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 optimum 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. These problems can be split into time periods, which may have periodic properties. If the time-span is large, solving the entire optimization problems results in large optimization models with enormous numbers of variables and constraints.

For this reason, large-scale instances with a long time horizon are usually solved using decomposition approaches. These reduce the problem size significantly in order to obtain computational tractability. However this typically comes at the expense of solution quality due to the fact that existing coupling constraints connecting subsequent time steps are ignored. Two of the most common decomposition techniques in linear optimization are Lagrangian Decomposition and Benders’ Decomposition. These methodologies can be applied to such a sequence of optimization problems as follows. If there are some constraints which link single time periods, Lagrangian Decomposition can be used to solve the entire problem. To do this, the coupling constraints are relaxed and the constraint violation is penalized. The result is a block-structured sub problem. Afterwards, the Lagrangian Function is optimized with state-of-the-art algorithms.

Alternatively, Benders’ Decomposition can be used if there are certain variables expressing e.g. some goods inventory amount, which can be fixed to specific values. The fixing of coupling variables also
results in a block-structured sub problem. Afterwards, the problem can be solved by state-of-the-art algorithms. Both procedures iteratively solve single-period optimization problems.

If the large problem can be split into time periods which are subsequently coupled, the resulting sub problems can be solved progressively. Optimizing each time step individually can result in decisions being made which are beneficial for the current time period, but may have negative effects on subsequent time periods. This could lead to overall solutions of low quality with respect to the objective function, both in theory and in practice. To compensate for this, in this paper we take into account full or partial information of one or more subsequent periods. This broadly describes the rolling-horizon approach, a decomposition technique that has been applied to many problem classes.

We focus on the rolling-horizon approach for a number of reasons. First, the computational efficiency of Lagrangian Decomposition or Benders’ Decomposition strongly depends on how many constraints are relaxed or variables are fixed, respectively. The number of coupling constraints or variables may be large. Thus, the computational effort of these decomposition approaches may not be significantly lower than solving the entire problem. Second, in many applications the priority may be to quickly obtain an acceptable solution for the current time period. While this is possible with the rolling-horizon approach, other decomposition techniques do not allow for this. Furthermore, Lagrangian Decomposition and Benders’ Decomposition share the disadvantage that they are restricted to special classes of optimization problems, e.g., they do not allow for integrality constraints in the single-period problems.

Here, we examine the rolling-horizon approach in terms of its solution quality. For this purpose, we propose examples which illustrate the generally poor quality of the solution. Furthermore, we discuss the choice of the length of the forward horizon. This specifies how many proximately following time periods should be included in the optimization process of the current period. We develop a theoretical framework for time decomposition methods and modify the rolling-horizon approach by adding additional constraints, such that we can state conditions under which the approach delivers solutions with quality guarantees. These include the condition that the ratios of optimal values under the worst start states to those under the best start states of the optimization problems are bounded. This ratio is then exactly the relative solution quality guaranteed by our approach. We further show that this ratio is small if certain requirements to the problem sequence are met.

We demonstrate empirically that our framework is applicable to a large class of optimization problems with an illustrative lot-sizing example and also by large-scale optimization problems emerging in airline operations. We investigate the cost optimal assignment of a set of heterogeneous aircraft to a set of scheduled flights. The aircraft have to fulfill specific safety regulations and must be maintained regularly. This problem is known as “integrated fleet and tail assignment problem with maintenance constraints” in the literature, hereafter referred to as the “tail-assignment problem” (TAP). The decomposition techniques we use have been applied to similar problems before. However, to the best of our knowledge they have not been investigated theoretically in the context of airline operations before. The computational results prove that solutions for realistic instances of TAPs obtained by the rolling-horizon approach are only a few percent away from optimality.

The remainder of this paper is organized as follows. In Section 2 we give a brief overview of relevant literature. In Section 3, 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 4 we briefly demonstrate the applicability of our algorithm to lot-sizing problems in production design. In Section 5 we introduce the optimization model for the TAP and describe how the
A ROLLING-HORIZON APPROACH FOR MULTI-PERIOD OPTIMIZATION

methodology from Section 3 is applied to solve TAPs. In Section 6, 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. A calculation for an instance of this size takes a few hours. Finally, in Section 7, we summarize the main results of this paper and offer suggestions for future research directions.

2. Literature

An extensive survey of the theory and application of the rolling-horizon approach can be found in Chand et al. (2002). Example problems include inventory management as described in Morton (1978), machine scheduling, for example in Ovacik and Uzsoy (1995), capacity expansion problems as in Ryan (1998) and dynamic lot-sizing, see e.g. Sung and Lee (1993). Some authors have also applied the rolling-horizon approach to transportation problems, e.g. Nielsen et al. (2012).

There are some theoretical convergence studies on the rolling-horizon approach. Bean and Smith (1984) proved that decision horizons, i.e., a set of time periods where the optimal variable values can be determined by observing a finite number of 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. Many theoretical results are tailored for specific problem classes. For example, Chand et al. (1990) and Bylka and Sethi (1992) proved that decision horizons exist for their version of a stationary non-discounted lot-sizing model.

The TAP was modeled early, e.g. in Abara (1989), for example. Some authors, like Hane et al. (1995), propose the fleet assignment, the assignment of aircraft types to flights, as a separate task. However, it was shown that a decomposition into fleet assignment and tail assignment may lead to poor results in some cases, see Barnhart et al. (1998). We therefore investigate the fleet and tail-assignment problem in an integrated manner. The string based modeling of this version of the TAP was developed by Barnhart et al. (1998). The number of variables is exponential in the input length and they also include constraints that ensure that aircraft are maintained regularly. This is necessary, as pointed out in Feo and Bard (1989). An alternative model for the TAP is offered by Haouari et al. (2012), using RLT-techniques to eliminate non-linearities. The size of the model is polynomial in the input length. This is the model we chose to model TAPs.

The rolling-horizon approach has been applied to the TAP before, for example by Sinclair et al. (2016). However, to the best of our knowledge, it has never been investigated theoretically in the context of airline operations. Additional references to the literature will be provided in the following chapters.

3. 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 \( \{P_0, ..., P_T\} \), defined below in (1). Each belongs to a time period \( t \in [T] := \{0, \ldots, T\} \), where \( T \in \mathbb{N}_0 \). For each time period \( t \), we further assume that every problem \( P_t \) has a set of variables with start-state variables \( \xi_t \) defined on a domain \( \Xi_t \) and a set of end-state variables \( \vartheta_t \) defined on a domain \( \Theta_t \). Interior variables \( x_t \) are neither start-state nor end-state variables and are defined on a domain \( X_t \). The start-state variables \( \xi_t \) and the end-state variables \( \vartheta_t \) 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: 
\[ \Theta_t \subseteq \Xi_{t+1} \text{ for all } t \in [T-1]. \]
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 \( f_t : X_t \rightarrow \mathbb{R} \) and constraint functions \( g_t : \Xi_t \times X_t \times \Theta_t \rightarrow \mathbb{R}^{m_t} \) 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

\begin{align*}
(1a) \quad & \min \ f_t(x_t) \\
(1b) \quad & \text{s.t. } g_t(\xi_t, x_t, \vartheta_t) \leq 0 \\
& \quad (\xi_t, x_t, \vartheta_t) \in \Xi_t \times X_t \times \Theta_t.
\end{align*}

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 \( t \in [T] \) 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 \( g_t(\xi_t, x_t, \vartheta_t) \leq 0 \).

In practice, this assumption is often not very restrictive. Many real-world optimization problems, for example in logistics, can be formulated such that feasibility is guaranteed. We define problems with fixed start states as \( P_t^{\xi} \), the problem with fixed end states as \( P_t^{\vartheta} \) 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 \( \mu \in \mathbb{N}_0 \), \( \mu + t \leq T \), starting at time period \( t \) as

\begin{align*}
(2a) \quad & \min \sum_{j=t}^{t+\mu} f_j(x_j) \\
(2b) \quad & \text{s.t. } g_j(\xi_j, x_j, \vartheta_j) \leq 0 \quad \forall j \in \{t, \ldots, t+\mu\} \\
(2c) \quad & \xi_j = \vartheta_{j-1} \quad \forall j \in \{t+1, \ldots, t+\mu\} \\
& \quad (\xi_j, x_j, \vartheta_j) \in \Xi_j \times X_j \times \Theta_j \quad \forall j \in \{t, \ldots, t+\mu\}.
\end{align*}

We define \( P_{t,\mu}^{\xi} \), \( P_{t,\mu}^{\vartheta} \) and \( P_{t,\mu}^{\xi,\vartheta} \) as above by adding constraints \( \xi_t = \xi \) or \( \vartheta_{t+\mu} = \vartheta \) and \( z_{t,\mu}, z_{t,\mu}^{\xi}, z_{t,\mu}^{\vartheta}, z_{t,\mu}^{\xi,\vartheta} \) as the optimal value of the corresponding multi-period problem. The index \( \mu \) can be suppressed if it is 0. We assume that optimal solutions exist for some of the optimization problems above:

(A4) Optimal solutions exist: For all \( t \in [T], \mu \in [T - t] \) the problems \( P_{t,\mu} \) and \( P_{t,\mu}^{\xi_t} \) for all \( \xi_t \in \Xi_t \) attain their optimal solutions.
Most real-world optimization problems can be modeled in such a way that optimal solutions exist. 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 \( P_0, T \) and a parameter \( \mu \in [T] \). 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 \( T - \mu + 1 \) combined \( \mu \)-period problems

**Algorithm 1** Rolling-horizon algorithm

1: **INPUT:** A sequence of minimization problems \( P_{0, T}, \mu \leq T + 1 \).
2: **OUTPUT:** A solution \((\xi_t, x_t, \vartheta_t)_{t \in [T]}\) of \( P_{0, T} \).
3: **procedure** FINITE FORESIGHT OPTIMIZATION\((P_{0, T}, \mu)\)
4: \((\xi_t, x_t, \vartheta_t)_{0 \leq t < \mu} \leftarrow \text{argmin} P_{0,\mu-1}\)
5: **for** \( t \in \{1, ..., T - \mu + 1\} \) **do**
6: \((\xi_j, x_j, \vartheta_j)_{t \leq j < t + \mu} \leftarrow \text{argmin} P_{t,\mu-1}\)
7: **end for**
8: **return** \((\xi_t, x_t, \vartheta_t)_{t \in [T]}\)
9: **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 \( T \in \mathbb{N} \). Let \( \Xi_t = \mathbb{N}_0 = \Theta_t \) and \( X_t = \mathbb{N}_0 \times \{0, 1\} \) for all \( t \in [T] \). Let \( P_t \) be defined as

\[
\begin{align*}
(3a) \quad & \min f_t(x_t) := (c + 1)1_{\mathbb{R}^+}(x_1^t) + \frac{x_1^t}{x_1^t + 1} + (c + \frac{t}{t + 1}) \cdot (1 - x_1^t) + 1 \\
(3b) \quad & \text{s.t.} \quad \xi_t + x_1^t - x_2^t = \vartheta_t \\
& \xi_t, x_t, \vartheta_t \in \Xi_t \times X_t \times \Theta_t
\end{align*}
\]

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

The problem can be interpreted as follows: At the beginning of each period \( t \in [T] \), \( \xi_t \) units of a certain good are in the inventory. In every period, we must decide whether to produce nothing \((x_1^t = 0, \text{ costs } 0)\) or to produce \( x_1^t \) units of the good \((\text{costs } (c + 1) + \frac{x_1^t}{x_1^t + 1})\). In every period, we also have to decide whether to consume one unit of a certain good \((x_2^t = 1, \text{ costs } 0)\) or not to consume \((x_2^t = 0, \text{ costs } c + \frac{t}{t+1})\).
number of units in the inventory after the period \((\vartheta_t)\) is equal to the number of units in the inventory before the period plus the units produced minus the units consumed.

Let \(T \in \mathbb{N}\) and \(\mu \in [T]_t\), \(t \in [T - \mu + 1]\). 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 \(z_{t,\mu-1}^* \leq c + \mu + 2\) 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 \(P_{t,\mu-1}\) 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 \((P_{0,T}, \mu)\). In the first iteration, the algorithm solves problem \(P_{0,\mu-1}\) and obtains the optimal solution \(x^1_j = 0\) for all \(j \in \{0, \ldots, \mu - 1\}\) and \(x^2_j = 1\) for all \(j \in \{0, \ldots, \mu - 1\}\), with \(f_0(x_0) = 1\) and \(\vartheta_0\) set to \(\mu - 1\). In the next iteration, the algorithm calculates the solution of \(P_{1,\mu-1}\). The unique optimal solution of \(P_{1,\mu-1}\) for an arbitrary \(t \in [T - \mu + 1]\) is setting \(x^1_j = 0\) for all \(j \in \{t, \ldots, t + \mu - 1\}\), setting \(x^2_j = 0\) and \(x^2_j = 1\) for all \(j \in \{t + 1, \ldots, t + \mu - 1\}\) with an objective value of \(c + \mu + 1 + \frac{1}{t+1}\). For this reason, the algorithm fixes \(x^1_j = 0, x^2_j = 0\) and \(\vartheta_t = \mu - 1\) for all periods \(t \in \{1, \ldots, T - \mu + 1\}\). Thus, \(f_1(x_t) \geq c\) for all periods except the first period and the last \(\mu - 1\) periods and

\[
\sum_{j=0}^{T} f_j(x_j) \geq (T - \mu + 1)c \geq (T - \mu + 1)\frac{c}{T+1} = (1 - \frac{\mu}{T+1})c.
\]

This means that for \(\mu << T\), 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.

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

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

\[
\begin{align*}
\min \quad & \sum_{t=0}^{T} c^1_t x_t + c^2_t (1 - x_t) \\
\text{s.t.} \quad & \xi_t = x_t = \vartheta_t & \forall t \in \{0, \ldots, T\} \\
& \xi_{t+1} = \vartheta_t & \forall t \in \{0, \ldots, T - 1\} \\
& (\xi_t, x_t, \vartheta_t) \in \Xi_t \times X_t \times \Theta_t & \forall t \in \{0, \ldots, T\} .
\end{align*}
\]

![Figure 2. Algorithm 1 applied to problem (3)](image)
For every feasible solution, \( x_0 = x_1 = \ldots = x_T \) holds. The decision made in time period 0 determines all following decisions.

Assume \( \sum_{t=0}^{T} c^1_t < \sum_{t=0}^{T} c^2_t \). Thus, \( x^* = (x^*_0, \ldots, x^*_T) = (1, \ldots, 1) \) is an optimal solution for the full time horizon \( \{0, \ldots, T\} \). Let \( \mu \in [T + 1] \). In the case \( \sum_{t=0}^{\mu-1} c^1_t < \sum_{t=0}^{\mu-1} c^2_t \), Algorithm 1 applied to \( (P_0, T) \) yields the overall optimal solution since \( x_0 \) is set to 1. Otherwise, Algorithm 1 applied to \( (P_0, T, \mu) \) yields the sub-optimal solution since \( x_0 \) is set to 0. For example, for \( c^1 = (1, 0, 2, 0, 2, \ldots) \) and \( c^2 = (0, 2, 0, 2, \ldots) \), these two cases alternate.

This illustrates the fact that Algorithm 1 generally does not calculate increasingly better solutions with increased values of \( \mu \). For a theoretical comparison we introduce the straightforward Algorithm 2. It takes a parameter \( \mu \in \mathbb{N} \) and a sequence of optimization problems \( P_{0,T} \). Algorithm 2 works as shown in Figure 3. The length of the optimization problem-sequence, \( T \), is not necessarily divisible by \( \mu \). For this reason, we introduce \( \beta := T + 1 - \lfloor T \rceil_\mu \), where \( \lfloor \cdot \rceil_\mu \) 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 \( (\xi_t, x_t, \vartheta_t)_{t \in \{0, \ldots, \beta-1\}} \). The variable \( \xi_0 \) is fixed to \( \vartheta_{\beta-1} \). Then this algorithm solves the multi-period problem \( P_{\beta-1, \mu-1} \) and again fixes its solution. The variable \( \xi_{\beta+1} \) is fixed to \( \vartheta_{\beta+1} \). This procedure is continued until the entire problem sequence is solved.

**Algorithm 2** Planning horizon decomposition

1. **INPUT**: A sequence of minimization problems \( P_{0,T}, \mu \leq T + 1 \).
2. **OUTPUT**: A solution \( (\xi_t, x_t, \vartheta_t)_{t \in \lfloor T \rceil} \) of \( P_{0,T} \).
3. **procedure** PLANNING HORIZON DECOMPOSITION\((P_{0,T}, \mu)\)
4. \( \beta \leftarrow T + 1 - \lfloor T \rceil_\mu \)
5. \( (\xi_t, x_t, \vartheta_t)_{0 \leq t < \beta} \leftarrow \text{argmin} P_{0,\beta-1} \)
6. for \( j \in \{0, \ldots, \lfloor T \rceil_\mu - 1\} \) do
7. \( (\xi_t, x_t, \vartheta_t)_{j \leq t < \mu(j+1) + \beta} \leftarrow \text{argmin} P_{\beta+1, \mu-1} \)
8. end for
9. return \( (\xi_t, x_t, \vartheta_t)_{t \in \lfloor T \rceil} \)
10. **end procedure**

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

Algorithm 1 and Algorithm 2 return solutions that are feasible, but in general not optimal. Algorithm 1 fixes end-state variables considering only \( \mu-1 \) of the subsequent time-period problems, while Algorithm

![Figure 3. Schematic depiction of Algorithm 2 for \( T = 7 \) and \( \mu = 3 \).](image-url)
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.

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

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

Even though this can be a severe restriction in general, there are many problems for which this inequality holds for adequately chosen $\varepsilon_\mu$. An example is the lot-sizing problem, where rejecting demand at the expense of paying high contractual penalties is allowed. For this setting, producing nothing is an option. Independent of the initial inventory situation, the sum of all contractual penalties of all periods is an expense of paying high contractual penalties is allowed. For this setting, producing nothing is an option. Thus, the denominator is bounded from below and an $\varepsilon_\mu$ exists. In Lemma 4, we give sufficient criteria for optimization problem-sequences guaranteeing that inequality \textbf{(A5)} holds.

Although in general it is difficult to find the best $\varepsilon_\mu$, 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 are the inventory levels, the worst-case start state is an inventory of zero. 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 $P_{t,\mu-1}^\xi$ through simple operations on the parameters of the optimization problems. Furthermore, an exact knowledge of $\varepsilon_\mu$ 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 $\xi$ also provide an estimate, which can be verified after the optimization problem-sequence has been solved. Lemma 2 states that assuming \textbf{(A1)} - \textbf{(A5)}, Algorithm 2 returns a feasible solution of a certain quality. Before we prove this, we state an auxiliary lemma.

\textbf{Lemma 1.} Let $T \in \mathbb{N}$ and $P_{0,T}$ be a sequence of coupled optimization problems. Let $z^{\cdot}$ denote the optimal values of the corresponding optimization problem and $\Theta$ denote the end-state set of the corresponding optimization problem. Let $\mu \in [T]$. Let $t \in [T - \mu + 1]$ 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

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

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

\textbf{Proof.} For $\mu \in [T], t \in [T - \mu + 1]$ 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 $z_{t,\mu'-1} + z_{t+\mu',\mu''-1}$ is for $\vartheta \in \Theta_{t+\mu'-1}$ exactly the optimal value of $P_{t,\mu-1}$ with the
additional constraint \( \partial_{t+\mu'-1} = \emptyset \). The second inequality follows.

For multi-period problems with fixed start or end states, the inequalities can be similarly shown. \( \square \)

**Lemma 2.** Let \( T \in \mathbb{N} \) and \( \mu \in [T + 1] \). Let \( P_0, T \) be a sequence of coupled optimization problems with corresponding start states \( \Xi_t \) and optimal values \( z^* \). Assume \((A1) - (A5)\), i.e.,

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

for an \( \varepsilon_{\mu} > 0 \). \( z_{t,\mu-1} \) is the optimal value of the \( \mu \)-period problem \( P_{t,\mu-1} \) with free start and end states. \( z^*_{t,\mu-1} \) is the optimal value of the \( \mu \)-period problem \( P^*_{t,\mu-1} \), with start states fixed to a \( \xi \in \Xi_t \). Then, the solution returned by Algorithm 2 started with \((P_0, T, \mu)\), \( x_t \) for \( t \in \{0, \ldots, T\} \), is an \( \varepsilon_{\mu} \)-optimal solution of the problem sequence. We define \( \varepsilon_{\mu} \)-optimal as

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

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

\[
z_{0,T} - \text{Lemma 1} \geq z_{0,\beta-1} + \sum_{j=0}^{k} z_{\mu j+\beta,\mu-1}.
\]

It further holds that

\[
\sum_{t=0}^{T} f_t(x_t) = \sum_{t=0}^{\beta-1} f_t(x_t) + \sum_{j=0}^{k} \sum_{l=\mu j+\beta}^{\mu(j+1)+\beta-1} f_t(x_t) \leq z_{0,\beta-1} + \sum_{j=0}^{k} z_{\mu j+\beta,\mu-1} \leq (1 + \varepsilon_{\mu}) z_{0,\beta-1} + \sum_{j=0}^{k} (1 + \varepsilon_{\mu}) z_{\mu j+\beta,\mu-1}.
\]

Equality \((*)\) holds, because Algorithm 2 determines the values \( x_t \) such that they are optimal for \( P^*_{t,\mu-1} \) for \( t \in \{j: j \in [T - \mu + 1], j = \beta + \mu l, l \in \mathbb{N}\} \). Taking \((5)\) and \((6)\) together, we obtain

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

This is equivalent to \( \varepsilon_{\mu} \)-optimality. \( \square \)

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 (3). The relative deviation of the objective function value can be calculated as in \((A5)\):

\[
\sup_{t \in [T-\mu+1], \xi \in \Xi_t} \frac{z^*_{t,\mu-1}}{z_{t,\mu-1}} \leq \frac{c + \mu + 2}{\mu} \leq (1 + \frac{c + 2}{\mu}).
\]
Hence, we can set \( \epsilon^2 + 2 = \varepsilon \mu \), which converges to 0 as \( \mu \) goes to \( \infty \). 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 \( \beta := T + 1 - \lfloor T \rfloor \mu \). Algorithm 3 generates the constraint

\[
\sum_{l=0}^{\beta-1} f_l(x_l) \leq (1 + \epsilon)z_{0,\beta-1}
\]

in iteration 0 with adequately chosen \( \epsilon \). The algorithm then proceeds almost as Algorithm 1 until iteration \( \beta - 1 \), where instead of solving \( P_{1,\mu-1}^{\beta-1} \), it solves these problems with (8) as additional constraints. This generates the constraint

\[
\sum_{l=j}^{j+\mu-1} f_l(x_l) \leq (1 + \epsilon)z_{j,\mu-1}
\]

in iteration \( \beta \) for \( j = 0 \) and adequately chosen \( \epsilon \). Algorithm 3 proceeds almost as Algorithm 1 until iteration \( \mu + \beta - 1 \). Instead of solving \( P_{1,\mu-1}^{\beta-1} \), it solves these problems with (9) as additional constraints. Parameter \( j \) is raised by 1. This procedure is continued until the entire problem sequence is solved. The constraints are added to every \( \mu \)-period problem \( P_{t,\mu-1} \) for \( t \in \{j, \ldots, j + \mu - 1\} \). 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 \( \mu \)-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:

\[
\sum_{l=j}^{t-1} f_l(x_l) + \sum_{l=t}^{j+\mu-1} f_l(x_l) \leq (1 + \epsilon)z_{j,\mu-1}
\]

A constant term, determined in the previous steps of the algorithm. This term is dependent on the variables \( x_l \) contained in \( P_{t,\mu-1} \).

A constant term determined in the \( (j+1) \)-th step of the algorithm.

It enforces the objective of the first few periods of a \( \mu \)-period problem to be below a certain value. To be precise, the left hand side of constraints (9) 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 \( \mu \)-period problems starting at every \( \mu \)-th problem. The left hand side is constrained by the \( (1 + \epsilon) \)-fold of the optimal value of the problem \( P_{j,\mu-1} \), \((1 + \epsilon)z_{j,\mu-1} \).

Parameter \( \epsilon \) has to be set to a value such that assumption (A5) holds for \( (\mu, \epsilon) \). If such a parameter can be determined, the solutions returned by Algorithm 3 are \((1 + \epsilon)\)-optimal. We already discussed the fact that determining such an \( \epsilon \), which is preferably small, may be a difficult task for some applications, so we propose an alternative. The right hand side of constraints (9) can be set to \((1 + \epsilon')z_{j,\mu-1}^{\vartheta_{j-1}} \) with \( \epsilon' \geq 0 \). 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 \((1 + \epsilon')(1 + \epsilon) \). The
practical behavior of Algorithm 3 is very similar to Algorithm 2 if \( \varepsilon' \) is very close to 0. It becomes more similar to Algorithm 1 as \( \varepsilon' \) 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 lemma.

**Lemma 3.** Let \( T \in \mathbb{N} \) and \( P_{0,T} \) be a sequence of coupled optimization problems. Let \( \mu \in \lfloor T + 1 \rfloor \), \( \varepsilon_{\mu} > 0 \) and \( \Xi_{t} \) denote the start state domain of \( P_{t} \). Let \( z^{*}_{t} \) denote the optimal value of the corresponding multi-period problem. Assuming (A1) - (A5), Algorithm 3 started with \( (P_{0,T}, \mu, \varepsilon_{\mu}) \) returns a feasible and \( \varepsilon_{\mu} \)-optimal solution, \( x_{t} \) for \( t \in \{0, \ldots, T\} \), of the problem sequence.

**Proof.** Since we assume (A1), (A3) and (A4) for \( \mu, P_{0,\mu-1} \) and \( P_{t+1,\mu-1}^{\vartheta_{t-1}} \) for all \( t \in \{1, \ldots, T - \mu + 1\} \), regardless of \( \vartheta_{t-1} \), have an optimal solution. Algorithm 3 started with \( (P_{0,T}, \mu, \varepsilon_{\mu}) \) is able to calculate a feasible and \( \varepsilon_{\mu} \)-optimal solution of \( P_{t+1,\mu-1}^{\vartheta_{t-1}} \) with the additional constraints (9) in every \( \mu \)-th iteration of the 11-th line of the algorithm. This is because inequality (A5) guarantees the existence of a solution of \( P_{t+1,\mu-1}^{\vartheta_{t-1}} \) fulfilling constraint (9). If there exists a solution \( (x_{t}, \ldots, x_{t+\mu-1}) \) of \( P_{t+1,\mu-1}^{\vartheta_{t-1}} \) with constraint (9), it can be expanded to a solution of \( P_{t+1,\mu-1}^{\vartheta_{t-1}} \) with constraint (9). Thus, the values \( \vartheta_{t} \) are set such that problem \( P_{t+1,\mu-1}^{\vartheta_{t-1}} \) with constraint (9) can be solved. This property is retained for the next \( \mu \) periods.
We deduce that Algorithm 3 always terminates in a finite number of steps, returning a feasible solution of $P_{0,T}$.

It remains to show that this sequence is $\varepsilon_\mu$-optimal. Constraint (8) implies that inequality $\sum_{t=0}^{\beta-1} f_t(x_t) \leq (1 + \varepsilon_\mu) z_{0,\beta-1}$ holds. Constraint (9) implies that for all $t \leq \lfloor \frac{T}{\mu} \rfloor - 2$ the inequality $\sum_{j=\mu t + \beta}^{\beta-1} f_j(x_j) \leq (1 + \varepsilon_\mu) z_{\mu t + \beta, \mu - 1}$ holds. Assumption (A5) implies $\sum_{t=T-\mu+1}^{T} f_t(x_t) \leq (1 + \varepsilon_\mu) z_{T-\mu+1, \mu - 1}$. For $k := \lfloor \frac{T}{\mu} \rfloor - 2$ and $\beta := T + 1 - \lfloor \frac{T}{\mu} \rfloor$ it holds that

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

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

Assuming (A5) for $(\mu, \varepsilon_\mu)$, Algorithm 3 determines solutions which are $\varepsilon_\mu$-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 $\mu$ goes to infinity and Assumption (A5) holds for all $(\mu, \varepsilon_\mu)$.

This statement is trivial for $\mu \geq T$. 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 $T$ 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 $\mu$ is chosen sufficiently large. This is stated in the following lemma.

**Lemma 4.** Let $(P_t)_{t \in \mathbb{N}}$ be an infinite sequence of optimization problems and let (A1) - (A4) be fulfilled. Let $\Xi_t$ denote their corresponding start states and $\Theta_t$ denote their corresponding end states. Let $z_{0,t,m}$ denote their corresponding optimal values. Let the solution values of $(m+1)$-period problems be bounded for an $m \in \mathbb{N}$:

\[
\alpha := \sup_{t \in \mathbb{N}, \xi_t \in \Xi_t, \vartheta_{t+m} \in \Theta_{t+m}} \frac{\xi_t \vartheta_{t+m}}{z_{t,m}^{0,\mu-1}} < \infty.
\]

Let the solution values of $\mu$-period problems grow uniformly to infinity with $\mu$:

\[
\lim_{\mu \to \infty} \inf_{t \in \mathbb{N}} z_{t,\mu-1} = \infty.
\]

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

\[
\sup_{t \in \mathbb{N}, \xi_t \in \Xi_t} \frac{z_{t,m}^{\xi_t, \vartheta_{t,m-1}}}{z_{t,\mu-1}} \leq (1 + \varepsilon_\mu)
\]

for all $\mu \in \mathbb{N}$.

**Proof 1.** 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

\[1 \leq \frac{z_{t,m}^{\xi_t, \vartheta_{t,m-1}}}{z_{t,\mu-1}} \leq \frac{\xi_t \vartheta_{t+m}}{z_{t,m}} + \frac{\vartheta_{t+m}}{z_{t,m}^{\mu-1}} \leq \frac{\alpha}{z_{t,m}^{\mu-1}} \leq (1 + \varepsilon_\mu).
\]
By choosing $\vartheta^*_{t+m} \in \Theta_{t+m}$ minimizing $z_{t+m+1,\mu-m-2}^*\vartheta^*_{t+m}$, we obtain for every $\mu > m$

$$\frac{\alpha + z_{t+m+1,\mu-m-2}^*\vartheta^*_{t+m}}{z_{t,\mu-1}} \leq \frac{\alpha + z_{t,m+1,\mu-m-2}^*\vartheta^*_{t+m}}{z_{t,\mu-1}} = 1 + \frac{\alpha}{z_{t,\mu-1}}$$

and for this reason

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

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

Although the conditions of Lemma 4 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 $\sup\{f(x_t) | x_t \in X_t, t \in \mathbb{N}\} < \infty$, condition (10) is fulfilled.

**Proof 2.** Clearly, $z_{t,\mu}^*\vartheta$ is uniformly bounded by $\sup_{t \in \mathbb{N}} \sup_{x_t \in X_t} f_t(x_t)$, implying condition (10) 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 $c \geq 0$. Then, condition (11) is fulfilled.

**Proof 3.** 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.

## 4. 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.
4.1. **Detailed model description of the capacitated lot-sizing problem.** A typical capacitated lot-sizing problem has the structure shown in (13). The model variables are summarized in Table 1.

\[
\begin{aligned}
(13a) & \quad \min h_{-1}s_{-1} + \sum_{t \in T}(p_t x_t + h_t s_t + q_t y_t + c_t z_t + f) \\
(13b) & \quad \text{s.t. } x_t \leq C_t y_t \quad \forall t \in T \\
(13c) & \quad s_{t-1} + x_t + z_t = d_t + s_t \quad \forall t \in T \\
(13d) & \quad x_t \in \mathbb{R}^+_{0,T} \\
(13e) & \quad s_t \in \mathbb{R}^+_{0,T} \\
(13f) & \quad y_t \in \{0, 1\}^{T}.
\end{aligned}
\]

<table>
<thead>
<tr>
<th>Set</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(T = {0, \ldots,</td>
<td>T</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(h_{-1} \in \mathbb{R}^+_{0} )</td>
<td>Storage costs per item of start state in period (t = -1)</td>
</tr>
<tr>
<td>(s_{-1} \in \mathbb{R}^+_{0} )</td>
<td>Fixed storage before period (t = 0) as start state</td>
</tr>
<tr>
<td>(h_t \in \mathbb{R}^+_{0} )</td>
<td>Storage costs per item from (t) to (t + 1)</td>
</tr>
<tr>
<td>(p_t \in \mathbb{R}^+_{0} )</td>
<td>Variable costs of production per item in period (t)</td>
</tr>
<tr>
<td>(q_t \in \mathbb{R}^+_{0} )</td>
<td>Fixed costs of production in period (t)</td>
</tr>
<tr>
<td>(c_t \in \mathbb{R}^+_{0} )</td>
<td>Penalty costs per item in period (t) if demand cannot be satisfied</td>
</tr>
<tr>
<td>(f \in \mathbb{R}^+_{0} )</td>
<td>Strategic unavoidable company costs per period</td>
</tr>
<tr>
<td>(d_t \in \mathbb{R}^+_{0} )</td>
<td>Demand in time period (t)</td>
</tr>
<tr>
<td>(C_t \in \mathbb{R}^+_{0} )</td>
<td>Capacity in time period (t)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_t \in \mathbb{R}^+_{0,T} )</td>
<td>Production volume in period (t)</td>
</tr>
<tr>
<td>(s_t \in \mathbb{R}^+_{0,T} )</td>
<td>Storage amount from (t) to (t + 1)</td>
</tr>
<tr>
<td>(y_t \in {0, 1} )</td>
<td>Production indicator in period (t)</td>
</tr>
<tr>
<td>(z_t \in \mathbb{R}^+_{0} )</td>
<td>Number of missing items to fulfill the demand of period (t)</td>
</tr>
</tbody>
</table>

**Table 1.** Input data for the capacitated lot-sizing problem

The objective (13a) 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 and strategic unavoidable costs per period. Constraint (13b) 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 (13c) 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. If there is not enough production in a given time spectrum, it is possible that the production is not sufficient to meet the demand of the subsequent period. There are different approaches to dealing with this issue: Firstly, it can be assumed that it is possible 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 \(z_t\). Secondly, it is possible to allow demand not to be met, which leads to penalties in terms of dissatisfied customers. Combined approaches leading to feasibility are also conceivable. There is also the concept of **backlogging**, which allows the producer to fulfill the demand of the present period through production in subsequent periods at the expense
of penalty payments. We decided for the penalty approach and adjust constraint (13c) by a variable $z_t \in \mathbb{R}^+_0$. Additionally we add $\sum_{t \in T} c_t z_t$, where $c_t$ denotes penalty costs per item in period $t$ if the demand cannot be satisfied. It is straightforward to see that (13c) couples the individual time periods by storing the inventory over the periods. In this model variant, exactly two consecutive time 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 omitted completely providing the capacity can be sufficiently adapted to demand and the time interval examined is sufficiently long. Nevertheless, problems arise if subsequent periods are only solved as LP-relaxations: The fixed costs of production will behave like variable costs and therefore falsify the assumed objective.

4.2. 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 3, this effect can be compensated for 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.

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:

1. **(L1)** The demand per period is bounded from above: $d_t^i \leq U < \infty \quad \forall t \in T$
2. **(L2)** All cost coefficients in the objective function are assumed to be non-negative.

(Even with bounded negative objective values, it is possible to apply Section 3, because constant values can be added to the individual time periods. This will not influence the solution process.)

It is possible to derive bounds for formula (A5) with $c = \max_{t \in T} c_t$:

$$\sup_{t \in [T-\mu+1], \xi \in \Xi_t} z^\xi_{t,\mu-1} \leq \mu (f + Uc) \quad \text{and} \quad \sup_{t \in [T-\mu+1]} z_{t,\mu-1} \geq \mu f > 0.$$

Therefore, we can estimate an upper bound for $\epsilon_{\mu}$:

$$\sup_{t \in [T-\mu+1], \xi \in \Xi_t} \frac{z^\xi_{t,\mu-1}}{z_{t,\mu-1}^{\xi_{t,\mu-1}}} \leq \frac{\mu (f + Uc)}{\mu f} = 1 + \frac{Uc}{f} = 1 + \epsilon_{\mu}.$$

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 the effects of a time-decomposition approach, a brief example will now be considered.

The example illustrates how the objective value can be affected when using the Rolling Horizon. To do this, we consider a lot-sizing problem with 4 time periods. It can be assumed that these are directly consecutive, possibly as part of a larger time horizon. Thus, the value of $s_{-1}$ is interpreted as $\xi_{-1}$ and the corresponding objective value will not be incorporated in the solution. This is also our assumption for the value of $\epsilon$. 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 $\mu = 3, \beta$ takes the value 1. For Algorithm 2 we choose $\epsilon = 21$. Initially, this may seem a high value, but the value has to express the ratio between an optimal value regarding an arbitrarily large $\xi_{-1}$ and an optimal value with $\xi_{-1} = 0$. 
Table 2. Example for a lot-sizing problem with $T = \{0, 1, 2, 3\}$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$t = 0$</th>
<th>$t = 1$</th>
<th>$t = 2$</th>
<th>$t = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_t$</td>
<td>28</td>
<td>25</td>
<td>29</td>
<td>33</td>
</tr>
<tr>
<td>$q_t$</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>$c_t$</td>
<td>750</td>
<td>750</td>
<td>750</td>
<td>750</td>
</tr>
<tr>
<td>$f$</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>$d_t$</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>$C_t$</td>
<td>40</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>solution approach</th>
<th>$t = 0$</th>
<th>$t = 1$</th>
<th>$t = 2$</th>
<th>$t = 3$</th>
<th>objective value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(\mu = 1)$ Algorithm 1,2,3</td>
<td>(20, 0, 1, 0)</td>
<td>(20, 0, 1, 0)</td>
<td>(20, 0, 1, 0)</td>
<td>(20, 0, 1, 0)</td>
<td>5460</td>
</tr>
<tr>
<td>$(\mu = 2)$ Algorithm 1</td>
<td>(40, 20, 1, 0)</td>
<td>(20, 20, 1, 0)</td>
<td>(20, 20, 1, 0)</td>
<td>(0, 0, 0, 0)</td>
<td>4790</td>
</tr>
<tr>
<td>$(\mu = 2)$ Algorithm 2,3</td>
<td>(40, 20, 1, 0)</td>
<td>(0, 0, 0, 0)</td>
<td>(40, 20, 1, 0)</td>
<td>(0, 0, 0, 0)</td>
<td>4060</td>
</tr>
<tr>
<td>$(\mu = 3)$ Algorithm 1,2,3</td>
<td>(20, 0, 1, 0)</td>
<td>(40, 20, 1, 0)</td>
<td>(20, 20, 1, 0)</td>
<td>(0, 0, 0, 0)</td>
<td>4670</td>
</tr>
<tr>
<td>$(\mu = 4)$ Algorithm 1,2,3</td>
<td>(40, 20, 1, 0)</td>
<td>(40, 40, 0, 0)</td>
<td>(0, 20, 0, 0)</td>
<td>(0, 0, 0, 0)</td>
<td>4020</td>
</tr>
</tbody>
</table>

Table 3 shows the different solutions with all algorithms varying over $\mu$. Clearly, for $\mu = 1$ all algorithms deliver the same solution, because every period will be solved separately. The same holds for $\mu = 4$, where the whole integrated problem is solved from the beginning. With $\mu = 3$ all solutions yield the same production scheme, as no one prefers to produce during the last period, and the constellation for the first three periods can be chosen integrated. The interesting case occurs with $\mu = 2$, where the decision of the first step regarding period $t = 0, 1$ has to be re-adjusted at immense cost when the optimal solution of periods $t = 1, 2$ comes into play.

We will offer insights into how to choose $\epsilon$ for Algorithm 3 in the case of loose start states and start states that are fixed by the calculations during the execution of Algorithm 3. Therefore, we wish to bound the objective value of time periods already solved by its optimal value in terms of start states, which are not yet fixed. It is easy to observe, that for $\mu = 2$, we receive $z_{t, \mu - 1} = 2f + s_t d_t = 140 \forall t \in \{0, 1, 2\}$. If no storage is available at the beginning of the calculation of two integrated periods, an approximation of the worst case objective value can be derived by $\mu(f + Uc)$ again, which takes the value 3080 in this example. Thus, $\epsilon$ can be chosen as $\frac{3080}{140} - 1 = 21$, which is also $\epsilon_\mu$ in this example. In absolute values, we will constrain the costs that can arise in two consecutive periods by the upper bound:

$$
\sum_{t}^{t+1} f_t(x_t) \leq (1 + \epsilon)z_{t,1} = (1 + \epsilon_\mu)z_{t,1} = (1 + 21) \times 140 = 3080 \quad \forall t \in \{0, 1, 2\}.
$$

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 $z_{t, \mu - 1}^{\xi_t \cdot}$, with the start state $\xi_t$ obtained during the solution process. In our example we can choose $\epsilon_{\text{rolling}} = 0.5$:

$$
\sum_{t}^{t+1} f_t(x_t) \leq (1 + \epsilon_{\text{rolling}})z_{t,1}^{\xi_t \cdot} = (1 + 0.5) \times z_{t,1}^{\xi_t \cdot} \quad \forall t \in \{0, 1, 2\}.
$$
The value of this bound is calculated during the solution process and is not given in advance. The bound \( \epsilon_{\text{rolling}} \) can be interpreted as the maximal flexibility of the rolling horizon approach.

With the example data in the first step of the rolling horizon, the full demand will be produced in period 1 to avoid additional fixed costs in period 1. Therefore all costs of 1970 come from the production \( x_0 = 40 \) in period 0, storage \( s_0 = 20 \) from period 0 to 1 and 40 are fixed costs in period 1. Then Algorithm 3 calculates a bound \((1 + 0.5) \times 2010 = 3015\) for the first two periods, which cannot be exceeded by rising costs in period 1 in the next optimization step. 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 1350 in period 1 and cannot be realized due to \( 1970 + 1350 > 3015 \) in Algorithm 3. Therefore the production takes place in period 2 with costs of 1370 and in period 1 the costs remain 40. A new bound for period \( t = 1, 2 \) is now given by \((1 + 0.5) \times 1410 = 2115\). In the next step, it is preferable to expand the production in period 2 to \( x_2 = 40 \), which results in total costs of 2050 < 2115 in period \( t = 1, 2 \). This solution is feasible.

Although this is a minimal example that does not take backlogging or multi-level production into account, the effect of the improved rolling horizon approach can be clearly seen. 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.

5. 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).

5.1. 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 4.

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 legs. \( N_k \in \mathbb{N} \) denotes the maximal number of available aircraft per fleet. The parameter \( x_{s,ki}^b \in \{0, 1\} \) 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 \( \sum_{i=1}^{N_k} x_{s,ki}^b \). Every aircraft of fleet \( k \) is allowed to fly at most \( t_k^M \in \mathbb{R}^+ \) 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 \( \omega_{s,ki}^b \in \mathbb{R}_0^+ \) equals the time elapsed since the last maintenance of aircraft \( i \) (\( i \in \{1, \ldots, N_k\} \))
Set | Description
--- | ---
$S$ | Set of airports
$[T] = \{1, \ldots, T\}$ | (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

$A^M_k$ | $A^M \cap A_k$
$A^{NM}_k$ | $A \setminus A^M$ 
$A^{NM}_k$ | $A^{NM} \cap A_k$

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 $k$
$x^b_{ski} \in \{0, 1\}$ | 1, if and only if $i$ or more aircraft of type $k$ are at station $s$ at time 0, otherwise 0.
$\omega^b_{ski} \in [0, t_k]$ | 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 4. Input data of TAP

of aircraft type $k$ at station $s$ if $i$ or more aircraft of type $k$ are at station $s$, otherwise, $\omega^b_{ski}$ 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 $A^M_k$ and $A^{NM}_k$, while a pair of flight legs is in $A^M_k$ if and only if there is enough time to execute a maintenance event. We further expand the connection networks by a start node $v^b_{ski}$ 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 $v^e_{ski}$ 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.

5.2. Description of the integrated fleet and tail assignment model. The TAP can be modeled as MIP (14), following Haouari et al. (2012). The model variables are summarized in Table 5. For every flight leg $l \in L$, model (14) 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.

\begin{align}
(14a) & \quad \min \sum_{l \in L} c_l z_l + \sum_{k \in K} \left( \sum_{a \in A_k} c_{ak} x_{ak} + \sum_{l \in L_k} c_l x_{lk} \right) \\
(14b) & \quad \text{s.t.} \quad \sum_{k \in K_l} x_{lk} + z_l = 1 \quad \forall l \in L \\
(14c) & \quad \sum_{a \in A_k, l = a^-} x_{ak} = x_{lk} \quad \forall l \in L_k, k \in K \\
(14d) & \quad \sum_{a \in A_k, l = a^+} x_{ak} = x_{lk} \quad \forall l \in L_k, k \in K \\
(14e) & \quad \sum_{a \in A_k, a^- = v_{ski}^b} x_{ak} = x_{s_ki}^b \quad \forall s \in S, k \in K, i \in [N_k] \\
(14f) & \quad \sum_{a \in A_k, a^+ = v_{ski}^e} x_{ak} = x_{s_ki}^e \quad \forall s \in S, k \in K, i \in [N_k] \\
(14g) & \quad x_{s_ki}^e \geq x_{s_ki}^{e(i+1)} \quad \forall s \in S, k \in K, i \in [N_k - 1] \\
(14h) & \quad \omega_{s_ki}^b = \omega_{s_ki} \quad \forall s \in S, k \in K, i \in [N_k] \\
(14i) & \quad \omega_{s_ki}^e = \omega_{s_ki} \quad \forall s \in S, k \in K, i \in [N_k] \\
(14j) & \quad \omega_{s_ki}^{e(i+1)} \geq \omega_{s_ki}^e \quad \forall s \in S, k \in K, i \in [N_k - 1] \\
(14k) & \quad \sum_{a \in A_k, a^- = l} \omega_{ak} - \sum_{a \in A_k, a^+ = l} \omega_{ak} = t_{lk} x_{lk} \quad \forall l \in L_k, k \in K \\
(14l) & \quad t_{a^-} x_{ak} \leq \omega_{ak} \leq t_{a^+} x_{ak} \quad \forall a \in A_k, k \in K
\end{align}

The binary variables $x_{ak}$ take the value 1 if and only if an aircraft of fleet $k$ is assigned to connection $a$. The variables $\omega_{ak}$ 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 $\omega$-variables are thus prevented from exceeding the maximal time allowed between two maintenance checks, $t_k$. The binary variables $x_{s_ki}^e$ provide information on whether at least $i$ aircraft of fleet $k$ terminate their last assigned flight leg at station $s$: $\sum_{i \in \{1, \ldots, N_k\}} x_{s_ki}^e = \max\{i | i = 0 \lor x_{s_ki}^e = 1\}$. The continuous variables $\omega_{s_ki}^e$ 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 (14a) 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 (14b) ensures that every flight leg is assigned to exactly one fleet or is otherwise canceled. Constraints (14c) and (14d) 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 (14e) 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 (14f) reflects the fact that every aircraft of type $k$ terminating at station $s$ increases the value of $x^e_{ski}$ by 1. Constraint (14g) 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 (14h) ensures that the $i$-th aircraft of fleet $k$ at station $s$ starts with a time-elapsed-since-maintenance of exactly $\omega^b_{ski}$, while constraint (14i) 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 $\omega^e_{ski}$ for an $i \in [N_k]$. Constraint (14j) is not necessary, but, like (14g) breaks the symmetry of equivalent solutions. Finally, constraint (14k) 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 (14l) 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.

### 5.3. 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 (14) 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 (14) can be established, interpreting the $x^b, \omega^b$ 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 $\Xi_t$ is the (reasonable) domain for $x^b, \omega^b$ variables of $P_t$. Analogously the set $\Theta_t$ is the (reasonable) domain for $x^e, \omega^e$ 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 Lemma 4 is applicable to a reasonably general class of TAPs. This would imply

<table>
<thead>
<tr>
<th>Variable</th>
<th>Domain</th>
<th>Indices</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z_l$</td>
<td>${0,1}$</td>
<td>$l \in L$</td>
<td>1 if and only if flight leg $l$ is canceled</td>
</tr>
<tr>
<td>$x_{sk}$</td>
<td>${0,1}$</td>
<td>$s \in S, k \in K$</td>
<td>1 if and only if leg $s$ is assigned to fleet $k$</td>
</tr>
<tr>
<td>$x_{ek}$</td>
<td>${0,1}$</td>
<td>$e \in E, k \in K$</td>
<td>1 if and only if flight leg $e$ is assigned to fleet $k$</td>
</tr>
<tr>
<td>$\omega_{ak}$</td>
<td>$[t_{a-1}, t_k]$</td>
<td>$a \in A, k \in K$</td>
<td>Time-elapsed-since-maintenance transmission of fleet $k$ on connection $a$</td>
</tr>
<tr>
<td>$x^e_{ski}$</td>
<td>${0,1}$</td>
<td>$s \in S, k \in K, i \in {1, \ldots, N_k}$</td>
<td>1 if and only if $i$ or more aircraft of type $k$ terminating at station $s$</td>
</tr>
<tr>
<td>$\omega^e_{ski}$</td>
<td>$[0, t_k - t_s]$</td>
<td>$s \in S, k \in K, i \in {1, \ldots, N_k}$</td>
<td>Time-elapsed-since-maintenance of aircraft $i$ of fleet $k$ terminating at station $s$</td>
</tr>
</tbody>
</table>

**Table 5. Model variables of TAP**
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 5 with conditions very likely to be fulfilled in practice. This lemma presents conditions that are sufficient, but not necessary, for the conditions of Lemma 4 to be fulfilled.

Lemma 5. Let \((P_t)_{t \in \mathbb{N}}\) be a sequence of TAPs defined with data structured as in Table 4 and modeled as in problem (14). Let \(\Xi_t\) and \(\Theta_t\) be the set of feasible start states and end states of the single-period TAP \(P_t\), and \(z_{\cdot \cdot}^{\cdot \cdot}\) be the optimal value function of the \(\mu\)-period TAP beginning at period \(t\), \(P_{t, \mu}\) as defined in Section 3, 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 \(\Delta\) 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 \(P_{0,n}^{\cdot \cdot}\) for every \(\vartheta \in \Theta_n\), for all \(n \in \mathbb{N}\)

(C5) start states \((x^b, \omega^b)\) 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 \((x^e, \omega^e)\) 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 Lemma 4, equations

\[
\alpha := \sup_{t \in \mathbb{N}, (x^b, \omega^b) \in \Xi_t, (x^e, \omega^e)_{t+m} \in \Theta_{t+m}} z_{\cdot \cdot}^{(x^b, \omega^b), (x^e, \omega^e)_{t+m}, \cdot \cdot} < \infty,
\]

and

\[
\lim_{\mu \to \infty} \inf_{t \in \mathbb{N}} z_{t, \mu - 1} = \infty,
\]

hold for an \(\alpha \leq C \sum_{k \in K} \left( |S| + t_k \right) N_k\) and \(m = \sum_{k \in K} \left( |S| + t_k \right) N_k\). These conditions imply that a sequence \((\varepsilon_\mu)_{\mu \in \mathbb{N}}\) converging to 0 with

\[
\sup_{t \in \mathbb{N}, (x^b, \omega^b) \in \Xi_t} \frac{z_{t, \mu - 1}}{z_{t, \mu - 1}} \leq (1 + \varepsilon_\mu)
\]

exists, such that Algorithm 2 or Algorithm 3 can find arbitrarily good solutions. Lemma 4 is therefore applicable to the TAP sequence.
Lemma 5 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 5.

**Proof 4.** According to Corollary 3, (C8) implies condition (11). Condition (10) 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 \(\lceil \frac{\Theta}{5} |S| + t_k \rceil\) time periods per aircraft and can be successively applied to every aircraft in the schedule, leading to an
upper bound for $\alpha$ in equation (10) according to (C8):

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

valid for $m = \sum_{k \in K} \left\lceil \frac{\Delta}{\delta} |S| + t_k \right\rceil N_k$. 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 $\alpha$ 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 $\alpha$. Computational results concerning real-world TAPs will be presented in Section 6.

5.4. Choice of $\mu$ in TAPs. We will discuss an example of a very small connection network in the context of tail assignment to provide a sense of the mechanism of choosing the right forward horizon. Figure 5 shows a connection network with seven flights, which is split into four time periods. The network starts at Airport 1, where two aircraft from different fleets can be used to execute all seven flights. In the case that no flight cancellations are allowed, it is immediately possible to see that flight connections $(A,F), (A,D)$ cannot be selected as otherwise flight $B$ must be canceled. The same holds with the connections $(B,G), (C,F)$ because of flight $E$ and $D$ respectively. The unselectable connections are drawn grey in Figure 5. Therefore only two flight sequences exist and the whole assignment depends on just one decision: Which aircraft will be assigned to flight $A$? Assuming a planning horizon of exactly one period, we wish to analyze the solutions in terms of changes in the lengths of the forward horizon. The special structure of the problem allows no more changes after fixing planning period 1.

In Table 6 it can be observed that the optimal assignment in planning period 1 varies according to $\mu$. Thus, $\mu = 1, 2, \mu = 4$ will lead to the overall optimal assignment, while $\mu = 3$ does not. Note that $\mu = 4$
Assignments of Resulting Objective Optimal

<table>
<thead>
<tr>
<th>Step Length ( \mu )</th>
<th>Assignments of (ABEF, CDG)</th>
<th>Value of first ( \mu ) periods</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>((f_1, f_2), (f_2, f_1))</td>
<td>10, 15</td>
<td>yes, no</td>
</tr>
<tr>
<td>2</td>
<td>((f_1, f_2), (f_2, f_1))</td>
<td>35, 40</td>
<td>yes, no</td>
</tr>
<tr>
<td>3</td>
<td>((f_1, f_2), (f_2, f_1))</td>
<td>65, 55</td>
<td>no, yes</td>
</tr>
<tr>
<td>4</td>
<td>((f_1, f_2), (f_2, f_1))</td>
<td>80, 85</td>
<td>yes, no</td>
</tr>
</tbody>
</table>

Table 6. Table of solutions of problem \( P_{1,\mu} \)

is equivalent to solving the fully integrated model. This example proves that longer forward horizons will not automatically deliver better objective values, especially when it comes to TAPs.

6. Computational Experiments

This section describes the computational experiments we performed as part of this study. The aim of the study is to determine whether large-scale tail-assignment instances can be solved with our rolling horizon approach. We therefore conduct a mean value study with 50 large instances and additionally look at four more precisely specified very large instances consisting of 1500 to 2300 flights each. 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.

6.1. Generating instances. An integrated fleet and tail assignment is performed here. This means that we assign aircraft of different fleets to flight legs simultaneously and without predetermining which type of aircraft serves which flight leg. Therefore, we have to deal with a much larger and more general problem class than classical TAP. In addition, we include maintenance planning as a further challenging optimization issue to cover as many strategic aspects as possible.

The underlying flight plan was recorded as a screenshot from a particular week of the airline Lufthansa. Flight numbers, start and destination, and planned flight times were detailed. The focus was on the inner-European area, which is served from the hub airports in Frankfurt (FRA) and Munich (MUC). The resulting weekly timetable comprised 1666 flight numbers with 10082 direct flights over 200 airports. The aircraft data included 13 different aircraft types and a total of 335 aircraft. In addition the entire fleet was reconstructed based on the data published online by Lufthansa.

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. Fixed kilometer operating cost rates are charged for different fleets. The maintenance parameters result from the duration of the maintenance event, the maximum flight time between two maintenance operations and the maximum number of take-offs between two maintenance operations.

To ensure the schedule can be adjusted to the chosen size per instance, the full schedule is filtered randomly for an almost complete network component, which incorporates the requested number of flights and airports. In this case, an almost complete network is determined by searching for an airport
combination whose complete number of all incoming and outgoing flights matches the desired number of
flights. Both hub airports FRA and MUC are integrated in every instance, since we allow maintenance
checks only at these airports. This filtering procedure is allowed to accept relatively small divergences
regarding the number of flights. This makes it possible to generate differentiated instances because
airport combinations that have already been selected can be excluded for subsequent instances.

After the schedule is set, 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.

6.2. 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. Care
was taken to ensure that no instance contained the same airport constellation. In addition, there are
further unique instance attributes due to the randomized input parameters. 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

<table>
<thead>
<tr>
<th>Instance name</th>
<th>#flights</th>
<th>#airports</th>
<th>#fleets</th>
<th>#aircraft</th>
</tr>
</thead>
<tbody>
<tr>
<td>LargeScale1</td>
<td>1516</td>
<td>16</td>
<td>6</td>
<td>53</td>
</tr>
<tr>
<td>LargeScale2</td>
<td>1690</td>
<td>18</td>
<td>7</td>
<td>64</td>
</tr>
<tr>
<td>LargeScale3</td>
<td>2110</td>
<td>22</td>
<td>7</td>
<td>74</td>
</tr>
<tr>
<td>LargeScale4</td>
<td>2310</td>
<td>25</td>
<td>7</td>
<td>80</td>
</tr>
</tbody>
</table>

Table 7. Table of large instances for computations

the problem is as follows: The flight plan is divided into individual 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. To realize this approach and remain computationally tractable, integrated planning
horizons combined with LP-relaxed forward horizons were solved.

6.3. 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 computer the rolling-horizon approach for integrated fleet and TAPs. The focus is on the following elements:

(1) comparison of runtime between solving the entire integrated problem and solving the problem with our rolling horizon approach.
(2) comparison of different parameter settings depending on instance size and forward-horizon extent.
(3) verification of the benefit of our rolling horizon approach when applied to large-scale, practically relevant instances.

In the first computation run, 50 instances with about 800 flights and 8 airports each were used and tested with different procedures. The results depicted in Figure 6 indicate that our approach is appropriate to find near-optimal solutions for large TAPs. Beyond that, it should be mentioned that all instances were also calculated fully integrated with less success. 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. In order to make a qualitative evaluation in the form of an optimality gap, the solution and the mean running time of each LP relaxation of the entire integrated problem was determined. The solution value found served as the lower limit of the objective function value and was used to compute the gaps shown in the Figure 6.

![Tail Assignment: Results of the Rolling Horizon](image)

**Figure 6.** Computations: Results Mean
In further computational experiments, additional interesting results were identified. In some instances, depending on the longer forward horizon, we were able to observe our theoretical results in practice in terms of the objective function value. One tends to assume that a longer forward horizon will deliver better overall results, but for structural reasons this is not necessarily the case, as noted in Section 3 and illustrated in Subsection 5.4. The computation showed that for the combination of 15 flights in the planning horizon and a relatively long forward horizon, the solution quality of the rolling horizon procedure drops with the horizon length, which suggests this effect occurs in practice. These observations were made when single instances with LP-relaxed forward horizon were analyzed. To reproduce this behavior, the forward-horizon parts of the problems were modeled integral, which strengthened the effect, as shown in Figure 7 for a selected instance. It can be seen that the objective function value does not always decrease with increasing forward horizon, but for example rises considerably in the transition horizon length of 90 to 105 flights. If we look at the future periods as integer problems, the difference becomes even more evident: The rolling-horizon approach with a horizon length of 120 flights performs significantly worse than all its predecessors up to 45 flights.

This effect occurs because the additional information causes the current solution to deviate too much from already assumed solution structures. It can be compensated theoretically by the constraints introduced for Algorithm 3 in Section 3. In practice, however, experiments showed that it is not effective to apply them to the TAP. It appears they are far too restrictive and lead to considerably worse solutions. As a practical alternative to avoid the aforementioned effect, we tried to incorporate discount functions to increase the weight of the current time period’s objective function in the multi-period problem. Two strictly falling weight distribution functions $d$ were used to find out whether they prove themselves in practice:

- $d(x) = 1 - \frac{x}{\mu}$, where $x$ denotes one specific forward-horizon period. This reflects a linear decline in the valuation of forward periods.
The same computation of Instance 48 with integer forward-horizon periods and different discount factors on the forward-horizon parts can be seen in Figure 8. As can easily be observed, the curve of the objective function values can be smoothed and the outlier for 120 forward-horizon flights is omitted. It is difficult to find out which method performs better in general settings, so this has to be determined individually for different applications and problem structures.

Since the test results of the mean study seemed promising, further experiments were carried out with even larger instances based on real data. The results can be seen in Figure 9. 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 to be calculated, the parameters of our

---

**Figure 9.** Computations: Solutions and Runtimes of real-world instances with planning and forward horizon measured in number of flights

- \( d(x) = 3 - 2x^2 \), where \( x \) denotes one specific forward period. This reflects a curved decline in the valuation of forward periods, which allocates more value to periods that are close to the planning horizon, while the decline in value becomes very steep in later periods.
rolling horizon approach were adjusted. Not all parameters led to a result within the set time of 23 hours. In this example, each calculation type failed exactly once, which proves that the problem structure can significantly influence the calculation time and the efficiency of the rolling horizon. The number of flights to be considered per period was 360 for all computations, whereby the number of flights to be fixed was varied. In three out of four approaches, the best approach resulted in a gap of less than 3%. The largest instance with an approximative gap over 9% falls far out of the scheme. In summary, the results show the effectiveness of the approach. We received solutions for realistic large-scale instances, which are integral. Hence, they can be implemented in practice, unlike LP-relaxed solutions, which can lead to ambiguous assignments. These solutions also include 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 number of flights, opens up the possibility of even solving entire six-monthly flight schedules in the future. The calculated applicable solutions do not depend on cyclic structural components and feasibility of upstream operational decisions. These solutions come with a provable guarantee. Even the seemingly large gap of 9.27% can be relativized easily: It can be explained by the fact that the LP relaxation, which serves as a lower bound, strongly underestimates the real optimal value. This in turn is possibly caused by an unsuitable flight structure, an insufficient number of flights being considered per period or a fleet structure that is too differentiated. Taking into account these conservative bounds, a single digit gap has been achieved compared to the optimal objective function value of the LP relaxation of the corresponding integrated problem.

7. 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. A topic for future research is the investigation of the behaviour of this approach when dealing with disturbed situations for which methodologies from optimization under uncertainty need to be applied.

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


