DeLoop: Decomposition-based Long-term Operational Optimization of energy systems with time-coupling constraints

Nils Baumgärtner[a], David Shu[b], Björn Bahl[a], Maike Hennen[a], Dinah Elena Hollermann[a] and André Bardow[a,b,*]

[a]*Institute of Technical Thermodynamics, RWTH Aachen University, Aachen, Germany*

[b]*Institute of Energy and Climate Research – Energy Systems Engineering (IEK-10), Forschungszentrum Jülich, Jülich, Germany*

**andre.bardow@ltt.rwth-aachen.de*

Abstract

Long-term operational optimization of energy systems results in challenging, large-scale problems. These large-scale problems can be directly decomposed into smaller subproblems, in the absence of time-coupling constraints and variables. However, time-coupling is common in energy systems, e. g. due to (seasonal) energy storage and peak-power prices. To solve time-coupled long-term operational optimization problems, we propose the method DeLoop for the Decomposition-based Long-term operational optimization of energy systems with time-coupling. DeLoop calculates feasible solutions (upper bounds) by decomposing the operational optimization problem into smaller subproblems. The solutions of these subproblems are recombined to obtain a feasible solution for the original long-term problem. To evaluate the quality of the feasible solutions, DeLoop computes lower bounds by linear programming relaxation. DeLoop iteratively decreases the number of subproblems and employs the Branch-and-Cut procedure to tighten the bounds. In a case study of an energy system, DeLoop converges fast, outperforming a commercial state-of-the-art solver by a factor of 32.

Keywords: large-scale MILP, complicating constraints, seasonal storage, net connection fees, emission targets, time-coupling, solution method

1. Introduction

On-site energy systems are increasingly widespread in both the industrial and residential sectors. Optimization of these energy systems can lead to significant economic and ecological benefits (Somma et al., 2015). The optimization of energy systems is a two-stage problem consisting of a design stage and an operational stage (Lin et al., 2016). In design optimization, both stages needs to be taken into account. Thus, design optimization also includes operational optimization, whereas operational optimization is solved based on a fixed design. In this work, we focus on the operational optimization of energy systems with a fixed design, as the proposed method is suited for optimal operation. However, to extend the scope to optimal design problems, the proposed method can be implemented into the second stage of two-stage algorithms that include the design optimization in the first stage , e.g. in Bahl et al. (2018a); Baumgärtner et al. (2019a); Baumgärtner et al. (2019b).

Typically, optimization problems of energy systems are formulated as a mixed-integer linear programming (MILP) problem (Elia and Floudas, 2014). For small-scale MILP problems, state-of-the-art solution software is able to quickly provide solutions (Grossmann, 2012). The computational effort of the operational problems, however, increases exponentially with the considered problem size; in fact, operational problems of energy systems are at least weakly NP-hard (Goderbauer et al., 2019). For operational optimization of energy systems, the size of a MILP problem depends on the number of considered time steps. Thus, the many time steps in long-term operational optimization lead to large-scale MILP problems. A common solution approach to solve these large-scale MILP problems is to split them into smaller periods (direct decomposition). The decomposed periods can be solved independently and often in acceptable time. However, direct decomposition is only possible if no time-coupling constraints or variables are present (Bradley et al., 1992). For industrial energy systems, time-coupling constraints and variables are very common, e. g. by peak-power prices, network connection fees, and cogeneration subsidies (Bischi et al., 2019). To still solve large-scale MILP operational problems with time-coupling constraints, various solution methods exist. Prominent solutions methods are given in Tab.1.

Table 1: Examples of solution methods for long-term operational optimizations with time-coupling constraints and variables

Method	**Examples**	**Strengths (+) / Limits (−)**
Model simplifications	Piacentino and Cardona (2008), Yokoyama (2013)	+ easily applicable − problem specific
Non-deterministic	Kazarlis et al. (1996), Park et al. (2000), Kavvadias and Maroulis (2010), Renaldi and Friedrich (2017), Bischi et al. (2019)	+ allows parallelization + accurate model − no quality measure
Decomposition	Yokoyama and Ito (1996), Al-Agtash and Su (1998), Rong et al. (2008), Nasrolahpour et al. (2016), Wang et al. (2016)	+ allows parallelization + accurate model + provide quality measure − slow convergence − complex formulation

Model simplifications solve such long-term operational problems by simplifying component models to reduce the complexity of the optimization problem. For example, Piacentino and Cardona (2008) relax binary variables, and Yokoyama (2013) converts an MINLP problem into an MILP problem. These simplifications are easy to implement and render the long-term optimization problem solvable, even when considering time-coupling constraints and variables. However, model simplifications limit physical accuracy of optimization models, which may lead to suboptimal or even infeasible solutions in practice.

To prevent suboptimal or infeasible solutions, physical accurate models might be necessary. For these more accurate optimization models, non-deterministic methods can be employed to solve the resulting complex optimization problems. Genetic algorithms have been used for unit commitment (Kazarlis et al., 1996), for capacity expansion (Park et al., 2000), and for trigeneration (Kavvadias and Maroulis, 2010). Bischi et al. (2019) use a rolling horizon approach including long-term foresight by typical weeks for the operational optimization considering cogeneration subsidies. Renaldi and Friedrich (2017) optimize the operation of short- and long-term storage systems by using 2 time grids, one with short and the other with long time intervals per time step. The advantage of these non-deterministic methods is that they normally yield good solutions for difficult problems and often allow fast computation, e. g., by parallelization. However, in general, it is not possible to evaluate the quality of the proposed solution (Hanne and Dornberger, 2017).

As an alternative, decomposition approaches can be used to obtain solutions of known solution quality. Decomposition methods have been applied to the optimization of energy systems with coupling constraints and variables. Problems with coupling constraints have been decomposed by Dantzig-Wolfe decomposition (Yokoyama and Ito, 1996) and Lagrangian relaxation (Al-Agtash and Su, 1998; Rong et al., 2008). To decompose problems with coupling variables, Benders' decomposition is suitable (Nasrolahpour et al., 2016). Modified decomposition methods can cope with both time-coupling constraints and variables in MILP optimizations. To solve problems with both coupling constraints and variables, Wang et al. (2016) apply a combination of Lagrangian relaxation and Benders' decomposition .

Although much progress has been made in automated Dantzig-Wolfe decomposition, it is still hard to automatically identify an appropriate decomposition. Further, solution times remain slow for many problems (Bergner et al., 2015). Lagrangian methods require good multipliers for fast convergence. However, the selection of Lagrangian multipliers is difficult, which has lead to the development of a large variety of methods (Conejo et al., 2006). Benders' decomposition has slow convergence due to many slow iterations. Although methods to improve convergence of Benders' decomposition have been proposed, its formulation and implementation remain complex (Rahmaniani et al., 2017).

In this paper, we propose a time-series **De**composition-based solution method for **L**ong-term **o**perational **op**timization problems (**DeLoop**). The proposed solution method DeLoop copes with both time-coupling constraints and variables in operational optimization.

DeLoop combines the strengths of the approaches in Tab.1: DeLoop is easily applicable, uses a physical accurate model, allows for parallelization, and shows fast convergence. A preliminary version of the proposed time-series decomposition-based solution algorithm DeLoop has been presented in a conference paper (Baumgärtner et al., 2019b). Here, we improve convergence of the method, update the case study, and describe all method details.

In Section 2, we state a generic long-term operational optimization problem with time-coupling constraints. The proposed method DeLoop is presented in detail in Section 3. In Section 4, we apply DeLoop to a real-world based industrial operational problem and validate the results by a computational study. We conclude our findings in Section 5.

2. Time-coupling constraints and variables in operational optimization

In this section, we state the operational problem for energy systems as a mixed-integer linear programming (MILP) problem. The MILP formulation is a common problem class investigated in literature for the optimization of energy systems (Elia and Floudas, 2014). The stated problem is based on the previous work of our group (Voll et al., 2013; Bahl et al., 2017, 2018b; Baumgärtner et al., 2019a). Here, we extend the model by time-coupling constraints: emission limits, seasonal storage, network connection fees, and peak-power prices. Commonly, these time-coupling constraints extend over one year of operation. However, our method can handle time-coupling constraints on arbitrary time horizons. For readability, we present here a simplified version of the problem. The detailed model formulation can be found in the Supplementary Material A.

We optimize the operational expenditures *OPEX* of the energy system Eq. (1)

$$\min_{\dot{V}_{n,t},\delta_{n,t},\dot{V}_{\text{grid}}^{\max},\dot{V}_{\text{grid},t},V_{n,t},x,y} OPEX$$

with

$$OPEX := \overbrace{\sum_{t\in\mathscr{T}}\left(\Delta t_t \sum_{n\in\mathscr{C}} c^{\text{o}}_{n,t}\cdot\frac{\dot{V}_{n,t}}{\eta_n}\right)}^{fuel/electricity\ costs} + \overbrace{\sum_{n\in\mathscr{C}} M_n^{\text{N}}}^{maintenance\ costs} + \underbrace{c_{\text{net}}^{\text{(low/high)}}\cdot\sum_{t\in\mathscr{T}}\Delta t_t \dot{V}_{\text{grid},t}}_{network\ connection\ fees} + \underbrace{c_p^{\text{(low/high)}}\cdot\dot{V}_{\text{grid}}^{\max}}_{peak-power\ costs} \tag{1}$$

The operational expenditures, Eq. (1), typically sum fuel and electricity costs, maintenance costs M_n^{N} for all components n, network connection fees, and peak-power costs. Fuel and electricity costs are the sum of the output power $\dot{V}_{n,t}$ divided by the efficiency η_n of every component $n \in \mathscr{C}$ for every time step t and multiplied by the specific operation cost $c^{\text{o}}_{n,t}$ and the duration Δt_t of a time step (Eq. (1)). The maintenance costs M for a unit n are constant and given in the Supplementary Material A. The network connection fees result from the electricity consumption $\sum_{t\in\mathcal{T}} \dot{V}_{\text{grid},t}$ multiplied by network connection fees $c_{\text{net}}^{\text{(low/high)}}$ that depend on the time of utilization. The peak power $\dot{V}_{\text{grid}}^{\max}$ multiplied by a peak-power price $\mathrm{c_p}$ results in additional operational expenditures *OPEX* (Eq. (1)). The peak-power price $\mathrm{c_p}$ also depends on the time of utilization.

The optimization problem is subject to constraints, which are presented in the following.

$$\sum_{n\in\mathcal{C}\setminus\mathcal{C}_{\text{stor}}} \dot{V}_{n,t} + \sum_{n\in\mathcal{C}_{\text{stor}}} (\dot{V}_{n,t}^{\text{out}} - \dot{V}_{n,t}^{\text{in}}) = \dot{E}_t, \qquad \forall t \in \mathcal{T}, \tag{2}$$

$$A_1\dot{V}_{n,t} + \widetilde{A}_1\delta_{n,t} \leq b_1, \qquad \forall t \in \mathcal{T}, \forall n \in \mathcal{C}, \tag{3}$$

$$A_2 x + \widetilde{A}_2 y \leq b_2, \qquad \forall t \in \mathcal{T}, \forall n \in \mathcal{C}, \tag{4}$$

$$V_{n,t} + \Delta t_t \cdot (\dot{V}_{n,t}^{\text{in}} - \dot{V}_{n,t}^{\text{out}}) = V_{n,t+1}, \qquad \forall t \in \mathcal{T}, \forall n \in \mathcal{C}_{\text{stor}}, \tag{5}$$

$$|\dot{V}_{\text{grid},t}| \leq \dot{V}_{\text{grid}}^{\max}, \qquad \forall t \in \mathcal{T}, \tag{6}$$

$$\mathrm{c_j} = \begin{cases} c_{\mathrm{j}}^{\text{high}} & ,\text{if } \frac{\sum_{t\in\mathcal{T}} \dot{V}_{\text{grid},t}\cdot\Delta \mathrm{t}_t}{\dot{V}_{\text{grid}}^{\max}} \leq 2500\text{ h} \\ c_{\mathrm{j}}^{\text{low}} & ,\text{otherwise} \end{cases} \qquad \mathrm{j} \in \{\text{net}, \mathrm{p}\}, \tag{7}$$

$$\sum_{t\in\mathcal{T}} \Delta\mathrm{t}_t \cdot \left[\left(\dot{V}_{\text{grid},t} \cdot c_{t,el}^{\text{CO2}}\right) + \sum_{n\in\mathcal{C}_{gas}} \left(\frac{\dot{V}_{n,t}}{\eta_n} \cdot c_{t,gas}^{\text{CO2}} \right) \right] \leq C_{\text{CO2}}^{\max}, \tag{8}$$

$$\delta_{n,t}, y \in \{0,1\}; x \in \mathbb{R}^a; \dot{V}_{n,t} \in \mathbb{R}; V_{n,t}, \dot{V}_{\text{grid}}^{\max} \in \mathbb{R}^+ \qquad \forall t \in \mathcal{T}, \forall n \in \mathcal{C}.$$

Eq. (2) represents the energy balance. The energy balance enforces that the component's output power $\dot{V}_{n,t}$ plus the net energy output of the storage units $\dot{V}_{n,t}^{\text{out}} - \dot{V}_{n,t}^{\text{in}}$ meet the energy demand $\dot{E}_t$ for every time step t. Eq. (3) is the generic representation of further (in)equalities with the coefficient matrices A_1, $\widetilde{A}_1$, and the vector b_1. These (in)equalities determine the component's binary on/off status $\delta_{n,t}$ and the current part-load performance. Surrogate Eq. (4) summarizes additional constraints such as constraints for describing unit behavior, e. g the photovoltaics output limitation, or the choice to either charge or discharge storage systems. The detailed MILP model of the low-carbon utility system, including part-load behavior, minimal loads, and linearizations, is presented in Supplementary Material A.

Long term operational optimization is challenging due to both time coupling constraints and variables which occur in the following equations. In the storage energy balance Eq. (5), the net energy input of the storage units $\dot{V}^{\text{in}}_{n,t} - \dot{V}^{\text{out}}_{n,t}$ couples the current storage level $V_{n,t}$ to the future storage level $V_{n,t+1}$. Thereby, the storage balance is a coupling equation for the entire time series.

Eq. (6) ensures that the peak power $\dot{V}^{\text{max}}_{\text{grid}}$ is the maximum power exchanged with the grid. Thus, $\dot{V}^{\text{max}}_{\text{grid}}$ is a time-coupling variable for the entire time series. The network connection fees as well as the peak-power price depend on the time of utilization, e. g in Germany (StromNEV, 2005). Eq. (7) determines the price $c_{\text{i}}^{\text{(low/high)}}$ as a function of the time of utilization. The dependence of prices on the time of utilizationis reformulated using MILP constraints, resulting in both time-coupling constraints and variables. Eq. (8) sets an upper limit for greenhouse gas emissions $C^{\text{max}}_{\text{CO2}}$ from electricity usage and all gas-consuming units $\mathbb{C}_{gas}$. This emission limit represents a coupling constraint for the entire time series.

Overall, Eq. (5-8) represents a long-term operational problem that includes long-term time-coupling constraints and variables. In the next section, we propose the time-series decomposition-based solution algorithm DeLoop to solve such long-term operational problems.

3. Time-series decomposition for long-term operational optimization

The proposed time-series decomposition-based solution algorithm, DeLoop, generates feasible solutions of long-term operational optimization problems with known solution quality. In this section, we first present the idea underlying DeLoop and then the details of the time-series decomposition.

3.1. Solution method

DeLoop provides lower and upper bounds for the operational optimization problem (Eq. (1-8)). Deloop computes these lower and uppd bounds in a parallel computing mode (Fig. 1). After each calculating of a new upper bound, we calculate the optimality gap ε and check if the desired optimality gap ε_{DeLoop} is satisfied.

$$\varepsilon := \frac{OPEX^{\text{upper bound}} - OPEX^{\text{lower bound}}}{OPEX^{\text{upper bound}}} \leq \varepsilon_{DeLoop}. \tag{9}$$

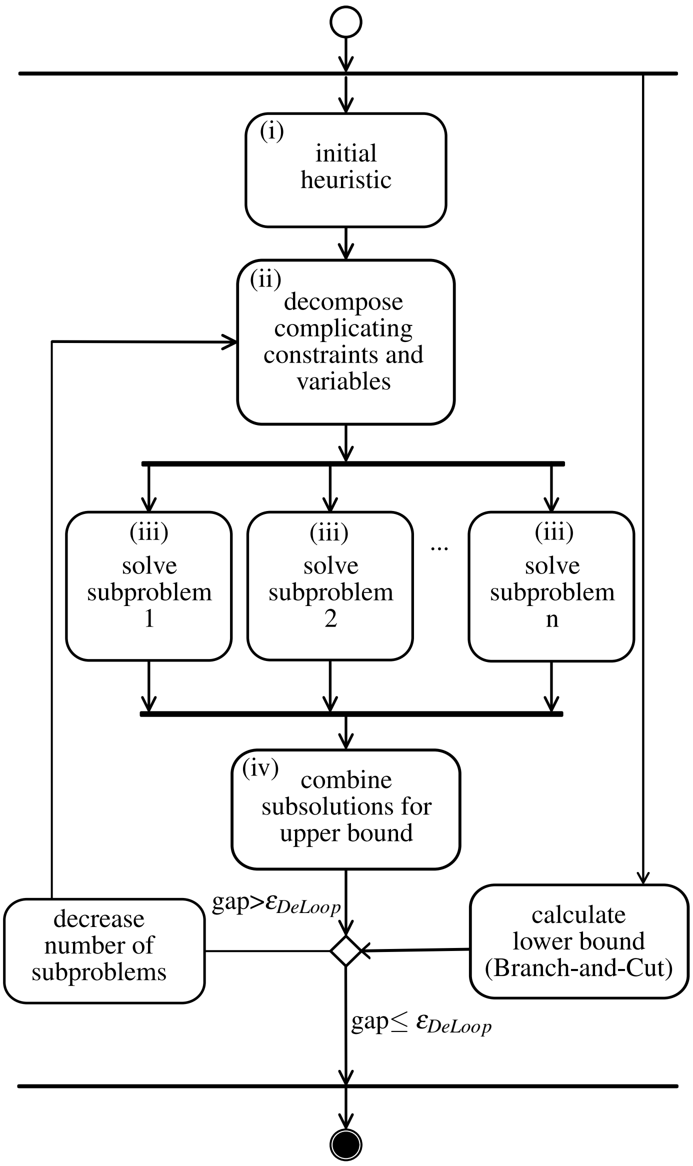

Figure 1: Overview of the proposed time-series decomposition method DeLoop to create feasible solutions of operational optimization problems with known solution quality.

Upper bounds are obtained from feasible solutions. These feasible solutions are computed by decomposing the original operational problem into smaller subproblems. The upper bound is tight if the impact of time-coupling constraints and variables is small. The tightness of the upper bound is commonly unknown prior to optimization. The upper bounds are tightened if the desired optimality gap ε_{DeLoop} is not satisfied. DeLoop tightens the upper bounds by iteratively decreasing the number of subproblems and thereby DeLoop improves the representation of time coupling within the decomposition. In general, the original long-term operation problem can be decomposed into any number of subproblems (less than or equal to the size of the time series T). Each of the subpoblems represents a subsequence of the full time series. However, the subproblems should preferably have a similar number of time steps to achieve similar solution times when solved in parallel. For simplicity, we only allow integer divisors of the time steps T as possible numbers of subproblems, which leads to subproblems of the same size. The algorithm for upper bounds is described in Section 3.2.

For the lower bounds, we relax all binary variables ($\delta_{n,t}, y \in \{0,1\}$) of the operational problem (Eq. (1-8)), thereby converting the complex MILP into an LP which can be solved efficiently. This solution of the relaxed problem serves as the first lower bound and is the root node for the Branch-and-Cut procedure. The lower bound is tight if the impact of binary decision variables is small. The tightness of the lower bound is commonly unknown prior to optimization. Subsequently to tighten the lower bound, the Branch-and-Cut procedure starts by branching on binary variables and cutting off branches that cannot improve the solution (IBM Cooperation, 2016).

After each calculated upper bound, DeLoop compares the current upper bound to the current lower bound to calculate the optimality gap ε. The iterative calculation of the lower and upper bounds stops when an optimality gap ε_{DeLoop} is satisfied. As a post-processing step, the solution quality can be further enhanced by warm-starting the original problem with the final solution of DeLoop.

3.2. Time-series decomposition for the upper bound

DeLoop obtains feasible solutions (upper bounds) of the original long-term operational problem (Eq. (1-8)) in four steps (Fig. 1):

(i) Initial heuristic to initialize the number of subproblems
(ii) Decomposition of time-coupling constraints and variables
(iii) Optimizing subproblems in parallel computing mode
(iv) Combining subsolutions to upper bound

In step (i), we select an initial number of subproblems. Subsequently, in step (ii), DeLoop decomposes the long-term operational problem including its complicating constraints and variables into subproblems. The decomposition into subproblems reduces the overall complexity of the operational problem. Thus, the subproblems can be solved efficiently in parallel computing mode in step (iii). In step (iv), the solutions of all subproblems are combined into a feasible solution of the original operational problems, resulting in an upper bound. In the following, we present the details of steps (i-iv) of the proposed time-series decomposition method DeLoop.

Step (i): Initial heuristic

Initialization of the decomposition method DeLoop requires an initial number of subproblems. In principle, one could always start with the maximum number of subproblems. In practice, we found it more efficient to identify an initial number of subproblems, leading to minimal computational time for the first solution. To identify a good estimate for this initial number, a heuristic is employed. The heuristic only defines the initial number for the decomposition. In our experience, the choice of the initial number only affects the convergence but not the final optimal solution. The algorithm ensures that the solution always satisfies the desired optimality gap. Increasing the number of subproblems decreases the size of the individual subproblems and therefore also the calculation time per subproblem. However, more subproblems have to be solved in parallel by a limited number of cores, which might again increase the overall computation time. Thus, the calculation time is minimal for a particular number of subproblems.

To efficiently identify a good number of subproblems, leading to low computational time to generate a first feasible solution, we employ a heuristic that: tests decreasing numbers of subproblems, starting with the maximum number of subproblems. For each tested decomposition, we decompose the problem into subproblems but solve only one of the resulting subproblems. The time for the decomposition and the solution time of the one subproblem are recorded and extrapolated to the full number of subproblems. DeLoop decreases the number of subproblems until the extrapolated time increases. Thereby, DeLoop estimates the number of subproblems, resulting in minimal calculation time at low computational costs. The identified number of subproblems is selected as the initial number of subproblems.

Step (ii): Decomposition of time-coupling constraints and variables

First, we decompose all complicating constraints and variables of the subproblems. The decomposition is generic and conducted automatically based on the type of time-coupling constraint or variable in 4 steps:

(1) For storage-like constraints (Eq. (5)), we fix start and end values of the storage level variable for each subproblem. To ensure feasibility, the end value of each subproblem has to be consistent with the start value of the subsequent subproblem. In the initial decomposition, the variables representing the start and end values are fixed to one identical but arbitrary value for all subproblems. In the subsequent decompositions, the fixed corresponding values are changed to the values resulting from the preceding optimization. By this adaptation, DeLoop iteratively improves the solution. We illustrate this iterative improvement for 5 and 4 subproblems in Fig. 2.

(2) Time-coupling variables like (Eq. (6-7)) are substituted by independent variables in every subproblem. Thus, for example, network connection fees and the peak-power prices are calculated for each subproblem separately. Thereby, the resulting subproblems become independent of each other.

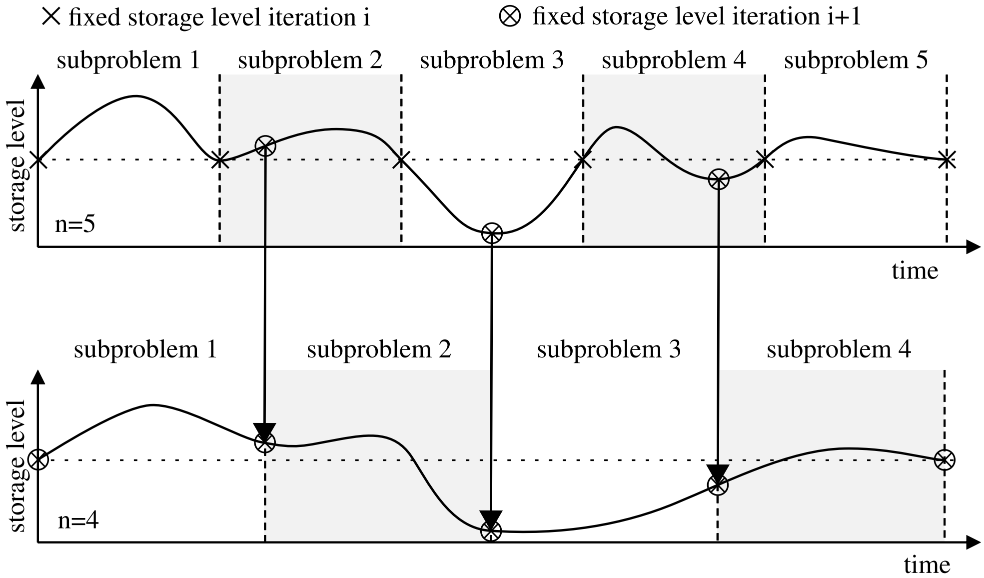

Figure 2: In DeLoop, start and end values for storage-like constraints are fixed in each subproblem. In the subsequent decomposition with fewer subproblems (n=4), these start and end values are taken from the previous optimization results with more subproblems (n=5). Storage levels are piecewise linear in the actual model.

(3) For time-coupling constraints, like the emission limits in Eq. (8), a fraction of the limit is allocated to each subproblem. However, equally distributed limits may lead to poor overall convergence or even infeasible subproblems. Therefore, DeLoop computes individual limits for each subproblem. To compute individual subproblem limits, e. g of emissions, we first identify the minimal possible emissions by an independent minimization of the emissions within each subproblem. Then, the total limit is divided proportionally according to the level of the minimal possible emissions of each subproblem.

(4) To improve performance, we extend each subproblem by aggregated time steps. The aggregated time steps represent the missing parts of the full time series in each subproblem, but only in an averaged manner. Furthermore, critical time steps can be added to each subproblem. Critical time steps are case-study specific and often unknown beforehand. For the optimization of the network connection fees and the peak-power price, the peak-power demand represents a critical time step. Thus, we add a critical time step for the expected peak-power demand. Here, the expected peak-power demand is approximated based on the power demand, the maximal power produced, and the minimal power consumed by all units. For detailed description of the identification of see peak power demand see the Supplementary Material B. Thus, in each subproblem, the corresponding time interval of the long-term operational optimization is solved with the full accuracy of the time series, and the rest with low accuracy. For the low-accuracy representation, we use 3 aggregated time steps for each subproblem: (1) one averaged time step for all time steps before the time interval corresponding to the subproblem, (2) one averaged time step between the subproblem and the critical time step, and (3) one averaged time step after the critical time step. Thus, 3 aggregated time steps are added to each subproblem, except for the first and last subproblem, where only 2 aggregated time steps are added. In Fig. 3, we illustrate this subproblem extension for 5 subproblems and the power demand.

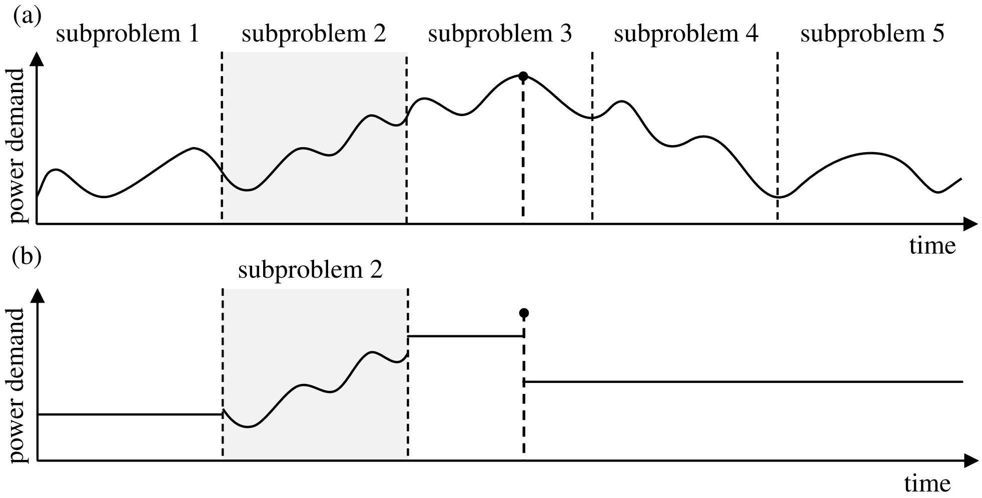

Figure 3: Illustration of the subproblem extension for 5 subproblems for the power demand time series. a) The full time series is decomposed into five intervals, each belonging to a subproblem. The time step with the peak-power demand is a potential critical time step and is highlighted by a dot. b) The time series of subproblem 2 is extended by the critical time step and by aggregated time steps represented by average values, respectively.

Step (iii): Optimizing subproblems in parallel computing mode

DeLoop decomposes the original problem into smaller and independent subproblems that can be quickly solved in parallel computing mode. To further enhance computational speed, the solution of the independent minimization of minimal emissions (step (ii)(3)) is used to warm-start the optimization of each subproblem.

Step (iv): Combining subsolutions to upper bound

In step (iv), the solutions of the subproblems are combined into a feasible solution for the original operational problem. For this purpose, the parts of the subsolutions with low accuracy are discarded, i.e., the added aggregated and critical time steps (step (ii) (4)). Subsequently, we merge all subsolutions: for each time interval, the subsolution is used that has solved this time intervall with the full accuracy. After these adaptations, the combination of all subproblems is a feasible solution. The combination yields a feasible solution as all time-coupling constraints and variables needed for feasibility have been decomposed and fixed to the same values in the subproblems, as described in step (ii).

Then, the time-coupling variables, such as the peak-power demand $\dot{V}_{\text{grid}}^{\text{max}}$, are recalculated by taking all merged subproblems. The recalculation contains only fixed parameters; thus, no further optimization is required. As the time-coupling constraints and variables have been fixed and the objective function is recalculated, the solution is an upper bound for the original operational problem.

In the next section, we apply DeLoop to a complex long-term operational optimization problem with time-coupling constraints and variables.

4. Case study

To validate the proposed time-series decomposition method, we apply DeLoop to a long-term operational MILP problem based on the synthesis problem presented in Baumgärtner et al. (2019a). We optimize an industrial energy system with 3 boilers, 5 compression chillers, 4 absorption chillers, 7 heat exchangers, 3 inverters, 1 combined heat and power engine, 1 photovoltaic system, 1 battery, and 2 storage tanks, one for hot and one for cold water (Fig. 4). A table with all capacities of the units can be found in Supplementary Material A. All results are generated using 4 Intel-Xeon CPUs at 3 GHz and 64 GB RAM. All MILP problems are solved using CPLEX 12.6.3.0 (IBM Cooperation, 2016), and the model is written in General Algebraic Modeling System 24.7.3 (GAMS Development Corporation, 2019). For this case study, 4 cores are employed to solve subproblems in parallel. We consider one year of operation, because time-coupling constraints and variables for industrial energy systems often span one year. The time-series have a two-hourly resolution for demands of steam, low-temperature heating, cooling, and electricity, as well as solar radiation and electricity grid prices (Fig. 5) from Baumgärtner et al. (2019a).

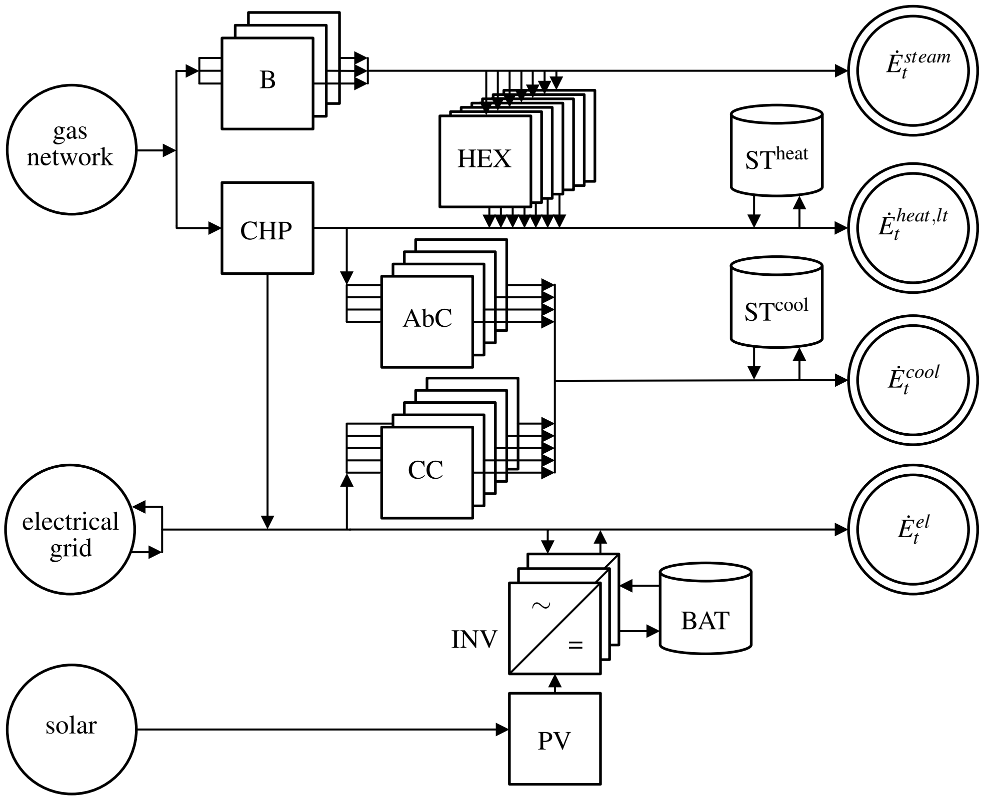

Figure 4: Visualization of the design of the energy system used in the case study, consisting of boilers (B), heat exchangers (HEX), a combined heat and power unit (CHP), absorption chillers (AbC), compression chillers (CC), inverters (INV), a photovoltaic system (PV), a heat storage unit (ST^{heat}), a cold storage unit (ST^{cool}), and a battery (BAT). A table with all capacities can be found in the Supplementary Material A.

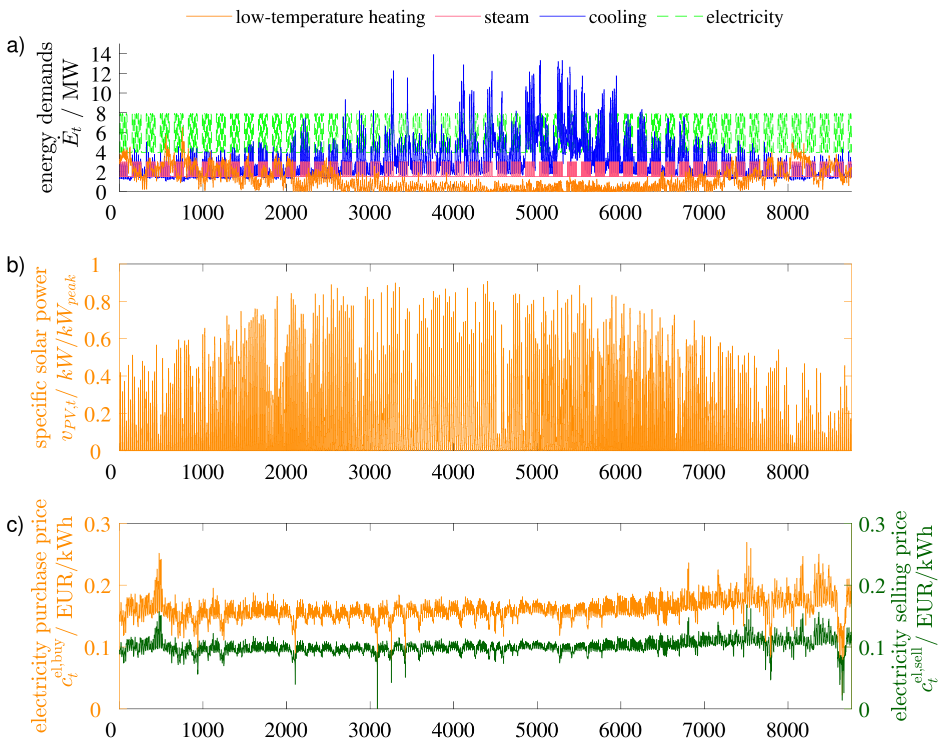

Figure 5: Full time series for the operational problem: a) energy demands, b) solar irradiation, and c) electricity prices for purchase and selling.

The original operational problem after the presolve contains $6 \cdot 10^5$ equations and $5 \cdot 10^5$ variables (incl. $2 \cdot 10^5$ binaries) with $18 \cdot 10^5$ nonzero elements. The long-term operational problem is highly coupled due to the storage and battery systems, an annual emission limit, peak-power prices, and network connection fees (Eq. (1-8)).

As benchmark, we intended to solve the original operational problem with the General-Column-Generation (GCG) solver (Gamrath and Lübbecke, 2010; Gleixner et al., 2018). The GCG solver decomposes coupled problems into independent subproblems and a connective master problem by a Dantzig-Wolfe decomposition and solves the decomposed problems via the Branch-Cut-and-Price algorithm. However, the GCG solver did not provide any decomposition besides the trivial decomposition in only one subproblem. Thus, as benchmark, we solve this one subproblem—ergo the original long-term operational problem—directly with CPLEX 12.6.3.0 (IBM Cooperation, 2016). In CPLEX, we employ the deterministic parallel mode such that all methods use the same number of computational cores. To validate the computational results, we repeat the calculation for 5 instances generated by statistical noise using Latin hypercube sampling on the data ($\pm$5%) (McKay et al., 2000).

DeLoop finds a feasible solution within a 2 % optimality gap in 5,656 s on average (Fig. 6). The average computational time of the benchmark takes 32 times longer (182,152 s) to obtain solutions of equal quality. On average, after 786 s, the proposed time-series decomposition method generates the first feasible solution with known solution quality, whereas on average, the benchmark takes 55 times longer (43,573 s) to provide the first feasible solution. Thus, the proposed decomposition method outperforms the benchmark in all instances by more than an order of magnitude.

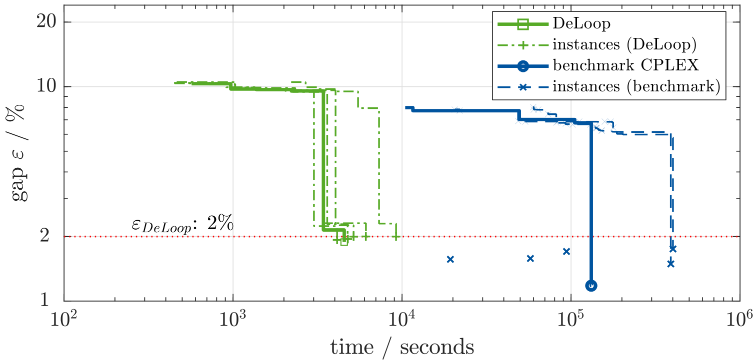

Figure 6: Gap ε of the proposed decomposition method DeLoop and the benchmark CPLEX as function of the solution time. Final optimality gap ε and solution time of all calculations are indicated by a marker. The required optimality gap ε_{DeLoop} of 2 % is marked in dotted red.

The solution method DeLoop is further analyzed by the lower bound and the upper bound of the operational expenditures *OPEX* of the original instance as a function of the solution time, Fig. 7. The linear-programming relaxation provides the first bound very quickly af-

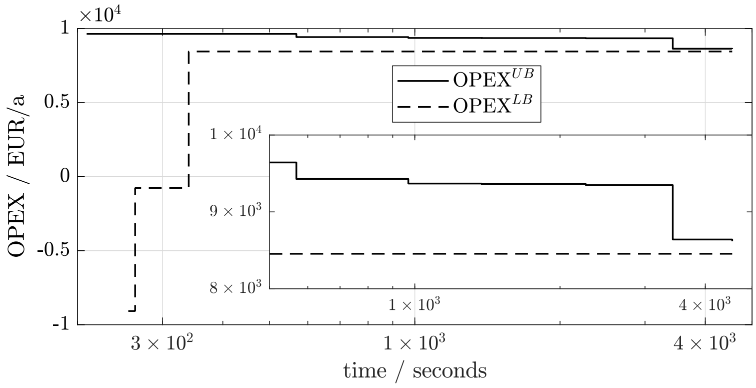

Figure 7: Operational expenditures *OPEX* of the lower and upper bounds of DeLoop as function of the solution time. The figure includes a zoom on the objective *OPEX* for better visualization of the improvement of the upper bound.
Note: The gap ε is evaluated after each iteration of the upper bound; thus, for the first feasible solution of the original problem after 209 s, no solution quality can be determined. Therefore, the first gap ε is computed for the second found feasible solution, compare Fig. 6.

ter 255 s. The first lower bound is the solution of the linear programming relaxation of the MILP, i.e. the root node of the Branch-and-Cut procedure. Within 340 s, the lower bound of the operational expenditures *OPEX* rises sharply twice to reach a value just 0.01 % below its final value. After 340 s, the lower bound is already a very tight relaxation of the original long-term operational problem.

For the upper bound, the initial heuristic (step (i)) composes the original problem into 219 subproblems. The solution of these 219 subproblems results in the first upper bound after 209 s. In subsequent iterations, the solution quality slightly improves as the number of subproblems is iteratively decreased. The improvement is due to subproblems being larger, which allows the peak power demands and its costs to be reduced and allows the distribution of the total emission limit among the subproblems to be improved. At 3425 s, with a decomposition into 30 subproblems, a large improvement in solution quality is achieved by reducing the network connection fees. At this point, a gap of 3 % is reached, which, for this case-study, is set as the starting gap for the post-processing warm-start procedure. The warm start satisfies the desired optimality gap of 2 % after a total computational time of 4,548 s. Considering fewer but longer subproblems enables DeLoop to better represent long-term effects. By the iterative solution approach, DeLoop identifies the maximal length necessary for the subproblems to accurately consider all long-term effects. Accurate consideration of long-term effects reduces operating costs by about 10 % in the considered case study, as the improvement of the upper bound shows.

The investigated case study shows that DeLoop is very time-efficient for handling time-coupling in long-term operational optimization problems. Although time-coupling is very common, not all of the presented time-coupling constraints and variables are always present at once in operational optimization problems. Therefore, we investigate the performance of DeLoop for less severe cases of time-coupling. For comparability, we also use CPLEX as benchmark. Again, to validate the computational results, we repeat the calculation for 5 additional instances with $\pm 5\%$ variation on the data.

We analyze the solution time for the original case study (*all*), and 3 further cases of less severe time-coupling, where we exclude different time-coupling equations from the original long-term operational problem: *no emission limits*, *no network fee*, and *neither* emission limits nor network fees, Fig. 8. All further cases still include time-coupling due to the storage and battery systems.

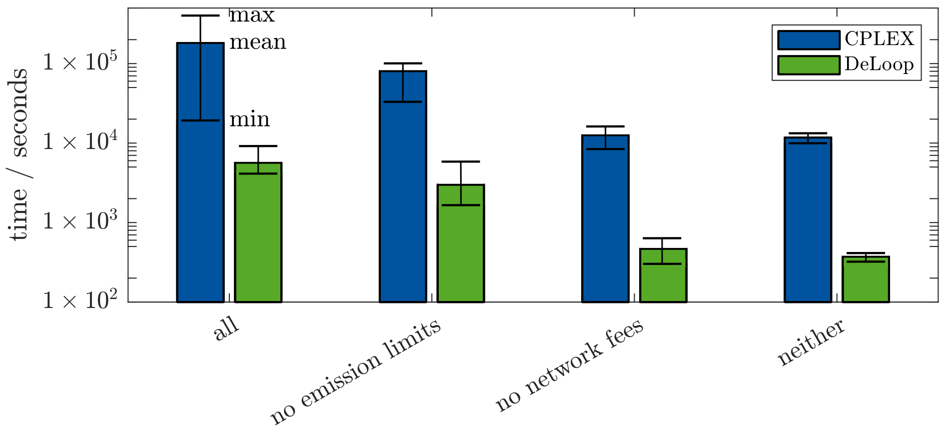

Figure 8: Solution time of the proposed decomposition method DeLoop and the benchmark with CPLEX for the investigated combinations of complicating constraints and variables: *all* includes all complicating constraints and variables (Eq. (1-8)); thus, *all* is the unaltered case study; *no emission limits* does not include the time-coupling constraint related to the emission limit (Eq. (1-7)); *no network fee* does not include the time-coupling constraints and variables related to the network connection fees (Eq. (1-5 and 8)); *neither* includes neither the time-coupling constraints and variables related to the emission limit nor related to the network connection fees (Eq. (1-5)). The top line of the bar represents the maximal solution time needed of the 6 investigated instances (marked by *max* at the very left bar). The bottom lime represents the minimal (*min*) and the middle line the average solution time (*mean*) of the 6 instances.

In all 6 instances of each of the investigated 3 cases with less severe time-coupling, DeLoop also outperforms the benchmark, here on average by a factor of 28. In the case of *no emission limits*, DeLoop requires 2992 s on average to solve the long-term operational problem, whereas the benchmark takes 80353 s. For the cases of *no network fees* and *neither*, DeLoop needs about 400 s to find a solution satisfying the optimality gap, whereas the benchmark takes 30 times as long (≈12000 s).

The investigation shows: combining multiple time-coupling constraints and variables leads to very challenging optimization problems; the network connection fees increase the solution time of the problem more than the emission limit as the network connection fees result in a combination of both a time-coupling constraints and variables; operational problems considering storage systems alone as time-coupling variable already result in long computational times.

Overall, the computational study shows that DeLoop enhances computational speed for long-term operational problems over a broad range of severeness of time-coupling.

5. Conclusions

Long-term operational optimization of energy systems is challenging. The complexity of the operational problem strongly increases due to time-coupling constraints and variables that are common in industrial applications. Additionally, time-coupling constraints prohibit the direct decomposition of long-term operational problems. As a consequence, long-term operational optimization is often not solvable within reasonable computational time or memory limits. In this paper, we propose a time-series decomposition method (DeLoop) providing feasible solutions with known solution quality. The method decomposes the original long-term operational problem with time-coupling constraints into smaller subproblems. These subproblems can be quickly solved in parallel computing mode. Subsequently, DeLoop recombines the subproblem solutions into a feasible solution of the original long-term operational problem while still representing a rigorous decomposition. The method is generally applicable to long-term operational problems considering time-coupling constraints and variables. In future work, DeLoop could be extended to design problems. Design problems introduce further time-coupling constraints and variables which cannot be directly solved using the suggested decomposition. However, DeLoop could be used in the underlying operational problems and thus be integrated into solutions frameworks including the design stage.

DeLoop is applied to an operational problem of an industrial energy system that exhibits both time-coupling constraints and variables. Time-coupling is due to storage systems, emission limits, peak-power prices, and network connection fees. First, the case study illustrates the importance of long-term operational planning, as significant cost reductions of about 10 % can be achieved. Second, the proposed method DeLoop provides fast convergence, outperforming a commercial solver in a large computational study on average by a factor of 32. Third, DeLoop also outperforms a commercial solver for less severely time-coupled problems on average by a factor of 28, showing the broad applicability of DeLoop. DeLoop is very time-efficient for handling time-coupling, thus renders real world long-term operational problems solvable. Thereby, DeLoop enables significant cost reductions for these real-world applications. In general, operational optimization of energy systems should consider both long-term constraints and forecast uncertainty. In this paper, we consider only certain predictions about future parameters. The integration of uncertainty would be an important extension. A potential extension could handle uncertainty via receding-horizon optimization, where the operation schedule is frequently optimized based on updated forecast data. As our method enables fast computation, DeLoop could be used in a frequent receding-horizon optimization to account for uncertainty. This concept will be explored in future work.

Acknowledgments

We thank Erik Mühmer, Simon Heising, and the GCG-development team for applying the Dantzig-Wolfe Decomposition to the case study. Furthermore we thank Tobias Gülker and Jonas Grossmann for their contribution to the case study.

References

Al-Agtash, S., Su, R., 1998. Augmented Lagrangian approach to hydro-thermal scheduling. IEEE Transactions on Power Systems 13 (4), 1392–1400.

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

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

Bahl, B., Söhler, T., Hennen, M., Bardow, A., 2018b. Typical periods for two-stage synthesis by time-series aggregation with bounded error in objective function. Frontiers in Energy Research 5, 35.

Baumgärtner, N., Delorme, R., Hennen, M., Bardow, A., 2019a. Design of low-carbon utility systems: Exploiting time-dependent grid emissions for climate-friendly demand-side management. Applied Energy 247, 755 – 765.

Baumgärtner, N., Temme, F., Bahl, B., Hennen, M., Hollermann, D. E., Bardow, A., 2019b. RiSES4 Rigorous Synthesis of Energy Supply Systems with Seasonal Storage by relaxation and time- series aggregation to typical periods. In: Stanek, W., Gładysz, P., Werle, S., Adamczyk, W. (Eds.), ECOS 2019 - Proceedings of THE 32ND INTERNATIONAL CONFERENCE ON EFFICIENCY, COST, OPTIMIZATION, SIMULATION AND ENVIRONMENTAL IMPACT OF ENERGY SYSTEMS At: JUNE 23-28, 2019, WROCLAW, POLAND.

Baumgärtner, N., Bahl, B., Hennen, M., Bardow, A., 2019a. Rises3: Rigorous synthesis of energy supply and storage systems via time-series relaxation and aggregation. Computers & Chemical Engineering 127, 127 – 139.

Baumgärtner, N., Shu, D., Bahl, B., Hennen, M., Bardow, A., 2019b. From peak power prices to seasonal storage: Long-term operational optimization of energy systems by time-series decomposition. In: Kiss, A. A., Zondervan, E., Lakerveld, R., Özkan, L. (Eds.), 29th European Symposium on Computer Aided Process Engineering. Vol. 46 of Computer Aided Chemical Engineering. Elsevier, pp. 703 – 708.

Bergner, M., Caprara, A., Ceselli, A., Furini, F., Lübbecke, M. E., Malaguti, E., Traversi, E., 2015. Automatic Dantzig–Wolfe reformulation of mixed integer programs. Mathematical Programming 149 (1), 391–424.

Bischi, A., Taccari, L., Martelli, E., Amaldi, E., Manzolini, G., Silva, P., Campanari, S., Macchi, E., 2019. A rolling-horizon optimization algorithm for the long term operational scheduling of cogeneration systems. Energy 184, 73 – 90.

Bradley, S. P., Hax, A. C., Magnanti, T. L., 1992. Applied mathematical programming, 19th Edition. Addison-Wesley, Reading, Mass., pp. 363-365.

Conejo, A. J., Castillo, E., García-Bertrand, R., Mínguez, R., 2006. Decomposition techniques in mathematical programming: Engineering and Science Applications. Springer-Verlag, Berlin, Heidelberg.

Elia, J. A., Floudas, C. A., 2014. Energy supply chain optimization of hybrid feedstock processes: a review. Annual Review of Chemical and Biomolecular Engineering 5, 147–179.

Gamrath, G., Lübbecke, M. E., 2010. Experiments with a generic Dantzig-Wolfe decomposition for integer programs. In: Festa, P. (Ed.), Experimental Algorithms. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 239–252.

GAMS Development Corporation, 2019. General Algebraic Modeling System (GAMS) Release 24.7.3.

Gleixner, A., Bastubbe, M., Eifler, L., Gally, T., Gamrath, G., Gottwald, R. L., Hendel, G., Hojny, C., Koch, T., Lübbecke, M. E., Maher, S. J., Miltenberger, M., Müller, B., Pfetsch, M. E., Puchert, C., Rehfeldt, D., Schlösser, F., Schubert, C., Serrano, F., Shinano, Y., Viernickel, J. M., Walter, M., Wegscheider, F., Witt, J. T., Witzig, J., 2018. The SCIP optimization suite 6.0. Tech. Rep. 18-26, ZIB, Takustr. 7, 14195 Berlin.

Goderbauer, S., Comis, M., Willamowski, F. J., 2019. The synthesis problem of decentralized energy systems is strongly NP-hard. Computers & Chemical Engineering 124, 343 – 349.

Grossmann, I. E., 2012. Advances in mathematical programming models for enterprise-wide optimization. Computers & Chemical Engineering 47, 2–18.

Hanne, T., Dornberger, R., 2017. Computational intelligence in logistics and supply chain management. Vol. 244 of International Series in Operations Research & Management Science. Springer, Switzerland.

IBM Cooperation, 2016. IBM ILOG and CPLEX Optimization and Studio and CPLEX User's and Manual and Version 12 and Release 7.

Kavvadias, K. C., Maroulis, Z. B., 2010. Multi-objective optimization of a trigeneration plant. Energy Policy 38 (2), 945–954.

Kazarlis, S. A., Bakirtzis, A. G., Petridis, V., 1996. A genetic algorithm solution to the unit commitment problem. IEEE Transactions on Power Systems 11 (1), 83–92.

Lin, F., Leyffer, S., Munson, T., 2016. A two-level approach to large mixed-integer programs with application to cogeneration in energy-efficient buildings. Computational Optimization and Applications 65 (1), 1–46.

McKay, M. D., Beckman, R. J., Conover, W. J., 2000. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics 42 (1), 55–61.

Nasrolahpour, E., Kazempour, S. J., Zareipour, H., Rosehart, W. D., 2016. Strategic sizing of energy storage facilities in electricity markets. IEEE Transactions on Sustainable Energy 7 (4), 1462–1472.

Park, J.-B., Park, Y.-M., Won, J.-R., Lee, K. Y., 2000. An improved genetic algorithm for generation expansion planning. IEEE Transactions on Power Systems 15 (3), 916–922.

Piacentino, A., Cardona, F., 2008. EABOT – Energetic analysis as a basis for robust optimization of trigeneration systems by linear programming. Energy Conversion and Management 49 (11), 3006–3016.

Rahmaniani, R., Crainic, T. G., Gendreau, M., Rei, W., 2017. The Benders decomposition algorithm: A literature review. European Journal of Operational Research 259 (3), 801–817.

Renaldi, R., Friedrich, D., 2017. Multiple time grids in operational optimisation of energy systems with short- and long-term thermal energy storage. Energy 133, 784 – 795.

Rong, A., Lahdelma, R., Luh, P. B., 2008. Lagrangian relaxation based algorithm for trigeneration planning with storages. European Journal of Operational Research 188 (1), 240 – 257.

Somma, M. D., Yan, B., Bianco, N., Graditi, G., Luh, P., Mongibello, L., Naso, V., 2015. Operation optimization of a distributed energy system considering energy costs and exergy efficiency. Energy Conversion and Management 103, 739 – 751.

StromNEV, 2005. Verordnung über die Entgelte für den Zugang zu Elektrizitätsversorgungsnetzen (Stromnetzentgeltverordnung -StromNEV). German.
URL `http://www.gesetze-im-internet.de/stromnev/StromNEV.pdf`

Voll, P., Klaffke, C., Hennen, M., Bardow, A., 2013. Automated superstructure-based synthesis and optimization of distributed energy supply systems. Energy 50, 374–388.

Wang, Q., McCalley, J. D., Zheng, T., Litvinov, E., 2016. Solving corrective risk-based security-constrained optimal power flow with Lagrangian relaxation and Benders decomposition. International Journal of Electrical Power & Energy Systems 75, 255–264.

Yokoyama, R., 2013. Optimal operation of a gas turbine cogeneration plant in consideration of equipment minimum up and down times. Journal of Engineering for Gas Turbines and Power 135 (7), 071801.

Yokoyama, R., Ito, K., 1996. A revised decomposition method for MILP problems and its application to operational planning of thermal storage systems. Journal of Energy Resources Technology 118 (4), 277–284.