction of the wavelet representation of the corresponding solution and thus of the number of DoFs used per process input and their distribution is given in Figure 6.

Iteration 0 starts with using only four DoFs per process input to represent the 128 hourly intervals of the scheduling horizon, i.e., the same value is assigned to a process input in the 32 h with highest electricity prices, etc. The characteristics of the price profile are however such that the most intuitive solution, i.e., filling the cavern in the 32 h with lowest electricity prices and emptying the cavern in the 32 h with highest prices, is not feasible. This explains the very low profit in iteration 0 (cf., Figure 4). Consequently, additional grid points are inserted to represent the times series, causing an increased number of DoFs for optimization in the subsequent iteration. Next, eight equally-distributed DoFs per process input are used that suffice to reproduce an intuitive behavior, i.e., a completely filled cavern after ∼50 h, when electricity prices rise steeply. From then on, only minor adjustments of the charging/discharging strategy occur that slightly tend to a finer resolution when representing intervals with higher prices. After four iterations, the finest resolution used in the linear mapping is an assignment of one DoF to four intervals with similar prices. The coarsest resolution is an assignment of one DoF to 16 intervals. It is assumed unlikely that the introduction of more than the final 16 DoFs per process input would lead to substantial improvements in the objective value. More generally speaking, we assume that 16 DoFs per process input suffice to exploit all relevant fluctuations of the electricity price, in spite of the large range of prices within the considered horizon.

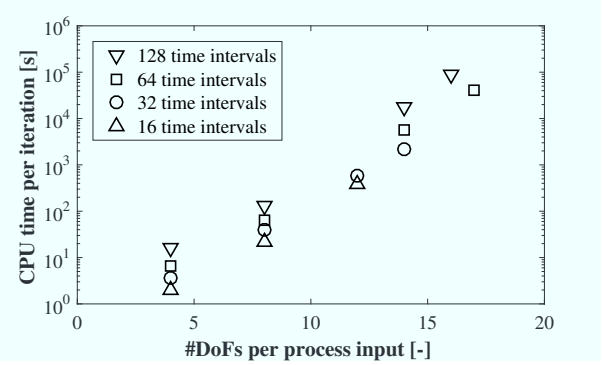

Figure 7: CPU time per iteration of the adaptive grid algorithm vs. DoFs for optimization for each of the process inputs. Note that there are four process inputs m_t^E, m_t^C, y_t^E, and y_t^C with equal mapping matrices. Symbols depict the varying scheduling horizons (upwards-pointing triangles: 16 h, circles: 32 h, squares: 64 h, downwards-pointing triangles: 128 h).

4.2.2 Scaling of the algorithm

As computational times represent the main bottleneck for the solution of nonlinear scheduling problems with time-variable electricity prices on relevant planning horizons, we briefly discuss the scaling of the presented algorithm. Here, one has to differentiate between the consideration of larger time horizons, i.e., more intervals, and the increase in the number of DoFs per process input to represent the time series. Note that in straight-forward solution approaches, the latter directly follows from the former. However, in case of the presented adaptive grid algorithm, the consideration of larger time horizons does not necessarily mean a proportional increase of optimization variables. Solely the number of constraints that need to be satisfied, i.e., the dimensionality of (6b), scales linearly with the length of the horizon.

In Figure 7, solution times for each individual iteration are depicted. The plot uses a logarithmic scale for the CPU times. If the number of intervals considered remains unchanged, the solution times approximately lie on a straight line, corresponding to an exponential scaling of the solution time when increasing the number of optimization variables. In fact, introduction of four additional DoFs per process input results in an increase of CPU time by a factor ten. This finding is valid for all considered scheduling horizons, thus giving parallel lines for the horizons. Importantly, the distance between the lines for N and $N+1$ is approximately constant for all N. This corresponds to a linear relation. More precisely, for a fixed number of DoFs representing the time series, consideration of twice as many intervals ($N' = N+1$), leads to a doubled solution time.

This scaling behavior is considered an important feature of the grid-adaptation, as it allows for the consideration of relevant planning horizons of several days to weeks even for complex nonlinear scheduling problems if an appropriate assignment of DoFs to time intervals is chosen and thus sufficiently few DoFs are exposed to the optimization algorithm.

5 Conclusion

We present an algorithm for the solution of discrete-time nonlinear scheduling problems with time-variable electricity prices. The algorithm is designed for a generic class of scheduling problems that are formulated in a reduced space. In particular, the algorithm is applicable to problems with linking between time intervals due to storage. The key idea of the algorithm is to substitute the original degrees of freedom of the optimization problem, which represent each time interval of the scheduling horizon individually, by new degrees of freedom with substantially reduced dimensionality. More precisely, one degree of freedom is assigned to multiple intervals of the scheduling problem that comprise similar electricity prices through a linear mapping. A systematic approach using a wavelet-based analysis iteratively adapts the assignment. Although we target the application of this algorithm to complex nonlinear programs, the general ideas might also be advantageous for the solution of mixed-integer linear scheduling problems.

We apply the grid-adaptation to the operational optimization of a compressed air energy storage plant. Substantial nonlinearities in the scheduling problem arise from the efficiency characteristics of the turbo machines. These are accounted for by using artificial neural networks as surrogate models. We show that the algorithm is able to identify feasible schedules that give similar objective values as a global search with consideration of the full temporal dimensionality, but requires substantially less time. In fact, this enables the consideration of scheduling horizons, which are prohibitively long without the grid-adaptation. Moreover, we find that the solution times scale approximately linearly with the horizon length when maintaining the same number of degrees of freedom in the optimization problem.

Important perspectives for future research can be identified. (i) It might be a promising improvement to a priori estimate the required number of degrees of freedom and their assignment to the intervals for a good initial representation of the time series. (ii) There are process features in literature which are not relevant in our case study, such as ramping and transitional constraints (more generally: constraints limiting the rate of change of the DoFs between subsequent time intervals). If such constraints are highly relevant for the optimal schedule, the performance of the algorithm could be impaired. Consequently, in presence of ramping and transitional constraints, tailored adaptation strategies might be beneficial. Along the same lines, a special treatment of binary optimization variables is desired that addresses their "on/off" behavior. (iii) Besides, there has been work in the field of dynamic optimization concerning the detection of the switching structure of the problem [55, 54, 1]. Adapting these methods to scheduling as well is a promising perspective. (iv) Finally, it is worth exploring different applications for the grid-adaptation. These will include scheduling problems subject to other time-variable input data, e.g., subject to a time-variable environmental impact (e.g., [28]). A further possible application lies in two-stage stochastic programs, where the same recourse action could be applied to multiple similar scenarios.

Acknowledgements The authors gratefully acknowledge the financial support of the Kopernikus project SynErgie by the Federal Ministry of Education and Research (BMBF) and the project supervision by the project management organization Projektträger Jülich. The authors further thank Dominik Bongartz, Jaromił Najman and Susanne Sass for fruitful discussions concerning the algorithm and helpful advice during the use of MAiNGO as well as Tim Kerkenhoff and Katrin Ostendorf for help with programming issues.

References

[1] E. Aydin, D. Bonvin, and K. Sundmacher. Toward fast dynamic optimization: An indirect algorithm that uses parsimonious input parameterization. *Industrial & Engineering Chemistry Research*, 57(30):10038–10048, 2018.

[2] B. Bahl, A. Kümpel, H. Seele, M. Lampe, and A. Bardow. Time-series aggregation for synthesis problems by bounding error in the objective function. *Energy*, 135:900–912, 2017.

[3] B. Bahl, J. Lützow, D. Shu, D. E. Hollermann, M. Lampe, M. Hennen, and A. Bardow. Rigorous synthesis of energy systems by decomposition via time-series aggregation. *Computers & Chemical Engineering*, 112:70–81, 2018.

[4] M. Baldea and I. Harjunkoski. Integrated production scheduling and process control: A systematic review. *Computers & Chemical Engineering*, 71:377–390, 2014.

[5] N. Baumgärtner, B. Bahl, M. Hennen, and A. Bardow. RiSES3:Rigorous synthesis of energy supply and storage systems via time-series relaxation and aggregation. *Computers & Chemical Engineering*, 2019.

[6] M. Beaudin, H. Zareipour, A. Schellenberglabe, and W. Rosehart. Energy storage for mitigating the variability of renewable electricity sources: An updated review. *Energy for Sustainable Development*, 14(4):302–314, 2010.

[7] L. T. Biegler. *Nonlinear Programming*. Society for Industrial and Applied Mathematics, 2010.

[8] D. Bongartz and A. Mitsos. Deterministic global optimization of process flowsheets in a reduced space using McCormick relaxations. *Journal of Global Optimization*, 69(4):761–796, 2017.

[9] D. Bongartz and A. Mitsos. Deterministic global flowsheet optimization: Between equation-oriented and sequential-modular methods. *AIChE Journal*, 65(3):1022–1034, 2019.

[10] D. Bongartz, J. Najman, S. Sass, and A. Mitsos. MAiNGO - McCormick-based algorithm for mixed-integer nonlinear global optimization.

[11] L. C. Bree, K. Perrey, A. Bulan, and A. Mitsos. Demand side management and operational mode switching in chlorine production. *In press: AIChE Journal*, 2018.

[12] P. M. Castro, I. Harjunkoski, and I. E. Grossmann. New continuous-time scheduling formulation for continuous plants under variable electricity cost. *Industrial & Engineering Chemistry Research*, 48(14):6701–6714, 2009.

[13] P. M. Castro, L. Sun, and I. Harjunkoski. Resource-task network formulations for industrial demand side management of a steel plant. *Industrial & Engineering Chemistry Research*, 52(36):13046–13058, 2013.

[14] A. D. Dimitriadis, N. Shah, and C. C. Pantelides. RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. *Computers & Chemical Engineering*, 21:1061–1066, 1997.

[15] J. Du, J. Park, I. Harjunkoski, and M. Baldea. A time scale-bridging approach for integrating production scheduling and process control. *Computers & Chemical Engineering*, 79:59–69, 2015.

[16] O. Exler and K. Schittkowski. A trust region SQP algorithm for mixed-integer nonlinear programming. *Optimization Letters*, 1(3):269–280, 2007.

[17] C. A. Floudas and X. Lin. Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers & Chemical Engineering*, 28(11):2109–2129, 2004.

[18] A. Ghobeity and A. Mitsos. Optimal time-dependent operation of seawater reverse osmosis. *Desalination*, 263(1):76–88, 2010.

[19] I. E. Grossmann and S. Lee. Generalized convex disjunctive programming: Nonlinear convex hull relaxation. *Computational Optimization and Applications*, 26(1):83–100, 2003.

[20] H. Hadera, I. Harjunkoski, G. Sand, I. E. Grossmann, and S. Engell. Optimization of steel production scheduling with complex time-sensitive electricity cost. *Computers & Chemical Engineering*, 76:117–136, 2015.

[21] I. Harjunkoski, C. T. Maravelias, P. Bongers, P. M. Castro, S. Engell, I. E. Grossmann, J. Hooker, C. Mendez, G. Sand, and J. Wassick. Scope for industrial applications of production scheduling models and solution methods. *Computers & Chemical Engineering*, 62:161–193, 2014.

[22] A. Hartwich and W. Marquardt. Dynamic optimization of the load change of a large-scale chemical plant by adaptive single shooting. *Computers & Chemical Engineering*, 34(11):1873–1889, 2010.

[23] B. B. Hubbard. *The world according to wavelets: the story of a mathematical technique in the making*. AK Peters, Ltd., 1996.

[24] M. G. Ierapetritou and C. A. Floudas. Effective continuous-time formulation for short-term scheduling. 1. Multipurpose batch processes. *Industrial & Engineering Chemistry Research*, 37(11):4341–4359, 1998.

[25] M. G. Ierapetritou and C. A. Floudas. Effective continuous-time formulation for short-term scheduling. 2. Continuous and semicontinuous processes. *Industrial & Engineering Chemistry Research*, 37(11):4360–4374, 1998.

[26] M. G. Ierapetritou, D. Wu, J. Vin, P. Sweeney, and M. Chigirinskiy. Cost minimization in an energy-intensive plant using mathematical programming approaches. *Industrial & Engineering Chemistry Research*, 41(21):5262–5277, 2002.

[27] M. H. Karwan and M. F. Keblis. Operations planning with real time pricing of a primary input. *Computers & Operations Research*, 34(3):848–867, 2007.

[28] M. T. Kelley, R. Baldick, and M. Baldea. Demand response operation of electricity-intensive chemical processes for reduced greenhouse gas emissions: Application to an air separation unit. *ACS Sustainable Chemistry & Engineering*, 7(2):1909–1922, 2019.

[29] M. T. Kelley, R. C. Pattison, R. Baldick, and M. Baldea. An MILP framework for optimizing demand response operation of air separation units. *Applied Energy*, 222:951–966, 2018.

[30] E. Kondili, C. C. Pantelides, and R. W. H. Sargent. A general algorithm for short-term scheduling of batch operations - I. MILP formulation. *Computers & Chemical Engineering*, 17(2):211–227, 1993.

[31] L. Kotzur, P. Markewitz, M. Robinius, and D. Stolten. Impact of different time series aggregation methods on optimal energy system design. *Renewable Energy*, 117:474–487, 2018.

[32] D. Kraft. On converting optimal control problems into nonlinear programming problems. In K. Schittkowski, editor, *Computational Mathematical Programming*, pages 261–280. Springer Berlin Heidelberg, 1985.

[33] D. Kraft. Algorithm 733: TOMP – fortran modules for optimal control calculations. *ACM Transactions on Mathematical Software*, 20(3):262–281, 1994.

[34] H. Lund and G. Salgi. The role of compressed air energy storage (CAES) in future sustainable energy systems. *Energy Conversion and Management*, 50(5):1172–1179, 2009.

[35] H. Lund, G. Salgi, B. Elmegaard, and A. N. Andersen. Optimal operation strategies of compressed air energy storage (CAES) on electricity spot markets with fluctuating prices. *Applied Thermal Engineering*, 29(5):799–806, 2009.

[36] X. Luo, J. Wang, M. Dooner, and J. Clarke. Overview of current development in electrical energy storage technologies and the application potential in power system operation. *Applied Energy*, 137:511–536, 2015.

[37] C. E. Lythcke-Jorgensen, M. Münster, A. V. Ensinas, and F. Haglind. A method for aggregating external operating conditions in multi-generation system optimization models. *Applied Energy*, 166:59–75, 2016.

[38] S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$. *Transactions of the American Mathematical Society*, 315(1):69–87, 1989.

[39] C. T. Maravelias. General framework and modeling approach classification for chemical production scheduling. *AIChE Journal*, 58(6):1812–1828, 2012.

[40] C. T. Maravelias and I. E. Grossmann. Minimization of the makespan with a discrete-time state-task network formulation. *Industrial & Engineering Chemistry Research*, 42(24):6252–6257, 2003.

[41] C. T. Maravelias and I. E. Grossmann. New general continuous-time state-task network formulation for short-term scheduling of multipurpose batch plants. *Industrial & Engineering Chemistry Research*, 42(13):3056–3074, 2003.

[42] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I — Convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976.

[43] S. Mitra, I. E. Grossmann, J. M. Pinto, and N. Arora. Optimal production planning under time-sensitive electricity prices for continuous power-intensive processes. *Computers & Chemical Engineering*, 38:171–184, 2012.

[44] S. Mitra, J. M. Pinto, and I. E. Grossmann. Optimal multi-scale capacity planning for power-intensive continuous processes under time-sensitive electricity prices and demand uncertainty. Part I: Modeling. *Computers & Chemical Engineering*, 65:89 –101, 2014.

[45] S. Mitra, L. Sun, and I. E. Grossmann. Optimal scheduling of industrial combined heat and power plants under time-sensitive electricity prices. *Energy*, 54:194–211, 2013.

[46] A. Mitsos, N. Asprion, C. A. Floudas, M. Bortz, M. Baldea, D. Bonvin, A. Caspari, and P. Schäfer. Challenges in process optimization for new feedstocks and energy sources. *Computers & Chemical Engineering*, 113:209–221, 2018.

[47] A. Mitsos, B. Chachuat, and P. Barton. McCormick-based relaxations of algorithms. *SIAM Journal on Optimization*, 20(2):573–601, 2009.

[48] P. Nahmmacher, E. Schmid, L. Hirth, and B. Knopf. Carpe diem: A novel approach to select representative days for long-term power system modeling. *Energy*, 112:430–442, 2016.

[49] R. C. Pattison, C. R. Touretzky, T. Johansson, I. Harjunkoski, and M. Baldea. Optimal process operations in fast-changing electricity markets: Framework for scheduling with low-order dynamic models and an air separation application. *Industrial & Engineering Chemistry Research*, 55(16):4562–4584, 2016.

[50] J. M. Pinto and I. E. Grossmann. A continuous time mixed integer linear programming model for short term scheduling of multistage batch plants. *Industrial & Engineering Chemistry Research*, 34(9):3037–3051, 1995.

[51] A. Prata, J. Oldenburg, A. Kroll, and W. Marquardt. Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. *Computers & Chemical Engineering*, 32(3):463–476, 2008.

[52] R. W. H. Sargent and G. R. Sullivan. The development of an efficient optimal control package. In J. Stoer, editor, *Optimization Techniques*, pages 158–168. Springer Berlin Heidelberg, 1978.

[53] G. Schilling and C. C. Pantelides. A simple continuous-time process scheduling formulation and a novel solution algorithm. *Computers & Chemical Engineering*, 20:1221–1226, 1996.

[54] M. Schlegel and W. Marquardt. Adaptive switching structure detection for the solution of dynamic optimization problems. *Industrial & Engineering Chemistry Research*, 45(24):8083–8094, 2006.

[55] M. Schlegel and W. Marquardt. Detection and exploitation of the control switching structure in the solution of dynamic optimization problems. *Journal of Process Control*, 16(3):275–290, 2006.

[56] M. Schlegel, K. Stockmann, T. Binder, and W. Marquardt. Dynamic optimization using adaptive control vector parameterization. *Computers & Chemical Engineering*, 29(8):1731–751, 2005.

[57] A. M. Schweidtmann, W. R. Huster, J. T. Lüthje, and A. Mitsos. Deterministic global process optimization: Accurate (single-species) properties via artificial neural networks. *Computers & Chemical Engineering*, 121:67–74, 2019.

[58] A. M. Schweidtmann and A. Mitsos. Deterministic global optimization with artificial neural networks embedded. *Journal of Optimization Theory and Applications*, 180(3):925–948, 2019.

[59] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.

[60] H. Teichgraeber and A. R. Brandt. Clustering methods to find representative periods for the optimization of energy systems: An initial framework and comparison. *Applied Energy*, 239:1283–1293, 2019.

[61] C. Tsay and M. Baldea. Learning latent variable dynamic models for integrated production scheduling and control. 2019. arXiv:1904.04796.

[62] C. Tsay, A. Kumar, J. Flores-Cerrillo, and M. Baldea. Optimal demand response scheduling of an industrial air separation unit using data-driven dynamic models. *Computers & Chemical Engineering*, 126:22–34, 2019.

[63] A. Tsoukalas and A. Mitsos. Multivariate McCormick relaxations. *Journal of Global Optimization*, 59(2):633–662, 2014.

[64] S. Velez and C. T. Maravelias. Multiple and nonuniform time grids in discrete-time MIP models for chemical production scheduling. *Computers & Chemical Engineering*, 53:70–85, 2013.

[65] A. Wächter and L. T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, 2006.

[66] Q. Zhang and I. E. Grossmann. Enterprise-wide optimization for industrial demand side management: Fundamentals, advances, and perspectives. *Chemical Engineering Research and Design*, 116(Supplement C):114–131, 2016.

[67] Q. Zhang, I. E. Grossmann, C. F. Heuberger, A. Sundaramoorthy, and J. M. Pinto. Air separation with cryogenic energy storage: Optimal scheduling considering electric energy and reserve markets. *AIChE Journal*, 61(5):1547–1558, 2015.

[68] Q. Zhang, I. E. Grossmann, A. Sundaramoorthy, and J. M. Pinto. Data-driven construction of convex region surrogate models. *Optimization and Engineering*, 17(2):289–332, 2016.

[69] Q. Zhang, M. F. Morari, I. E. Grossmann, A. Sundaramoorthy, and J. M. Pinto. An adjustable robust optimization approach to scheduling of continuous industrial processes providing interruptible load. *Computers & Chemical Engineering*, 86:106–119, 2016.

[70] Q. Zhang, A. Sundaramoorthy, I. E. Grossmann, and J. M. Pinto. A discrete-time scheduling model for continuous power-intensive process networks with various power contracts. *Computers & Chemical Engineering*, 84:382–393, 2016.

A Further details on the grid-adaptation algorithm

A.1 Fast wavelet transform

The specific case of the wavelet transform for piecewise-constant signals (i.e., signals in the Haar-Basis that naturally occur in scheduling with quasi-stationary models) is presented below.

Let the argument of the wavelet transform (5) be denoted by $(\tilde{d}^{s*}_{i,1}, \dots, \tilde{d}^{s*}_{i,T})^T$, i.e.

$$(\tilde{d}^{s*}_{i,1}, \dots, \tilde{d}^{s*}_{i,T})^T = \boldsymbol{B}^s \cdot (\hat{d}^{s*}_{i,1}, \dots, \hat{d}^{s*}_{i,\hat{T}})^T.$$

Note that $\tilde{d}^{s*}_{i,1}$ does not denote the optimal value of d_i in the first interval, but rather in the interval with the highest electricity price. This vector is now rearranged and multiplied with the matrix $\boldsymbol{H}$.

$$\begin{pmatrix} v^s_{1,1} & \dots & v^s_{1,\frac{T}{2}} \\ v^s_{2,1} & \dots & v^s_{2,\frac{T}{2}} \end{pmatrix} = \boldsymbol{H} \cdot \begin{pmatrix} \tilde{d}^{s*}_{i,1} & \dots & \tilde{d}^{s*}_{i,T-1} \\ \tilde{d}^{s*}_{i,2} & \dots & \tilde{d}^{s*}_{i,T} \end{pmatrix}$$

$$\boldsymbol{H} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

The first row $(v^s_{1,1}, \dots, v^s_{1,\frac{T}{2}})^T$ thus contains (scaled) pairwise averages of the solution vector and the second row $(v^s_{2,1}, \dots, v^s_{2,\frac{T}{2}})^T$ (scaled) pairwise differences, which are the wavelet coefficients $\lambda^s_{N-1,b}$.

$$(v^s_{2,1}, \dots, v^s_{2,\frac{T}{2}})^T = (\lambda^s_{N-1,1}, \dots, \lambda^s_{N-1,\frac{T}{2}})$$

This procedure is repeated N-times setting $(v^s_{1,1}, \dots, v^s_{1,\frac{T}{2}})^T$ as the new $(\tilde{d}^{s*}_{i,1}, \dots, \tilde{d}^{s*}_{i,T})^T$ and recursively calculating the $\lambda^s_{a,b}$'s. In the N'th iteration, the remaining pairwise average then equals $\lambda^s_{-1,0}$, which corresponds to the average of the entire time series.

A.2 Grid point deletion and insertion algorithms

Let $\boldsymbol{\lambda}^s$ denote a vector containing all wavelet coefficients $\lambda^s_{a,b}$. Then, a suitable criterion for deletion that uses a threshold $0 \leqslant e_d \leqslant 1$ reads:

$$|\lambda^s_{a,b}| \leqslant e_d \left\|\boldsymbol{\lambda}^s\right\|_2 \quad \forall a, b.$$

If a wavelet coefficient $\lambda^s_{a,b}$ is selected for deletion, the corresponding basis function $\psi_{a,b}$ will be discarded when deriving the matrix $\boldsymbol{B}^{s+1}$ for the next iteration (cf., Appendix A.3).

For insertion, only the boundary coefficients are considered. The coefficient $\tilde{\lambda}^s_{a,b}$ is called a boundary coefficient if $\tilde{\lambda}^s_{a+1,2b} = 0 \vee \tilde{\lambda}^s_{a+1,2b+1} = 0$ holds. Let the boundary coefficients further be ordered, such

that $|\tilde{\lambda}^s_{a,b,j}| \geqslant |\tilde{\lambda}^s_{a,b,j+1}|$ and let there be vectors $\tilde{\boldsymbol{\lambda}}^s$ containing all boundary coefficients and $\tilde{\boldsymbol{\lambda}}^s_j$ containing the j highest boundary coefficients. Then, we apply the following criterion for insertion:

$$\left\|\tilde{\boldsymbol{\lambda}}^s_j\right\|_2 \geqslant e_i \left\|\tilde{\boldsymbol{\lambda}}^s\right\|_2 \quad \forall j.$$

If a wavelet coefficient $\lambda^s_{a,b}$ is selected for insertion, the two basis function $\psi_{a+1,2b}$ and $\psi_{a+1,2b+1}$ will be included when deriving the matrix $\boldsymbol{B}^{s+1}$ for the next iteration (cf., Appendix A.3).

Selecting the same coefficient for both deletion and insertion during one iteration is excluded. Furthermore, we only allow for deletion of boundary coefficients.

A.3 Illustration of the construction of the mapping matrix

The matrix $\boldsymbol{B}^{s+1}$ is constructed in a recursive manner, starting from $\boldsymbol{B}^{s+1} = 1$ and $a = 0$ and iterating from $b = 0$ to $b = 2^a - 1$. A wavelet basis function $\psi_{a,b}$ that shall be considered for $\boldsymbol{B}^{s+1}$ adds both a row and a column to the matrix. A neglected basis function $\psi_{a,b}$ in contrast only adds a new row. For instance, if $\psi_{0,0}$ shall be used, the matrix is adjusted to:

$$\boldsymbol{B}^{s+1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

and we set $a = 1$. Now, we assume that - in accordance with the example in Section 3 - $\psi_{1,0}$ shall be used, whereas $\psi_{1,1}$ shall be unused. The first adjustment then results in:

$$\boldsymbol{B}^{s+1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

The second step gives:

$$\boldsymbol{B}^{s+1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

The procedure is repeated N times. Thereby, a matrix $\boldsymbol{B}^{s+1}$ is finally obtained with 2^N rows and as many columns as wavelet basis functions that shall be considered in iteration $s + 1$.