Abstract

Prior to the day-of-operations, airlines solve the Tail Assignment Problem (TA), in which specific aircraft (*tails*) are assigned to multi-day sequences of flights (*aircraft rotations*) and an associated *maintenance plan* is built, in which specific maintenance activities for each tail are assigned to specific airports on specific days. On the day-of-operations, however, system disruptions may require changes to be made to this tail assignment which leads to disruptions of the associated maintenance plan. We consider the maintenance recovery problem (MRP), which seeks to recover a disrupted maintenance plan. In our approach, we consider two variations of the problem. In the first, we only consider the most imminent ("next") required maintenance activity (*check*) for each aircraft, and try to push these checks as far into the future as possible (i.e. minimize the number of such checks that must fall in a short-term time horizon). This lays a foundation for the more realistic, and more complex, second variation in which we develop a solution methodology for a specified recovery horizon throughout which we seek to minimize the total number of maintenance checks required to attain feasibility. This variation is significantly more challenging, in part because the deadlines of checks are not pre-determined, but rather depend on the scheduling of earlier checks. We present two different formulations (based on *pure-* and *augmented-rotations*) to address this second variation, discuss the challenges associated with solving these different formulations, and develop solution techniques for each to achieve tractability. We conclude by providing computational results to demonstrate the effectiveness of our approach.

Methods for Improving Aircraft Maintenance Recovery

July 14, 2015

1 Introduction

1.1 Overview

Regular aircraft maintenance is paramount for safe operation. It is not just recommended by the aircraft manufacturer, but mandated by the government. In the United States, the Federal Aviation Administration (FAA) provides strict guidelines for all *maintenance checks*. Airlines must operate within these guidelines and may not place an aircraft into passenger service if a maintenance check has not been performed by its deadline.

Because aircraft are an airline's most expensive set of assets, their maintenance is an integral part of airline operations — and an expensive one. Aircraft maintenance accounts for a large portion of the operating expenses of any airline. For example, in 2010, Delta Airlines spent a total of $1.6 billion USD, about 5% of its operating expenses, to maintain its fleet of approximately 750 aircraft [Delta, 2010], while US Airways spent approximately 6.5% of its operating expenses in 2010 on aircraft maintenance. [USAirways, 2010].

During the planning phase, an airline decides a set of flights for a given day. Subsequently, the airline determines all flights which will be flown by the same aircraft (*tail*) for a particular day (*a line-of-flight or LOF*). Several lines-of-flight together form *rotations*, which are then assigned to a particular aircraft.

An individual aircraft's maintenance requirements are a function of three factors: flight hours, cycles and calendar days. Flight hours refer to the difference between off-block and on-block time, while a cycle refers to a take-off and subsequent landing (one cycle) and calendar days refer to a day-count. *Maintenance counters* define the accounting standards in aircraft maintenance. Such counters generally include: 1) *flight-hours*, 2) *cycles* and 3) *calendar days*. Here, flight-hours refer to the *off-block* to *on-block* time, i.e. the time the aircraft departs the origin gate until it arrives at the destination gate. Aircraft cycles count the number of take-offs and landings performed by an aircraft over the course of a specific time horizon.

The frequency of required maintenance based on these factors is, in turn, determined by the rotation assigned to the aircraft. Thus, when an airline designs a rotation, a *maintenance opportunity* (a block of time at a station where a maintenance check can be performed) must exist when one of these factors approaches a specific limit. For example, a maintenance check for a Boeing 737-300 might be required after every 40 flight hours. Thus, when an airline designs a rotation for an aircraft (during the planning phase), maintenance opportunities must exist every 40 hours (in this example) to ensure the plan's overall feasibility.

If an airline decides to change the routing for a particular aircraft, the maintenance requirements for that aircraft may change and the maintenance plan must be adapted. More specifically, a change in the rotation may require a particular aircraft to incur additional (unplanned) maintenance counters. In addition to changes in the maintenance counters, a rotation change may also remove maintenance opportunities, i.e. an aircraft may no longer overnight at a maintenance station as scheduled. In either case, maintenance recovery must be performed to ensure all aircraft do not exceed their maintenance limits and allow adequate opportunity for completion of maintenance checks.

In addition to the fact that maintenance deadlines are constantly changing as a function of the activities that are actually completed by each tail during operations, aircraft maintenance is further complicated by the resource restrictions that exist within a network. First, aircraft maintenance is limited to certain airports (*stations*) within an airline's network. Depending on the airline, such maintenance stations may exist only at the airline's major hubs, or alternatively, the airline has a designated maintenance hub. In addition to maintenance being limited to specific locations, maintenance stations feature a finite amount of capacity governed by the available man-hours that may be used to perform a maintenance check.

As we will demonstrate, the frequent changes in the deadline of maintenance checks due to changes in the planned rotations, along with limited availability of maintenance opportunities and resources, make maintenance planning and recovery difficult problems to solve. In the remainder of this paper, we uncover the difficulties when solving the maintenance recovery problem, addressing issues such as finding an initial solution for a short time horizon quickly and subsequently increasing the time horizon to provide a comprehensive maintenance plan.

1.2 Problem Description

Currently, most major airlines recover from maintenance disruptions manually. That is, a team of route planners is responsible for ensuring that aircraft reach maintenance stations before the deadline of a specific maintenance check. Given this manual process of finding appropriate lines-of-flight for aircraft, opportunity exists for both automation and optimization.

In our approach, we determine a recovered maintenance assignment subject to the most recent schedule changes that have been realized. More specifically, when an airline does an over-the-day tail swap, cancels flights, or in other ways performs operational recovery steps, the aircraft are disrupted from their current rotation. This in turn also leads to disruptions of the associated maintenance plan. We consider the *Maintenance Recovery Problem (MRP)* in which, at the end of the day, we build new rotations for each flight, and an associated maintenance plan that is feasible relative to these rotations. Note that in building new rotations, we only change the sequencing of the underlying lines-of-flights. The LOFs themselves are not changed, due to the importance of the pre-assigned aircraft turns in other aspects of the system's operations.

In our approach, we consider two variations of the problem. In the first, we only consider the next check for each aircraft, and try to push these checks as far into the future as possible. In the second, we consider the more realistic problem in which we minimize the *total* number of maintenance checks needed within a given time horizon to attain feasibility in a recovery horizon of multiple days. As we will demonstrate, this problem is considerably more difficult due to the interaction that exists between maintenance checks, i.e. the assignment of one check in turn determines the deadline for subsequent checks. We present two different formulations (based on *pure-* and *augmented-rotations*) to address the problem of determining the fewest maintenance checks needed to achieve feasibility within a recovery plan, discuss the challenges associated with solving these different formulations, and develop solution techniques for each to achieve tractability. We conclude by providing computational results to demonstrate the effectiveness of our approach.

1.3 Research Contributions

The contributions of this research are two-fold. First, we provide a new approach to solving the maintenance recovery problem in which we address the most immediate maintenance concerns, while delaying the non-critical maintenance checks to the future. Second, we develop an optimization model to schedule not only the most immediate, but all future maintenance checks within a give time horizon. As we will demonstrate, this problem is considerably more difficult due to the interaction that exists between maintenance checks, i.e. one check determines the deadline for subsequent checks.

The remainder of the paper is organized as follows. In §2, we provide an overview of the airline planning process and current maintenance operations. We also provide the context for the maintenance recovery problem. In §3 we introduce related literature. Next, in §4 we solve a simplified version of the MRP in which we only consider the immediate (next) maintenance requirements for each aircraft. In §5, we solve the recurring maintenance recovery problem which not only provides immediate recovery, but also offers insights into workload capacity planning for future maintenance checks. Finally, we offer conclusions and future research ideas in §6.

2 Airline Planning and Maintenance Operations

2.1 Airline Planning Process

To begin the planning process, an airline sets its schedule of daily-repeating flights for a fixed time period, for example, a quarterly schedule. Given the set of flights, the *fleet assignment problem* is then solved, assigning each flight a specific aircraft type. Once fleeting is completed, the schedule can be decomposed by fleet type and, for each fleet type, the *crew scheduling problem* and maintenance routing problems are solved.

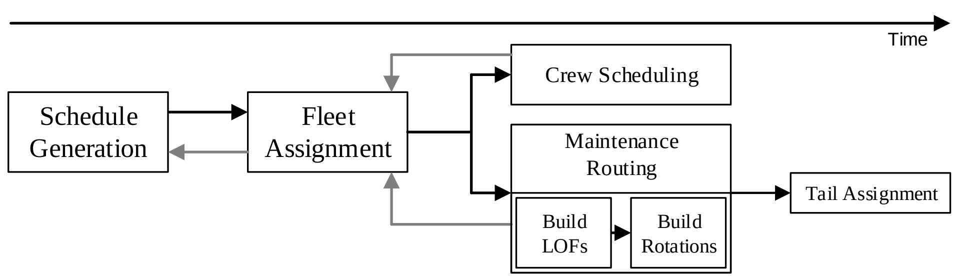


Figure 1: Typical Airline Planning Process

Within maintenance routing, the lines-of-flight are first constructed. These lines-of-flight are then used to construct maintenance-feasible aircraft rotations. Initially, the rotations are not assigned to specific aircraft (and, in fact, may be modified on a daily basis once the operation of the schedule begins), but they provide a mechanism for ensuring that a maintenance-feasible set of rotations exists before the schedule is set. This process is illustrated in Figure (1).

2.2 Maintenance Planning

To perform maintenance planning, it is important to realize the connection between a day-long sequence of flights that will be flown by a single, common aircraft, i.e. LOF, and its implication on aircraft maintenance. A line-of-flight contains a set of flights, each with a value for a respective maintenance counter that when summed defines the number of flight-hours and cycles for the entire LOF.

Next, each type of aircraft is required to periodically undergo a set of *maintenance checks*, where the frequency depends on the values of different maintenance counters. When solving the Tail Assignment Problem, feasibility is established by guidelines set forth by the Federal Aviation Administration (FAA) in conjunction with aircraft manufacturers in the United States. For example, an aircraft might be required to receive a small-size (A) check every 40 flying-hours to continue legal operation. Although our approach is not limited to specific checks, we consider those checks common to most commercial airlines, including A, B and C checks, each with different limits and requirements, in our computational experiments.

A maintenance check is further defined by a man-hour requirement at a maintenance-capable station in the flight network. This requirement comes in two forms. First, for a check to be completed at a maintenance station, the aircraft must be on the ground for a sufficiently-long time window, e.g. 8 hours (typically overnight). Furthermore, each check features a specific number of man-hours that must be allocated (from the total available capacity at a maintenance station) for this particular check. It should be noted that each station provides only a finite number of man-hours of available maintenance capacity on any given day. Both of these limits (time-window and available capacity) must be observed for a maintenance check to take place.

It should be noted that we define a maintenance check relative to the state of the system on the day-of-operations, i.e. *time-zero*. In other words, given the current set of counters of a specific aircraft and its planned future rotation, we can identify the *next* required maintenance check of each type that must be completed by each aircraft. For example, suppose we have an aircraft that at the beginning of the day holds 32 flight-hours on its clock. Furthermore, we have two possible choices for line-of-flight assignment for the day, as seen in Figure (2). In this case, we could assign line-of-flight #2 with 7.25 hours, but not line-of-flight #1 with 9 hours as this would exceed the 40 hour limit for an A-check.

We note that in the first problem that we address, our focus is to schedule the first of each (recurring) type of check — that is, we only schedule one check of each type for each aircraft as necessary to ensure a maintenance-feasible plan. In the second problem, we expand this into all maintenance checks required for specific future time horizon.

Finally, aircraft can only receive maintenance at specific stations in the flight network. These maintenance stations provide man-power capacity for a subset of maintenance checks that may be

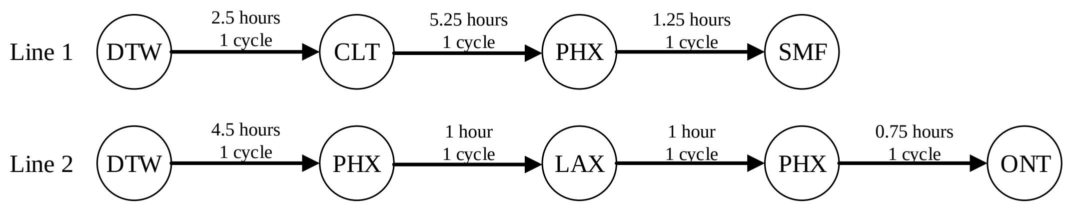


Figure 2: Various lines-of-flight with different maintenance counters

performed at the respective station. When an aircraft visits one of these stations during an overnight stop, we refer to this as a *maintenance opportunity*. That is, if a tail is scheduled to spend the night at a maintenance station (maintenance is generally performed overnight when aircraft utilization is low) which is equipped to perform maintenance for the appropriate fleet type and of the appropriate check, then an opportunity exists.

2.3 Maintenance Recovery Process

Upon reaching the day-of-operations, the airline planning process has ended and the implementation stage has begun. However, upon execution, this particular plan may change due to unforeseen circumstances. For example, a station manager may request to swap two aircraft to mitigate a delay situation. When such a swap occurs, aircraft will generally trade the entire remainder of their respective lines-of-flight, which alters the remaining rotation and thus may make the current maintenance plan infeasible, either because the maintenance counters will change or because the tail will no longer reach an intended maintenance station at the right time.

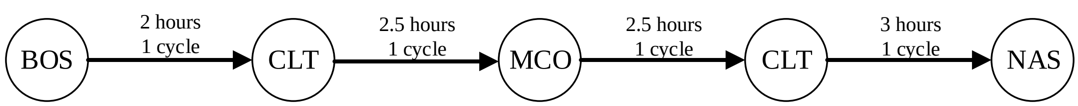
For example, in Figure (3), we have the current status of two tails in our network that are candidates for a swap. Suppose that due to a disruption, one of these tails has to perform the rotation shown in Figure (4). In such a case we note the dramatically different values for the final state of each of the tails at the conclusions of these rotations. More specifically, we note that Tail #2 would now quickly approaching its cycle limit if we assigned it to the new LOF that contains four cycles over the course of a day.

In a disruption scenario, an airline will modify aircraft rotations. With the set of tails with their current maintenance state and a set of LOFs that need to be covered in the future, we can formulate and solve an optimization problem in which we first attain a feasible a solution that minimizes the number of maintenance requirements. That is, we begin with information about the tails' current

Tail 1		Current	Limit
A-check	Hours (H):	4	40
	Cycles (C):	2	10

Tail 2		Current	Limit
A-check	Hours (H):	15	40
	Cycles (C):	4	10

Figure 3: Current state of two tails in a network



Tail 1		Current	Limit
A-check	Hours (H):	10	40
	Cycles (C):	4	10

Tail 2		Current	Limit
A-check	Hours (H):	25	40
	Cycles (C):	9	10

Figure 4: Example final state of each aircraft with assignment of line-of-flight

state at the end of the day (location and counts towards checks). In the first version of the problem, we build new rotations with accompanying maintenance plans to cover the most imminent check of each type. We then extend the problem definition to consider a pre-defined time horizon. In both versions of the problem, we enforce capacity limits that exist at each maintenance facility.

2.4 Sample Recovery Allocation

To further illustrate the maintenance recovery process, we consider a small instance of this problem. Consider a tail, Tail #1, on the beginning of the day-of-operation. The tail's current maintenance counter status is shown in Figure (5). For simplicity, we only consider A and B-checks in this example. Figure (6) presents two possible rotations for this tail with respective maintenance opportunities. Based on the selected rotation and its current status, the tail must complete an A-check in the near future. Upon execution, the maintenance recovery problem can choose the first rotation and thus perform the A-check at the termination of line-of-flight 3 at the end of day 3 as seen in Maintenance Assignment 1. This would reset the counters for the A-check at the beginning of day 4 as shown in the diagram.

Alternatively, the A-check could also take place at the termination of line-of-flight 2 with the

Tail #1		Current	Limit
A-check	Hours (H):	0	40
	Cycles (C):	0	10
B-check	Hours (H):	20	100
	Cycles (C):	10	50

Figure 5: Tail #1: Beginning maintenance counter status

assignment of the second rotation. This example illustrates the various choices that exist when scheduling a maintenance check for an aircraft through the assignment of various rotations. Furthermore, this example demonstrates the types of choices the optimizer will have when scheduling rotations with interspersed maintenance checks.

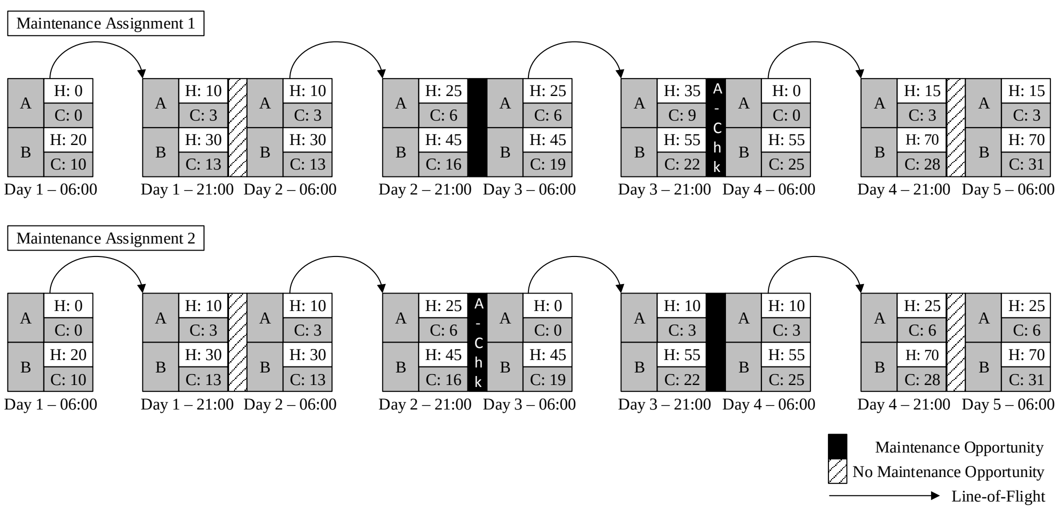


Figure 6: MRP rotation assignment decision example

3 Literature Review

Airline maintenance planning has been studied extensively. The survey paper by Gopalan and Talluri [1998] provides a general overview of the aircraft (tail) assignment problem. One of the earlier formulations of the tail assignment problem is illustrated in Clarke et al. [1997]. The authors in this paper consider the assignment of aircraft to flights without maintenance planning implications for man-power capacity planning. Additional work by Cohn and Barnhart [2003] integrates the flight-crew planning process with aircraft maintenance routing to derive synergies in the planning process.

In Sriram and Haghani [2003] the authors investigate maintenance planning by assigning aircraft to flight legs so as to minimize the cost of the maintenance that is performed. The proposed solution approach uses a heuristic that is computationally fast. On the other hand, in Yan et al. [2004] the authors address the problem of manpower supply planning. Aircraft are scheduled to receive their respective A- and B-checks given the manpower that is available at certain maintenance stations in the network. The authors solve their problem using a mixed-integer programming approach.

Applying a Lagrangian relaxation solution approach to the tail assignment problem is explored in Daskin and Panayotopoulos [1989]. Here, the authors use a Lagrangian relaxation approach to determine the assignment of aircraft to routes with the objective of maximizing profit. The authors note that the Lagrangian approach is effective at providing a bound on the optimal solution and when paired with heuristic approaches provides a good solution to the assignment problem.

Airline recovery models have received more recent attention. Work from Sarac et al. [2006] focuses on solving a one-day operational maintenance routing problem. Authors employ a branch-and-price approach to solve an optimization problem that minimizes total daily maintenance costs, while ensuring all aircraft remain within legal flying parameters. In Abdelghany et al. [2008], the authors pose a holistic airline schedule recovery framework by which various operational constraints are taken into consideration. Here, the authors pre-process a set of recovery networks that can be implemented as over-the-day disruptions take place. The authors note that the effectiveness of this approach is governed by the flights/aircraft ratio, an indicator of the possible number of assignments and thus a surrogate for the size of the optimization problem. In Eggenberg et al. [2010], the authors focus on a total recovery solution including passengers and crew rotations.

In our approach we focus strictly on the tail assignment and its maintenance implications, while placing emphasis on solution speed and the level of maintenance coverage. That is, we only consider aircraft maintenance implications and thus can build a recovery plan quickly. In our approach, we operate at the lines-of-flight level and are able to solve longer horizon models because crew and through-connections are not changed. To achieve this, we implement decompositions approaches including column-generation.

4 First Maintenance Check Problem

In this section, we focus on maintenance recovery from the perspective of the *first* maintenance check. That is, at most one check is scheduled for each type for each tail over the recovery horizon. In the *First Maintenance Check Problem* (FMCP), the objective is to assign maintenance-feasible rotations to each aircraft, minimizing the number of maintenance checks performed over the planning horizon while still ensuring that no aircraft violates its most imminent check of each type and ensuring that each LOF is covered and maintenance station capacity is observed. We emphasize that in our solution approach we only modify the sequences of LOFs that form each rotation, but not the LOFs themselves. Changing LOFs would force the airline to incur additional costs by redefining crew rotations, as well as other strategic decisions related to LOF construction.

Our formulation for this problem is based on two sets of decision variables. First, we determine whether or not to assign specific, pre-defined rotations (i.e. sequences of LOFs) to specific tails. Second, we determine where within their assigned rotations each tail receives its first check of each type. The notation for this model is as follows.

Sets

T	The set of tails.
D	The set of days in the planning horizon.
M	The set of maintenance stations in the network.
R	The set of all possible rotations.
$R(t) \subseteq R$	The set of all rotations that can be assigned to tail t, $\forall t \in T$.
L	The set of lines-of-flight.
C	The set of check types.

Parameters

θ_c	The amount of maintenance man-hour capacity required for maintenance check c, $\forall c \in C$.
ν_{rdm}	A binary parameter that is 1 if rotation r overnights on day d of the planning horizon at maintenance station m, $\forall r \in R, \forall d \in D, \forall m \in M$.
δ_{lr}	A binary parameter that indicates if line-of-flight l is contained in rotation r, $\forall r \in R, \forall l \in L$.
ρ_{md}	The available capacity at station m on day d in the planning horizon, $\forall m \in M, \forall d \in D$.
r_{dt}^{cr}	A binary parameter that is 1 if rotation r requires the first check of type c to be completed by day d for tail t, $\forall d \in D, \forall t \in T, \forall c \in C, \forall r \in R(t)$. The choice of rotation implies the deadline for the first maintenance check.

Variables

x_{tr}	a binary variable that is 1 if rotation r is assigned to tail t, $\forall t \in T, \forall r \in R(t)$.
w_{tcmd}	a binary variable that is 1 if tail t receives the *first* check of type c on day d of the planning horizon at station m, $\forall t \in T, \forall d \in D, \forall c \in C, \forall m \in M$.

Objective:

$$\min \sum_{t \in T} \sum_{c \in C} \sum_{m \in M} \sum_{d \in D} w_{tcmd} \tag{1}$$

Subject to:

$$\sum_{t \in T} \sum_{r \in R(t)} \delta_{lr} x_{tr} = 1 \qquad \forall l \in L \quad (\gamma_l) \tag{2}$$

$$\sum_{r \in R(t)} x_{tr} = 1 \qquad \forall t \in T \quad (\theta_t) \tag{3}$$

$$\sum_{m \in M} \sum_{\substack{d_1 \in D: \\ d_1 \leq d}} w_{tcmd_1} - \sum_{r \in R(t)} r^{cr}_{dt} x_{tr} \geq 0 \qquad \forall t \in T, \forall d \in D, \forall m \in M, \forall c \in C \quad (\alpha_{tdmc}) \tag{4}$$

$$w_{tcmd} - \sum_{r \in R(t)} \nu_{rdm} x_{tr} \leq 0 \qquad \forall t \in T, \forall m \in M, \forall d \in D, \forall c \in C \quad (\beta_{tmdc}) \tag{5}$$

$$\sum_{c \in C} \sum_{t \in T} \theta_c w_{tcmd} - \rho_{md} \leq 0 \qquad \forall m \in M, \forall d \in D \quad (\kappa_{md}) \tag{6}$$

$$x_{tr} \in \{0, 1\} \qquad \forall t \in T, \forall r \in R(t) \tag{7}$$

$$w_{tcmd} \in \{0, 1\} \qquad \forall t \in T, \forall c \in C, \forall m \in M, \forall d \in D \tag{8}$$

In this approach, the objective function in equation (1) minimizes the total number of maintenance checks to be performed during the planning horizon by deciding which rotation to assign to each tail in fleet. The input set of candidate rotations are constructed by definition to be maintenance feasible, meaning that each rotation includes at least one maintenance opportunity prior to the deadline (which depends on the rotation) of its first maintenance check. Note that, for some rotations, it will not be necessary to perform the first maintenance check within the time horizon; selection of such rotations is what allows the objective function to vary.

In constraint (2), we require that all lines-of-flight in the schedule are covered. Next, in constraint (3) all tails are permitted to perform exactly one rotation. Constraint (4) requires that if a rotation requires a maintenance check by a particular deadline, then this check must be scheduled on (or before) this particular deadline. The second term determines which day the check is due on, as a function of the chosen rotation, and the first term then ensures that the check occurs by that due date. Constraint (5) requires that a maintenance check can only be scheduled on a given day at a given station if the tail is assigned to a rotation that overnights at that station on that day. Finally, constraint (6) requires that station capacity constraints are observed. Constraint (7) and (8) require variable integrality. In addition, dual variables for all constraints are given in parenthesis.

Solving the First Maintenance Check Problem

Note that this formulation has a very large number of variables, including one for every feasible aircraft rotation. The vast majority of these variables will have value zero in the optimal solution, however, suggesting the value a column-generation based approach.

In the traditional column-generation framework, the restricted master problem is solved which provides dual (pricing) variables to determine if new columns should be included in the problem. A secondary optimization problem, the pricing problem, is then solved to determine the additional columns (in our case rotations) to include. An example of this is seen in Barnhart et al. [1998].

In our approach, we are actually able to enumerate all valid rotations that can be formed by combining the input lines-of-flight. As noted before, we are not changing the construction of LOFs, but rather focus on the combinations of LOFs to form rotations. This reduces the complexity of the optimization problem by reducing the overall decision space required by the model.

Although it is tractable to enumerate all feasible rotations, it is not practical to explicitly include them in the integer program. Instead, we use the dual variables of the primal problem to directly to compute the reduced cost of all additional rotations in our candidate set. Any rotations with negative reduced cost are then included in the next iteration.

Using the dual-variables from the restricted master problem we can determine the reduced-cost for all additional rotations as follows:

$$\begin{aligned}\overline{c_{rt}} = 0 - \sum_{l \in L} \delta_{lr} \gamma_l - \theta_t + \sum_{t \in T} \sum_{m \in M} \sum_{d \in D} \sum_{c \in C} r(d,t,c,r) \alpha_{tcmd} \\ + \sum_{t \in T} \sum_{m \in M} \sum_{d \in D} \sum_{c \in C} \nu_{rdm} \beta_{tmdc}\end{aligned} \tag{9}$$

We apply the resulting dual variables directly to the set of off-line generated rotations, which is saved from one iteration to the next. Any rotation that has a negative reduced cost will then be included in the restricted master LP problem, which is subsequently resolved to obtain new dual variables. This process continues until either no rotations provide a negative reduced cost or all rotations have been included in the master problem. At this point, we solve the restricted master problem, with all variables that have been priced into the problem, as an IP. Note that this is heuristic, as we are not pricing out new rotation at each node of the branch-and-bound tree.

We apply our approach to real-world data from a major domestic US carrier data. We consider a data set for a medium size airline with a 5-day planning window, including 71 aircraft and 2,176 flights over this horizon. A total of 104 iterations are required to solve the LP relaxation of FMCP for this instance.

Given the iterative nature of the column-generation solution algorithm, with each step additional rotations are included in the restricted master problem. The effectiveness of our algorithm is a function of the number of rotations that must be included to attain a solution. Figure (7) illustrates the number of rotations that must be included to attain a feasible solution. For our medium size airline, at the optimal solution, we include $15,048$ rotations out of a possible $856,537$ rotations.

For comparison purposes only, we illustrate both the LP and IP objective function value during each iteration. That is, at each iteration of the restricted master problem, we solve both the LP and the corresponding IP, to investigate gaps and run times. Once the LP has been solved to optimality, we solve the IP one final time, with all generated columns, to find a final (heuristic) solution to the problem.

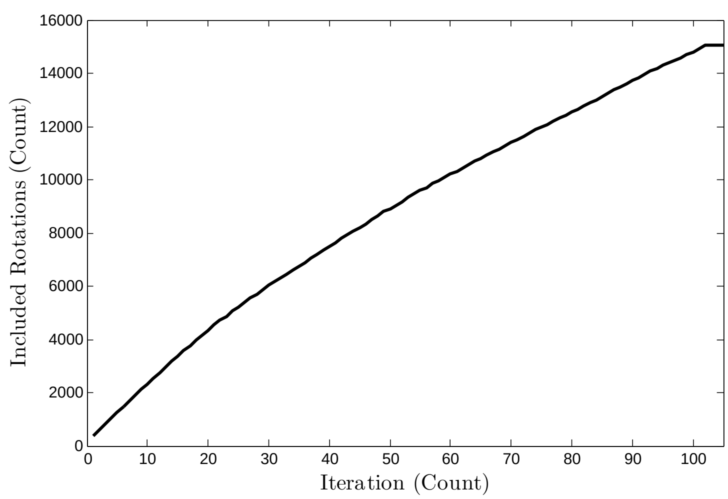


Figure 7: Number of rotations generated by iteration, cumulatively

Finally, we evaluate our approach in terms of overall runtime. We solve many iterations of the restricted master problem in sequence, each one adding new rotations to the set of included candidate rotations. With each of these subsequent problem instances, the solution time to determine the optimal solution will increase due to the increase in the overall size of the problem from the inclusion of additional rotations. However, as illustrated in Figure (8), each iteration for our instance of the problem solves in a matter of seconds, which demonstrates the effectiveness of our approach.

As noted in the figure, the solution to the IP is also relatively fast and requires at most 45 seconds. Due to the random starting nature of the branching process, we note some variation in the solution time, however, on average, it tends to increase with each iteration count. For our data instance we can solve the the FMCP in about two minutes (81 seconds for the LP column-generation process and 45 seconds for the IP branching process) using this modified column-generation approach.

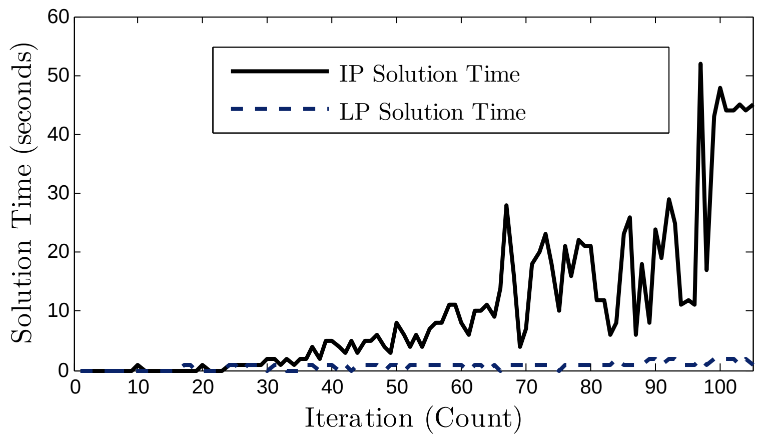


Figure 8: Required solution time per iteration of column generation algorithm

In this section, we provided a heuristic solution to the problem of scheduling the *first* maintenance check, while choosing a particular rotation for each tail. We solve this problem quickly (2 minutes) and provide a maintenance feasible assignment for a 5-day horizon. However, scheduling only the first check ignores the underlying recurrence of a maintenance check. That is, the decision of scheduling the first A-check impacts the decision on the second A-check, which in turn affects the third A-check and so on. In the next section, we consider the problem in which we not only consider the next check, but all subsequent checks that fall into the planning horizon.

5 Recurring Maintenance Check Problem

5.1 Recurring Maintenance Checks

The recovery problem we have discussed up to this point is a simplification of the real-world problem, in that we only consider the first (i.e. next due) maintenance check of each type, ignoring any future checks that may be required. Ignoring such future checks, however, poses several difficulties.

Omitting all future checks may lead to invalid, maintenance-infeasible tail assignments. More specifically, if we only consider the next check, there is no guarantee that a future maintenance check can be reached, because the underlying network structure may not allow the aircraft to reach a subsequent maintenance station when required. Furthermore, by disregarding future checks, we also ignore maintenance station capacity.

Our work so far has provided us with insights which we will now apply to solve the more realistic (and more challenging) problem in which we consider not only the first, but subsequent maintenance checks as well. Scheduling subsequent checks increases the problem complexity, not only because the total number of maintenance checks increases, but also because the decision regarding the deadline of one check determines the deadline of the subsequent check and so on. We refer to this problem as the Recurring Maintenance Problem (*RMP*).

To illustrate the increased complexity of scheduling recurring maintenance checks, consider the following example. Figure (9) shows the current state of the maintenance counters on a particular tail, Tail #1, which has just completed an A-check. Given this status, we can assign several possible rotations each with their associated maintenance plan that specifies the times as to when the next, and all subsequent, A-checks will take place over a five day planning horizon.

Tail 1		Current	Limit
A-check	Hours (H):	0	40
	Cycles (C):	0	10

Figure 9: Tail 1: Beginning A-check status

As noted in Figure (10), the first maintenance plan schedules an A-check on Day 3, satisfying the 40-hour flight limit. The decision to schedule the check on day 3 then enables the next A Check to take place on day 5. In contrast, the second maintenance plan performs the first A-check one day earlier. This then reduces the time before the second check, which now would be infeasible on day 5.

Performing this check early also forces the second subsequent check to be performed a day earlier on day 4. There are several additional plans (not-illustrated) in this scenario, where checks are performed earlier than specified by the deadline. Note, that in this case, plan 1 provides for a higher utilization of the assigned aircraft between maintenance, while the second plan may be desirable in cases of limited maintenance capacity. In other words, plan 2 could be more desirable if day 4 (from plan 1) would contain a large number of other checks from other aircraft.

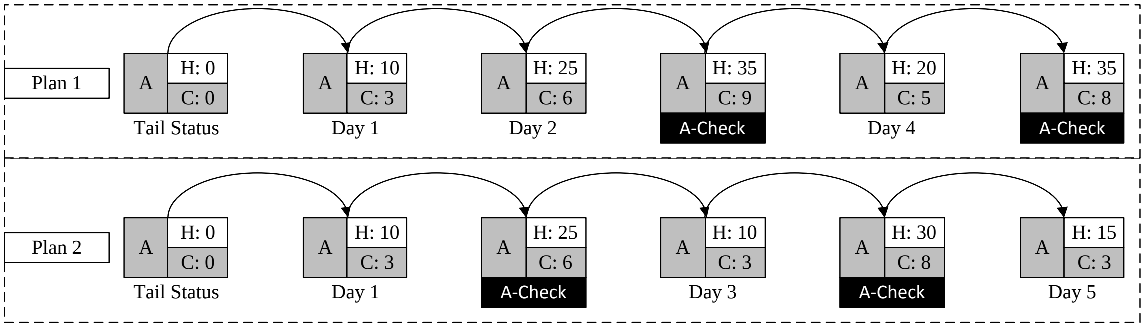


Figure 10: Maintenance A-check assignments with recurrence

5.2 Pure Rotation-based Approach

5.2.1 Pure Rotation-based Formulation

To solve the recurrent maintenance check problem, we explore two alternatives which we will refer to as: 1) the *pure rotation-based approach* and 2) the *augmented rotation-based approach.* The difference between these two approaches manifests itself in the variable definition. For the pure rotation-based approach, the primary decision variable only represents information about the assigned rotations, while in the augmented approach, the primary decision variable represents maintenance scheduling information, in addition to the rotation information.

To formulate the pure rotation-based approach, we introduce the following (additional) notation.

Sets

$N(c)$ The maximum number of checks of a given type c required over a planning horizon, $\forall c \in C$. This is computed by the total days in the horizon divided by shortest maintenance counter.

Parameters

$r(d_e, d_s, t, c, r)$ This parameter is 1 if tail t last received maintenance of check type c on d_s and will require its subsequent check by day d_e if assigned to rotation r, $\forall d_e, d_s \in D, \forall t \in T, \forall c \in C, \forall r \in R(t)$.

$m(r, d) \in M$ An binary parameter that provides the station where rotation r overnights on day d of the planning horizon, $\forall r \in R, \forall d \in D$.

Variables

y_{tr} a binary variable that is 1 if rotation r is assigned to tail t, $\forall t \in T, \forall r \in R(t)$.

w^n_{tdc} a binary variable that is 1 if tail t receives the n-th check of type c on day d of the planning horizon, $\forall t \in T, \forall d \in D, \forall c \in C, \forall n \in N$.

v_{tmdc} a binary variable that is 1 if maintenance on tail t is performed at station m on day d of check type c, $\forall t \in T, \forall m \in M, \forall d \in D, \forall c \in C$.

Objective:

$$\min \sum_{t\in T}\sum_{m\in M}\sum_{d\in D}\sum_{c\in C} v_{tmdc} \tag{10}$$

Subject to:

$$\sum_{t\in T}\sum_{r\in R(t)} \delta_{lr} y_{tr} = 1 \qquad \forall l \in L \quad (\delta_l) \tag{11}$$

$$\sum_{r\in R(t)} y_{tr} = 1 \qquad \forall t \in T \quad (\theta_t) \tag{12}$$

$$\sum_{\substack{d_1\in D:\\ d_1\le d}} w^1_{td_1c} - \sum_{r\in R(t)} r(d,0,t,c,r) y_{tr} \ge 0 \qquad {}^1 \tag{13}$$

$$\sum_{\substack{d_2\in D:\\ d_2>d_1\\ d_2\le d}} w^n_{tcd_2} - \left(w^{n-1}_{tcd_1} + \sum_{r\in R(t)} r(d,d_1,t,c,r) y_{tr} - 1 \right) \ge 0 \qquad {}^2 \tag{14}$$

$$v_{tmdc} - \left(\sum_{n\in N} w^n_{tdc} + \sum_{\substack{r\in R(t)\\ m(r,d)=m}} y_{tr} - 1 \right) \ge 0 \qquad {}^3 \tag{15}$$

$$\sum_{c\in C}\sum_{t\in T} \theta_c v_{tmdc} - \rho_{md} \le 0 \qquad \forall m \in M, \forall d \in D \quad (\beta_{md}) \tag{16}$$

$$y_{tr} \in \{0,1\} \qquad \forall t \in T, \forall r \in R(t) \tag{17}$$

$$v_{tmdc} \in \{0,1\} \qquad {}^4 \tag{18}$$

$$w^n_{tdc} \in \{0,1\} \qquad {}^5 \tag{19}$$

[1] $\forall t \in T, \forall d \in D, \forall c \in C \quad (\xi_{tdc})$

[2] $\forall t \in T, \forall d \in D, \forall {}^{d_1 \in D}_{d_1 < d}, \forall c \in C, \forall n \in N(c) : n > 1 \quad (\kappa_{tdd_1cn})$

[3] $\forall t \in T, \forall m \in M, \forall d \in D, \forall c \in C \quad (\alpha_{tmdc})$

[4] $\forall t \in T, \forall m \in M, \forall d \in D, \forall c \in C$

[5] $\forall t \in T, \forall n \in N, \forall d \in D, \forall c \in C$

The objective function in equation (10) minimizes the total number of maintenance checks that are completed during the planning horizon. In constraint (11), we ensure line-of-flight coverage. In constraint (12) we require that each tail is assigned to a rotation.

The next set of constraints are used to enforce maintenance limits on assigned rotations. Constraint (13) requires a maintenance check to take place if the rotation assigned requires this check to be completed. For example, if a particular rotation is assigned that requires a maintenance check to be completed on or before day 2, then the first part of the constraint ($w^1_{td_1c}$) ensures that a maintenance check is scheduled anywhere between the beginning of the planning horizon and day 2. Similarly, constraints (14) require future checks to be performed relative to the deadline established by the timing of the preceding check. Constraint (15) connects the day and station of a maintenance check to the rotation that is selected. Constraint (16) enforces station capacity limits. Finally, constraint (17) to (19) require variable integrality. In addition, dual variables for all constraints are given in parenthesis.

5.2.2 Solving the Pure Rotation-based Approach

We solve the pure rotation-based approach using column-generation, providing a subset of all possible rotations to the problem, solving the master problem and pricing out any additional rotations and including those with a negative reduced cost. As with the First Maintenance Check Problem from §4, in this approach we enumerate the set of all possible rotations, and then price out all of these rotations at each iteration to determine which ones to include in the restricted master problem. Using the dual variables from (11) to (15), the reduced cost function is as follows:

$$\begin{aligned}
\bar{c_{rt}} = \quad & 0 - \sum_{l \in L} \delta_{lr} \beta_l - \theta_t + \sum_{d \in D} \sum_{c \in C} r(d, 0, t, c, r) \xi_{tdc} \\
& + \sum_{d \in D} \sum_{d_1 < d} \sum_{c \in C} \sum_{n \in N} r(d, d_1, t, c, r) k_{tdd_1cn} + \sum_{m \in M} \sum_{d \in D} \sum_{c \in C} m(r, d) \alpha_{tmdc} \qquad (20)
\end{aligned}$$

We first pre-compute the set of all possible rotations, Ω_t^P. Next, we solve the restricted master problem above using a linear-programming solver. Subsequently, we price all rotations in the passive set of additional rotations Ω_t^P and include those with negative reduced cost. Finally, we take the final LP solution of the restricted master problem and branch on the y_{tr} variables to obtain an integer solution.

We construct all possible rotations corresponding to the length of the planning horizon that can be formed by linking lines-of-flight that can be connected (i.e. arrive and depart at the same station within a given time window). We will refer to the total set of rotations as the passive set (Ω_t^P) from which rotations are included in the restricted master problem. With this set of rotations, we are now ready to solve the pure-rotation based approach.

5.2.3 Pure Rotation-based Computational Results

The pure rotation-based approach is solved using a column-generation approach in which rotations are pre-processed. To initialize the restricted master problem, we start with a set of columns we know to satisfy all the constraints of the optimization problem. In practice, however, these columns are not feasible because the lines-of-flight from one day to the next may not actually connect. In other words, such a rotation is technically infeasible, however, it can be used to satisfy the maintenance requirements. These columns provide an initial basic feasible solution, however, and we apply a suitably large penalty in the objective function to guarantee that they will ultimately be pivoted out of the solution to the restricted master.

As outlined in Figure (11), we continue to price additional rotations with each iteration. We loop through the candidate rotations for each tail, pricing them out with the current dual values and, for each tail, stop as soon as we find a negative rotation.

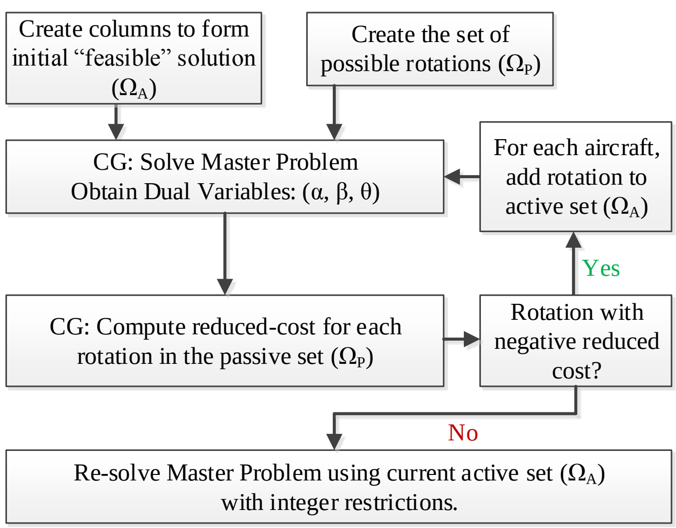
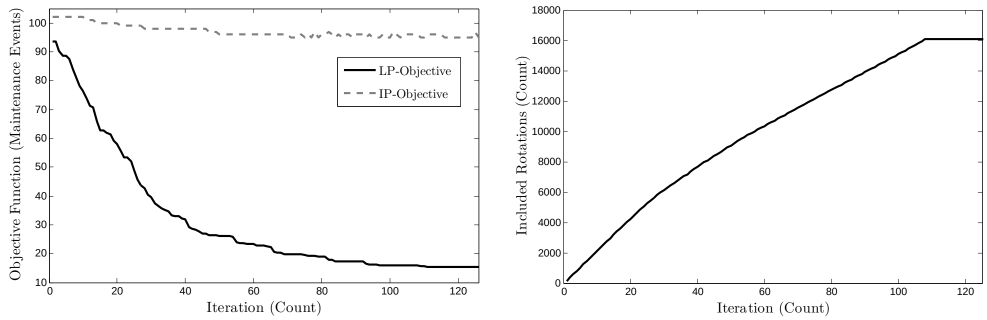


Figure 11: Column generation approach using pre-processed rotations

The results are shown in Figure (12) where we solve the restricted master problem (LP), and for comparison purposes, the corresponding restricted master IP. We emphasize that these results are time constrained. That is, each IP iteration was permitted to run for a maximum of 600 seconds of



(a) Objective function value (maintenance counts excluding penalty) progression

(b) Rotations included in restricted master

Figure 12: Solution approach to pure-rotation based model

CPU time or until it reached a solution, whichever came first. Note that, after iteration 41, as seen in Figure (13),all IPs corresponding to the current LP fail to solve within 600 seconds.

Furthermore, when comparing the IP solution to the LP solution, we note a growing divergence in objective function value as the number of iterations increase. In other words, solving the LP relaxation for the pure rotation-based approach does not appear to be an effective bound on the total number of maintenance checks that must be completed from the IP solution.

Therefore, we suggest that the pure rotation-based approach is not viable for most instances of the maintenance recovery problem with recurring checks. It motivates us, however, to consider an alternative formulation, the augmented-rotation based model, which combines all rotation information into a single variable."

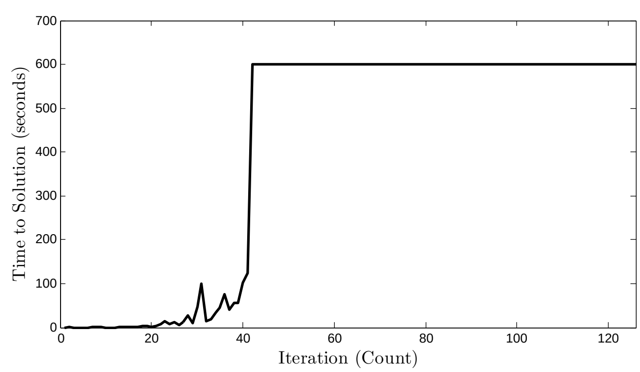


Figure 13: Pure rotation-based solution time results for IP solution.

5.3 Augmented Rotation-based Approach

5.3.1 Augmented Rotation-based Formulation

In the pure rotation-based approach, sequences of lines-of-flight are strung together to form multi-day sequences of flights and additional variables and constraints are used to enforce maintenance checks. In the augmented rotation-based approach, we build the maintenance checks directly into the rotation. We use the variable v_{tr} to determine which rotation is assigned to a particular tail t. However, in this case, a rotation consists not only of a set of lines-of-flight, but also includes a set of maintenance checks, including their respective scheduled day and station, that will be performed if such a rotation is selected. As a result, this formulation requires a larger number for variables, but the number of constraints is reduced.

Sets

R^A	The set of all augmented rotations.
$R^A(t) \subseteq R^A$	The set of all augmented rotations that can be assigned to tail t, $\forall t \in T$.

Parameters

κ_r	The number of maintenance checks that are contained within rotation r, $\forall r \in R$.
ξ_{crsd}	A binary parameter that indicates whether rotation r contains a check of type c at station s on day d of the planning horizon, $\forall c \in C, \forall r \in R^A(t), \forall s \in M, \forall d \in D$.

Variables

v_{tr}	a binary variable that is 1 if augmented-rotation r is assigned to tail t, $\forall t \in T, \forall r \in R^A(t)$.

Objective:

$$\min \sum_{t \in T} \sum_{r \in R^A(t)} \kappa_r v_{tr} \tag{21}$$

Subject to:

$$\sum_{r \in R^A(t)} v_{rt} = 1 \qquad \forall t \in T \quad (\theta_t) \tag{22}$$

$$\sum_{t \in T} \sum_{r \in R^A(t)} \delta_{lr} v_{tr} = 1 \qquad \forall l \in L \quad (\beta_l) \tag{23}$$

$$\sum_{t \in T} \sum_{r \in R^A(t)} \sum_{c \in C} \theta_c \xi_{crsd} v_{rt} \leq \rho_{sd} \qquad \forall s \in M, \forall d \in D \quad (\alpha_{sd}) \tag{24}$$

$$v_{tr} \in \{0, 1\} \qquad \forall t \in T, \forall r \in R^A(t) \tag{25}$$

The objective in equation (21) minimizes the total number of maintenance checks performed over the planning horizon. Constraint (22) ensures that each tail is assigned a particular rotation. Constraint (23) requires each line-of-flight over the course of the planning horizon to be covered. Finally constraint (24) ensures that the capacity of a maintenance block, measured in available man-hours, is not exceeded. In addition, dual variables for all constraints are given in parenthesis. Note that, relative to the pure rotation-based approach, the augmented rotation-based approach has far fewer constraints, because all maintenance feasibility issues are embedded within the variable definition, but far more variables.

5.3.2 Solving the Augmented Rotation-based Formulation

In section (5.2.2), we discussed the exhaustive set of rotations used to solve the pure rotation-based approach. The augmented rotation-based approach builds on this approach by interspersing maintenance tasks as necessary.

Starting with an exhaustive set of rotations, for the augmented rotation-based formulation, we most now convert each rotation into a collection of augmented rotations by inserting maintenance checks. In Figure (14), we provide a possible rotation to demonstrate our algorithm which will insert maintenance checks where appropriate. In this rotation, an aircraft flies five lines-of-flight over the course of a five day horizon. As noted on the diagram, each line-of-flight has a certain number of hours (H) and cycles (C) associated with it. For simplicity of this example, we ignore the calendar days maintenance counter. Moreover, in this example, each station at the end of the day features the opportunity to perform aircraft maintenance.

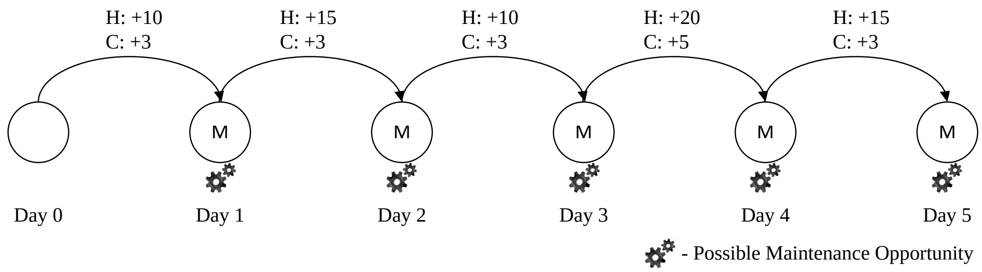


Figure 14: Sample conjunction of five lines-of-flight

To determine the set of possible rotations for a particular tail, we begin by defining the tail properties as seen in Figure (15) below. This indicates that we are building rotations for Tail #1, which just completed an A-Check, and thus its maintenance counters for this check are set to zero. In addition, this tail still has 50 flight hours and 30 cycles before it requires its next B-Check.

Tail #1		Current	Limit
A-check	Hours (H):	0	40
	Cycles (C):	0	10
B-check	Hours (H):	50	80
	Cycles (C):	30	40

Figure 15: Tail information used for generation

Figure (16) shows four augmented rotations associated with the basic rotation from Figure (14). Rotation 1 features the first maintenance check on day 2, as this is the farthest the aircraft can go without a B-check given the characteristics (flying hours and cycles) of the lines-of-flight. In addition, an A-check follows the next day, because of the flight hour and cycle count. For rotation 2, given that we are allowed to perform maintenance at most one day early, the B-check is now scheduled for day 1, while the A-checks remain the same. For rotation 3, not only is the B-check moved one day forward, but now the A-check could also be performed one day earlier than required. However, performing this A-check early requires another A-check to be added on day 4. Finally, in rotation 4, the A-check that was added on day 4 for rotation 3 can also be performed a day earlier. As such, in this rotation, the A-checks follow each other directly. As the application of the problem expands to larger data sets, one could argue that rotation 4, while feasible, performs two successive A-checks, which may be undesirable from a capacity planning perspective and therefore should be eliminated. In our approach, however, we do not eliminate such rotations.

In theory, the number of augmented rotations for any given basic rotation and aircraft (with its current set of characteristics relative to the different types of maintenance checks) can be exponential,

Figure 16: Sample of possible rotations including maintenance

with the option of scheduling or not scheduling each possible maintenance check at each possible overnight at a maintenance station. In practice, this is too large to be tractable. In addition, many of these augmented rotations will either be maintenance-infeasible (e.g. by leaving too much time between maintenance checks) or highly undesirable and unlikely to be included in an optimal solution.

We therefore limit the augmentations that we include in our computational experiments. In our approach, we incorporate three business requirements into the rotation generation process. First, we define whether or not a maintenance check will be scheduled at the end of the planning horizon. This allows of the scheduling of a maintenance check at the end of the recovery horizon even if maintenance is not necessary for the aircraft. Next, we restrict the number of days early maintenance can be performed. Finally, we define a parameter that controls whether two maintenance checks can be scheduled on the same day.

The augmented rotation-based approach contains a large number of variables, even for relatively short planning horizons. In Figure (17), we graphically demonstrate the impact of each additional day of the planning horizon on the number of possible augmented rotations. However, in our solution approach, we will not solve a problem containing all variables simultaneously. Instead, we use a similar approach as to the one posed for the pure rotation-based approach previously, whereby we use a column-generation approach to price out only those variables that improve the objective.

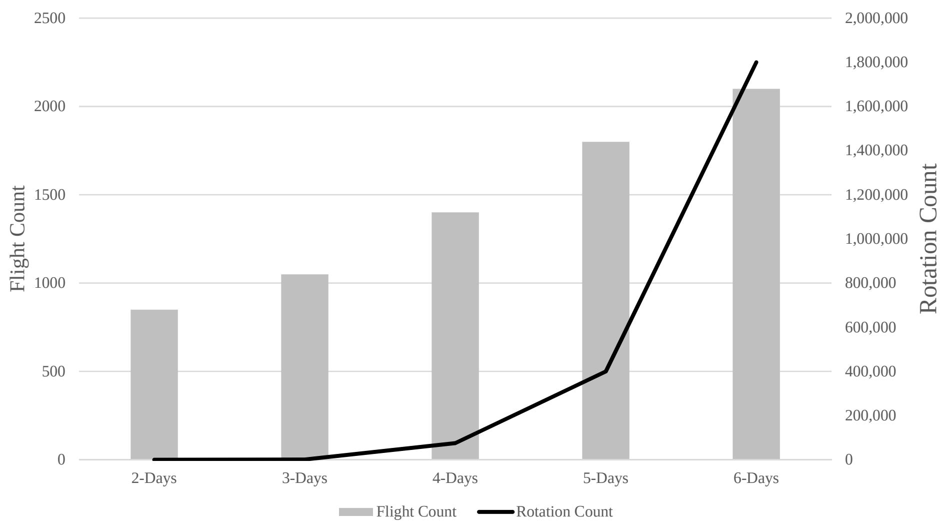


Figure 17: Passive-set (Ω_t^P) variable count

We begin by solving the optimization problem with a subset of all possible augmented rotations, the active-set (Ω_t^A). With these dual-variables, we note the reduced-cost equation that will be used to price each of the additional rotations from passive set. The passive set includes all feasible rotations and their respective maintenance tasks. This reduced-cost equation is shown in (26). Note that the dual-variable θ_t has an implied 0/1 multiplier. That is, the dual-variable θ_t is only applied if the proposed rotation can actually be flown by tail t for which it is being considered. Otherwise, this variable is not considered in the reduced-cost calculation. As noted earlier, if a $\overline{c_{rt}}$ is negative, then this variable is included in the active-set and the master problem is resolved. We iterate between solving the optimization problem (master problem) and the pricing problem until we cannot find a rotation that has a negative reduced-cost.

$$\overline{c_{rt}} = \kappa_r - \sum_{c \in C} \sum_{s \in M} \sum_{d \in D} \xi_{crsd} \alpha_{sd} - \sum_{l \in L(t)} \delta_{lr} \beta_l - \theta_t \tag{26}$$

5.3.3 Augmented Rotation-based Approach Results

We implemented our approach using the C++ programming language with CPLEX as our solver. System specifications for our computational experiments include an Intel Core i7 processor with a

2.8GHz clock speed and 8GB of main memory. Our implementation is based on serial execution and does not take advantage of multiple processors (or processor cores). However, note that the pricing problem, as illustrated in our approach, is an independent problem. More specifically, solving a pricing problem for a particular tail does not involve any cross-tail constraints. As such, solving such a problem lends itself to parallelization for improved computational performance.

We now provide computational results for both the pure rotation-based and augmented rotation-based problems illustrated earlier in this section. These results suggest that only the augmented rotation-based approach provides a solution to the recurring maintenance check problem in a reasonable amount of computation time.

In Figure (18), we illustrate the effectiveness of our approach. That is, this figure illustrates the convergence of the algorithm to a solution.

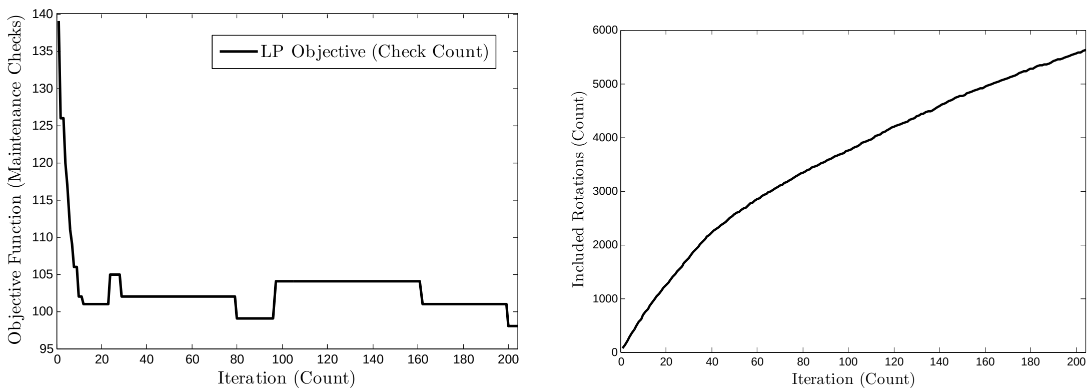
To initialize the restricted master problem, we employ the same technique as in the pure rotation-based approach. That is, we start with a set of columns we know to satisfy all the constraints of the optimization problem. Given the fact that some non-feasible rotations are included as part of the solution approach, the objective function (excluding the penalty-coefficients) can actually *increase* during the column-generation process. That is, we find a rotation in the passive set that, when included, reduces the overall objective function value (by removing an infeasible rotation with an artificially high objective function coefficient), however, given this rotation the total number of maintenance checks actually increases. Since we apply a suitably large penalty in the objective function, these columns will eventually be pivoted out of the solution to the restricted master.

Solving the five-day horizon problem is done over the course of 204 iterations of the modified column-generation solution algorithms. The objective function shown here is the total number of checks that are included in the final set of assigned rotations.

Thus far we solved the maintenance recovery problem with recurring checks. Based on our results, the augmented-rotation-based approach appears most successful in solving the problem both from a feasibility and speed-to-solution perspective.

6 Conclusions and Future Work

We have considered the maintenance recovery problem (MRP), which seeks to recover a disrupted maintenance plan. In our approach, we evaluated two variations of the problem. In the first, we only considered the most imminent ("next") required maintenance activity for each aircraft, and try to push these checks as far into the future as possible (i.e. minimize the number of such checks



(a) Maintenance check count value progression

(b) Required rotations in solution

Figure 18: Solution approach to augmented-rotation-based model

that must fall in a short-term time horizon). This provided a foundation for the more realistic, and more complex, second variation in which we develop a solution methodology for a specified recovery horizon throughout which we seek to minimize the total number of maintenance checks required to attain feasibility.

We presented two different formulations (based on *pure-* and *augmented-rotations*) to address this second variation, discussed the challenges associated with solving these different formulations, and developed solution techniques for each to achieve tractability.

As we have shown, each of these problems can be effectively solved using a column-generation approach in which the rotations are augmented to include maintenance checks within each actual rotation. Furthermore, we can solve the rolling horizon model using two different objectives to assist in the maintenance man-power planning process, which illustrates an extension to this particular problem.

As for future work, we believe that there is applicability of our models not only for recovery, but for maintenance planning as well. More specifically, as we solve the augmented rotation-based approach, we realize the work that is performed by all aircraft over the course of the planning period. These line-of-flight assignments imply the checks that must be realized, much like the maintenance recovery problem. However, if solved for a longer horizon, the maintenance capacity required at each station can be noted, as seen in Figure (19). Based on these requirements, maintenance man-power can be accurately predicted despite changing airline plans. This is because our recovery approach reveals the cyclical nature of maintenance checks. That is, we emphasize less the *specific* tail that rotates through a maintenance station, but rather that *a* tail rotates through a maintenance station.

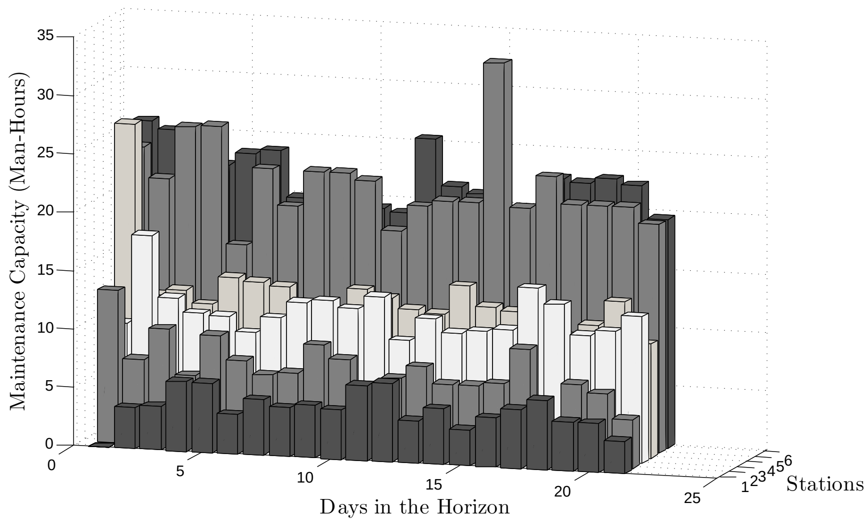


Figure 19: Maintenance capacity planning based on rolling horizon approach

References

Khaled F. Abdelghany, Ahmed F. Abdelghany, and Goutham Ekollu. An integrated decision support tool for airlines schedule recovery during irregular operations. *European Journal of Operational Research*, 185(2):825 – 848, 2008. ISSN 0377-2217.

Cynthia Barnhart, Ellis Johnson, George Nemhauser, Martin Savelsbergh, and Pamela H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46 (3):316–329, 1998.

Lloyd Clarke, Ellis Johnson, George Nemhauser, and Zhongxi Zhu. The aircraft rotation problem. *Annals of Operations Research*, 69(1):33–46, 1997. doi: 10.1023/A:1018945415148.

Amy M. Cohn and Cynthia Barnhart. Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research*, 51(3):387–396, 2003. ISSN 0030364X.

Mark S. Daskin and Nicholaos D. Panayotopoulos. A Lagrangian Relaxation Approach to Assigning Aircraft to Routes in Hub and Spoke Networks. *Transportation Science*, 23(2):91–99, 1989.

Delta. Annual 10-k company filing. 2010.

Niklaus Eggenberg, Matteo Salani, and Michel Bierlaire. Constraint-specific recovery network for solving airline recovery problems. *Computers & Operations Research*, 37(6):1014 – 1026, 2010. ISSN 0305-0548.

Ram Gopalan and Kalyan T. Talluri. The aircraft maintenance routing problem. *Operations Research*, 46(2):260–271, 1998. ISSN 0030364X.

Abdulkadir Sarac, Rajan Batta, and Christopher Rump. A branch-and-price approach for operational aircraft maintenance routing. *European Journal of Operations Research*, 3(175):1850–1869, 2006.

Chellappan Sriram and Ali Haghani. An optimization model for aircraft maintenance scheduling and re-assignment. *Transportation Research Part A: Policy and Practice*, 37(1):29 – 48, 2003. ISSN 0965-8564.

USAirways. Annual 10-k company filing. 2010.

Shangyao Yan, Ta-Hui Yang, and Hsuan-Hung Chen. Airline short-term maintenance manpower supply planning. *Transportation Research Part A: Policy and Practice*, 38(9-10):615 – 642, 2004. ISSN 0965-8564.