A ROLLING-HORIZON APPROACH FOR MULTI-PERIOD OPTIMIZATION

LUKAS GLOMB, FRAUKE LIERS, FLORIAN RÖSEL (CA)

All authors: FAU Erlangen-Nuremberg, Department of Mathematics, Cauerstraße 11, 91058 Erlangen, Germany

Abstract. Mathematical optimization problems including a time dimension abound. For example, logistics, process optimization and production planning tasks must often be optimized for a range of time periods. Usually, these problems incorporating time structure are very large and cannot be solved to global optimality by modern solvers within a reasonable period of time. Therefore, the so-called rolling-horizon approach is often adopted. This approach aims to solve the problem periodically, including additional information from proximately following periods. In this paper, we first investigate several drawbacks of this approach and develop an algorithm that compensates for these drawbacks both theoretically and practically. As a result, the rolling horizon decomposition methodology is adjusted to enable large scale optimization problems to be solved efficiently. In addition, we introduce conditions that guarantee the quality of the solutions. We further demonstrate the applicability of the method to a variety of challenging optimization problems. We substantiate the findings with computational studies on the lot-sizing problem in production planning, as well as for large-scale real-world instances of the tail-assignment problem in aircraft management. It proves possible to solve large-scale realistic tail-assignment instances efficiently, leading to solutions that are at most a few percent away from a globally optimal solution.

Keywords. Large scale optimization, time decomposition, Rolling Horizon, Lot Sizing, Tail Assignment

1. Introduction

Many real-world optimization problems contain complex coupled decisions over a large time-span, often with periodic properties. Time-span is not necessarily meant literally, it is rather a surrogate for decisions that repeat in a similar but usually not in an exactly identical way. For a large time-span, the full optimization problem may contain an enormous number of variables and constraints and typically cannot be solved in an integrated fashion within reasonable time. Beyond taking advantage of problem specific aspects, an often applied solution methodology is to exploit the underlying time structure in decomposition approaches.

Recently, some authors developed and compared decomposition based solution approaches which take into account the full problem sequence at once. These are capable of calculating feasible or optimal solutions or valid bounds. Most of the approaches are developed and demonstrated for specific problem settings, but in principle many of the ideas could be transferred to various optimization problems. In Degraeve and Jans (2003), Akartunalı and Miller (2012), Akartunalı et al. (2016) and Fragkos et al. (2016), the authors investigate efficient methodologies to obtain lower bounds and feasible solutions for different optimization problems, utilizing e.g. Dantzig-Wolfe Decomposition, Lagrangian Decomposition and polyhedral combinatorics to exploit the underlying time structure.

E-mail address: lukas.glomb@fau.de, frauke.liers@fau.de, florian.roesel@fau.de (CA's email).

Although these are powerful tools, there are settings for which the aforementioned decomposition approaches are not applicable. Problem data is often known only a certain amount of time in advance. This situation naturally leads to an approach called rolling-horizon. The underlying optimization problem is repeatedly modeled for a certain period of time (forecast horizon), depending on how much data is available, then solved, and then shifted forward by a short period of time (decision horizon), while all variables slipping out of the forecast horizon are considered as fixed in subsequent iterations. In this work, we present general rolling horizon decomposition frameworks together with theoretical results on the quality of the obtained solutions. It is shown that for challenging applications, the obtained methods lead to efficient algorithms, obtaining high-quality solutions for huge instances that otherwise would be unsolvable.

The rolling-horizon approach has recently been applied to a large number of optimization problems. This includes lot-sizing problems (Mohammadi and Poursabzi (2014)), operating room scheduling (Addis et al. (2016)), energy storage management (Cruise et al. (2019)), stochastic supply chain management (Fattahi and Govindan (2020)), disruption management for railway networks (Nielsen et al. (2012)) and management of goods transportation processes (Bertazzi and Maggioni (2018)). An excellent summary of rolling-horizon based solution approaches can be found in Chand et al. (2002). As mentioned earlier, some future parameters may not be known exactly. Some works, e.g. Georghiou et al. (2019) and Postek and Hertog (2016), provide frameworks for the handling of uncertain forward parameters. The rolling-horizon approach can often be considered as a variant of a relax-and-fix heuristic, as in Stadtler (2003), where it has been applied to a difficult variant of lot-sizing.

Originally developed as a primal heuristic to determine feasible solutions for arbitrary (mixed) integer optimization problems, relax-and-fix is closely related to the rolling-horizon heuristic when applied to mixed-integer programs with time structure. In its basic version Wolsey (1998), the approach chooses a subset of the integer variables of the underlying problem, relaxes the integrality constraints for the remaining integer variables, solves the problem and fixes the variables in the subset to the obtained solution values. This is iterated until the problem is solved or infeasibility is detected. The approach has been applied to many optimization problems emerging in practice: difficult variants of lot-sizing problems (Mohammadi et al. (2010)), energy system management (Yilmaz et al. (2020)), periodic capacitated arc routing (Oliveira and Scarpin (2020)), crew scheduling for rapid transit networks (Fuentes et al. (2018)), smart vehicle communication network management (Ribeiro et al. (2019)), closed-shop scheduling (Kelly and Mann (2004)), data security applications (Baena et al. (2015)) and wireless communication (Malandrino et al. (2013)).

Though for some problem settings quite similar, the two approaches are not the same. Designed as a heuristic approach to quickly find feasible solutions for (mixed) integer programs, the only necessary condition for applicability of relax-and-fix heuristics is the existence of integer variables, while rolling-horizon heuristics are applied to possibly non-integer problems with time structure. A relax-and-fix approach typically takes into account more information than a rolling-horizon algorithm such as constraints and continuous as well as relaxed integer variables after the considered forward horizon. This can lead to better solutions, but also to possibly much larger sub problems, since all variables in the entire time horizon have to be incorporated, although the integrality constraints are relaxed. Then, these problems might be much harder to solve than the corresponding rolling horizon sub problems. An additional side effect is that the quality of the solutions obtained by relax-and-fix methods highly depend on the formulation. Tight formulations can lead to high-quality solutions, but loose formulations can lead to arbitrarily bad

integral variable assignments in each step. Solutions obtained by the rolling-horizon approach depend on the decomposition in sub problems, but not on the formulation, provided that the same linking variables appear in the formulations. Hence, a loose formulation may indeed lead to inferior runtime for the rolling-horizon approach, but does not lead to worse solutions. Hence, which approach to apply depends on the application, and some are solved more effectively by relax-and-fix, some others by rolling-horizon. Due to the formulation independency, the rolling-horizon approach is easier to analyze than relax-and-fix methods, since no formulation for the problem under consideration has to be specified. Contributions to the application of rolling-horizon and relax-and-fix heuristics abound. Typically, the theoretical results on the behavior of the heuristics rely on specific structures of the underlying problem, and not much is known about theoretical properties of general decomposition algorithms.

Some prior works focus on theoretical results for rolling-horizon heuristics, and deliver results which are taylored for specific problem classes. For example, Chand et al. (1990) and Bylka and Sethi (1992) proved that the rolling-horizon approach delivers exact solutions for their version of a stationary non-discounted lot-sizing model.

A more general result regarding rolling-horizon heuristics can be found in Bean and Smith (1984). The authors proved that sets of time periods where the optimal variable values can be determined by observing a finite number of subsequent time periods, for infinite time optimization problems exist in some cases. Two conditions must be met: a finite number of single time-period policies and an objective function that is discounted with a higher rate than its uniformly exponentially bounded growth.

In Sethi and Sorger (1991), a framework is proposed where general time structured optimization problems are solved with a rolling-horizon approach, taking into account the costs emerging from information procurement. A dynamic programming algorithm is developed to solve the resulting control problems.

More recently, in Absi and van den Heuvel (2019) a worst case analysis of a relax-and-fix approach applied to an uncapacitated single-item lot-sizing problem is conducted. As the fixing of integer variables is in chronological order, the approach can be considered as a rolling-horizon heuristic. It turns out that it is not possible to find guarantees on the obtained results. This is similar to some of the general results presented in Section 2. The theorems from Absi and van den Heuvel (2019) however are formulated specifically for the relax-and-fix heuristic, which enables to adapt continuous variables of periods with already fixed integer variables.

Rolling-horizon approaches are well suited for a relevant optimization problem in aircraft operations that is called tail-assignment problem (TAP). It is frequently solved by commercial passenger and cargo airlines. Given a flight plan, the latter assigns aircraft to flights. A good summary of recent developments in this and related fields is Aggarwal et al. (2019). Current developments in this research field can be found in e.g. Borndörfer et al. (2010), Ahmed et al. (2018), Froyland et al. (2013), Maher (2016), Sherali et al. (2013), Shao et al. (2017). Solving realistic sized TAP instances within a reasonable amount of time is a challenging task in itself. Within our rolling-horizon approach, we used the model Haouari et al. (2012) with some slight adaptations as it leads to the most efficient solution approaches.

Rolling-horizon heuristics were used for the practical solution of TAP Sinclair et al. (2016), but theoretical results are yet missing. For the rolling-horizon approaches presented here, a theoretical analysis of the solution quality is possible, and huge tail-assignment instances can be solved.

Our contribution is twofold. We present a new rolling-horizon algorithm that - in contrast to typical approaches - incorporates additional constraints. The latter ensure that the algorithm does not too often

sacrifice solution quality in the current period. Integrating these constraints, it is possible to derive solution quality guarantees. These are not valid for the classical rolling-horizon setting. This is demonstrated with an example. On the practical side, we demonstrate the ability of the algorithm to solve large optimization problems by applying it to realistic instances of the tail-assignment problem. We benchmark our computational results against a standard approach that first solves a fleet assignment problem (without maintenance considerations and aircraft routing) and afterwards a number of single-fleet tail assignment problems, taking these considerations into account. Computational results show that rolling-horizon outperforms fleet/tail assignment decomposition by far. We show that our algorithm is applicable to other optimization problems by solving an easy lot-sizing example.

The remainder of this paper is organized as follows. In Section 2, we define optimization problems with time structure, introduce decomposition methodologies and derive conditions under which solutions with provable solution quality can be obtained. In Section 3 we briefly demonstrate the applicability of our algorithm to lot-sizing problems. In Section 4 we introduce the optimization model for the TAP and describe how the methodology from Section 2 is applied to solve TAPs. In Section 5, we present extensive computational studies on instances derived from flight plans from a large German airline. We demonstrate that we are able to find solutions less than 10 percent from optimality for problems containing more than 2000 flights. The decomposition methods only need a few hours for solving these huge instances that are otherwise unsolvable. We conclude in Section 6 with a brief summary.

2. A rolling-horizon framework for optimization problems with time structure

In this chapter, we give a formal introduction into our version of the rolling-horizon approach. The objects taken into account are finite sequences of coupled optimization problems $\{P_0, ..., P_{\mathbf{T}}\}$, defined below in (1). Each belongs to a time period $t \in [\mathbf{T}] := \{0, \dots, \mathbf{T}\}$, where $\mathbf{T} \in \mathbb{N}_0$. For each time period t, we further assume that every problem P_t has a set of variables with start-state variables ξ_t defined on a domain Ξ_t and a set of end-state variables ϑ_t defined on a domain Θ_t. Interior variables x_t are neither start-state nor end-state variables and are defined on a domain X_t. The start-state variables ξ_t and the end-state variables ϑ_t connect the current time period and the previous or subsequent time period, respectively. We assume that the end-state set of the current time period is contained in the start-state set of the subsequent period:

(A1) Every end state of the current time period is a start state for the subsequent time period: $\Theta_t \subseteq \Xi_{t+1}$ for all $t \in [\mathbf{T} - 1]$.

This assumption is often satisfied in real-world problems, for example if the end states encode inventory levels. Every problem P_t has an objective function $f_t \colon X_t \to \mathbb{R}$ and constraint functions $g_t \colon \Xi_t \times X_t \times \Theta_t \to \mathbb{R}^{m_t}$ for appropriate $m_t \in \mathbb{N}$. We define the optimization problem P_t for a single time period with unconstrained start states and end states as

$$\min \quad f_t(x_t) \tag{1a}$$

$$\text{s.t.} \quad g_t(\xi_t, x_t, \vartheta_t) \leq 0 \tag{1b}$$

$$(\xi_t, x_t, \vartheta_t) \in \Xi_t \times X_t \times \Theta_t.$$

We assume wlog. that the objective function f_t depends only on interior variables. It is not too difficult to bring an optimization problem in this format. The constraint functions g_t depend on start states, end

states and interior variables.

We require the following two assumptions that concern P_t for all $t \in [\mathbf{T}]$ to be met:

(A2) Objective functions are non-negative: $f_t(x_t) \geq 0$ for all $x_t \in X_t$.

This assumption is often fulfilled in real-world problems, for example if non-negative costs are minimized. If the objective function of a problem is not non-negative and the underlying problem is not unbounded, we can add a constant term to the objective to make it non-negative.

(A3) Every start state can be completed to a feasible solution: For all $\xi_t \in \Xi_t$ there is $(x_t, \vartheta_t) \in X_t \times \Theta_t$ such that $g_t(\xi_t, x_t, \vartheta_t) \leq 0$.

This assumption may be restrictive. Take for example an instance of a capacitated lot-sizing problem with large demand at the end that exceeds production capacities at the end. Assume that start/end states are inventory levels that may range from 0 to a maximum level in every period. It then can happen that for some t a start inventory of 0 does not lead to a feasible solution. However, in this case the assumption can be restored as follows. Relaxing the demand satisfaction constraint and allowing shortage at the cost of appropriate penalties, or allowing to satisfy demand by buying at the market draws the problem feasible, regardless of the choice of the initial demand. In the single item case, another way to satisfy the assumption is to recursively determine minimum inventory level $\tilde{\xi}_t$, evaluating the formula

$$\tilde{\xi}_{\mathbf{T}+1} = 0,\ \tilde{\xi}_t := \max(0, \tilde{\xi}_{t+1} + d_t - C_t) \text{ for } t \in \{\mathbf{T}, \dots, 1\}, \tag{2}$$

where period demand is d_t and production capacity is C_t for period $t \in [\mathbf{T}]$. Of course, in many cases it would neither be legitimate to relax the problem setting nor be possible to construct the start state sets easily such that the assumption holds, though, it is also clear that in many cases it holds.

We define problems with fixed start states as $P_t^{\xi,\cdot}$, the problem with fixed end states as $P_t^{\cdot,\vartheta}$ and the problem with fixed start states and end states as $P_t^{\xi,\vartheta}$, respectively, for $\xi \in \Xi_t$ and $\vartheta \in \Theta_t$ by simply adding the constraint $\xi_t = \xi$, $\vartheta_t = \vartheta$ or both.

We further define the multi-period problem, $P_{t,\mu}$ for $\mu \in \mathbb{N}_0,\ \mu + t \leq \mathbf{T}$, starting at time period t as

$$\begin{aligned}
&\min && \sum_{j=t}^{t+\mu} f_j(x_j) && && \text{(3a)}\\
&\text{s.t.} && g_j(\xi_j, x_j, \vartheta_j) \leq 0 && \forall j \in \{t, ..., t+\mu\} && \text{(3b)}\\
& && \xi_j = \vartheta_{j-1} && \forall j \in \{t+1, ..., t+\mu\} && \text{(3c)}\\
& && (\xi_j, x_j, \vartheta_j) \in \Xi_j \times X_j \times \Theta_j && \forall j \in \{t, ..., t+\mu\}. &&
\end{aligned}$$

We define $P_{t,\mu}^{\xi,\cdot}$, $P_{t,\mu}^{\cdot,\vartheta}$ and $P_{t,\mu}^{\xi,\vartheta}$ as above by adding constraints $\xi_t = \xi$ or $\vartheta_{t+\mu} = \vartheta$ and $z_{t,\mu}$, $z_{t,\mu}^{\xi,\cdot}$, $z_{t,\mu}^{\cdot,\vartheta}$, $z_{t,\mu}^{\xi,\vartheta}$ as the optimal value of the corresponding multi-period problem. The index μ can be suppressed if it is 0. Taking assumptions **(A1)**, **(A2)** and **(A3)** together, it is already guaranteed that problems $P_{t,\mu}$ and $P_{t,\mu}^{\xi,\cdot}$ have at least one feasible solution, and that each feasible solution has a non-negative finite value. In general, it cannot be assumed that an optimum solution exists. For example if $P_t := \min_{x \geq 0} x_1$ s.t. $x_1 x_2 = 1$, all three above assumptions hold, but obviously an optimal solution is never attained. However, typically a real-world optimization problem can be modeled so that an optimum is attained, we assume that the aforementioned problems have optimal solutions.

(A4) Optimal solutions exist: For all $t \in [\mathbf{T}], \mu \in [\mathbf{T} - t]$ the problems $P_{t,\mu}$ and $P_{t,\mu}^{\xi_t,\cdot}$ for all $\xi_t \in \Xi_t$ attain their optimal solutions.

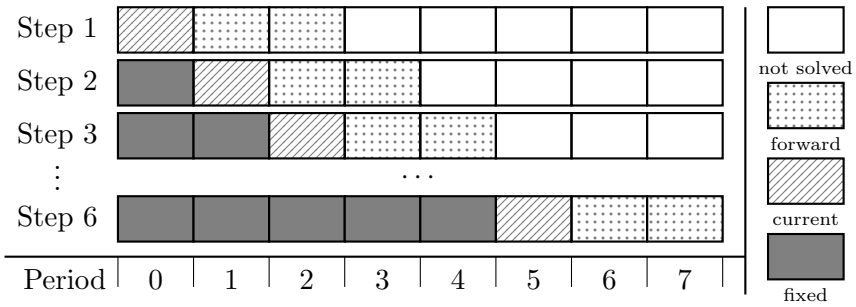

FIGURE 1. Schematic depiction of Algorithm 1 for $\mathbf{T} = 7$ and $\mu = 3$.

For problems of this kind, the rolling-horizon approach often delivers high quality solutions in practice. The procedure is formally defined in Algorithm 1 and works as follows: Algorithm 1 takes the problem sequence $P_{0,\mathbf{T}}$ and a parameter $\mu \in [\mathbf{T}]$. It solves $P_{0,\mu-1}$ and the solution of period 0, $(\xi_0, x_0, \vartheta_0)$ is fixed. It solves $P_{1,\mu-1}$ and the solution of period 1, $(\xi_1, x_1, \vartheta_1)$ is fixed. This process is continued until the entire problem sequence has been solved. Algorithm 1 solves $\mathbf{T} - \mu + 1$ combined μ-period problems

Algorithm 1 Rolling-horizon algorithm

INPUT: A sequence of minimization problems $P_{0,\mathbf{T}}$, $\mu \leq \mathbf{T} + 1$.
OUTPUT: A solution $(\xi_t, x_t, \vartheta_t)_{t\in[\mathbf{T}]}$ of $P_{0,\mathbf{T}}$.
procedure FINITE FORESIGHT OPTIMIZATION($P_{0,\mathbf{T}}, \mu$)
    $(\xi_t, x_t, \vartheta_t)_{0\leq t<\mu} \leftarrow argmin\, P_{0,\mu-1}$
    for $t \in \{1, ..., \mathbf{T} - \mu + 1\}$ **do**
        $(\xi_j, x_j, \vartheta_j)_{t\leq j<t+\mu} \leftarrow argmin\, P^{\vartheta_{t-1},\cdot}_{t,\mu-1}$
    end for
    return $(\xi_t, x_t, \vartheta_t)_{t\in[\mathbf{T}]}$
end procedure

and Algorithm 1 is shown schematically in Figure 1. The solid gray blocks represent time periods with an already fixed solution. The lined blocks represent the time period, for which the solution is to be fixed in the current step. The dotted blocks represent the time periods that are also taken into account in the current step and the white blocks represent periods not taken into account in the current step. Algorithm 1 comes with no theoretical solution quality. This is illustrated in Example 1.

Example 1. *Let $\boldsymbol{T} \in \mathbb{N}$. Let $\Xi_t = \mathbb{N}_0 = \Theta_t$ and $X_t = \mathbb{N}_0 \times \{0, 1\}$ for all $t \in [\boldsymbol{T}]$. Let P_t be defined as*

$$\text{(4a)} \qquad \min \quad f_t(x_t) := (c+1)\mathbb{1}_{\mathbb{R}^+}(x_t^1) + \frac{x_t^1}{x_t^1+1} + (c + \frac{t}{t+1}) \cdot (1 - x_t^2) + 1$$

$$\text{(4b)} \qquad s.t. \quad \xi_t + x_t^1 - x_t^2 = \vartheta_t$$
$$\xi_t, x_t, \vartheta_t \in \Xi_t \times X_t \times \Theta_t$$

for an arbitrarily large $c \in \mathbb{R}$.

The problem can be interpreted as follows: At the beginning of each period $t \in [\boldsymbol{T}]$, ξ_t units of a certain good are in the inventory. In every period, we must decide whether to produce nothing ($x_t^1 = 0$, costs 0) or to produce x_t^1 units of the good (costs $(c+1) + \frac{x_t^1}{x_t^1+1}$). In every period, we also have to decide whether to consume one unit of a certain good ($x_t^2 = 1$, costs 0) or not to consume ($x_t^2 = 0$, costs $c + \frac{t}{t+1}$). The number of units in the inventory after the period (ϑ_t) is equal to the number of units in the inventory before the period plus the units produced minus the units consumed.

Let $[\boldsymbol{T}] \in \mathbb{N}$ and $\mu \in [\boldsymbol{T}], t \in [\boldsymbol{T} - \mu + 1]$. Let $z_{t,\mu-1}$ be the optimal value of $P_{t,\mu-1}$. It holds that $z_{t,\mu-1} \geq \mu$

since the objective of every single period is at least 1. *It also holds that* $z^{\xi_t,\cdot}_{t,\mu-1} \leq c+\mu+2$ *for all* $\xi_t \in \Xi_t$ *since* $x^1_t = \mu$, $x^1_j = 0$ *for* $j \in \{t+1,\ldots,t+\mu-1\}$ *and* $x^2_j = 1$ *for* $j \in \{t,\ldots,t+\mu-1\}$ *is a feasible solution of* $P^{\xi_t}_{t,\mu-1}$ *for all* $\xi_t \in \Xi_t$. *The objective value is exactly* $c+\mu+1+\frac{\mu}{\mu+1}$. *We investigate the sequence of solutions generated by Algorithm 1 started with* $(P_{0,\boldsymbol{T}},\mu)$. *In the first iteration, the algorithm solves problem* $P_{0,\mu-1}$ *and obtains the optimal solution* $x^1_j = 0$ *for all* $j \in \{0,...,\mu-1\}$ *and* $x^2_j = 1$ *for all* $j \in \{0,...,\mu-1\}$, *with* $f_0(x_0) = 1$ *and* ϑ_0 *set to* $\mu-1$. *In the next iteration, the algorithm calculates the solution of* $P^{\mu-1,\cdot}_{1,\mu-1}$. *The unique optimal solution of* $P^{\mu-1,\cdot}_{t,\mu-1}$ *for an arbitrary* $t \in [\boldsymbol{T}-\mu+1]$ *is setting* $x^1_j = 0$ *for all* $j \in \{t,...,t+\mu-1\}$, *setting* $x^2_t = 0$ *and* $x^2_j = 1$ *for all* $j \in \{t+1,...,t+\mu-1\}$ *with an objective value of* $c+\mu+\frac{t}{t+1}$. *For this reason, the algorithm fixes* $x^1_t = 0$, $x^2_t = 0$ *and* $\vartheta_t = \mu-1$ *for all periods* $t \in \{1,...,\boldsymbol{T}-\mu+1\}$. *Thus,* $f_t(x_t) \geq c$ *for all periods except the first period and the last* $\mu-1$ *periods and*

$$\frac{\sum_{j=0}^{\boldsymbol{T}} f_j(x_j)}{z_{0,\boldsymbol{T}}} \geq \frac{(\boldsymbol{T}-\mu+1)c}{\boldsymbol{T}+1} = (1-\frac{\mu}{\boldsymbol{T}+1})c.$$

This means that for $\mu << \boldsymbol{T}$, *Algorithm 1 delivers solutions with a value of roughly* c *times the optimal solution value. Since we have chosen* c *arbitrarily, we can construct examples where the algorithm delivers solutions with an arbitrarily high optimality gap.*

Remark 1. *Example 1 also works when* Ξ_0 *is set to* $\{0\}$ *or any single element subset of* $\mathbb{N}$ *between* 0 *and* μ. *The only difference is that both algorithms fix* x^1_0 *to* $\mu-\xi_0$ *instead of* 0. *This results in a start state for period* 1 *of* $\xi_1 = \mu-1$ *as before. All further procedures and their theoretical investigation are analogous.*

The example shows that solutions obtained by Algorithm 1 are generally far away from an optimal solution, no matter how large the parameter μ is chosen to be.

A result similar to that from Example 1 has been presented in Absi and van den Heuvel (2019). The authors showed that their relax-and-fix approach (with fixing in chronological order) applied to an easy lot sizing problem cannot deliver solution guarantees. The reason for the bad performance of the relax-and-fix method is that continuous variables of periods with fixed binary decisions can adapt, i.e., a rolling-horizon based method fixing also continuous variables would perform pretty well on their counterexample. Both the relax-and-fix algorithm and the rolling-horizon algorithm may often take suboptimal decisions in a considered period. Example 1 complements that from Absi and van den Heuvel (2019) as it demonstrates that a rolling-horizon method can be suboptimal. Indeed, our instance is easily solved by a trivial decomposition which demonstrates that the suboptimality of the approach stems from the "rolling" character alone.

Increasing parameter μ does not always lead to a better overall solution. This is the case if taking information about additional time periods into account leads to a wrong decision, which is irreversible. This is demonstrated by the following brief example.

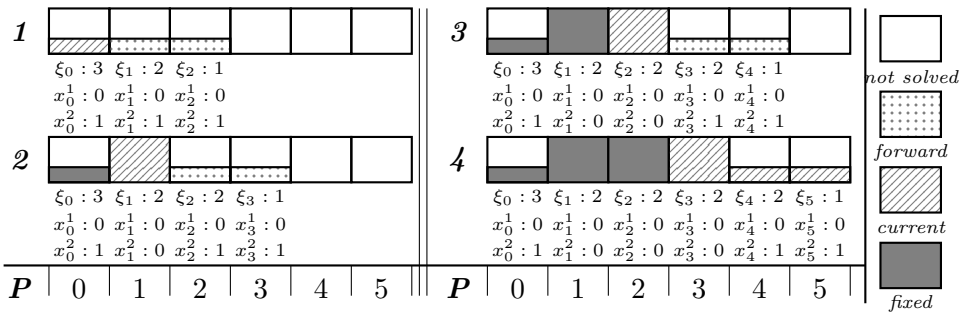

FIGURE 2. Algorithm 1 applied to problem (4)

Example 2. *Let* $\boldsymbol{T} \in \mathbb{N}$. *Let* $\Xi_t := X_t := \Theta_t := \{0,1\}$ *and* $c_t^1, c_t^2 \in \mathbb{R}^+$ *for all* $t \in [\boldsymbol{T}]$. *Let* $P_{0,\boldsymbol{T}}$ *be defined as*

$$\min \quad \sum_{t=0}^{\boldsymbol{T}} c_t^1 x_t + c_t^2(1-x_t) \tag{5a}$$

$$s.t. \quad \xi_t = x_t = \vartheta_t \qquad \forall t \in \{0,...,\boldsymbol{T}\} \tag{5b}$$

$$\xi_{t+1} = \vartheta_t \qquad \forall t \in \{0,...,\boldsymbol{T}-1\} \tag{5c}$$

$$(\xi_t, x_t, \vartheta_t) \in \Xi_t \times X_t \times \Theta_t \qquad \forall t \in \{0,...,\boldsymbol{T}\}.$$

For every feasible solution, $x_0 = x_1 = ... = x_{\boldsymbol{T}}$ *holds. The decision made in time period* 0 *determines all following decisions.*
Assume $\sum_{t=0}^{\boldsymbol{T}} c_t^1 < \sum_{t=0}^{\boldsymbol{T}} c_t^2$. *Thus,* $x^* = (x_0^*, ..., x_{\boldsymbol{T}}^*) = (1,...,1)$ *is an optimal solution for the full time horizon* $\{0,...,\boldsymbol{T}\}$. *Let* $\mu \in [\boldsymbol{T}+1]$. *In the case* $\sum_{t=0}^{\mu-1} c_t^1 < \sum_{t=0}^{\mu-1} c_t^2$, *Algorithm 1 applied to* $(P_{0,\boldsymbol{T}}, \boldsymbol{T})$ *yields the overall optimal solution since* x_0 *is set to* 1. *Otherwise, Algorithm 1 applied to* $(P_{0,\boldsymbol{T}}, \mu)$ *yields the sub-optimal solution since* x_0 *is set to* 0. *For example, for* $c^1 = (1,0,2,0,2,...)$ *and* $c^2 = (0,2,0,2,0,...)$, *these two cases alternate.*

Similar results were also shown in Absi and van den Heuvel (2019) for relax-and-fix heuristics. We complement it here with a result for rolling-horizon heuristics. This illustrates the fact that Algorithm 1 generally does not calculate increasingly better solutions with increased values of μ. For a theoretical comparison we introduce the straightforward Algorithm 2. It takes a parameter $\mu \in \mathbb{N}$ and a sequence of optimization problems $P_{0,\mathbf{T}}$. Algorithm 2 works as shown in Figure 3. The length of the optimization problem-sequence, $\mathbf{T}$, is not necessarily divisible by μ. For this reason, we introduce $\beta := \mathbf{T} + 1 - \lfloor \mathbf{T} \rfloor_\mu$, where $\lfloor \cdot \rfloor_\mu$ is the operation of rounding down to the next multiple of $\mu \in \mathbb{N}$. Algorithm 2 first solves the multi-period problem $P_{0,\beta-1}$ to optimality and fixes the optimal solution $(\xi_t, x_t, \vartheta_t)_{t \in \{0,...,\beta-1\}}$. The variable ξ_β is fixed to $\vartheta_{\beta-1}$. Then this algorithm solves the multi-period problem $P_{\beta,\mu-1}^{\vartheta_{\beta-1},\cdot}$ and again fixes its solution. The variable $\xi_{\beta+\mu}$ is fixed to $\vartheta_{\beta+\mu-1}$. This procedure is continued until the entire problem sequence is solved.

Algorithm 2 Planning horizon decomposition

INPUT: A sequence of minimization problems $P_{0,\mathbf{T}}$, $\mu \leq \mathbf{T}+1$.
OUTPUT: A solution $(\xi_t, x_t, \vartheta_t)_{t \in [\mathbf{T}]}$ of $P_{0,\mathbf{T}}$.
procedure PLANNING HORIZON DECOMPOSITION($P_{0,\mathbf{T}}, \mu$)
    $\beta \leftarrow \mathbf{T} + 1 - \lfloor \mathbf{T} \rfloor_\mu$
    $(\xi_t, x_t, \vartheta_t)_{0 \leq t < \beta} \leftarrow argmin\, P_{0,\beta-1}$
    for $j \in \{0,...,\lfloor \frac{\mathbf{T}}{\mu} \rfloor - 1\}$ **do**
        $(\xi_t, x_t, \vartheta_t)_{\mu j+\beta \leq t < \mu(j+1)+\beta} \leftarrow argmin\, P_{\mu j+\beta,\mu-1}^{\vartheta_{\mu j+\beta-1},\cdot}$
    end for
    return $(\xi_t, x_t, \vartheta_t)_{t \in [\mathbf{T}]}$
end procedure

Observation 1. *Let* $\boldsymbol{T} \in \mathbb{N}$. *Let* $\mu \in [\boldsymbol{T}+1]$. *Let* $P_{0,\boldsymbol{T}}$ *be a sequence of coupled optimization problems. Assuming* ***(A1)*** *-* ***(A4)****, Algorithm 1 and Algorithm 2 started with* $(P_{0,\boldsymbol{T}}, \mu)$ *return a feasible solution of* $P_{0,\boldsymbol{T}}$.

Algorithm 1 and Algorithm 2 return solutions that are feasible, but in general not optimal. Algorithm 1 fixes end-state variables considering only $\mu - 1$ many of the subsequent time-period problems, while Algorithm 2 fixes end-state variables considering none of the subsequent time-period problems. Both algorithms may make decisions in the current step that appear to be good, but lead to poor overall solutions. To ensure quality guarantees for the obtained solutions can be derived, we require an additional assumption.

(A5) The ratio of the solution value of the μ-period problem $P^{\xi,\cdot}_{t,\mu-1}$, $z^{\xi,\cdot}_{t,\mu-1}$, with start-state variables fixed to the worst possible $\xi \in \Xi$ and the μ-period problem $P_{t,\mu-1}$, $z_{t,\mu-1}$, with the best possible start-state variable realizations, is bounded by $(1+\varepsilon_\mu) \geq 1$:

$$\sup_{t\in[\mathbf{T}-\mu+1],\xi\in\Xi_t} \frac{z^{\xi,\cdot}_{t,\mu-1}}{z_{t,\mu-1}} \leq (1+\varepsilon_\mu).$$

This can indeed be a severe restriction in general. Often, Assumption **(A5)** can be satisfied by allowing a potential constraint violation at the price of very high penalty costs. This may have the consequence that the bound ε_μ is large for realistic sizes of μ. This makes it impossible to exploit the property effectively. However, there are many problems for which this inequality holds for adequately chosen ε_μ. One of those is presented in the following example.

Example 3. *Let a lot sizing problem be defined by the tuple (demand, production limit, setup costs, unit production costs, unit holding costs, holding limit, unit lost sales costs) per period. Hence* $(d_t, u_t^{prod}, q_t, p_t, h_t, u_t^{inv}, c_t), t \in [\boldsymbol{T}]$ *defines the instance. An instance (which can be found in our digital appendix) with* $u_t^{prod} = 4$, $q_t = 100$, $p_t = 20$, $h_t = 0$, $u_t^{inv} = 3.5$, $c_t = 100 + 0.001t$ *and* d_t *randomly drawn based on a distribution on the interval* $[0.9, 1.1] + 0.01t$ *for* $t \in [\boldsymbol{T}] = [100]$ *is taken into account. The values for* ε_μ*, as stated in* ***(A5)****, can be calculated easily (because of the lack of holding costs) by dividing the optimal value of each* μ*-period problem with empty start storage through the optimal value of the same problem with unrestricted start storage. The resulting values can be obtained from Figure 4, which shows that the value of* ε_μ *drops to less than* 15% *even for relatively small values of* μ*.*

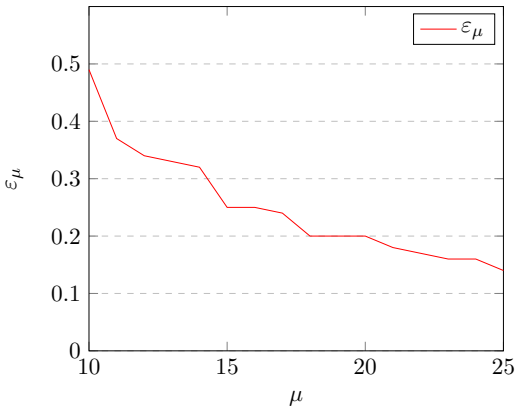

FIGURE 4. Relation $\mu \mapsto \varepsilon_\mu$.

The last example is also the motivation for Theorem 2 below. Although in general it is difficult to find the best ε_μ, we will give arguments that good bounds can often be obtained. In many real-world applications, the worst start state for a problem sequence is self-evident, for example in lot-sizing, where the start states

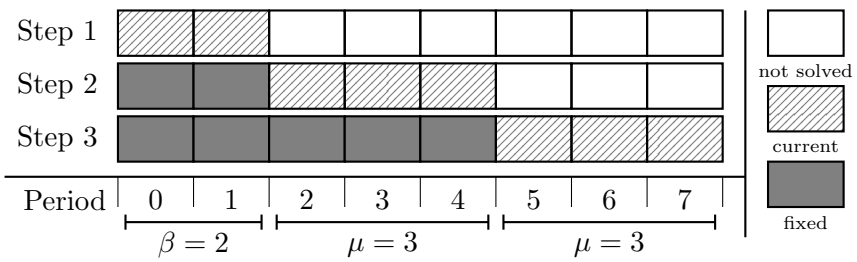

FIGURE 3. Schematic depiction of Algorithm 2 for $\mathbf{T} = 7$ and $\mu = 3$.

are the inventory levels, the worst-case start state is an inventory of zero, if reducing excess inventory is cheap. This reduces the max-min optimization problems to single-stage optimization problems. It is also frequently the case that the single-period problems of the optimization problem-sequence all share a similar structure. This makes it possible to generate upper and lower bounds for the optimal values of the optimization problems $P_{t,\mu-1}$ and $P^{\xi_t,\cdot}_{t,\mu-1}$ through simple operations on the parameters of the optimization problems. Furthermore, an exact knowledge of ε_μ is not necessary to apply the algorithms we describe to optimization problem sequences, an upper bound suffices. As an alternative, random sampling of some t and corresponding ξ also provide an estimate, which can be verified a posteriori after the optimization problem-sequence has been solved. Lemma 2 states that assuming **(A1)** - **(A5)**, Algorithm 2 returns a feasible solution of a certain quality. Before we prove this, we state an auxiliary lemma.

Lemma 1. *Let $\boldsymbol{T} \in \mathbb{N}$ and $P_{0,\boldsymbol{T}}$ be a sequence of coupled optimization problems. Let $z^{\cdot,\cdot}_{\cdot,\cdot}$ denote the optimal values of the corresponding optimization problem and $\Theta_{\cdot}$ denote the end-state set of the corresponding optimization problem. Let $\mu \in [\boldsymbol{T}]$. Let $t \in [\boldsymbol{T} - \mu + 1]$ and $\mu', \mu'' \in \mathbb{N}, \mu' + \mu'' = \mu$. Let $\vartheta \in \Theta_{t+\mu'-1}$. Then it holds that*

$$z_{t,\mu-1} \geq z_{t,\mu'-1} + z_{t+\mu',\mu''-1}.$$

It further holds that

$$z_{t,\mu-1} \leq z^{\cdot,\vartheta}_{t,\mu'-1} + z^{\vartheta,\cdot}_{t+\mu',\mu''-1}.$$

Analogous inequalities hold for problems with fixed start states or fixed end states.

Proof. For $\mu \in [\mathbf{T}]$, $t \in [\mathbf{T} - \mu + 1]$ and $\mu', \mu'' \in \mathbb{N}, \mu' + \mu'' = \mu$, the expression $z_{t,\mu'-1} + z_{t+\mu',\mu''-1}$ is exactly the optimal value of $P_{t,\mu-1}$ relaxed by removing the constraint $\vartheta_{t+\mu'-1} = \xi_{t+\mu'}$. The first inequality follows.
The expression $z^{\cdot,\vartheta}_{t,\mu'-1} + z^{\vartheta,\cdot}_{t+\mu',\mu''-1}$ is for $\vartheta \in \Theta_{t+\mu'-1}$ exactly the optimal value of $P_{t,\mu-1}$ with the additional constraint $\vartheta_{t+\mu'-1} = \vartheta$. The second inequality follows.
For multi-period problems with fixed start or end states, the inequalities can be similarly shown. □

Lemma 2. *Let $\boldsymbol{T} \in \mathbb{N}$ and $\mu \in [\boldsymbol{T} + 1]$. Let $P_{0,\boldsymbol{T}}$ be a sequence of coupled optimization problems with corresponding start states Ξ_t and optimal values $z^{\cdot,\cdot}_{\cdot,\cdot}$. Assume **(A1)** - **(A5)**, i.e.,*

$$\sup_{t \in [\boldsymbol{T}-\mu+1], \xi \in \Xi_t} \frac{z^{\xi,\cdot}_{t,\mu-1}}{z_{t,\mu-1}} \leq (1 + \varepsilon_\mu)$$

for an $\varepsilon_\mu > 0$. $z_{t,\mu-1}$ is the optimal value of the μ-period problem $P_{t,\mu-1}$ with free start and end states. $z^{\xi,\cdot}_{t,\mu-1}$ is the optimal value of the μ-period problem $P^{\xi,\cdot}_{t,\mu-1}$, with start states fixed to a $\xi \in \Xi_t$.
Then, the solution returned by Algorithm 2 started with $(P_{0,\boldsymbol{T}}, \mu)$, x_t for $t \in \{0, ..., \boldsymbol{T}\}$, is an ε_μ-optimal solution of the problem sequence. We define ε_μ-optimal as

$$\sum_{t \in [\boldsymbol{T}]} f_t(x_t) \leq (1 + \varepsilon_\mu) z_{0,\boldsymbol{T}}.$$

Proof. f_t is the objective function of period problem P_t for all $t \in [\mathbf{T}]$. $z_{t,\mu-1}$ is the optimal value of the corresponding μ-period problem. $z^{\xi,\cdot}_{t,\mu-1}$ is the optimal value of the corresponding μ-period problem with start state fixed to ξ. Let x_t be the interior variable values of the solution determined by the algorithm and $k := \lfloor \frac{\mathbf{T}}{\mu} \rfloor - 1$. Let $\beta = \mathbf{T} + 1 - \lfloor \mathbf{T} \rfloor_\mu$. Then the following holds:

$$z_{0,\mathbf{T}} \overset{\text{Lemma 1}}{\geq} z_{0,\beta-1} + \sum_{j=0}^{k} z_{\mu j+\beta,\mu-1}. \tag{6}$$

It further holds that

$$\sum_{t=0}^{\mathbf{T}} f_t(x_t) = \sum_{t=0}^{\beta-1} f_t(x_t) + \sum_{j=0}^{k} \sum_{l=\mu j+\beta}^{\mu(j+1)+\beta-1} f_l(x_l) \overset{(*)}{=} z_{0,\beta-1} + \sum_{j=0}^{k} z^{\vartheta_{\mu j+\beta-1},\cdot}_{\mu j+\beta,\mu-1} \overset{\textbf{(A5)}}{\leq} (1+\varepsilon_\mu) z_{0,\beta-1} + \sum_{j=0}^{k} (1+\varepsilon_\mu) z_{\mu j+\beta,\mu-1}. \tag{7}$$

Equality $(*)$ holds, because Algorithm 2 determines the values x_t such that they are optimal for $P^{\vartheta_{t-1},\cdot}_{t,\mu-1}$ for $t \in \{j \colon j \in [\mathbf{T}-\mu+1], j = \beta+\mu l, l \in \mathbb{N}\}$. Taking (6) and (7) together, we obtain

$$\frac{\sum_{t=0}^{\mathbf{T}} f_t(x_t)}{z_{0,\mathbf{T}}} \leq \frac{(1+\varepsilon_\mu) z_{0,\beta-1} + \sum_{j=0}^{k} (1+\varepsilon_\mu) z_{\mu j+\beta,\mu-1}}{z_{0,\beta-1} + \sum_{j=0}^{k} z_{\mu j+\beta,\mu-1}} = (1+\varepsilon_\mu). \tag{8}$$

This is equivalent to ε_μ-optimality. □

Even under assumption **(A5)**, a solution determined by Algorithm 1 generally does not satisfy any quality guarantee. Indeed, Example 1 is a counterexample for which **(A5)** holds. In all but the first period and the last few iterations, Algorithm 1 makes a suboptimal decision in the current period. This preserves the ability to make optimal decisions in future periods. However, the future decisions are never realized. Therefore, Algorithm 1 delivers deficient solutions for example problem (4). The relative deviation of the objective function value can be calculated as in **(A5)**:

$$\sup_{t \in [\mathbf{T}-\mu+1], \xi \in \Xi_t} \frac{z^{\xi,\cdot}_{t,\mu-1}}{z_{t,\mu-1}} \leq \frac{c+\mu+2}{\mu} \leq (1 + \frac{c+2}{\mu}).$$

Hence, we can set $\frac{c+2}{\mu} =: \varepsilon_\mu$, which converges to 0 as μ goes to ∞. Thus, there are optimization problems for which Algorithm 1 delivers feasible solutions arbitrarily far away from optimality, even under assumption **(A5)**. This means, the straightforward sequential approach is, at least under assumption **(A5)**, in theory better than the rolling-horizon Algorithm 1. This theoretical result differs from practical experience. Algorithm 1 usually delivers better results since it fixes the end states to beneficial values by taking knowledge about subsequent periods into account. Hence, we will adapt this algorithm to obtain Algorithm 3. While the new algorithm preserves the good practical properties of Algorithm 1, it delivers solution quality guarantees as strong as those provided by Algorithm 2.

The difference to Algorithm 1 is the following. Let $\beta := \mathbf{T} + 1 - \lfloor \mathbf{T} \rfloor_\mu$. Algorithm 3 generates the constraint

$$\sum_{l=0}^{\beta-1} f_l(x_l) \leq (1+\varepsilon) z_{0,\beta-1} \tag{9}$$

in iteration 0 with adequately chosen ε. The algorithm then proceeds almost as Algorithm 1 until iteration $\beta - 1$, where instead of solving $P^{\vartheta_{t-1}}_{t,\mu-1}$, it solves these problems with (9) as additional constraints. This generates the constraint

$$\sum_{l=j}^{j+\mu-1} f_l(x_l) \leq (1+\varepsilon) z_{j,\mu-1} \tag{10}$$

in iteration β for $j = \beta$ and adequately chosen ε. Parameter j is interpretable as the start period of each safeguarding section and takes values in $\{0, \beta, \beta+\mu, \beta+2\mu, \dots\}$. The indexing in Algorithm 3 slightly differs. Algorithm 3 proceeds almost as Algorithm 1 until iteration $\mu + \beta - 1$. Instead of solving $P^{\vartheta_{t-1}}_{t,\mu-1}$,

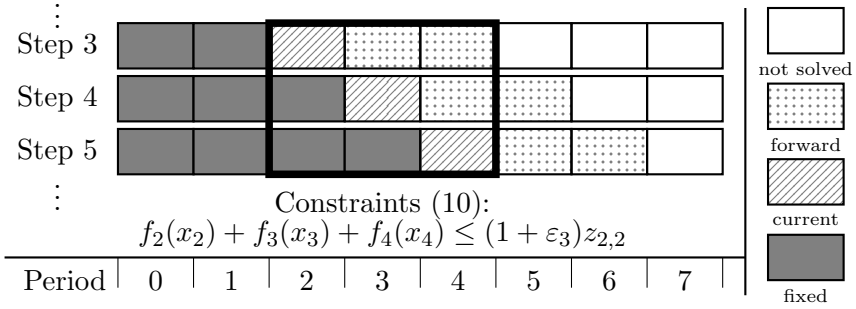

FIGURE 5. Schematic depiction of Algorithm 3 for $\mathbf{T} = 7$ and $\mu = 3$.

it solves these problems with (10) as additional constraints. Parameter j is raised by μ. This procedure is continued until the entire problem sequence is solved. The constraints are added to every μ-period problem $P_{t,\mu-1}$ for $t \in \{j, \dots, j+\mu-1\}$. If added to a problem $P_{t,\mu-1}$ with $t > j$, it contains variables $x_l, l < t$, which are not contained in the corresponding μ-period problem. These variables are simply the values the algorithm has fixed them to in previous steps. The constraint added to $P_{t,\mu-1}$ thus consists of three parts:

$$\underbrace{\sum_{l=j}^{t-1} f_l(x_l)}_{\substack{\text{A constant term,}\\ \text{determined in the previous}\\ \text{steps of the algorithm.}}} + \underbrace{\sum_{l=t}^{j+\mu-1} f_l(x_l)}_{\substack{\text{This term is dependent}\\ \text{on the variables } x_l\\ \text{contained in } P_{t,\mu-1}.}} \leq \underbrace{(1+\varepsilon)z_{j,\mu-1}}_{\substack{\text{A constant term}\\ \text{determined in the } (j+1)\text{-th}\\ \text{step of the algorithm.}}} .$$

It enforces the objective of the first few periods of a μ-period problem to be below a certain value. To be precise, the left hand side of constraints (10) consists of the objective value of the already fixed solution variables plus the objective depending on the yet to be fixed variables of the combined μ-period problems starting at every μ-th problem. The left hand side is constrained by the $(1+\varepsilon)$-fold of the optimal value of the problem $P_{j,\mu-1}$, $(1+\varepsilon)z_{j,\mu-1}$.

Parameter ε has to be set to a value such that assumption **(A5)** holds for (μ, ε). If such a parameter can be determined, the solutions returned by Algorithm 3 are $(1+\varepsilon)$-optimal. We already discussed the fact that determining such an ε, which is preferably small, may be a difficult task for some applications, so we propose an alternative. The right hand side of constraints (10) can be set to $(1+\varepsilon')z_{j,\mu-1}^{\vartheta_{j-1},\cdot}$ with $\varepsilon' \geq 0$. Parameter $\vartheta_{j-1} \in \Theta_{j-1}$ is the end state of the problem P_{j-1}, which has been determined and fixed in the previous step of the algorithm. The guaranteed solution quality is then $(1+\varepsilon')(1+\varepsilon)$. The practical behavior of Algorithm 3 is very similar to Algorithm 2 if ε' is very close to 0. It becomes more similar to Algorithm 1 as ε' grows.

Algorithm 3 provides feasible solutions with quality guarantees which are as strong as those provided by Algorithm 2. This is stated in the following theorem.

Theorem 1. *Let $\boldsymbol{T} \in \mathbb{N}$ and $P_{0,\boldsymbol{T}}$ be a sequence of coupled optimization problems. Let $\mu \in [\boldsymbol{T}+1]$, $\varepsilon_\mu > 0$ and Ξ_t denote the start state domain of P_t. Let $z_{\cdot,\cdot}^{\cdot,\cdot}$ denote the optimal value of the corresponding multi-period problem. Assuming **(A1)** - **(A5)**, Algorithm 3 started with $(P_{0,\boldsymbol{T}}, \mu, \varepsilon_\mu)$ returns a feasible and ε_μ-optimal solution, x_t for $t \in \{0, ..., \boldsymbol{T}\}$, of the problem sequence.*

Proof. Since we assume **(A1)**, **(A3)** and **(A4)** for μ, $P_{0,\mu-1}$ and $P_{t,\mu-1}^{\vartheta_{t-1},\cdot}$ for all $t \in \{1, \dots, \mathbf{T}-\mu+1\}$, regardless of ϑ_{t-1}, have an optimal solution. Algorithm 3 started with $(P_{0,\mathbf{T}}, \mu, \varepsilon_\mu)$ is able to calculate a feasible and ε_μ-optimal solution of $P_{t,\mu-1}^{\vartheta_{t-1},\cdot}$ with the additional constraints (10) in every μ-th iteration of the 11-th line of the algorithm. This is because inequality **(A5)** guarantees the existence of a solution of

Algorithm 3 Improved rolling-horizon algorithm

INPUT: A sequence of minimization problems $P_{0,\mathbf{T}}$, $\mu \leq \mathbf{T}$, $\varepsilon > 0$.
OUTPUT: A solution $(\xi_t, x_t, \vartheta_t)_{t\in[\mathbf{T}]}$ of $P_{0,\mathbf{T}}$.
procedure CAREFUL FINITE FORESIGHT OPTIMIZATION($P_{0,\mathbf{T}}, \mu, \varepsilon$)
    $\beta \leftarrow \mathbf{T} + 1 - \lfloor \mathbf{T} \rfloor_\mu$
    $(\xi_t, x_t, \vartheta_t)_{0\leq t<\mu} \leftarrow argmin\, P_{0,\mu-1}$s.t. $\sum_{l=0}^{\beta-1} f_l(x_l) \leq (1+\varepsilon) z_{0,\beta-1}$
    for $t \in \{1, ..., \beta\}$ **do**
        $(\xi_j, x_j, \vartheta_j)_{t\leq j<t+\mu} \leftarrow argmin\, P_{t,\mu-1}^{\vartheta_{t-1}}$s.t. $\sum_{l=0}^{\beta-1} f_l(x_l) \leq (1+\varepsilon) z_{0,\beta-1}$
    end for
    for $t \in \{0, ..., \lfloor \frac{\mathbf{T}}{\mu} \rfloor - 2\}$ **do**
        for $j \in \{\beta+1, ..., \mu+\beta\}$ **do**
            $(\xi_l, x_l, \vartheta_l)_{\mu t+j\leq l<\mu(t+1)+j} \leftarrow argmin\, P_{\mu t+j,\mu-1}^{\vartheta_{\mu t+j-1}}$s.t. $\sum_{l=\mu t+\beta}^{\mu(t+1)+\beta-1} f_l(x_l) \leq (1+\varepsilon) z_{\mu t+\beta,\mu-1}$
        end for
    end for
    return $(\xi_t, x_t, \vartheta_t)_{t\in[\mathbf{T}]}$
end procedure

$P_{t,\mu-1}^{\vartheta_{t-1},\cdot}$ fulfilling constraint (10). If there exists a solution $(x_t, ..., x_{t+\mu-1})$ of $P_{t,\mu-1}^{\vartheta_{t-1},\cdot}$ with constraint (10), it can be expanded to a solution of $P_{t+1,\mu-1}^{\vartheta_t,\cdot}$ with constraint (10). Thus, the values ϑ_t are set such that problem $P_{t+1,\mu-1}^{\vartheta_t,\cdot}$ with constraint (10) can be solved. This property is retained for the next μ periods. We deduce that Algorithm 3 always terminates in a finite number of steps, returning a feasible solution of $P_{0,\mathbf{T}}$.

It remains to show that this sequence is ε_μ-optimal. Constraint (9) implies that inequality $\sum_{t=0}^{\beta-1} f_t(x_t) \leq (1+\varepsilon_\mu) z_{0,\beta-1}$ holds. Constraint (10) implies that for all $t \leq \lfloor \frac{\mathbf{T}}{\mu} \rfloor - 2$ the inequality $\sum_{j=\mu t+\beta}^{\mu(t+1)+\beta-1} f_j(x_j) \leq (1+\varepsilon_\mu) z_{\mu t+\beta,\mu-1}$ holds. Assumption **(A5)** implies $\sum_{t=\mathbf{T}-\mu+1}^{\mathbf{T}} f_t(x_t) \leq (1+\varepsilon_\mu) z_{\mathbf{T}-\mu+1,\mu-1}$. For $k := \lfloor \frac{\mathbf{T}}{\mu} \rfloor - 2$ and $\beta := \mathbf{T} + 1 - \lfloor \mathbf{T} \rfloor_\mu$ it holds that

$$\frac{\sum_{t=0}^{\mathbf{T}} f_t(x_t)}{z_{0,\mathbf{T}}} \leq \frac{\sum_{t=0}^{\beta-1} f_t(x_t) + \sum_{t=0}^{k} \sum_{j=\mu t+\beta}^{\mu(t+1)+\beta-1} f_j(x_j) + \sum_{t=\mathbf{T}-\mu+1}^{\mathbf{T}} f_t(x_t)}{z_{0,\beta-1} + \sum_{t=0}^{k} z_{\mu t+\beta,\mu-1} + z_{\mathbf{T}-\mu+1,\mu-1}} \leq$$
$$\frac{(1+\varepsilon_\mu)(z_{0,\beta-1} + \sum_{t=0}^{k} z_{\mu t+\beta,\mu-1} + z_{\mathbf{T}-\mu+1,\mu-1})}{z_{0,\beta-1} + \sum_{t=0}^{k} z_{\mu t+\beta,\mu-1} + z_{\mathbf{T}-\mu+1,\mu-1}} = (1+\varepsilon_\mu),$$

analogously to the proof of Lemma 2. This completes the proof. □

Assuming **(A5)** for (μ, ε_μ), Algorithm 3 determines solutions which are ε_μ-optimal.

The value of $\varepsilon_\mu > 0$ could be large. We propose some sufficient criteria for sequences of optimization problems such that $(\varepsilon_\mu)_{\mu\in\mathbb{N}}$ exists and has two properties: It converges to zero if μ goes to infinity and Assumption **(A5)** holds for all (μ, ε_μ). It generalizes the ideas presented already in Example 3.

This statement is trivial for $\mu \geq \mathbf{T}$. We now interpret our finite-period optimization problem as a part of an infinite problem sequence. We prove our result for these infinite problem sequences. The result can be transferred to finite problem sequences. The finite problem sequence consisting of $\mathbf{T}$ subsequent single-period problems of the infinite sequence can then be solved by the algorithms proposed earlier. If a sequence $(\mu, \varepsilon_\mu)_{\mu\in\mathbb{N}}$ with the aforementioned properties exists, then Algorithm 2 and Algorithm 3 find arbitrarily good solutions, provided that parameter μ is chosen sufficiently large. This is stated in the following lemma.

Theorem 2. *Let $(P_t)_{t\in\mathbb{N}}$ be an infinite sequence of optimization problems and let **(A1) - (A4)** be fulfilled. Let Ξ_t denote their corresponding start states and Θ_t denote their corresponding end states. Let $z_{\cdot,\cdot}^{\cdot}$ denote their corresponding optimal values. Let the solution values of $(m+1)$-period problems be bounded for an $m \in \mathbb{N}$:*

$$\alpha := \sup_{t\in\mathbb{N},\xi_t\in\Xi_t,\vartheta_{t+m}\in\Theta_{t+m}} z_{t,m}^{\xi_t,\vartheta_{t+m}} < \infty, \tag{11}$$

Let the solution values of μ-period problems grow uniformly to infinity with μ:

$$\lim_{\mu\to\infty} \inf_{t\in\mathbb{N}} z_{t,\mu-1} = \infty. \tag{12}$$

Then a sequence $(\varepsilon_\mu)_{\mu\in\mathbb{N}}$ exists, which converges to 0 and has the property

$$\sup_{t\in\mathbb{N},\xi_t\in\Xi_t} \frac{z_{t,\mu-1}^{\xi_t,\cdot}}{z_{t,\mu-1}} \leq (1+\varepsilon_\mu) \tag{13}$$

for all $\mu \in \mathbb{N}$ with $\mu \geq m+1$ and $\inf_{t\in\mathbb{N}} z_{t,\mu-1} > 0$.

Proof. For an arbitrary $t \in \mathbb{N}$, $\xi_t \in \Xi_t$, $\mu > m$ and $\vartheta_{t+m} \in \Theta_{t+m}$ it holds that

$$1 \leq \frac{z_{t,\mu-1}^{\xi_t,\cdot}}{z_{t,\mu-1}} \overset{\text{Lemma 1}}{\leq} \frac{z_{t,m}^{\xi_t,\vartheta_{t+m}} + z_{t+m+1,\mu-m-2}^{\vartheta_{t+m},\cdot}}{z_{t,\mu-1}} \overset{(11)}{\leq} \frac{\alpha + z_{t+m+1,\mu-m-2}^{\vartheta_{t+m},\cdot}}{z_{t,\mu-1}}.$$

By choosing $\vartheta_{t+m}^{*\mu} \in \Theta_{t+m}$ minimizing $z_{t+m+1,\mu-m-2}^{\vartheta_{t+m},\cdot}$, we obtain for every $\mu > m$

$$\frac{\alpha + z_{t+m+1,\mu-m-2}^{\vartheta_{t+m}^{*\mu},\cdot}}{z_{t,\mu-1}} = \frac{\alpha + z_{t+(m+1),\mu-(m+2)}}{z_{t,\mu-1}} \leq \frac{\alpha + z_{t,\mu-1}}{z_{t,\mu-1}} = 1 + \frac{\alpha}{z_{t,\mu-1}}$$

and for this reason

$$1 \leq \sup_{t\in\mathbb{N},\xi_t\in\Xi_t} \frac{z_{t,\mu-1}^{\xi_t,\cdot}}{z_{t,\mu-1}} \leq 1 + \frac{\alpha}{\inf_{t\in\mathbb{N}} z_{t,\mu-1}}.$$

The right hand side converges to 1 as μ goes to ∞, since $\inf_{t\in\mathbb{N}} z_{t,\mu-1} \to \infty$ holds. □

Condition (11) states that regardless of the start state in the current period, every end state of the period m steps in advance can be reached. The optimal value of the corresponding optimization problem $P_{t,m}^{\xi_t,\vartheta_{t+m}}$ is at most α. For problem sequence (5) this is not true: If the start state of the current period t is 0, problem $P_{t,m}^{0,1}$ is infeasible for all $m \in \mathbb{N}$, since in this problem sequence all state variables are equal. Value α is interpreted as ∞ in this case. Condition (11) holds for Example 3. We emphasize that Theorem 2 only guarantees convergence for μ larger than m.

Although the conditions of Theorem 2 are formulated in a general way, it can easily be shown that they are satisfied in some special cases that may be likely to occur in practice.

Observation 2. *If for all $t \in \mathbb{N}$ the problem $P_t^{\xi,\vartheta}$ has a solution for all $(\xi,\vartheta) \in \Xi_t \times \Theta_t$ and $\sup\{f_t(x_t) | x_t \in X_t, t \in \mathbb{N}\} < \infty$, condition (11) is fulfilled.*

Proof. Clearly, $z_t^{\xi,\vartheta}$ is uniformly bounded by $\sup_{t\in\mathbb{N}} \sup_{x_t\in X_t} f_t(x_t)$, implying condition (11) for $m = 0$. □

Observation 3. *Let $(P_t)_{t\in\mathbb{N}}$ be a sequence of coupled optimization problems with $z_t \geq c$ for a $c > 0$. Then, condition (12) is fulfilled.*

Proof. It holds that

$$\lim_{\mu\to\infty} \inf_{t\in\mathbb{N}} z_{t,\mu-1} \geq \lim_{\mu\to\infty} c\mu = \infty.$$

□

In summary, we have shown that the rolling-horizon approach comes with no solution quality guarantee and have adapted it to ensure that appropriate solution quality guarantees are available under particular assumptions. We showed that these assumptions hold if the underlying problem sequence has some properties.
In the next sections, we will demonstrate that our approach can be applied to various classes of optimization problems.

3. Solving lot-sizing problems with rolling-horizon

A prominent example for optimization problems with time structure is the lot-sizing problem [see Pochet and Wolsey (2006)], which deals with the decision regarding the current size of a production lot depending on storage and demand. Given consecutive time periods of production, a company has to decide on the amount of items, spare parts or end products that should be manufactured in each time period. This decision heavily depends on how demand behaves now and in future. It is necessary to consider which variable and fixed production costs are incurred and what capacities are available to manufacture the products. In addition, the individual time periods are linked via inventories, which are associated with storage costs. The option of storing items over time can compensate effects of demand fluctuations or reduced capacities due to technical failures. For the sake of simplicity, we investigate the capacitated single-item case without resources. Nevertheless, it can be extended to more complex situations.

3.1. **Model description of the capacitated lot-sizing problem.**

A typical single-item capacitated lot-sizing problem has the structure shown in (14). The model variables are summarized in Table 1.

$$\min \sum_{t\in T}(p_t x_t + h_t s_t + q_t y_t + c_t w_t) \tag{14a}$$

$$\text{s.t.} \quad x_t \le C_t y_t \qquad \forall t \in T \tag{14b}$$

$$s_t + x_t + w_t = d_t + s_{t+1} \qquad \forall t \in T \tag{14c}$$

$$x, s \in \mathbb{R}_{0,+}^{|T|} \tag{14d}$$

$$w_t \in [0, d_t] \qquad \forall t \in T \tag{14e}$$

$$y \in \{0,1\}^{|T|}. \tag{14f}$$

The objective (14a) minimizes the fixed and variable costs of production per time period t, as well as the corresponding storage costs for production in advance. It also includes penalty costs for demand that cannot be met. Constraint (14b) ensures that whenever production takes place in period t, the corresponding binary variable y_t takes the value 1 and otherwise 0. This variable in turn controls the fixed costs in the objective function. C_t represents the upper bound on the individual capacity in period t if $y_t \neq 0$. Constraint (14c) is called the demand-satisfaction constraint. It ensures that the amount of storage at the beginning of a time period and production in this time period equals the corresponding demand plus the storage at the end of the time period. In this optimization model, it is necessary to avoid feasibility problems. Hence we allowed to purchase missing production units at a high price from competitors in order to meet one's own demand with the help of an additional variable w_t at costs c_t, which can also be interpreted as lost-sales costs. The purchase amount is not higher than the current demand, ensured by constraint (14e). It is straightforward to see that (14c) couples the individual time periods by storing the inventory over the periods. In this model variant, exactly two consecutive time

Set	**Description**
$T = \{0, ..., \lvert T-1 \rvert\}$	Set of consecutive time periods
Parameter	**Description**
$s_0 \in \mathbb{R}_0^+$	Fixed storage amount at the beginning of period $t = 0$ as start state
$h_t \in \mathbb{R}_0^+$	Storage costs per item from t to $t+1$
$p_t \in \mathbb{R}_0^+$	Variable costs of production per item in period t
$q_t \in \mathbb{R}_0^+$	Fixed costs of production in period t
$c_t \in \mathbb{R}_0^+$	Costs to buy one item in period t from competitors
$d_t \in \mathbb{R}_0^+$	Demand in time period t
$C_t \in \mathbb{R}_0^+$	Capacity in time period t
Variable	**Description**
$x_t \in \mathbb{R}_0^+$	Production volume in period t
$s_t \in \mathbb{R}_0^+$	Storage amount at the beginning of period t
$y_t \in \{0, 1\}$	Production indicator in period t
$w_t \in \mathbb{R}_0^+$	Number of missing items to fulfill the demand of period t

TABLE 1. Input data for the capacitated lot-sizing problem

periods are always connected. Taking into account information about subsequent periods can be very effective in the case of hard demand bounds, since feasibility issues can be omited completely providing the capacity can be sufficiently adapted to demand and the time interval examined is sufficiently long.

3.2. **Application of rolling-horizon.** Due to the nature of the problem, rolling-horizon approaches have been used to solve it [see e.g. Stadtler (2003)]. Since the time period is linked to the storage stock, all theoretical considerations about time decomposition can also be applied here: Limited foresight can lead to arbitrarily bad production plans, since incomplete problem information can negatively influence current production. As discussed in Section 2, this effect can be compensated by deviation bounds in order to limit excessive reactive measures, and by an appropriate choice of the forward-horizon length. The start- and end-variables are part of the objective function, which is theoretically acceptable, as it is possible to model duplicates to delete them from the objective function.
Next, we show the benefit of applying our algorithm in a prototypical fashion. To ensure our theoretical results can be applied, we need to make the following two assumptions, which will be fulfilled naturally in real-world instances:

- **(L1)** Costs are bounded from below: $h_t, p_t, c_t \geq P > 0 \quad \forall t \in T$
- **(L2)** The demand per period is bounded: $0 < L \leq d_t \leq U < \infty \quad \forall t \in T$
- **(L3)** Producing is cheaper than purchasing: $p_t \leq c_t \quad \forall t \in T$

Assumption **(L1)** and **(L2)** imply that the objective of each period is at least $L \cdot P$. It is possible to derive bounds for formula **(A5)** with $c = \max_{t \in T} c_t$, $h := \max_{t \in T} h_t$ and $S := \sum_{t \in T} d_t$ if 0 is a feasible end state for every period:

$$\sup_{t \in [\mathbf{T}-\mu+1], \xi \in \Xi_t} z_{t,\mu-1}^{\xi,\cdot} \leq \sup_{\xi \in [0,S]} \mu(h\xi + \max\{0, U - \frac{\xi}{\mu}\}c) \leq \mu \max\{hS, Uc\} \tag{15}$$

and

$$\sup_{t \in [\mathbf{T}-\mu+1]} z_{t,\mu-1} \geq \mu LP > 0. \tag{16}$$

Therefore, we can estimate an upper bound for ε_μ:

$$\sup_{t \in [\mathbf{T}-\mu+1], \xi \in \Xi_t} \frac{z_{t,\mu-1}^{\xi,\cdot}}{z_{t,\mu-1}} \leq \frac{\mu \max\{hS, Uc\}}{\mu LP} = \frac{\max\{hS, Uc\}}{LP}. \tag{17}$$

It should be noted that in reality, the bound can be much tighter when making a case distinction between potential production and penalty costs per period. For the sake of simplicity, we do not go into detail here.

To illustrate that Algorithm 3 may deliver better solutions than Algorithm 1, both are applied to an example in the following. We constructed a toy example lot-sizing problem. This is designed to show that applying Algorithm 1 leads to inferior results than Algorithm 3. We consider a lot-sizing problem with

Parameter	$t=0$	$t=1$	$t=2$	$t=3$
p_t	28	25	29	33
h_t	3	3	3	3
q_t	800	800	800	800
c_t	75	75	75	75
d_t	20	20	20	20
C_t	40	40	40	40

TABLE 2. Example for a lot-sizing problem with $T = \{0, 1, 2, 3\}$

4 time periods. It can be assumed that these are directly consecutive, possibly as part of a larger time horizon. Thus, $\{s_0\}$ is interpreted as Ξ_0 and the corresponding objective value will not be incorporated in the solution. This is also our assumption for the value of ε_μ. To ensure that the effects to be shown occur in the shortest possible time horizon, we point out that $\beta = 0$ in the case that $\mu = 1, 2$. For Algorithm 3 we either evaluate formula (17) to obtain an estimation of $\varepsilon_\mu = 24$ for $\mu = 2$. Alternatively, it is straight-forward to see that a start storage of 40 units is the best start state for each two-period problem with an objective value of 180, which leads to an estimation of ε_2 of $\frac{3000}{180} - 1 = 15\frac{2}{3}$, with the numerator of Inequality (17). Initially, this may seem a high value, but the value has to express the ratio between an optimal value regarding an arbitrarily large ξ_t and an optimal value with $\xi_t = 0$. Additionally, although the solution quality guarantee is not low enough to conclude proper results for this instance, we will demonstrate in the following that $\varepsilon_2 = 15\frac{2}{3}$ is small enough to obtain a better optimal solution from Algorithm 3 (in its original form) than from Algorithm 1.

Table 3 shows the different solutions with all algorithms varying over μ, considering a start inventory

solution approach	variable solution values (x_t, s_t, y_t, w_t)				objective value
	t = 0	t = 1	t = 2	t = 3	
$(\mu = 1)_{\text{Algorithm 1,2,3}}$	$(20, 0, 1, 0)$	$(20, 0, 1, 0)$	$(20, 0, 1, 0)$	$(20, 0, 1, 0)$	5500
$(\mu = 2)_{\text{Algorithm 1}}$	$(40, 0, 1, 0)$	$(20, 20, 1, 0)$	$(20, 20, 1, 0)$	$(0, 20, 0, 0)$	4780
$(\mu = 2)_{\text{Algorithm 2,3}}$	$(40, 0, 1, 0)$	$(0, 20, 0, 0)$	$(40, 0, 1, 0)$	$(0, 20, 0, 0)$	4000
$(\mu = 4)_{\text{Algorithm 1,2,3}}$	$(40, 0, 1, 0)$	$(40, 20, 0, 0)$	$(0, 40, 0, 0)$	$(0, 20, 0, 0)$	3960

TABLE 3. Solution of Lot-Sizing example with $T = \{0, 1, 2, 3\}$

of 0. Clearly, for $\mu = 1$ all algorithms deliver the same solution, because every period will be solved seperately. The same holds for $\mu = 4$, where the whole integrated problem is solved from the beginning. In case $\mu = 2$ the decision of the first step regarding period $t = 0, 1$ is re-adjusted by Algorithm 1 at immense costs when the additional information of period $t = 2$ comes into play.

On the other hand, Algorithm 3 is not allowed to produce something in period 1 (that makes the RH solution expensive), because the following inequality holds:

$$\sum_{t=0}^{1} f_t(x_t) \leq (1 + \varepsilon_\mu) z_{0,1} = (1 + 15\frac{2}{3}) \times 180 = 3000 \quad \forall t \in \{0, 1, 2\}. \tag{18}$$

All algorithms would decide to produce 40 units in period 0. Therefore costs of 1920 come from the production $x_0 = 40$, $y_0 = 1$ in period 0 and storage costs of 60 caused by $s_1 = 20$ at the beginning of

period 1. In step 2, a classical rolling-horizon approach starts the production in period 1, which yields the best solution for the observed horizon. This comes with costs of 1360 in period 1. According to Inequality (18), Algorithm 3 has a bound of 3000 for the objective value of the first two periods. Hence, this solution is excluded for Algorithm 3. Therefore the production takes place in period 2 with costs of 1380 and in period 1 the costs remain 60. In the next step, it is preferable to expand the production in period 2 to $x_2 = 40$ with additional costs of 580 and additional storage costs in period 3 of 60, which results in total costs of 2020 in period $t = 2, 3$. The total costs of the different algorithms are shown in Table 3. It has been shown that Algorithm 3 calculates better solutions than Algorithm 1 for this example.

In practical applications, the relative rolling deviation from the solutions produced by the rolling-horizon itself is of interest, since no additional optimal values have to be calculated aside from those that are automatically calculated. The idea is to create a bound regarding $z^{\xi_t,\cdot}_{t,\mu-1}$, with the start state ξ_t obtained during the solution process.

Therefore, we want to complete the prior observations by the conduction of a study with ten 100-period

Instance	**Algorithm 2**	**Algorithm 1**	**Algorithm 3**				
			$\varepsilon = 0.2$	$\varepsilon = 0.15$	$\varepsilon = 0.1$	$\varepsilon = 0.07$	$\varepsilon = 0.05$
1	0.0741	0.1663	0.1110	0.0893	0.0864	0.0581	**0.0475**
2	0.0817	0.2388	0.0927	0.1240	0.0900	0.0818	**0.0729**
3	0.0891	0.2149	0.1380	0.0918	0.0767	0.0742	**0.0532**
4	0.0754	0.1734	0.1157	0.0953	0.0786	0.0725	**0.0636**
5	0.0803	0.1609	0.0869	0.0928	0.0831	**0.0641**	0.0701
6	0.0778	0.1885	0.1431	0.0831	0.0925	**0.0704**	0.0862
7	0.0817	0.2022	0.1231	0.0971	0.0702	**0.0537**	0.0600
8	0.0869	0.1565	0.1364	0.1182	0.1038	0.0911	**0.0662**
9	0.0886	0.1930	0.1210	0.1027	0.0823	0.0703	**0.0640**
10	0.0856	0.1488	0.1197	0.1018	0.0957	**0.0628**	0.0691

TABLE 4. Optimality gap for different solution approaches applied to lot sizing instances

lot-sizing instances, solved in 7-day steps, demonstrating that the modified Algorithm 3 is capable of solving realistic optimization problems. The instances have been created randomly, very similar to the instances used in Example 3. Input parameters are $C_t = 4$, $q_t = 100$, $p_t = 20$, $h_t = 0$, $c_t = 100 + 0.001t$ and d_t randomly drawn based on a distribution on the interval $[0.9, 1.1] + 0.01t$ for $t \in [\mathbf{T}] = [100]$. Additionally, we want to bound the holding variable by $u^{inv}_t = 3.5$ from above. For these instances, the solution quality guarantee is for the relatively small $\mu = 7$ at $\varepsilon_\mu \in [.85, 1.0]$ for all of the instances created, i.e., it is relatively large compared to the solutions calculated by the algorithms. However, applying the algorithms 1, 2 and 3 (in its modified form) leads to optimality gaps summarized in Table 4. It is clearly visible that the latter algorithm with appropriately chosen parameter ε by far outperforms the conventional rolling-horizon approach (Algorithm 1) and also outperforms the sequential approach (Algorithm 2).

The effect of the improved rolling-horizon approach can be clearly seen already from the prototypical examples shown here. Time decomposition approaches become interesting when larger problem characteristics are considered, i.e., with several products, intermediate products and different correlated production steps. In addition, there are also considerations relating to the production process: If the production of a good exceeds the time span of a production planning period, the planning of this period can also depend on more than just the directly adjacent periods. As we have seen, rolling-horizon is an appropriate approach to solve large scale production planning instances in a reasonable amount of time, under some mild assumptions to avoid feasibility issues.

4. Solving tail-assignment problems with rolling-horizon

It is possible to apply the rolling-horizon approach to the so called tail-assignment problem (TAP), a mathematical model which aims to assign a set of aircraft to a set of scheduled line flights while minimizing the overall costs. In this section, the basic single-period tail assignment model is described in detail. Since it best fits our purpose, we adjust a flight connection based model introduced by Haouari et al. (2012).

4.1. Description of notation and input data.

Before we introduce our optimization model, we give a brief overview of input sets, parameters and model variables used, summarized in Table 5.
The first input component of the TAP is the flight plan. It consists of a finite set of flight legs L that connect finitely many airports (or stations) S and has a time horizon $[0, T]$, with $T \in \mathbb{N}$. For example, if the time is measured in minutes and the duration of the planning period is one day, T equals 1440 time units. Furthermore, the finite set K describes the available fleets to be assigned to serve the flight

Set	**Description**
S	Set of airports
$[T] = \{1, ..., T\}$	(Discretized) time horizon
K	Set of aircrafts or aircraft types (we call this a fleet)
L	Set of flight legs
A	Set of flight connections (possible consecutive flights)
L_k	Set of flight legs flyable by fleet k
K_l	Set of fleets able to fly flight leg l
K_a	Set of fleets able to fly connection a
A_k	Set of flight connections flyable by fleet k. $(l_i, l_j) = (a^-, a^+) = a \in A_k \Leftrightarrow a^-, a^+ \in L_k$
A^M	Subset of A. Set of flight connections with a maintenance possibility
A_k^M	$A^M \cap A_k$
A^{NM}	$A \setminus A^M$
A_k^{NM}	$A^{NM} \cap A_k$
Parameter	**Description**
$N_k \in \mathbb{N}$	Number of available aircraft of fleet k
$c_l \in \mathbb{R}_0^+$	Costs of canceling flight leg l
$c_{lk} \in \mathbb{R}_0^+$	Costs of assignment of fleet k to flight leg l
$c_{ak} \in \mathbb{R}_0^+$	Costs of fleet k serving connection a
$t_k \in \mathbb{R}_0^+$	Maximum allowed flying time of fleet k without a maintenance event taking place
$t_s \in \mathbb{R}_0^+$	Flying time necessary to get back from s to the next maintenance station.
$t_l^{dep} \in [0, T]$	Scheduled time of departure of flight leg l
$t_l^{arr} \in [0, T]$	Scheduled time of arrival of flight leg l
$t_k^M \in \mathbb{R}_0^+$	Maintenance duration of fleet k
$x_{ski}^b \in \{0, 1\}$	1, if and only if i or more aircraft of type k are at station s at time 0, otherwise 0.
$\omega_{ski}^b \in [0, t_k]$	Time elapsed since maintenance of i-th aircraft of type k at station s at time 0
$\vartheta_l \in [0, T]$	Nominal flying time of flight leg l

Table 5. Input data of TAP

legs. $N_k \in \mathbb{N}$ denotes the maximal number of available aircraft per fleet. The parameter $x_{ski}^b \in \{0, 1\}$ is equal to 1 if and only if i or more aircraft of type k are at station s at the beginning, otherwise it is equal to 0. Thus, the number of aircraft of fleet k at station s at the beginning of the time horizon is equal to $\sum_{i=1}^{N_k} x_{ski}^b$. Every aircraft of fleet k is allowed to fly at most $t_k \in \mathbb{R}^+$ time units without a

maintenance event. The maintenance duration is $t_k^M \in \mathbb{R}^+$ time units and the maintenance costs are $c_k^M \in \mathbb{R}^+$ monetary units. The parameter $\omega_{ski}^b \in \mathbb{R}_0^+$ equals the time elapsed since the last maintenance of aircraft i ($i \in \{1, ..., N_k\}$) of aircraft type k at station s if i or more aircraft of type k are at station s, otherwise, ω_{ski}^b is 0.

Every flight leg $l \in L$ has a scheduled departure time $t_l^{dep} \in [0, T]$, a scheduled arrival time $t_l^{arr} \in [0, T]$ and a departure and an arrival station. These four attributes can serve as unique characterizers for a flight leg. A flight leg also has a nominal flying time $\vartheta_l := t_l^{arr} - t_l^{dep}$. Since the model is acyclic, we assume $T \geq \max_{l \in L} t_l^{arr}$.

Every pair of flight legs that is consecutively executable by fleet k is contained in the set A_k. Since the set A_k consists of pairs of flight legs $l \in L_k$, (L_k, A_k) can be interpreted as a directed (multi-) graph which we call the connection network of fleet k. The set A_k can be partitioned into two subsets A_k^M and A_k^{NM}, while a pair of flight legs is in A_k^M if and only if there is enough time to execute a maintenance event. We further expand the connection networks by a start node v_{ski}^b for each aircraft numbered by i of the corresponding fleet k at each station s. Every start node is connected with every flight leg in L_k, starting at station s. We expand the connection network by termination nodes v_{ski}^e for each station s and every aircraft i of any fleet k. The termination nodes are connected from flight legs in L_k that arrive at s and from every start node with the same fleet and station.

Every flight leg $l \in L$ can be canceled at costs of c_l monetary units. The assignment costs of fleet k to flight leg l are represented by c_{lk} monetary units, the connection assignment costs of assigning fleet k to flight legs l_1 and l_2 consecutively are $c_{(l_1 l_2)k}$ or c_{ak}, $a = (l_1, l_2)$ monetary units. We assume the fleet-flight assignment costs of maintenance arcs already include the maintenance costs of the corresponding fleet.

4.2. **Description of the integrated fleet and tail assignment model.** The TAP can be modeled as MIP (19), following Haouari et al. (2012). The model variables are summarized in Table 6. For every flight leg $l \in L$, model (19) contains a binary variable z_l that is equal to 1 if the flight leg is canceled and 0 otherwise. Furthermore, the model contains a binary variable x_{lk} for every reasonable combination of fleets and flight legs that is 1 if fleet $k \in K$ is assigned to flight leg $l \in L_k$ and 0 if it is not.

$$\min \sum_{l \in L} c_l z_l + \sum_{k \in K} \Big(\sum_{a \in A_k} c_{ak} x_{ak} + \sum_{l \in L_k} c_{lk} x_{lk} \Big) \tag{19a}$$

$$\text{s.t.} \sum_{k \in K_l} x_{lk} + z_l = 1 \qquad \forall l \in L \tag{19b}$$

$$\sum_{a \in A_k : l = a^-} x_{ak} = x_{lk} = \sum_{a \in A_k : l = a^+} x_{ak} \qquad \forall l \in L_k, k \in K \tag{19c}$$

$$\sum_{a \in A_k : a^- = v_{ski}^b} x_{ak} = x_{ski}^b \qquad \forall s \in S, k \in K, i \in [N_k] \tag{19d}$$

$$\sum_{a \in A_k : a^+ = v_{ski}^e} x_{ak} = x_{ski}^e \qquad \forall s \in S, k \in K, i \in [N_k] \tag{19e}$$

$$x_{ski}^e \geq x_{sk(i+1)}^e \qquad \forall s \in S, k \in K, i \in [N_k - 1] \tag{19f}$$

$$\sum_{a \in A_k : a^- = v_{ski}^b} \omega_{ak} = \omega_{ski}^b \qquad \forall s \in S, k \in K, i \in [N_k] \tag{19g}$$

$$\sum_{a \in A_k : a^+ = v_{ski}^e} \omega_{ak} = \omega_{ski}^e \qquad \forall s \in S, k \in K, i \in [N_k] \tag{19h}$$

$$\omega^e_{ski} \geq \omega^e_{sk(i+1)} \qquad \forall s \in S, k \in K, i \in [N_k - 1] \tag{19i}$$

$$\sum_{a \in A_k : a^- = l} \omega_{ak} - \sum_{a \in A^{NM}_k : a^+ = l} \omega_{ak} = \vartheta_l x_{lk} \qquad \forall l \in L_k, k \in K \tag{19j}$$

$$t_{a^-} x_{ak} \leq \omega_{ak} \leq t_k x_{ak} \qquad \forall a \in A_k, k \in K \tag{19k}$$

The binary variables x_{ak} take the value 1 if and only if an aircraft of fleet k is assigned to connection a.

Variable	Domain	Indices	Description
z_l	$\{0,1\}$	$l \in L$	1 if and only if flight leg l is canceled
x_{lk}	$\{0,1\}$	$l \in L_k, k \in K$	1 if and only if leg l is assigned to fleet k
x_{ak}	$\{0,1\}$	$a \in A_k, k \in K$	1 if and only if connection a is assigned to fleet k
ω_{ak}	$[t_{a^-}, t_k]$	$a \in A_k, k \in K$	Time-elapsed-since-maintenance transmission of fleet k on connection a
x^e_{ski}	$\{0,1\}$	$s \in S, k \in K,$ $i \in \{1, ..., N_k\}$	1 if and only if i or more aircraft of type k terminating at station s
ω^e_{ski}	$[0, t_k - t_s]$	$s \in S, k \in K,$ $i \in \{1, ..., N_k\}$	Time-elapsed-since-maintenance of aircraft i of fleet k terminating at station s

TABLE 6. Model variables of TAP

The variables ω_{ak} can be interpreted as the time elapsed since the last maintenance event has taken place on the aircraft that is assigned to flight leg a^-, but only if the corresponding connection variable x_{ak} has value 1, otherwise, the variable is 0. The ω-variables are thus prevented from exceeding the maximal time allowed between two maintenance checks, t_k. The binary variables x^e_{ski} provide information on whether at least i aircraft of fleet k terminate their last assigned flight leg at station s: $\sum_{i \in \{1,...,N_k\}} x^e_{ski} = \max\{i | i = 0 \vee x^e_{ski} = 1\}$. The continuous variables ω^e_{ski} can be interpreted as the time-elapsed-since-maintenance of aircraft i of fleet k terminating its last flight at station s. The objective function (19a) is to minimize the total assignment costs, including flight cancellation costs, fleet to flight leg assignment costs, fleet through values and fleet maintenance costs (which are contained in the c_{ak} coefficients). Constraint (19b) ensures that every flight leg is assigned to exactly one fleet or is otherwise canceled. Constraints (19c) make certain that for every flight leg exactly one predecessor and one successor connection variable with the same fleet is set to 1 if it is assigned to this fleet, and all connections are set to 0 if it is not assigned to this fleet. Constraint (19d) ensures that the amount of aircraft of type k starting their route at station s is exactly the amount of aircraft of type k that are there at the beginning of the period. Constraint (19e) reflects the fact that every aircraft of type k terminating at station s increases the value of x^e_{ski} by 1. Constraint (19f) is not necessary, but prevents the existence of equivalent solutions which can be produced by simply exchanging the values of x^e_{ski} and x^e_{skj} for $i, j \in [N_k]$. Thus, it serves as a symmetry breaking function. Constraint (19g) ensures that the i-th aircraft of fleet k at station s starts with a time-elapsed-since-maintenance of exactly ω^b_{ski}, while constraint (19h) guarantees that the time-elapsed-since-maintenance of an aircraft of fleet k at station s at the end of the planning period is exactly ω^e_{ski} for an $i \in [N_k]$. Constraint (19i) is not necessary, but, like (19f) breaks the symmetry of equivalent solutions. Finally, constraint (19j) ensures that the time-elapsed-since maintenance along a single aircrafts route is increased by the flying time of the flight leg at every flight leg along non-maintenance connections. Constraint (19k) ensures that no maintenance time transmission variable exceeds the maximum allowed flying time t_k.

We are now able to describe in detail how we solve a long time TAP by decomposing it into multiple short time TAPs by slightly different means.

4.3. **Application of rolling-horizon.** In this section we prove the applicability of the rolling-horizon decomposition technique to solve TAPs with a large time horizon.

Before we do this, we briefly describe the procedure to split a long-time tail-assignment instance intuitively as described in (19) to obtain a sequence of coupled optimization problems. The set of flight legs L is ordered by departure time and then partitioned into subsets of consecutive flight legs. For each subset, a separate TAP of type (19) can be established, interpreting the x^b, ω^b parameters as variables. For each aircraft, the information on which flight leg was the last flown is transferred implicitly. These smaller TAPs can be interpreted as single-period problems P_t. The set Ξ_t is the (reasonable) domain for x^b, ω^b variables of P_t. Analogously the set Θ_t is the (reasonable) domain for x^e, ω^e variables of P_t, and the set X_t denotes the domain of the remaining variables of P_t.

When two or more consecutive single-period problems are merged, the merging constraints $(x^e, \omega^e)_{t-1} = (x^b, \omega^b)_t$ can be reinterpreted as a set of connection variables and corresponding constraints contained in the original, large TAP. Hence, a combined multi-period problem is also simply a TAP.

It remains to show that Theorem 2 is applicable to a reasonably general class of TAPs. This would imply that our rolling-horizon approach solves long-time TAPs with provable solution quality bounds. This is a non-trivial task, since there are simple examples for TAPs where bounds cannot be derived. For this reason, we introduce Lemma 3 with conditions very likely to be fulfilled in practice. This lemma presents conditions that are sufficient, but not necessary, for the conditions of Theorem 2 to be fulfilled.

Lemma 3. *Let $(P_t)_{t\in\mathbb{N}}$ be a sequence of TAPs defined with data structured as in Table 5 and modeled as in problem (19). Let Ξ_t and Θ_t be the set of feasible start states and end states of the single-period TAP P_t, and $z^{\cdot,\cdot}_{t,\mu-1}$ be the optimal value function of the μ-period TAP beginning at period t, $P_{t,\mu}$ as defined in Section 2, with fixed or free border states. If the following conditions hold:*

- *(C1) if the schedule contains a flight from airport A to airport B, it contains flights from A to B. Furthermore, there exists $\Delta \in \mathbb{R}^+$ such that the flights are regularly repeated with a maximum time difference of Δ time units*
- *(C2) if a flight from airport A to airport B can be flown by an aircraft type k, this also applies to every flight from A to B and every flight from B to A*
- *(C3) if there is a flight path from airport A to airport B for aircraft type k, then there is also a flight path from B to A for this type of aircraft*
- *(C4) the set of start states of the first period contains exactly one element. The start-state set of $(n+1)$-th period equals the end-state set of period n. There is a feasible solution of $P^{\cdot,\vartheta}_{0,n}$ for every $\vartheta \in \Theta_n$, for all $n \in \mathbb{N}$*
- *(C5) start states (x^b, ω^b) are pairs of location and time elapsed since maintenance such that it is possible to get back to a maintenance station along a path consisting of scheduled flight legs*
- *(C6) end states (x^e, ω^e) are pairs of location and time elapsed since maintenance that they can be reached by a just maintained aircraft at a maintenance station along a path consisting of scheduled flight legs*
- *(C7) every time period has a minimum duration of $\delta > 0$ time units*
- *(C8) assignment costs per period are globally bounded from above by a constant $C \in \mathbb{R}$ and from below by a constant $c > 0$*

then, the conditions of Theorem 2, equations

$$\alpha := \sup_{t\in\mathbb{N},(x^b,\omega^b)_t\in\Xi_t,(x^e,\omega^e)_{t+m}\in\Theta_{t+m}} z_{t,m}^{(x^b,\omega^b)_t,(x^e,\omega^e)_{t+m}} < \infty, \tag{8}$$

and

$$\lim_{\mu\to\infty}\inf_{t\in\mathbb{N}} z_{t,\mu-1} = \infty, \tag{9}$$

hold for an $\alpha \leq \frac{C}{c}\sum_{k\in K}\lceil\frac{\Delta}{\delta}|S|+t_k\rceil N_k$ *and* $m = \sum_{k\in K}\lceil\frac{\Delta}{\delta}|S|+t_k\rceil N_k$. *These conditions imply that a sequence* $(\varepsilon_\mu)_{\mu\in\mathbb{N}}$ *converging to* 0 *with*

$$\sup_{t\in\mathbb{N},\xi_t\in\Xi_t}\frac{z_{t,\mu-1}^{\xi_t,\cdot}}{z_{t,\mu-1}} \leq (1+\varepsilon_\mu) \tag{10}$$

exists, such that Algorithm 2 or Algorithm 3 can find arbitrarily good solutions. Theorem 2 is therefore applicable to the TAP sequence.

Lemma 3 implies that the rolling-horizon approach delivers solutions with arbitrarily good solution quality bounds when applied to sufficiently long finite problem sequences. Although it has many conditions, they are likely to be met in TAPs relevant for practical use. Condition (C1) and (C2) hold, since it is common for a regular airline schedule to contain flights which are repeated daily or weekly. An aircraft of a certain type has the ability to fly a specified maximal distance without stops. We assume the same distance for flights between the same airports, independent of their directions. Thus, every fleet, which can be assigned to a flight leg from airport A to airport B is also able to return from B to A. Schedules for which (C3) does not hold are simply not reasonably operable, since aircraft used to execute flights from A to B without the ability to return will get stuck at airport B. This behavior is caused by the Hub-and-Spoke structure most airline networks are based on. Condition (C4) is plausible, since at the beginning of the schedule, the start state is fixed, because every aircraft has a current location and has to be maintained at some time. These attributes can obviously not be changed without additional costs. The second part of the condition can be assumed implicitly, since it is again not plausible to assume end states which cannot be obtained. Conditions (C5) and (C6) are also plausible due to the fact that it is clearly not beneficial to allow aircraft start configurations that prevent some aircraft from continuing with their flights because they cannot reach a maintenance station. It is also not plausible to consider aircraft end configurations which cannot be reached. Furthermore, these two conditions are easy to check. Conditions (C7) and (C8) hold for all practically relevant flight schedules, since they always contain a finitely bounded number of flight legs per constant time span whose operating costs will not exceed a threshold C. Furthermore, operation of parts of the flight schedule will not be free, so the lower cost bound c is also plausible. We will now prove the Lemma 3.

Proof. According to Corollary 3, (C8) implies condition (12). Condition (11) can be obtained as follows. According to (C1), (C2) and (C3), the set of airports can be divided into subsets of airports for every aircraft type k. These subsets have the property that every airport can be reached along a path consisting of scheduled flights from every other airport in the subset with aircraft type k, regularly and infinitely often.

If such a subset, or component, does not contain a maintenance station, (C5) ensures that the only acceptable start state is to have zero aircraft of type k at these stations. The only acceptable end states are the same as a result of (C6). This is due to the fact that aircraft of type k are not able to reach a

maintenance station from these stations along a flight leg path or to reach one of these stations along a flight leg path from a maintenance station. The start/end-state sets of the periods assign a unique number of aircraft of type k to these components, since every start state must be reachable from the first start state, according to (C4), and there is no flight flyable by aircraft type k that connects the components. It remains to show that every end state can be reached from every start state in a bounded number of periods. This holds, since an aircraft can be transferred with arbitrary time elapsed since maintenance to every other location within the same component, with minimal time elapsed since maintenance. This can be done by assigning it along the shortest path (in terms of flying time consumption) to a maintenance station, which is possible according to (C6), and then assigning it to the shortest path (in terms of flying time consumption) to the desired location. According to (C1) and (C7), this takes at most $\lceil \frac{\Delta}{\delta}|S| + t_k \rceil$ time periods per aircraft and can be successively applied to every aircraft in the schedule, leading to an upper bound for α in equation (11) according to (C8):

$$\alpha \leq \frac{C}{c} \sum_{k \in K} \lceil \frac{\Delta}{\delta}|S| + t_k \rceil N_k$$

valid for $m = \sum_{k \in K} \lceil \frac{\Delta}{\delta}|S| + t_k \rceil N_k$. This completes the proof. □

We have shown, that from a theoretical point of view, it makes sense to apply the rolling-horizon approach to large tail assignment problems. The bound which was determined for α in the proof is a theoretical bound. However, it should be mentioned that in practice, considerably tighter bounds can be calculated based on problem-specific characteristics. Furthermore, if a larger variety of corrective actions, such as deadhead flights for aircraft transfer, is allowed, the proof simplifies, such that the result can straightforwardly be shown to have a considerably lower bound for α. Computational results concerning real-world TAPs will be presented in Section 5.

5. Computational Experiments

This section describes the computational experiments we performed as part of this study. The aim is to determine whether large-scale tail-assignment instances can be solved with our rolling-horizon approach. We conduct a mean value study with 50 large instances and look at four larger instances consisting of 1500 to 2300 flights each. All instances Glomb and Rösel (2020) are publicly available in case interested authors want to use them. First we will explain how the instances were created and describe their structure. Then we detail our test procedure and the different techniques used. Finally we compare our results and discuss whether and in which cases our method is able to find solutions with a low optimality gap.

5.1. **Generating instances.** An integrated fleet and tail assignment is performed here as introduced in Section 4, including various aircraft fleets and obligatory maintenance. The underlying flight plan was recorded as a screenshot from a particular week of the airline Lufthansa and consists inter alia of flight numbers, start and destination, as well as planned flight times. The majority of the flights are inner-European, hence they are served from the hub airports in Frankfurt (FRA) and Munich (MUC). The aircraft fleet was reconstructed based on the data published online by Lufthansa.
The resulting weekly timetable comprises 1666 flight numbers with 10082 direct flights, connecting a network of over 200 airports. The aircraft data includes 13 aircraft types and a total of 335 aircraft.
These basic data sets are always used and completed appropriately to create the instances. First, boundary parameters are set that roughly characterize the instance. The two most important are the number of

flights to be considered and the number of airports to be included. To prevent single fleets from being too small, an additional lower bound is defined, which specifies the minimum number of aircraft of a single type that must be used if the fleet is to be included in the instance. We also control the cost factors of the instance by giving specific intervals from which a random cost rate can be drawn. This includes the cost per passenger incurred in the event of an overbooking or complete cancellation of a flight and fixed kilometer operating cost rates for different fleets. Furthermore, the duration of a maintenance event, the maximum flight time between two maintenance operations and the maximum number of take-offs between two maintenance operations are specified. The full schedule is filtered randomly for a sub network which contains all flights connecting the airports FRA, MUC, and a dedicated, instance specific set of other airports. We asserted not to construct the same instance twice. After the schedule is created, a reduced fleet assignment is performed on this schedule. This is done for the purpose of using suitable parts of the entire fleet to serve the plan. We do not incorporate this solution in our computations and therefore compute a completely new fleet assignment later. The data concerning the schedule, the airports and the aircraft are now fixed and will be added according to the parameters specified at the beginning. The turn-around time depends linearly on distance and demand of the first and the second flight of the corresponding connection. Costs per kilometer rates depend linearly on the size of the corresponding aircraft, measured in seats. The maximal flying time without maintenance of an aircraft type is linearly dependent on the range of this aircraft type. Through values per connection are dependent on the amount of time between the first and the second flight of the connection. Furthermore, an initial value for the time since last maintenance event is determined randomly for each aircraft, such that the aircraft is still able to feasibly reach a maintenance station.

5.2. **Description of the computational evaluation.** For the computational study, the rolling-horizon approach is applied to the TAP. For the calculations, two different test runs were carried out. First, 50 instances of comparable size and airport structure were solved. The results of the calculations are averaged and evaluated. For the second calculation, 4 large instances were created, which cover considerably more flights and airports, but are based on the same input parameters. These instances are listed in Table 7. The general approach to solving the problem is as follows: The flight plan is divided into individual

Instance name	**#flights**	**#airports**	**#fleets**	**#aircraft**
LargeScale1	1516	16	6	53
LargeScale2	1690	18	7	64
LargeScale3	2110	22	7	74
LargeScale4	2310	25	7	80

TABLE 7. Table of large instances for computations

time intervals. The partitioning is performed dynamically based on the number of flights contained in the interval. The allocation is used to decide how many flights are to be fixed in the calculation of a single time interval. The following time intervals are included in the optimization of the currently relevant interval to ensure that the current solution can be appropriately connected to the future assignments. This procedure corresponds exactly to Algorithm 3.

5.3. **Results of our computations.** We implemented all algorithms in Python 3.7.1 and solved the MIPs with Gurobi 8.1.0, see Gurobi Optimization (2020). All computations were executed on a 4-core machine with a Xeon E3-1240 v6 CPU (3.7GHz base frequency) and 32 GB RAM. This section presents the computational results for the previously described methods to compute the rolling-horizon approach for integrated fleet and TAPs. The focus is on the following elements:

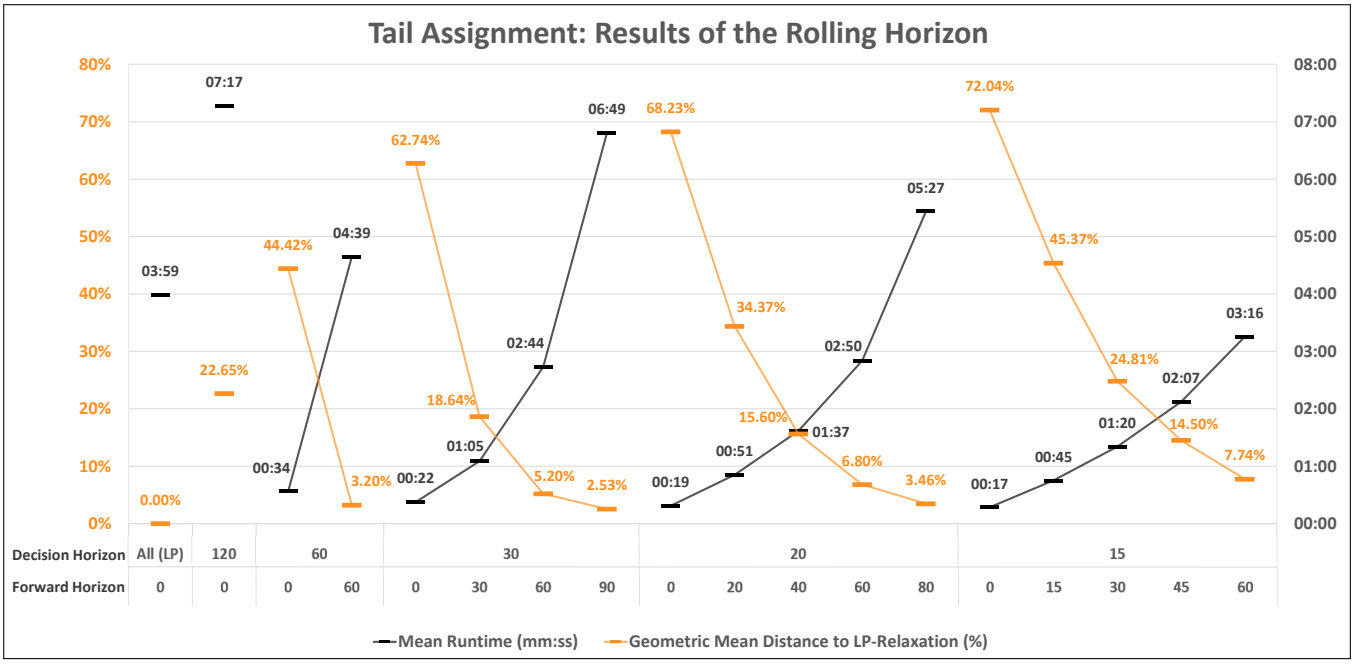

FIGURE 6. Computations: Results Mean

(1) To demonstrate suitability of RH-based approaches to solve tail assignment problems, a comparison of runtime between solving the entire integrated problem and solving the problem using Algorithm 1 as well as Algorithm 2, as well as a widely used heuristic approach, is conducted.
(2) Since the results of Section 3 indicates that Algorithm 3 is better suited than Algorithm 1 to solve special lot-sizing instances, we want to test if this result can be seen as well in other instances. Hence, a comparison of solution quality and runtime obtained by Algorithm 1, Algorithm 3 with varying safeguarding parameters and a widely used heuristic approach is conducted.
(3) It is demonstrated that Algorithm 3 can be used to solve large-scale, practically relevant instances.

In the first computation run, 50 instances with about 800 flights at 8 airports each were used and tested with different procedures. In order to be able to produce meaningful test runs in all categories, it must be estimated which horizon classification provides meaningful results within the overall time frame. To determine this, a small computational study was conducted in which a classical rolling-horizon approach was used. To realize the rolling-horizon, the time horizon was not chosen equidistant, but depending on the contained flights. The decision horizon comprises the number of flights that must be fixed in an iteration and correspond to the *current* time-period in Section 2. The forward horizon consists of additional flights, which are taken into consideration to determine a good solution but are not fixed in the current iteration. In order to obtain a lower bound on the optimal value for each instance, each was solved including all time periods (therefore without rolling-horizon) relaxing all integrality constraints. The reason is that the entire time horizon as an integrated integer model could not be solved in a reasonable time and therefore either no solutions or no optimal solutions were produced: The results were as follows with an upper time limit of 23 hours: Of 50 instances, only 23 reached a feasible solution. The mean optimality gap of these solutions was 54.87%. None of the instances was solved up to optimality. The results of the described test runs are shown in Figure 6 and give first computational insights to answer question (1): It indicates that in a standard rolling-horizon approach, it is appropriate to determine near-optimal solutions for large TAPs, while the integrated problem cannot be solved. Clearly, including 120 flights achieves good results: Once in the configuration with 60 flights in the planning horizon and 60 in the forward horizon and once

with only 30 planning flights and 90 in the forward horizon. With respect to these insights, we use these two horizon settings to test Algorithm 3 with different selections of the safeguarding parameters in order to demonstrate goal (2). Since the calculation of the fully integrated MIPs is not comparable with the results from the rolling-horizon approach, they are now compared with heuristics that are common for this problem. These heuristics decompose into fleet allocation and a subsequent aircraft routing with fixed fleet allocations. This decomposition is usually used in the planning phase of airlines. Sometimes, this decomposition leads to difficult TAPs with maintenance potentials due to unbalanced fleet sizes or to infeasible maintenance plans. To keep the test runs comparable despite these disadvantages, the maximum runtime of the routing in the heuristic was limited to one hour. Whenever no solution was found or the subsequent problem was infeasible, the test runs were removed from the mean value study.

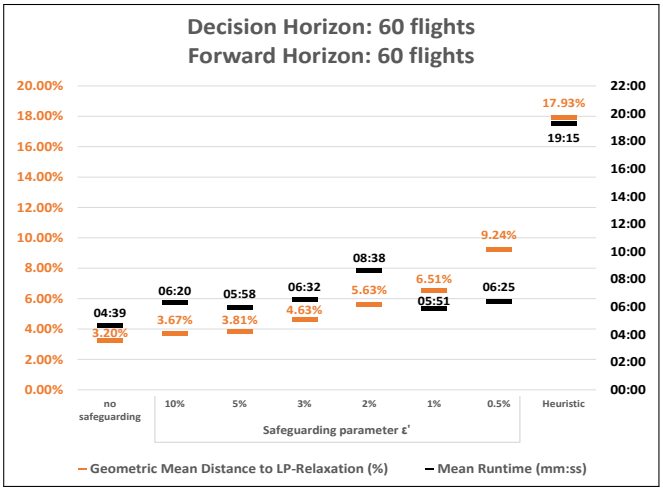

FIGURE 7. Optimality gaps for different safeguarding parameters and a heuristic solution with 60 flights in the decision horizon and 60 flights in the forward horizon

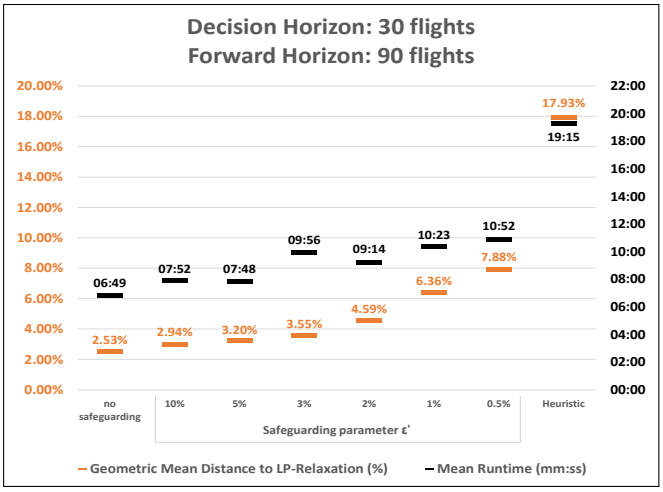

FIGURE 8. Optimality gaps for different safeguarding parameters and a heuristic solution with 30 flights in the decision horizon and 90 flights in the forward horizon

Previous test runs have shown that the empirical lower bound of ε_μ, which is in this case not easy to determine exactly, is usually above 16% and in some instances even significantly higher. Therefore, we had to apply the alternative version of Algorithm 3, mentioned in Section 2. It can guarantee a good gap for the corresponding instances in case that the safeguarding parameter ε' is chosen adequately. Due to the complexity of the problem and the structure of the instances considered, it is not easy to adequately determine the safeguarding parameter for each instance. Many conclusions can be drawn from the results of the computational study: Figure 7 and Figure 8 show the results of a computational study with our two favorite horizon settings, which allows the safeguarding parameter to vary between 10% and 0.5%. In addition, the value of the solution without the safeguarding parameter and the value of the heuristic solution are plotted in order to demonstrate the results for goal (2). First, it can be noted that the classical rolling-horizon approach performs best on average. This is often the case in practical applications, but as we have already shown, this is not true for every instance we investigated. In the midfield of the computational study, the safeguarding rolling horizon proposed is found. The worst results in terms of runtime and solution quality are provided by the classical heuristic, which is often used in applications. The established separation of fleet assignment and tail assignment hence is not a high quality method. As so often in the actual application, the instances do not run into possible extreme situations for the classical rolling-horizon approach, making it unnecessary to set a safeguarding parameter. However, the algorithm

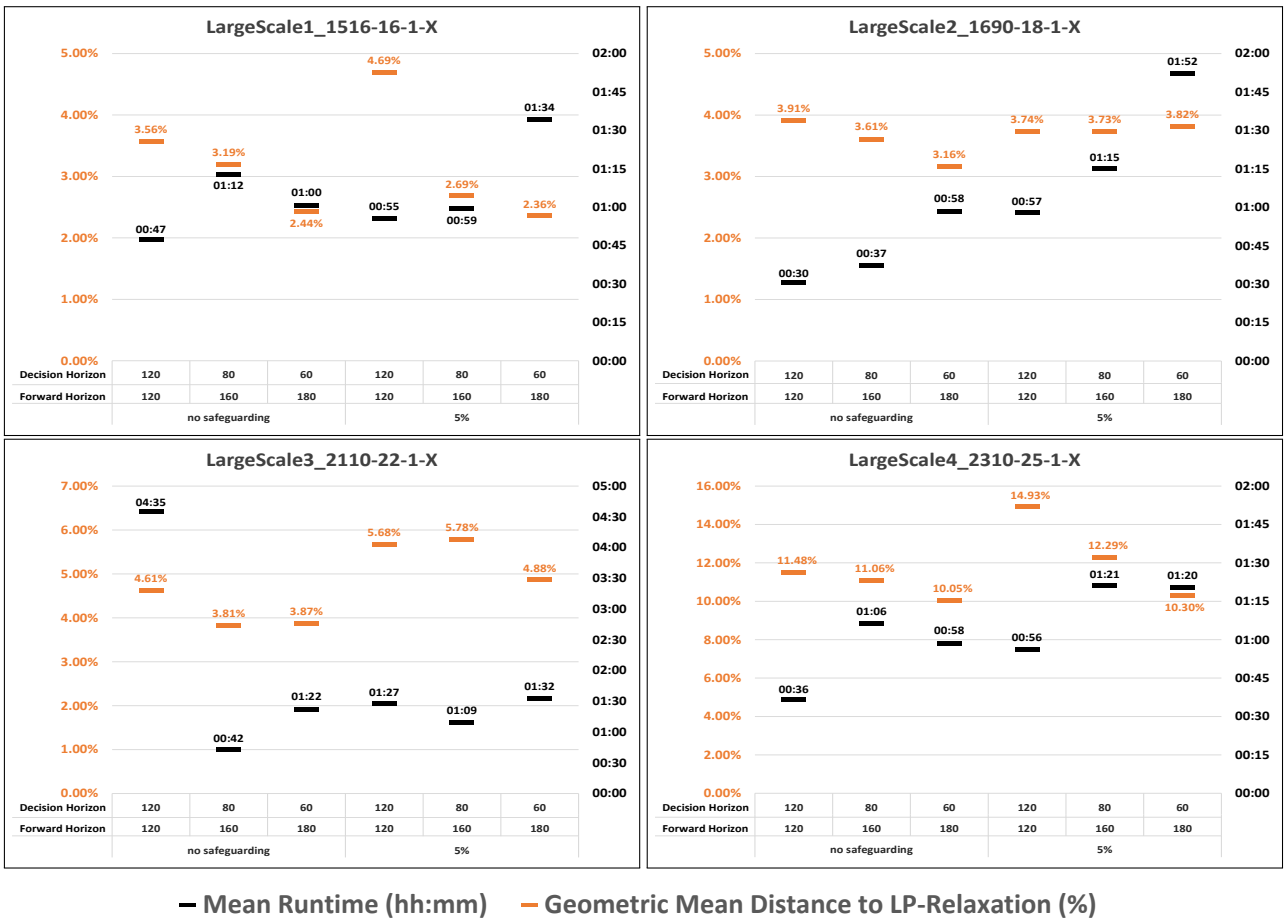

FIGURE 9. Computations: Solutions and Runtimes of real-world instances with decision horizon and forward horizon measured in number of flights

does not rely on the good behavior of the instances and can specify a quality guarantee for the solutions. The price of obtaining these guarantee stays acceptably low. We demonstrated in section 3, that there exist some applications and problem structures where this safeguarding leads to significantly better results than the classical rolling-horizon approach. For many of the tail-assignment instances considered in our mean-value study, classical rolling-horizon is not the best solution approach. Investigating the setting with 30 flights per period and three periods in advance, in 66% of the mean value study instances, the best safeguarding rolling horizon approach has obtained better results than the classical rolling horizon approach. In the setting with 60 flights per period and one period in advance, Algorithm 3 is superior for 72% of the instances. When setting the safeguarding parameter ε' to 0.1, in the setting 30 flights/3 periods in advance, in 48% of the mean value study instances the safeguarding rolling horizon approach obtains better solutions than the classical rolling horizon approach, in the setting 60 flights/1 period in advance, it is 58%. This is not reflected in the mean value study. Hence, in the case where efficient parameters ε' can be guessed, Algorithm 3 is competitive. This has also been demonstrated in the study conducted for easy lot-sizing problems in Section 3. Figure 7 also shows that a conservative choice of ε' can even accelerate the solution process.

Since the test results of the mean study is promising, further experiments on the computational study were carried out with even larger instances based on real data. The latter has been performed in order to demonstrate goal (3) The results can be seen in Figure 9. It turned out that an integrated model cannot be solved for any of these large instances, since an out-of-memory error occurred when the problem was set

up. The calculation of the LP-relaxation solution overall flights took between 40 minutes and 3.5 hours. Due to the large number of flights in the overall plan, the parameters of our rolling-horizon approach were adjusted. The number of flights per period was 240 for all computations, whereby the number of flights to be fixed was varied. All calculations were performed both without safeguarding parameter and with a safeguarding parameter of 5%. In general, the standard algorithm performs better here as well. However, the behavior described above can be confirmed: Although ε_μ is also beyond 15% in the instances considered here, the safeguarding constraint in Instance 1 improved the runtime and the solution for 80 flights in the decision horizon and 160 flights in the forward horizon. In the constellation with 120 flights in both the decision and forward horizon, the runtime improvement is very significant. This can be due to large fluctuations in subsequent decisions, which lead to unfavorable maintenance procedures and runtime extensions in the non-constrained case. In summary, the results show the effectiveness of the approach. We received solutions for realistic large-scale instances, which are integral, including fleet assignment and maintenance planning aspects. Neither the fleet assignment nor the maintenance planning had to be considered separately, but could be directly incorporated into the TAP. The ability to solve these strategic problem instances in less than 12 hours with an approach, that linearly scales in the length of the considered time frame, opens up the possibility of even solving entire six-monthly flight schedules. The calculated applicable solutions do not depend on cyclic structural components and feasibility of upstream operational decisions. These solutions come with a provable guarantee. The improved rolling-horizon proved, that the safeguarding constraint is not only applicable to ensure provable gaps, but also to improves solutions and runtimes in special problem settings.

6. Conclusion, Outlook

This paper has provided a new rolling-horizon algorithm and derived conditions under which the algorithm delivers provable near-optimal solutions for sequences of coupled optimization problems. Some easy to check properties of problem sequences sufficient for the conditions to be fulfilled were provided and the applicability of the technique to various problem classes was shown, e.g. to lot-sizing problems. We further showed that this technique can be used to solve large-scale TAPs, and transferred our theoretical solution quality bounds to the context of TAPs. Extensive computational study demonstrated that the rolling-horizon approach is a highly effective means of solving real-world tail assignment instances in a reasonable amount of time to near-optimality. Some forward parameters must be considered as uncertain in some application settings, for example for tail assignment problems with uncertain demand or weather prognoses. In this case, it is of interest to research how the rolling-horizon approach can be extended so that it is able to hedge against uncertainties.

Acknowledgements

This research was conducted within the framework of the research project OPs-TIMAL, financed by the German Federal Ministry of Economic Affairs and Energy (BMWi).

References

Absi, N. and van den Heuvel, W. (2019). Worst case analysis of relax and fix heuristics for lot-sizing problems. *European Journal of Operational Research*, 279(2):449–458.

Addis, B., Carello, G., Grosso, A., and Tànfani, E. (2016). Operating room scheduling and rescheduling: a rolling horizon approach. *Flexible Services and Manufacturing Journal*, 28(1-2):206–232.

Aggarwal, D., Saxena, D. K., and Emmerich, M. T. (2019). Interdependence and integration among components of the airline scheduling process: A state-of-the-art review.

Ahmed, M. B., Mansour, F. Z., and Haouari, M. (2018). Robust integrated maintenance aircraft routing and crew pairing. *Journal of Air Transport Management*, 73:15–31.

Akartunalı, K., Fragkos, I., Miller, A. J., and Wu, T. (2016). Local cuts and two-period convex hull closures for big-bucket lot-sizing problems. *INFORMS Journal on Computing*, 28(4):766–780.

Akartunalı, K. and Miller, A. J. (2012). A computational analysis of lower bounds for big bucket production planning problems. *Computational Optimization and Applications*, 53(3):729–753.

Baena, D., Castro, J., and González, J. A. (2015). Fix-and-relax approaches for controlled tabular adjustment. *Computers & Operations Research*, 58:41–52.

Bean, J. C. and Smith, R. L. (1984). Conditions for the existence of planning horizons. *Mathematics of Operations Research*, 9(3):391–401.

Bertazzi, L. and Maggioni, F. (2018). A stochastic multi-stage fixed charge transportation problem: Worst-case analysis of the rolling horizon approach. *European Journal of Operational Research*, 267(2):555–569.

Borndörfer, R., Dovica, I., Nowak, I., and Schickinger, T. (2010). Robust tail assignment.

Bylka, S. and Sethi, S. (1992). Existence and derivation of forecast horizons in a dynamic lot size model with nondecreasing holding costs. *Production and Operations Management*, 1(2):212–224.

Chand, S., Hsu, V. N., and Sethi, S. (2002). Forecast, solution, and rolling horizons in operations management problems: A classified bibliography. *Manufacturing & Service Operations Management*, 4(1):25–43.

Chand, S., Sethi, S. P., and Proth, J.-M. (1990). Existence of forecast horizons in undiscounted discrete-time lot size models. *Operations Research*, 38(5):884–892.

Cruise, J., Flatley, L., Gibbens, R., and Zachary, S. (2019). Control of energy storage with market impact: Lagrangian approach and horizons. *Operations Research*, 67(1):1–9.

Degraeve, Z. and Jans, R. (2003). Improved lower bounds for the capacitated lot sizing problem with set up times.

Fattahi, M. and Govindan, K. (2020). Data-driven rolling horizon approach for dynamic design of supply chain distribution networks under disruption and demand uncertainty. *Decision Sciences*.

Fragkos, I., Degraeve, Z., and De Reyck, B. (2016). A horizon decomposition approach for the capacitated lot-sizing problem with setup times. *INFORMS Journal on Computing*, 28(3):465–482.

Froyland, G., Maher, S. J., and Wu, C.-L. (2013). The recoverable robust tail assignment problem. *Transportation Science*, 48(3):351–372.

Fuentes, M., Cadarso, L., and Marín, Á. (2018). A fix & relax matheuristic for the crew scheduling problem. *Transportation research procedia*, 33:307–314.

Georghiou, A., Tsoukalas, A., and Wiesemann, W. (2019). Robust dual dynamic programming. *Operations Research*, 67(3):813–830.

Glomb, L. and Rösel, F. (2020). Problem instances for fleet assignment / tail assignment / maintenance planning. `https://faubox.rrze.uni-erlangen.de/getlink/fiEp4xZJQXPqK9wK42c1H1hs/a_rolling_horizon_approach_for_multi_period_optimization.zip`.

Gurobi Optimization, L. (2020). Gurobi optimizer reference manual.

Haouari, M., Shao, S., and Sherali, H. D. (2012). A lifted compact formulation for the daily aircraft maintenance routing problem. *Transportation Science*, 47(4):508–525.

Kelly, J. D. and Mann, J. L. (2004). Flowsheet decomposition heuristic for scheduling: a relax-and-fix method. *Computers & chemical engineering*, 28(11):2193–2200.

Maher, S. J. (2016). Solving the integrated airline recovery problem using column-and-row generation. *Transportation Science*, 50(1):216–239.

Malandrino, F., Casetti, C., and Chiasserini, C. (2013). A fix-and-relax model for heterogeneous lte-based networks. In *2013 IEEE 21st International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 308–312.

Mohammadi, M., Ghomi, S. F., Karimi, B., and Torabi, S. A. (2010). Rolling-horizon and fix-and-relax heuristics for the multi-product multi-level capacitated lotsizing problem with sequence-dependent setups. *Journal of Intelligent Manufacturing*, 21(4):501–510.

Mohammadi, M. and Poursabzi, O. (2014). A rolling horizon-based heuristic to solve a multi-level general lot sizing and scheduling problem with multiple machines (mlglsp_mm) in job shop manufacturing system. *Uncertain Supply Chain Management*, 2(3):167–178.

Nielsen, L. K., Kroon, L., and Maróti, G. (2012). A rolling horizon approach for disruption management of railway rolling stock. *European Journal of Operational Research*, 220(2):496–509.

Oliveira, J. D. and Scarpin, C. T. (2020). A relax-and-fix decomposition strategy based on adjacent nodes applied to the periodic capacitated arc routing problem (pcarp). *IEEE Latin America Transactions*, 18(03):573–580.

Pochet, Y. and Wolsey, L. A. (2006). *Production planning by mixed integer programming*. Springer Science & Business Media.

Postek, K. and Hertog, D. d. (2016). Multistage adjustable robust mixed-integer optimization via iterative splitting of the uncertainty set. *INFORMS Journal on Computing*, 28(3):553–574.

Ribeiro, C. C., Santos, T. d. A., and de Souza, C. C. (2019). Multicast routing under quality of service constraints for vehicular ad hoc networks: mathematical formulation and a relax-and-fix heuristic. *International Transactions in Operational Research*, 26(4):1339–1364.

Sethi, S. and Sorger, G. (1991). A theory of rolling horizon decision making. *Annals of Operations Research*, 29(1):387–415.

Shao, S., Sherali, H. D., and Haouari, M. (2017). A novel model and decomposition approach for the integrated airline fleet assignment, aircraft routing, and crew pairing problem. *Transportation Science*, 51(1):233–249.

Sherali, H. D., Bae, K.-H., and Haouari, M. (2013). A benders decomposition approach for an integrated airline schedule design and fleet assignment problem with flight retiming, schedule balance, and demand recapture. *Annals of Operations Research*, 210(1):213–244.

Sinclair, K., Cordeau, J.-F., and Laporte, G. (2016). A rolling horizon heuristic for aircraft and passenger.

Stadtler, H. (2003). Multilevel lot sizing with setup times and multiple constrained resources: Internally rolling schedules with lot-sizing windows. *Operations Research*, 51(3):487–502.

Wolsey, L. A. (1998). *Integer programming. Series in Discrete Mathematics and Optimization*. Wiley-Interscience New Jersey.

Yilmaz, H. Ü., Mainzer, K., and Keles, D. (2020). Improving the performance of solving large scale mixed-integer energy system models by applying the fix-and-relax method. In *2020 17th International Conference on the European Energy Market (EEM)*, pages 1–5. IEEE.