Scheduling a Capacity-Constrained Aircraft Repair Shop with Dynamic Repair Arrivals[1]

Maliheh Aramon Bajestani and J. Christopher Beck
Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ontario, Canada
{maramon,jcb}@mie.utoronto.ca

Abstract

We address a dynamic repair shop scheduling problem in the context of military aircraft fleet management where the goal is to maintain a full complement of aircraft over the long-term. A number of flights, each with a requirement for a specific number and type of aircraft, are already scheduled over a long horizon. We need to assign aircraft to flights and schedule repair activities while considering the flights requirements, repair capacity, and aircraft failures. The number of aircraft awaiting repair dynamically changes over time due to failures and it is therefore necessary to recompute the repair schedule online. To solve the problem, we view the dynamic repair shop as successive static repair scheduling sub-problems over shorter time periods. We propose a complete approach based on the logic-based Benders decomposition to solve the static sub-problems, and design different rescheduling policies to schedule the dynamic repair shop. Computational experiments demonstrate that the Benders model is able to find and prove optimal solutions seven times faster (geometric mean) than an integer programming approach. The rescheduling approach having both aspects of scheduling over a longer horizon and quickly adjusting the schedule increases aircraft available by 13% compared to the approaches having either one of the aspects alone.

1 Introduction

For many industries with expensive machinery, repairing a failed machine is significantly more economical than replacing it. For example, it is far too expensive for a railroad or airline company to keep idle trains or planes on-hand to replace failed machines. The need for repair, however, generates a set of new decisions: "How many repair shops are needed?", "How many repair resources (e.g., repairpersons) should be allocated to each repair shop?", "Where should the repair shops be located?", and "When should each repair be done?". In this paper, we study the problem of scheduling a dynamic repair shop to answer the last question. When aircraft fail, the management process must dynamically react by scheduling and rescheduling repair activities. A repair schedule capable of dealing with uncertainty and adjusting to unexpected events has a substantial effect on maximizing the aircraft availabilities.

Motivated by the work of Safaei et al. [32, 33], we study the problem of scheduling a military aircraft repair shop. In the problem, a number of flights are planned over a long horizon. Every flight, called *a wave*, has a requirement for a specific number of aircraft of different types. At the beginning, an aircraft is either ready for a pre-flight check or is awaiting repair in the repair shop. Aircraft flow over a long horizon is illustrated in Figure 1. The goal is to determine an assignment of aircraft to waves and a schedule of repair jobs that will maximize flight coverage, the extent to which the aircraft requirements of the flights are met. When an aircraft fails during a pre- or post-flight check, it enters the repair shop and must be incorporated into the current repair schedule. Each aircraft failure corresponds to a set of independent repair activities with known characteristics such as processing times and resource requirements to be scheduled on repair resources with a limited capacity.

The main idea in our solution approach is to view the dynamic repair shop as successive static sub-problems over shorter time periods. A solution of the static sub-problem determines an assignment of aircraft to flights and a schedule of repair jobs maximizing the flight coverage. When a failed aircraft enters the repair shop while

[1]This paper is a combined and extended version of a conference paper [2] and workshop paper [1] that have already appeared. The current paper extends these works in the following ways: the static problem is proved to be NP-complete; it is then generalized to its dynamic counterpart; all experiments are re-run to analyze the impact of different rescheduling strategies on when and how the scheduling and rescheduling should be done; and substantial new analysis of the experimental results is done.

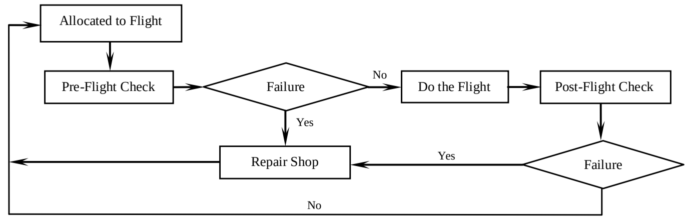

Figure 1: A flow chart representing aircraft flow among waves, checks, and the repair shop over a long horizon.

the previous repair schedule is still under execution, we reschedule the repair activities by solving a new static sub-problem.

In this paper we explore several techniques to solve the static sub-problem: mixed integer programming (MIP); logic-based Benders decomposition (LBBD) using either MIP or constraint programming (CP); a dispatching heuristic motivated by the Apparent Tardiness Cost (ATC) dispatching rule; and two hybrid approaches which integrate the dispatching heuristic with complete approaches (MIP and LBBD). We then design three different rescheduling policies based on how the static sub-problems are connected. For example, we schedule the repair activities over a time period containing three flights, and reschedule after each flight.

We perform two separate empirical studies. The first indicates that the integration of the dispatching heuristic and LBBD results in the lowest mean run-time of the techniques tested to optimally schedule the repair shop especially as the number of aircraft increases. The second experiment demonstrates that both defining the static scheduling problem over a longer horizon and rescheduling more frequently provide the flights with on average 13% higher coverage than either one of them alone.

The remainder of the paper is organized as follows: We define the problem, and provide an overview of the relevant literature in Section 2. Section 3 defines a number of solution approaches for the static sub-problem and presents the details of proposed rescheduling policies for scheduling the dynamic repair shop. The computational results on the performance of different scheduling techniques and on how and when scheduling and rescheduling should be done are described in Section 4. A discussion of our results is presented in Section 5. We end with conclusion and directions for the future work in Section 6.

2 Background

In this section, the formal definition of the problem is given and the relevant literature is reviewed.

2.1 Problem Definition

Figure 2 is a snapshot of the problem at time 0, where circles represent aircraft. A number of flights (five are shown) and their corresponding pre- and post-flight checks are already scheduled over a long horizon. It is assumed that the total number of aircraft is constant over a long horizon. A number of aircraft (three in the diagram) are ready for the pre-flight check while others are currently in the repair shop awaiting repair before they can proceed to a pre-flight check. Failure is only detected during a check and we assume that a check will always correctly assess the status of an aircraft.

The goal is to assign aircraft to waves to maximize coverage while at the same time creating a feasible repair schedule. The scheduling problem is under the constraints that the repair shop has limited capacity and the aircraft are subject to breakdown. We assume that once an aircraft fails, it goes to the repair shop and waits until its repair operations are performed.

We use the following notation to represent the problem.

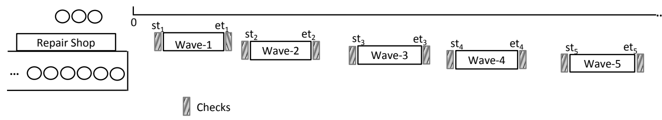

Figure 2: Snapshot of the problem at time 0 over a long horizon.

- N is the set of aircraft. λ_n is the failure rate of the aircraft $n \in N$.
- K is the set of aircraft types. For each aircraft type $k \in K$, there are A_k aircraft ready (i.e., not in the repair shop) at time 0. $\bar{\lambda}_k$ is the mean failure rate over all aircraft of type k, and I_k denotes the set of aircraft type k.
- R is the set of repair resources (called *trades*). The maximum capacity of trade $r \in R$ is C_r.
- W is the set of waves. Each wave, $w \in W$ has a start-time, st_w, and an end-time et_w. Each wave requires a_{kw} aircraft of type k.
- J is the set of existing jobs in the repair shop. Each job is associated with a specific aircraft type. M_r is the set of jobs requiring trade r. Each job might require more than one trade to be completed. The processing time of job j on trade r is p_{jr} and c_{jr} is the capacity of trade r required by job j.

To find the probability of failure of an aircraft in pre- and post-flight checks for a specific wave, we need to track the complete history of the aircraft. For example, assume that a given aircraft is repaired and assigned to the first wave. There are three paths: the aircraft fails the pre-flight check; the aircraft passes the pre-flight check, flies the wave, and fails the post-flight check; or the aircraft passes the pre-flight check, flies the wave, and passes the post-flight check. Therefore, the availability of the aircraft for the second wave can be represented as a random variable whose expected value depends on the probability of these three different paths and the scheduling decisions to repair the failed aircraft before the second wave. Similarly the availability of the aircraft for subsequent waves depends on its entire path through the checks and waves. As the number of waves and aircraft increase, the size of the state space will become prohibitive. Furthermore, the scheduling decisions themselves impact the aircraft histories: the probability that an aircraft is available for the third wave is different depending on if it was repaired in time for the first wave or only for the second wave. The details on calculating the failure probabilities are presented in Section 3.2.1.

As the complexity of the problem has not been shown, we provide a short proof that it is NP-hard in Section 3.1.

2.2 Literature Review

This section provides necessary background on repair shop scheduling problem and the logic-based Benders decomposition.

2.2.1 Repair Shop Scheduling

Repair shops have been mainly studied as a machine-repairman problem [18, 34]. A general machine-repairman problem has a set of workers and set of machines that are subject to failures and therefore need repair. As the number of workers is less than the number of machines, it is necessary to allocate the repair jobs to the workers with the goal of optimizing a given performance measure (e.g., the total expected machine downtime) over the long-term. Derman et al. [12] did the early work on solving the scheduling problem of a repair shop with a single repairman. They showed that repairing the failed machines in non-decreasing order of failure rate stochastically maximizes the number of working machines. The literature on the scheduling of a repair shop was then extended by considering multiple repairmen, preemptive and non-preemptive repair, and different failure and repair distributions among the others. A comprehensive review of the literature on the scheduling of a repair system is provided in [23].

The analytical models developed in the literature to guarantee the optimality in the long-term often do not consider the combinatorics of the real scheduling problems and typically result in some sort of static repair policy such as that found by Derman et al. However, in our problem, the optimization of scheduling performance at discrete time points (i.e., before each flight) is of interest. The processing time and the resource requirement of the repair activities become known when they enter the repair shop. Therefore, we believe that a better performance can be achieved by dealing directly with the combinatorics and explicitly scheduling the repair shop to meet the waves. To handle the uncertain and combinatorial structure of the scheduling problems, we use the dynamic scheduling approach.

Dynamic scheduling is the methodology developed in the scheduling literature where the operational uncertainties like machine breakdowns or the unexpected arrival of new orders prevent the execution of the schedule as planned [3, 25]. The dynamic scheduling problem is generally viewed as a collection of linked static sub-problems. Taking this view makes the algorithms developed for the static scheduling problems applicable because these algorithms can deal with the combinatorics of the scheduling problems and can optimize the quality of the schedule at discrete time points, i.e., for each static sub-problem. However, as these algorithms cannot completely deal with the operational uncertainties, a real-time disruption requires either modification of the schedule to permit execution or improving the quality of schedule considering the most recently revealed information. The process of modifying the previous schedule is called rescheduling [37, 3, 6]. How to solve static sub-problems and how to connect them using rescheduling strategies are the main aspects of the dynamic scheduling research.

Although we are not aware of using dynamic scheduling in a repair system, it has been successfully applied to a variety of scheduling problems including single machine [26, 25, 11], parallel machines [36, 27], and job shop [30, 24, 38].

Another area of literature with similarities to our problem is the operational level maintenance scheduling problem where a schedule for the maintenance activities has to be found such that the sum of maintenance costs is minimized. The focus is on the operational level, determining which maintenance activity is performed in what time periods [7]. Started with the early work of Wagner et al. [39], this literature was then extended through developing appropriate mathematical models and effective solution approaches for a variety of applications [14, 17, 7, 16].

Safaei et al. [32] modeled the static problem addressed here as an operational level maintenance scheduling problem using MIP. Their MIP includes an assignment problem and two network problems: the former assigns the aircraft to the waves and the latter calculates the expected number of available aircraft for the waves as well as the expected number of available workers for the repair jobs. They later extended their work by using slightly different MIP model where the time-index approach is used to enforce the workforce availability constraint and they verified the validity of their model by a number of instances under different combinations of workforce sizes [33].

The difference between the static problem addressed in this paper and the previous works on operational maintenance scheduling is that our objective function (flight coverage) depends not only on the scheduling decisions but also on the outcomes of the pre- and post-flight checks. These two quite different components of the problem motivated the decomposition approach, logic-based Benders decomposition, reviewed below.

2.2.2 Logic-based Benders Decomposition

As a generalization of classical Benders decomposition [5, 15], the logic-based Benders decomposition (LBBD) approach [22, 21] partitions the problem into a master problem (MP) and a set of sub-problems (SPs). The former is a projection of the global model to a subset of decision variables, denoted y, and the constraints and objective function components that involve only y. The rest of the decision variables, x, define the sub-problems. Solving a problem by Benders decomposition involves iteratively solving the MP to optimality and using the solution to fix the y variables, generating the sub-problems. The inference duals [19] of the SPs are solved to find the tightest bound on the global cost function that can be derived from the original constraints and the current MP solution. If this bound is greater than or equal to the current MP solution (assuming a maximization problem), the MP solution and the SP solutions constitute a globally optimal solution. Otherwise, a constraint, called a "Benders cut" is added to the MP to express the violated bound and another iteration is performed.

Representing the relaxation of SPs in the MP, and designing a strong Benders cut are of great importance in decreasing the computational effort to identify the globally feasible and optimal solution. The former results in MP solutions which are likely to satisfy the SPs, and the later rules out a large number of MP solutions in each iteration [20].

Logic-based Benders decomposition has been shown to be effective in a wide range of problems including scheduling [4, 19, 20], facility and vehicle allocation [13], and queue design and control problems [35].

3 Solution Approach

The main idea of our solution approach is to view the dynamic problem as linked successive static sub-problems. As noted above, this is a common approach in dynamic scheduling. This view results in a rescheduling strategy based on scheduling static sub-problems over shorter time periods. Therefore, we have two sub-goals: how to solve and how to connect the static sub-problems. In this section we first show that the static problem is NP-hard and then discuss different solution techniques for solving it. We then explain three various rescheduling strategies designed to connect the static sub-problems.

3.1 The Complexity of the Static Problem

We establish the NP-hardness of the static problem by reduction from a two-dimensional knapsack problem which is strongly NP-hard [8].

Theorem 1. *The static problem is NP-hard.*

Proof. An instance I of a two-dimensional knapsack problem (2KP) consisting m small rectangles, the jth having a width x_j, a height y_j and a profit 1, is considered. The objective is to pack the small rectangles in a bigger one with a width X and a height Y maximizing the profit such that the rectangles do not overlap. From this instance, an instance of the static problem can be formulated such that there are $|W|$ flights, m failed aircraft in the repair shop ($|N| = m$), and one repair resource ($|R| = 1$) with capacity $C = Y$. The set of existing jobs in the repair shop is $J = \{1, 2, ..., m\}$, each having the processing time $p_j = x_j$ and resource requirement $c_j = y_j$. The start time of the first flight is $st_1 = X$ and the probability of aircraft failure in both checks is 0. To maximize the flight coverage, we need to maximize the number of jobs repaired for the first flight as the probability of failure is 0. The problem of scheduling as many jobs as possible before the first flight corresponds to the problem of maximizing the profit in the 2KP problem. As 2KP is strongly NP-hard [8], we conclude that the static problem is strongly NP-hard. □

3.2 Scheduling Techniques

The details of different solution techniques are provided below. In our preliminary experimentation, a pure constraint programming model performed very poorly and so we did not pursue it.

3.2.1 Mixed Integer Programming

We propose a novel mixed integer programming model based on the common *time-indexed* formulation [29]. This model is different from and significantly faster than those of Safaei et al. [32, 33]. In this section, without loss of generality, we interpret W as the set of waves in the current static sub-problem, and define D to be an ordered set of due dates consisting of the wave start-times plus a big value, B, sorted in ascending order. We distinguish aircraft based on their type and use a recursive equation (Eqn (4)) to estimate expected number of available aircraft and flight coverage. The failure rate, $\bar{\lambda}_k = \frac{\sum_{n \in I_k} \lambda_n}{|I_k|}$, is used to calculate the probability of failure associated with each aircraft type during pre- and post-flight checks, respectively: $\xi_k^{pre} = (1 - e^{-\alpha \bar{\lambda}_k})$ and $\xi_k^{post} = (1 - e^{-\beta \bar{\lambda}_k})$, where $\alpha < \beta$ reflects deterioration of the aircraft through use. Note that, the conditions of a reliability function in the extreme values of failure rates hold true for the functions defined above. If the failure rate goes to 0, the probability of failure equals 0, and if the failure rate goes to ∞, the probability of failure equals 1.

The variables are defined in Table 1 and the model is shown in Figure 3. We refer to this model as *MIP*.

Var.	Definition
Z_{kw}	The number of aircraft of type k assigned to fly in wave w
x_{ij}	$x_{ij} = 1$ if the ith due date is assigned to job j
st_{jr}	The start-time of job j on trade r
U_{kw}	The number of aircraft of type k whose repair due date is st_w
E_{kw}	The expected number of available aircraft of type k for wave w
et_{jr}	The end-time of job j on trade r

Table 1: The decision variables (top) and inferred variables (bottom) for the MIP model.

The objective function (1) maximizes the number of aircraft assigned to a wave subject to a bound on the number of aircraft required and the expected number available (Constraint (5)). Equation (2) calculates the number of aircraft of type k whose repair due date is st_w. Equation (3) calculates the expected number of available aircraft for the first wave. Equation (4) calculates the expectation for the other waves. The first term includes those aircraft available but not used for the previous wave and those newly arrived from the repair shop. The second term sums over all aircraft that become available because they have completed waves since the previous wave started. Constraint (6) ensures that exactly one due date is assigned to each job. As noted earlier, the ordered set, D, includes the wave start-times and a big value B. If a failed aircraft cannot be repaired in time for one of the waves, its due date is assigned to B so as to not to constrain the static sub-problem. Equation (7) calculates the end-time of the jobs. The end-time of each job is guaranteed to be less than or equal to the assigned due date by constraint (8). Constraint (9) enforces the capacity limit of each trade, where t denotes each time point during which job j executes.

Z_{kw}, the number of aircraft of type k that are assigned to fly in wave w, is a true decision variable: we can choose to send fewer aircraft on a wave than are currently (in expectation) available. In contrast, E_{kw} is the expected number of aircraft of type k available for wave w and is based on the probabilistic outcomes of previous waves and the number of newly repaired aircraft (U_{kw}).

The objective function (1) is not the expected wave coverage because each wave has specific plane requirements and the maximum wave coverage for each wave is 1. If the expected number of available aircraft E_{kw} is more than the requirement a_{kw} for a given wave, the extra aircraft do not fly the wave and so do not have any contribution to the coverage. By not flying “extra” planes, we do not decrease the probability that they will be available for the next wave.

3.2.2 Logic-based Benders Decomposition

As the static problem requires making two different decisions, assigning aircraft to the waves and scheduling repair jobs for failed aircraft, a decomposition approach may be well suited. A logic-based Benders decomposition (LBBD) method can be formulated where the master problem assigns aircraft to waves to maximize wave coverage and the sub-problems create the maintenance schedules given the due dates derived from the master problem solution. We propose three variations: *Benders-MIP* and *Benders-MIP-T*, where the master problems are solved using MIP, the latter with a tighter sub-problem relaxation; and *Benders-CP*, with a constraint programming-based master problem. All models use CP for the scheduling sub-problems.

$$
\begin{aligned}
\text{max.} & \sum_{w=1}^{|W|}\sum_{k=1}^{|K|} Z_{kw} && (1)\\
\text{s.t.}\ & U_{kw} = \sum_{j\in I_k} x_{ij},\ \text{if } d_i = st_w && (2)\\
& E_{k1} = (A_k + U_{k1})(1-\xi_k^{pre}),\ \forall k && (3)\\
& E_{kw} = (E_{k(w-1)} - Z_{k(w-1)} + U_{kw})(1-\xi_k^{pre}) + \\
& \quad \sum_{v=1}^{w-1} Z_{kv}(1-\xi_k^{post})(1-\xi_k^{pre}), \\
& \quad \text{if } st_{w-1} < et_v \le st_w, \forall w \neq 1, k && (4)\\
& Z_{kw} \le \min(E_{kw}, a_{kw}),\ \forall k, w && (5)\\
& \sum_{i=1}^{|D|} x_{ij} = 1,\ \forall j && (6)\\
& st_{jr} + p_{jr} = et_{jr}, \forall j, r && (7)\\
& et_{jr} \le \sum_{i=1}^{|D|} x_{ij} d_i,\ \forall j, r && (8)\\
& \sum_{j\in M_r} c_{jr} \le C_r, \text{if } st_{jr} \le t < et_{jr},\ \forall t, r && (9)\\
& x_{ij} \in \{0,1\},\ \forall i, j && (10)\\
& 0 \le E_{kw} \le |N|,\ \forall k, w && (11)\\
& st_{jr}, et_{jr} \in \mathbb{Z}^+ \cup \{0\},\ \forall j, r && (12)\\
& Z_{kw} \in \mathbb{Z}^+ \cup \{0\}, Z_{kw} \le |N|,\ \forall k, w && (13)
\end{aligned}
$$

Figure 3: The Global MIP Model for the Static Repair Shop Scheduling Problem.

The Due-Date Assignment Master Problem (DAMP): MIP Model To formulate the master problem as a MIP model, we use a binary variable x_{ij} for each $j \in J$ and $i \in D$ with the same meaning as in the global MIP model. A MIP formulation of DAMP is as follows:

$$
\begin{aligned}
\text{max. } & \text{Objective (1)} \\
\text{s.t. } & \text{Constraints (2) to (6), (10), (11), (13)} \\
& \sum_{j\in M_r} c_{jr} p_{jr} \le C_r \max_{j\in M_r,\ i\in D} (x_{ij} d_i),\ \forall r && (14)\\
& \text{MIP cuts} && (15)
\end{aligned}
$$

The master problem incorporates a number of the constraints in the global MIP model. It does not represent the start-times of jobs nor does it fully represent the capacity of the trades. As is common in Benders decomposition, the master problem includes a relaxation of the sub-problems (Constraints (14)) and Benders cuts (Constraints (15)).

The Sub-problem Relaxation Constraint (14) is the relaxation of the capacity of a trade, expressing a limit on the area of jobs that can be executed. The limit is defined using the area bounded by the capacity of the trade and the time interval $[0, M]$ where M is the maximum due date assigned to the jobs on the trade. This relaxation is due to Sadykov & Wolsey [31].

As there are a relatively small number of waves, we can tighten the relaxation by enforcing an analogous limit on multiple intervals: $[0, st_w]$ for each wave w, plus $[0, B]$. For each interval, the sum of the areas of the jobs whose assigned due date is less than or equal to the end-time of the interval must be less than the available area. Formally, the tighter relaxation replaces Constraint (14) with:

$$\sum_{j \in M_r,\ \sum_{i=1}^{|D|} x_{ij} d_i \leq st_w} c_{jr} p_{jr} \leq st_w C_r, \quad \forall r, w$$

The Benders Cuts Before defining the cut formally, we demonstrate the intuition with an example. Consider a due date set, $D = \{14, 17, 20, 100\}$, and, for a given trade with five jobs, the current master solution: $x_{21} = 1, x_{12} = 1, x_{43} = 1, x_{14} = 1,$ and $x_{15} = 1$. Job 1 is assigned to the second due date, 17, job 2 has the first due date, 14, and so on. If the current solution is infeasible due to the resource capacity of the trade, then we know that at least one of the jobs must have a later due date than it has in the current master solution. We can, therefore, constrain the sum of the consecutive x_{ij} up to and including the ones currently assigned to 1 to be one less than the number of jobs. In our example, the cut would be:

$$\begin{aligned}&(x_{11} + x_{21}) + (x_{12}) + \\ &(x_{13} + x_{23} + x_{33} + x_{43}) + (x_{14}) + (x_{15}) \leq 5 - 1\end{aligned}$$

These variables represent the possible due dates less than or equal to those currently assigned for all jobs. By constraining these variables to be at most one less than the number of jobs, at least one job must be assigned a later due date. Formally, assume that in iteration h, the solution of the DAMP assigns a set, Q, of due dates to the jobs on trade r. Assume further that there is no feasible solution on trade r with the assignments in Q. The cut after iteration h is:

$$\sum_{j \in M_r} \sum_{i \in I_{jh}^r} x_{ij} \leq |M_r| - 1, \quad \forall r \tag{16}$$

where $I_{jh}^r = \{i' | i' \leq i, \text{ and } x_{ij}^h = 1\}$ is the set of due dates indices less than or equal to the due date index assigned to job j and $|M_r|$ is the number of jobs on trade r. The validity of this cut is proved in Section 3.2.5.

The Due-Date Assignment Master Problem: CP Model We also formulate the MP using CP. Let d_j be the variable corresponding to the due date for job j.

max. Objective (1)

s.t. Constraints (3) to (5), (11), (13)

$$\text{GCC}([U_{kw}], [st_1, st_2, ..., st_{|W|}], [d_{j \in I_k}]),\ \forall k, w \tag{17}$$

$$\sum_{j \in M_r} c_{jr} p_{jr} \leq C_r \max_{j \in M_r}(d_j),\ \forall r \tag{18}$$

$$d_j \in \{st_1, st_2, ..., st_{|W|}, B\} \tag{19}$$

CP cuts

Constraint (17) defines a global cardinality constraint for each aircraft type. The global cardinality constraint enforces that the cardinality variables (U_{kw}) count the number of times that each value in the due date set appears over the due date variables (d_j). Constraint (18) guarantees that the sum of processing areas for the set of jobs on the same trade does not exceed the maximum available area.

The CP cut is based on the same reasoning as the MIP cuts. If J_r is the set of job indices on trade r and the assigned set of due dates is not a feasible solution for the SP, the cut will guarantee that in the next iteration at least one of the assigned due dates will have a greater value. Formally, the cut is:

$$\bigvee d_j > d_j^h, \quad \forall j \in J_r \tag{20}$$

where d_j^h is the due date assigned to job j in iteration h and $\bigvee$ represents the logical-or constraints.

Repair Scheduling Sub-problem Given a set of due dates assigned to the jobs on a trade, the goal of the repair scheduling sub-problem (RSSP) is to assign start-times to the jobs to satisfy the due dates and the trade capacity. The RSSP for each trade can be modeled using cumulative constraints [19]. We use a CP formulation:

$$\begin{aligned} &\text{cumulative}([t_j|d_j^h], [p_{jr}|d_j^h], [c_{jr}|d_j^h], C_r), \ \forall r \\ &0 \leq t_j \leq d_j^h - p_{jr}, \ \forall j, r \end{aligned} \tag{21}$$

where t is an array of variables such that t_j is the start-time of job j, d is an array of values such that d_j^h is the due date assigned to job j in master problem in iteration h. The parameters p_{jr}, c_{jr}, C_r are as defined above. Constraint (21) enforces the time windows: the job cannot be started later than $d_j^h - p_{jr}$.

3.2.3 A Dispatching Heuristic

Since the static problem is NP-hard, solving it to optimality may be prohibitive. We therefore investigate a heuristic approach, inspired by the Apparent Tardiness Cost (ATC) heuristic, a composite dispatching rule that is typically applied to single machine scheduling with the sum of weighted tardiness objective [28]. The heuristic computes a ranking index for each job and sorts the jobs in ascending order of the index. The heuristic then iterates through the jobs, scheduling each job at its earliest available time. The ranking index we use is as follows:

$$I_j = ST(k_j) \exp(-\frac{FN_j}{FC_j}), \quad \forall j$$

If we let k_j denote the type of aircraft j, then $ST(k_j)$ is the start-time of the first wave that requires an aircraft of type k_j. FN_j is the fraction of the total number of aircraft of type k_j required by the first wave that requires k_j, and FC_j is the maximum proportion of the capacity needed by job j over all its required trades, as follows.

$$FC_j = \max_r(\frac{p_{jr}c_{jr}}{ST(k_j)C_r}), \quad \forall r$$

Intuitively, the earlier the start-time of the first relevant wave, the higher proportion of aircraft required by that wave, and the lower the proportion of capacity required before the wave, then the sooner the job will be scheduled. The exponential function is used to place more weight on the start-time.

In preliminary experiments, we investigated a number of different dispatching heuristics. The chosen heuristic was clearly the best of these.

3.2.4 Hybrid Heuristic-Complete Approaches

A hybrid heuristic-complete approach in which the heuristic solution provides a lower bound for the maximization objective (Equation (1)) may improve the performance of the complete approaches. Therefore, a simple hybrid first runs the dispatching heuristic and then uses the objective value as the initial (incumbent) solution. Assume that the heuristic finds a solution, S, with $f(S)$ as the number of aircraft assigned to waves. Any of the complete approaches can now be modified by adding the following constraint:

$$\sum_{w=1}^{|W|} \sum_{k=1}^{|K|} Z_{kw} \geq f(S)$$

3.2.5 Theoretical Results

To guarantee the finite convergence of an LBBD model to a globally optimal solution, the Benders cuts must be valid and the master decision variables must have finite domains. A Benders cut is valid in a given iteration, h, if and only if (1) it excludes the current globally infeasible assignment in the master problem without (2) removing any globally optimal assignments [9]. The former guarantees the finite convergence and the later guarantees the optimality. As the decision variables in DAMP have a finite domain, it is sufficient to prove the satisfaction of the two conditions.

Theorem 2. *Cut* (16) *is valid.*

Proof. For condition (1), for the sub-problem in iteration h on trade r, by definition:

$$\sum_{j \in M_r} \sum_{i \in I_{jh}^r} x_{ij} = |M_r|$$

Consequently, cut (16) excludes the current assignment of master problem.

For condition (2), consider a global optimal solution S that does not satisfy cut (16) as generated in iteration h. As the cut states that at least one job must have a greater due date than it had in h, to violate the cut, all jobs in S must have equal or lesser due dates than they had in iteration h. However, because the sub-problem was infeasible in iteration h, any sub-problem with only equal or lesser due dates must also be infeasible as the available capacity on the trade is the same or less. Therefore, S must be infeasible and we contradict the assumption that S is globally optimal.

Therefore, the cut is valid. □

An analogous argument holds for cut (20).

3.3 Rescheduling Strategies

The dynamic repair shop problem over the long horizon can be viewed as static scheduling sub-problems over successive time periods. Let's assume that we start repairing the failed aircraft and assigning them to the waves based on the computed schedule at time 0. A wave might start while a repair is under way in the repair shop. If some aircraft fails the pre-flight check, it goes to the repair shop. Each failed aircraft requires a set of repair activities with known processing times and resource requirements. At the repair shop, some of the previously failed aircraft might be already repaired, some might be under repair, and others might be awaiting repair. Once the failed aircraft enter the repair shop, we have a new static repair scheduling sub-problem where the set of existing jobs (J) and the number of aircraft not in the repair shop (A_k) is updated. The set of existing jobs includes the recently failed aircraft and the previously failed aircraft whose repairs are still under way or are not yet started. The new static sub-problem has an added constraint, namely that the repairs currently under way cannot be disrupted.

We connect the static sub-problems using three different policies denoted as P_{ij} where i and j define the length of scheduling horizon and the frequency of rescheduling in number of waves, respectively. In all three policies, we schedule the repair activities, observe the aircraft failures, and respond to failures by rescheduling the repair activities.

The three policies discussed here are:

- P_{11}: This policy has a scheduling horizon with a length of one wave and reschedules after every wave. In Figure 4, we show that P_{11} schedules one wave at a time ($i = 1$) and reschedules after each wave ($j = 1$).
- P_{31}: This policy has a scheduling horizon with a length of three waves and reschedules after every wave. In contrast to P_{11}, for P_{31} (Figure 4), the scheduling horizon is three waves but rescheduling is still done after each wave.
- P_{33}: This policy has a scheduling horizon with a length of three waves and reschedules after every third wave (Figure 4).

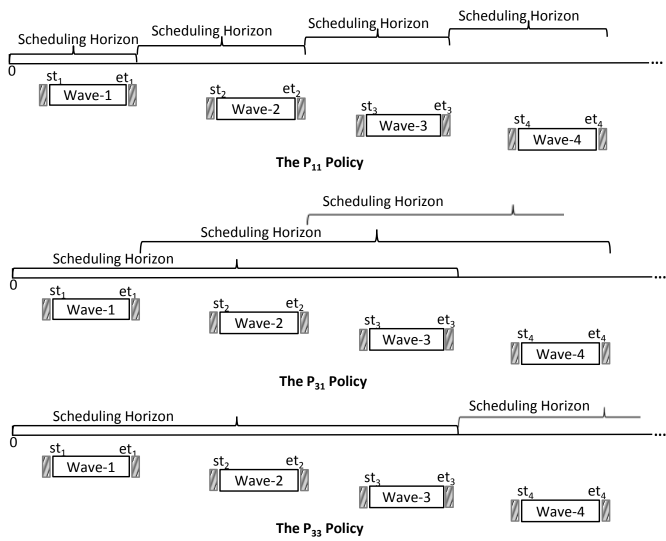

Figure 4: The rescheduling policies.

To model the dynamic events, we simulate the aircraft failures in pre- and post-flight checks. Every aircraft either passes or fails each check. If the aircraft fails, a new set of repair activities with known processing times and resource requirements is added to the repair shop. If the aircraft passes, it flies the wave. The repair is assumed to be minimal, returning the failure rate to what it was just before the failure. Minimal repair is one of the standard repair models in the maintenance literature [40]. To model aircraft deterioration, we increase the failure rate of the aircraft by γ percent each time it flies a wave. For example, consider that λ_n is the initial failure rate of the aircraft $n \in N$. Its failure rate after flying w waves without failure will be equal to $\lambda_n(1 + \gamma)^w$.

4 Computational Experiments

In this section, we present two separate empirical studies. The first study compares the scheduling techniques experimentally and presents insights into each algorithm's performance through a deeper analysis of the results.

The second study investigates the impact of using different scheduling techniques and rescheduling policies on the observed wave coverage.

4.1 Experimental Results on Scheduling Techniques

This sub-section describes the experiment comparing different solution techniques for scheduling the static repair shop.

4.1.1 Experimental Setup

The problem instances have 10 to 30 aircraft (in steps of 1), 3 or 4 trades, and 3 or 4 waves. Five instances for each combination of parameters are generated, resulting in 420 instances (21 total aircraft counts by 2 trade counts by 2 wave counts by 5 instances).

Aircraft The number of aircraft types is equal to $\frac{|N|}{5}$, where $|N|$ is the number of aircraft. The aircraft are randomly assigned to different types with uniform probability. The number of aircraft of type k is n_k. The failure rate for each aircraft is randomly chosen from the uniform distribution $[0, 0.1]$. The failure rate for aircraft of type k, $\bar{\lambda}_k$, is the mean failure rate over all aircraft of type k. The values of α and β, the coefficients in the probability of aircraft failure in pre- and post-flight checks as described in Section 3.2.1, are 1 and 3, respectively.

Waves The plane requirement for each aircraft type for each wave is randomly generated from the integer uniform distribution $[1, n_k]$. The length of each wave is drawn with uniform probability from $[3, 5]$. To make an instance loose enough to permit feasible solutions yet tight enough to be challenging, a lower bound on the B values is needed. The sum of the processing areas of the jobs in each trade, r, divided by the trade capacity is denoted by S_r. $LB = \max_r(S_r)$ is a lower bound on the time required to schedule the jobs and we set $B = 1.2 \times LB$. The end-time for each wave, et_w, is generated as $et_{|W|} = B - rand[0, 3]$ for the final wave, $|W|$, and $et_w = st_{w+1} - rand[0, 3]$ for $w < |W|$.

Trades The capacity limit for each trade is set at $C_r = 10$.

Repair Jobs Eighty percent of the aircraft are in the repair shop at the beginning, resulting in $|J| = 0.8|N|$ repair jobs. The jobs are randomly assigned to the trades with replacement such that the number of jobs per trade is equal to $|J|/2$. Each job requires at least one trade and some require more than one trade. The capacity of trade r used by job j, c_{jr}, is drawn from $[1, 10]$ while the processing time, p_{jr}, is drawn from $[r, 10r]$: jobs on trades with lower indices have shorter processing times than those on trades with higher indices.

All experiments were run with a 7200-second time limit. The MIP solver is IBM CPLEX 12.1 and the CP solver is IBM ILOG Solver/Scheduler 6.7.

4.1.2 Experimental Results

Figure 5 shows scatter-plots of run-times of the four complete approaches. Both axes are log-scale, and the points below the line $y = x$ indicate a lower run-time for the algorithm on the y-axis. The numbers in the boxes indicate the number of points below or above the line. Run-times are counted as equal if they differ less than 10%.

The graphs indicate a clear benefit for Benders-MIP over Benders-CP and MIP and for Benders-MIP-T over Benders-MIP while the performance comparison of Benders-CP and MIP is more even. Table 2 presents further data, sorted in ascending percentage of unsolved problems, for all algorithms.

Benders-MIP vs. MIP The Benders-MIP approach achieves a better run-time than MIP on 63% of the test problems and a worse run-time on 16%, achieving a lower mean run-time and solving a higher proportion of the problem instances. When the time horizon is short, the MIP approach is faster due to the tightness of the time-index model. With larger horizons and more jobs, the number of time-index variables grows, substantially reducing the performance.

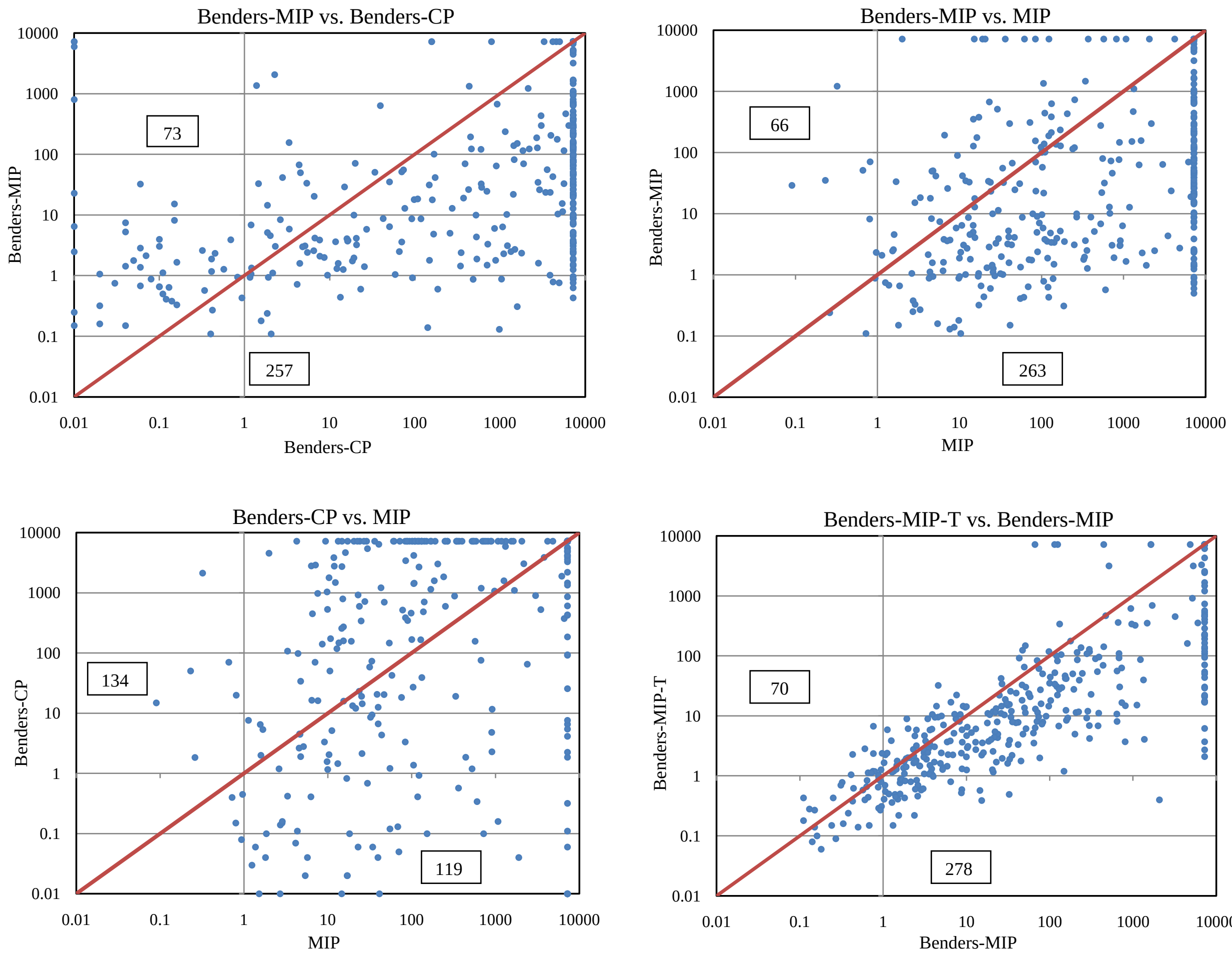

Figure 5: Run-times of the four complete models.

Benders-CP vs. MIP The Benders-CP approach does better than MIP in terms of run-time on 28% of the instances while performing worse on 32%. When the CP master problem can be solved within the time limit, Benders-CP is superior to the MIP. However, Table 2 favors MIP in terms of the overall performance.

Benders-MIP vs. Benders-CP The Benders-MIP approach achieves a better run-time than Benders-CP on just over 61% of the test problems, performing worse on about 17%. The branching heuristics for Benders-CP (smallest to largest due date) often lead to an initial feasible master solutions with tighter due dates than the initial master solution in Benders-MIP. The tighter, globally infeasible initial solution means that the CP-based model requires more than twice as many iterations to find a globally feasible solution. However, when the initial master solutions are identical, Benders-CP often finds and proves optimality faster than Benders-MIP. We believe this is due to the different forms of the cut, but further research is necessary to fully understand this point. These observations suggest, however, that more intelligent branching heuristics may improve Benders-CP, perhaps, to the point of being better than Benders-MIP.

Method	Mean Time (s)	# Iter.	MP Time (s) per Iter.	SP Time (s) per Iter.	% Unsolved
Benders-MIP-T-Hybrid	995.25	14.15	71.24	0.13	11.90
Benders-MIP-T	1155.34	14.97	78.01	0.13	14.52
Benders-MIP	1842.61	16.75	111.81	0.08	23.10
MIP-Hybrid	3338.24	-	-	-	42.86
MIP	3566.19	-	-	-	46.90
Benders-CP	4181.72	34.71	123.04	0.02	52.86
Dispatching Rule	0.01	-	-	-	65.71

Table 2: The mean run-time, the mean number of master problem iterations, the mean run-time spent on the master problem and the sub-problems per iteration, and the percentage of unsolved problems for all approaches.

Benders-MIP-T vs. Benders-MIP The tighter relaxation in Benders-MIP-T clearly speeds up LBBD: Benders-MIP-T has a better run-time than Benders-MIP on 66% of problems instances and worse on only 17%. The mean run-time, the percentage of unsolved problems, and the number of iterations decrease by 40%, 40%, and 11%, respectively. The solve time for the master problem decreases considerably compared to Benders-MIP, while the sub-problem run-time increases. This latter observation is because many sub-problems for Benders-MIP that can be quickly proved insoluble by the initial propagation of CP sub-problem model, violate the tighter relaxation in the Benders-MIP-T master problem. Therefore, in the tighter model, the sub-problem solver is not called on these "easy" sub-problems, increasing the mean run-time per sub-problem.

Incomplete and Hybrid Approaches The dispatching heuristic is fast, finding a feasible solution to all problems in an average of 0.096 seconds. However, it finds (but, of course, does not prove) an optimal solution in only 34% of the instances and Benders-MIP-T finds and proves optimality for these instances in 30.4 seconds on average. It seems that the heuristic can find the optimal solution only when the problem instance is relatively easy. The mean quality of the heuristic solution is 5% from optimal. In industries with expensive assets, such a reduction in solution quality can translate to costly under-use of a valuable resource (e.g., a fighter aircraft costs in the vicinity of 100 million dollars).

To evaluate the effect of combining the dispatching heuristic with the complete approaches, we examine using the hybrid heuristic-complete approach. A smaller feasible set is the direct consequence of defining a bound on the cost function. As the MIP model searches the feasible set, while LBBD methods explore the infeasible space, one intuition is that the MIP model should benefit more from using the heuristic solution. However, solving the master problem in LBBD requires searching in a relaxed feasible space and therefore the heuristic starting solution may also speed solving. Furthermore, as most of the LBBD run-time is taken up in solving the master problem, any such speed up is likely to impact the overall LBBD run-time.

Figure 6 and Table 2 show the effect of bounding the complete approaches with the dispatching heuristic solution. MIP-Hybrid achieves a better (worse) run-time than MIP on 35% (21%) of the problem instances. Benders-MIP-T-Hybrid has a better run-time than Benders-MIP-T on 60% of the instances and a worse run-time on 27%. The mean run-time decreases by 6% and 14% in MIP-Hybrid and Benders-MIP-T-Hybrid, respectively. As Table 2 indicates both hybrid methods are able to solve more problems to optimality. However, bootstrap paired-t tests [10] indicate that there is no significant difference in mean run-time at $p \leq 0.01$ for either hybrid.

Scalability Figure 7 shows our results as the number of aircraft per wave increases. We aggregate results by truncating $\frac{|N|}{|W|}$ and using the instances with three waves and both three and four trades. Note that each point represents 30 problem instances except $x = 3$ which has only 20 problems instances. We omitted $x = 10$ as we only have 10 problem instances for that point. The y-axis is log-scale.

The results show that the LBBD variations outperform the other techniques across all ratios. Interestingly, the

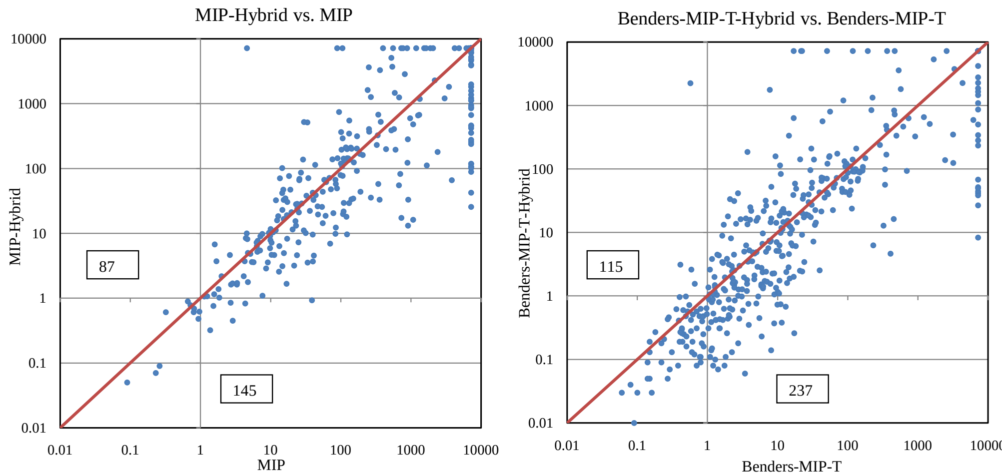

Figure 6: Run-time of heuristic-complete approaches vs. complete approaches.

LBBD hybrid approach that includes the dispatching heuristic shows increasing performance with more aircraft per wave. This result suggests that the lack of statistical significance noted above, is mainly due to the results on the instances with a small number of aircraft per wave.

Note that, the underperformance of the LBBD hybrid approach at $x = 3$ ratio is due to the out-lier of the instance being timed-out while it is solved in 17 seconds excluding the dispatching heuristic. The added constraint in the hybrid approach makes it substantially hard to prove the optimality in the time limit for this particular problem instance.

Summary The following observations on the performance of scheduling techniques are supported by this empirical study.

- The LBBD approach combining mixed integer programming and constraint programming outperforms the mixed integer programming model. Benders-MIP is over 7 times faster than MIP, on average, with a geometric mean time ratio of 7.62. The time ratio for a given instance is calculated as the MIP run-time divided by Benders-MIP run-time.

- A tighter relaxation speeds up LBBD. Benders-MIP-T has a run-time over 3 times faster than Benders-MIP. (The geometric mean time ratio is 3.11).

- A dispatching heuristic can provide the optimal solution for the easy problem instances. However, the mean percent relative error of heuristic is almost 5% overall, indicating that the dispatching rule by itself is not effective enough for industries with high equipment cost.

- Both MIP and LBBD benefit somewhat from a simple hybridization with the dispatching heuristic. While neither benefit is statistically significant overall, on problems with a larger number of aircraft per wave, the heuristic appears to have a strong positive impact on the LBBD approach.

4.2 Experimental Results on Rescheduling Strategies

This sub-section describes experiment investigating the impact of applying different scheduling techniques and rescheduling policies in a dynamic repair shop.

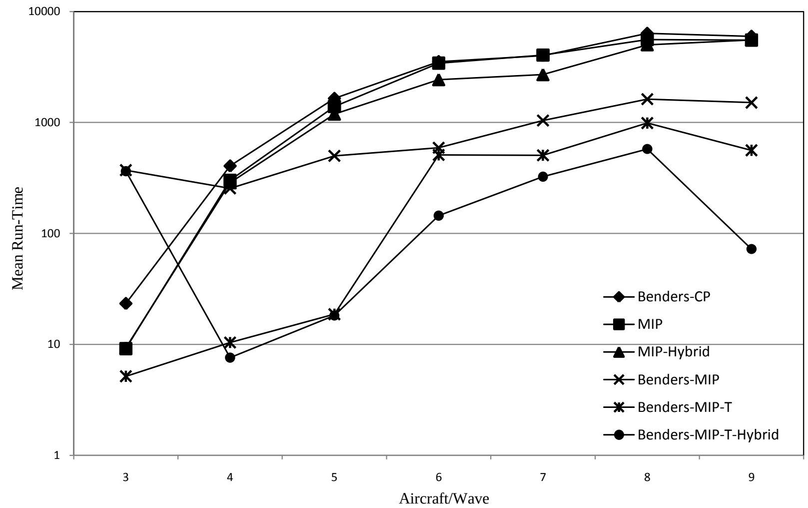

Figure 7: Mean run-time vs. number of aircraft per wave ($|W| = 3$).

4.2.1 Experimental Setup

For our problem instances, the number of aircraft, the number of trades, and the total number of waves are set to $\{10, 15, 20, 25, 30\}$, $\{4\}$, and $\{30\}$ respectively. Each combination has 5 instances for a total of 25 instances. Each instance is simulated 10 times. The parameters of the problem instances are generated as in Section 4.1 with the following changes:

Aircraft The initial failure rate for each aircraft is randomly chosen from the uniform distribution of $[0, 0.5]$. The failure rate of an aircraft is increased by $\gamma = 5\%$ each time it is used.

Repair Jobs Repair jobs that entered repair shop after time 0 are randomly assigned to the trades. The probability of assigning a job to each trade is considered as 0.5.

Waves The start time of each wave is generated as $st_1 = rand[\frac{B}{3}, \frac{B}{2}]$ for the first wave, and $st_w = et_{w-1} + rand(0, 40)$ for $1 < w \leq 30$. As mentioned earlier the total number of waves is 30. The value of B is calculated as in Section 4.1.

Dynamic events To simulate an aircraft failure, we generate a random value from the uniform distribution $[0, 1]$ for each aircraft at each check. If the random value is less than the aircraft's probability of failure, the aircraft fails; otherwise, it passes. The aircraft's probability of failure in pre- and post-flight checks are calculated using $(1 - e^{-\alpha\lambda_n})$ and $(1 - e^{-\beta\lambda_n})$, respectively. As mentioned earlier, λ_n is the failure rate of aircraft, $n \in N$, which increases by $\gamma = 5\%$ each time the aircraft flies a wave. Note that passing the pre-flight check of a wave does not necessarily mean that the aircraft flies the wave. If the number of available aircraft is more than the requirements, the aircraft that fly are randomly selected from those that passed the pre-flight check to meet the requirements.

We experiment with three techniques including MIP, Benders-MIP-T, and dispatching rule discussed in Section 3.2 and the time-limit to schedule the repair activities in each scheduling horizon is 600 seconds. We execute the best feasible schedule found before the time-limit if an algorithm times out. In the case that Benders-MIP-T times out, the schedule created by the dispatching heuristic is executed as Benders-MIP-T cannot create a feasible schedule when it times-out.

As in the static problem, the scheduling uses IBM CPLEX 12.1 and IBM ILOG Solver/Scheduler 6.7, and the simulation is coded in C++.

4.2.2 Experimental Results on Rescheduling Strategies

We discuss our results to answer a number of different questions.

Question 1 What is the impact of using a complete technique vs. the dispatching heuristic on the mean observed flight coverage?

We expect a complete technique to achieve higher flight coverage because it incorporates known information on uncertainty into scheduling the repair activities, while the dispatching heuristic does not have this property.

Figure 8 shows the mean observed coverage up to flight $w \in \{1, 2, ..., 28\}$ for different scheduling techniques over all three policies. The mean observed coverage up to flight w is $O_w = \frac{\sum_{i=1}^{w} \nu_i}{w}$, where ν_i denotes the coverage of flight i. As illustrated, Benders-MIP-T achieves about a 13% higher mean coverage over all flights than either MIP or the dispatching heuristic. The MIP algorithm also takes the probabilistic information into account when creating a repair schedule, but it results in flights with almost the same coverage as the dispatching heuristic alone. There are two reasons for this performance. First, MIP times out without finding a feasible solution on 47% of scheduling problems. The dispatching heuristic is then used to create the repair schedule. Second, as shown in Table 3, the mean percentage of available aircraft for the first flight using a given rescheduling policy over all problem instances in MIP and heuristic is less than Benders-MIP-T.

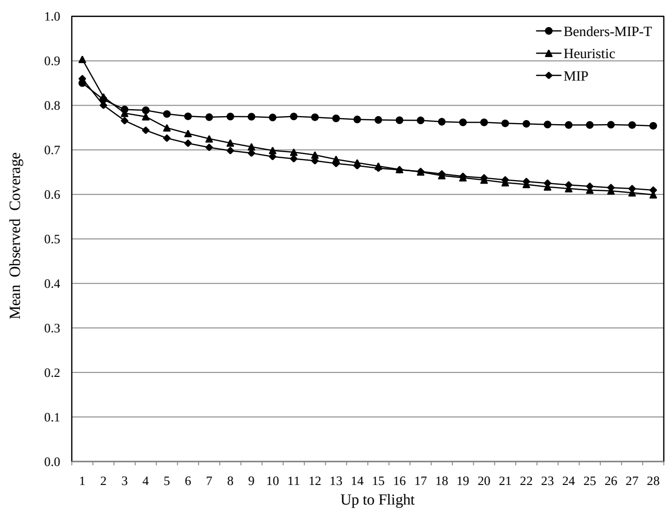

Figure 8: Mean observed coverage for different scheduling techniques.

Method	P_{11}	P_{31}	P_{33}
Benders-MIP-T	54	63	50
MIP	43	44	40
Heuristic	41	42	46

Table 3: The mean percentage of available aircraft for the first flight.

To investigate the impact of having more aircraft available earlier, we define $\rho(P_{ij}) = \frac{\sum_{k=1}^{S} \rho_k}{S}$ for each instance and each policy (25 instances and 3 policies), where ρ_k and S denote the percentage of aircraft available for the first flight in the k-th static sub-problem and S is the number of static sub-problems in P_{ij} policy. For example, in P_{31} policy, the first static sub-problem includes flights 1, 2, and 3. Then, ρ_1 is the number of aircraft available for flight 1 divided by the total number of aircraft in the system. The second static sub-problem schedules for flights 2, 3, and 4. Therefore, ρ_2 is equal to the number of aircraft available for flight 2 divided by the total number of aircraft. We follow the same procedure to find ρ_k for all 28 static sub-problems in P_{31} policy. We can find $\rho(P_{11})$ and $\rho(P_{33})$ using the same argument considering that the number of static sub-problems are 30 and 10, respectively.

Intuitively, making more aircraft available earlier increases the coverage, even though the number of pre-flight checks that aircraft go through increases in expectation. The quick adjustment of the schedule makes some of the failed planes again available before the start time of the next flight. Figure 9 illustrates the mean coverage difference, over all flights, between Benders-MIP-T and MIP for each problem instance where the x-axis represents the difference between the mean percentage of aircraft available for the first flight, for each instance and policy. As the difference between the percentage of aircraft available for the first flight increases in Figure 9, the difference between the mean flight coverage also increases. The high value of Pearson's correlation coefficient, $r = 0.798$, provides evidence for the relationship between schedules that repair aircraft earlier and achieving high coverage.

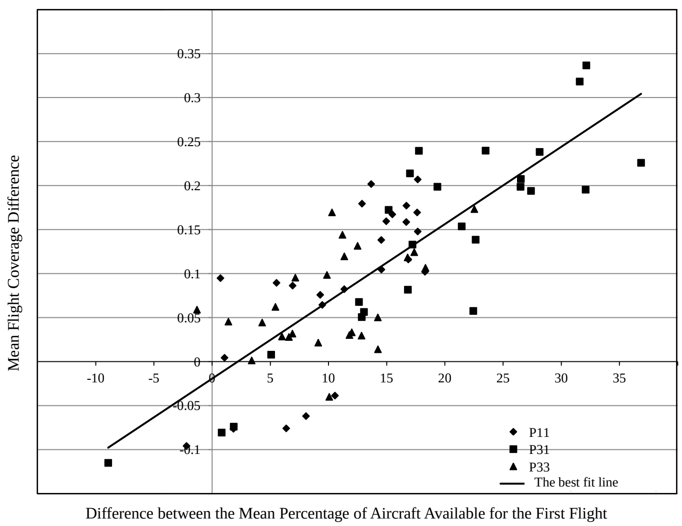

Figure 9: The mean coverage difference between Benders-MIP-T and MIP for each instance and each policy.

Question 2 Does P_{31} policy provide the flights with higher coverage than P_{33} and P_{11} policies using the optimization technique Benders-MIP-T?

We expect that P_{31} with Benders-MIP-T will produce better coverage because it schedules over a longer horizon and adjusts the schedule as soon as aircraft failures occur. Although P_{31} with the dispatching heuristic also responds quickly to the aircraft failures, it does not incorporate the length of the scheduling horizon into the ranking index for repair activities and always repairs the aircrafts for the earliest possible time.

Figure 10 shows the mean observed coverage for different policies using Benders-MIP-T up to flight $w \in \{1, 2, .., 28\}$. As illustrated, in the short-term (i.e., for the first three flights), the P_{11} policy outperforms, but the P_{31} policy then leads to consistently higher coverage over a longer term.

Figure 11 displays the cumulative percentage of the flights with a coverage less than or equal to ω for Benders-MIP-T and the dispatching heuristic, where ω denotes the values on the x-axis. The best performing approach will have a fewer flights with a low coverage and more flights with a high coverage. Therefore, its curve will be closer

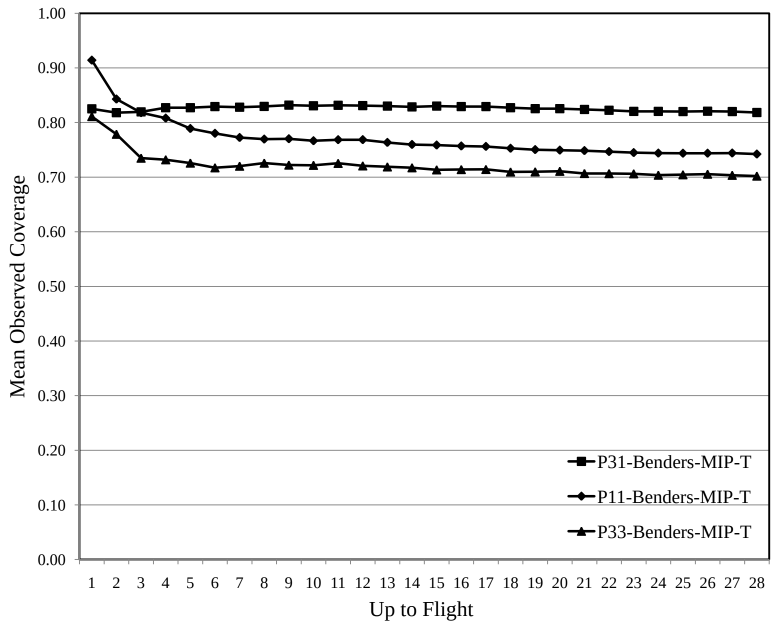

Figure 10: Mean observed coverage for different policies using Benders-MIP-T.

to the lower right-hand corner. As illustrated, in Benders-MIP-T, P_{31} performs better than the two other policies. In contrast, in the dispatching heuristic, P_{31} results in flights with the same coverage as P_{11}.

Question 3 Is a quicker reaction to the dynamic events more important than scheduling over a longer horizon?

The P_{31} policy changes the repair schedule after each flight and trades-off the coverage among three consecutive flights by scheduling over a longer horizon. In contrast, the P_{11} policy schedules for one flight and reacts after each flight while the P_{33} policy reasons over a longer term without a quick response to the dynamic events.

As already shown in Figure 10, the P_{31} policy results in a higher mean coverage. The superiority of policy P_{31} indicates that both features of quick response to the dynamic events and long-term reasoning contribute to the overall performance, but the question is which one contributes more.

When the failure rate is higher, the probability of aircraft being diagnosed as failed in pre- and post- flight checks is higher. Therefore, the arrival rate of the aircraft to the repair shop is higher and the previously constructed schedule is more likely not to be executed as is. In such a system, scheduling for a longer future is not useful as the future does not behave as expected. Our conjecture is when the failure rate is low, constructing a longer schedule and trading-off among the flights can contribute to the increase in coverage in the long term. However, when the failure rate is high, quicker reaction and adjusting to the real situation sooner has more contribution to the flight coverage.

To test our conjecture, consecutive flights are partitioned into buckets of size 3, and the mean coverage difference between P_{11} and P_{13} policies over each bucket is shown in Figure 12 in different uncertainty conditions where the failure rate of aircraft is randomly drawn from uniform distributions of $[0, 0.7]$, $[0, 0.5]$, $[0, 0.3]$ and $[0, 0.1]$, respectively. As shown, the observation supports our conjecture and the difference between P_{11} and P_{31} increases when the failure rates stochastically increases. Note that, the failure rates chosen from uniform distributions of $[0, a]$ are stochastically greater than those chosen from $[0, b]$, if $a > b$. The difference in coverage between two policies is 7.3%, 5.8%, 4.8% and 0.4% in Figure 12 as the failure rate stochastically decreases. Therefore, we can conclude that when the failure rate is stochastically high, early adjustment to the real situation contributes more to the increase in the flight coverage.

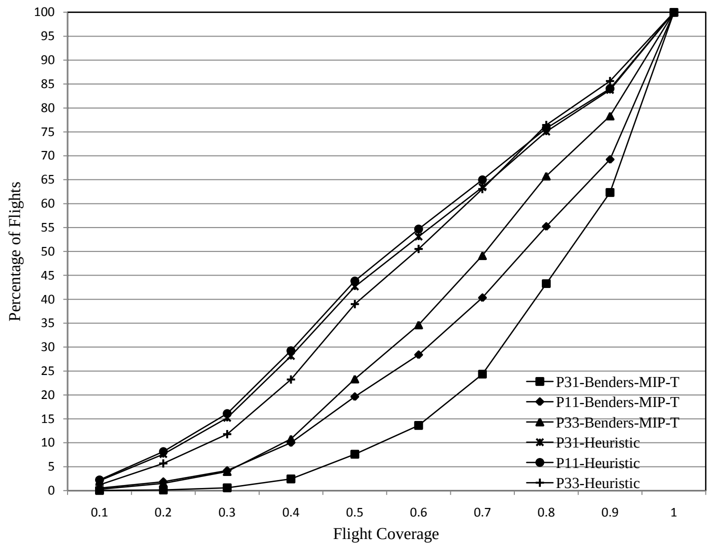

Figure 11: The percentage of flights with a coverage less than or equal to ω, where ω denotes the values on the x-axis.

Summary The following observations on how and when the scheduling and rescheduling should be done are supported by the second empirical study:

- Benders-MIP-T provides the flights with a higher coverage than the dispatching heuristic.
- The P_{31} policy results in flights with a higher coverage than P_{11} policy using Benders-MIP-T and with almost the same coverage using the dispatching heuristic.
- Scheduling over a longer horizon and quickly adjusting the schedule based on the real events are the features contributing to the increase in the observed coverage. Furthermore, it is shown that the quick reaction to the dynamic events is more important than the long scheduling horizon when the aircraft are more likely to fail in pre- and post-flight checks.

5 Discussion

The experimental results demonstrate that incorporating both probabilistic and execution time reasoning into the schedule of repair activities results in a higher system performance. We showed that the decomposition technique, LBBD, considering the known probabilistic information about uncertainty over a longer scheduling horizon and the rescheduling policy taking the advantage of truly up-to-date information more frequently increase the flight coverage on average by 13%, increasing the utilization of the valuable resources. Since the repair activities share the repair resources, trading-off the coverage among the flights through scheduling over a longer horizon increases the available aircraft in the long-term. If there were not common repair resources, we would expect that P_{31} and P_{11} perform the same. Furthermore, as the out-come of each check is uncertain, exploiting the recent revealed information provides the flights with higher coverage. If the probability of aircraft failure in pre- and post-flight checks were 0, we would expect the same performance of P_{31} and P_{33}.

We also showed that either scheduling over longer horizon or rescheduling more frequently after a certain point yields almost no gains. The performance of such approaches depends on how much and how fast the state of the

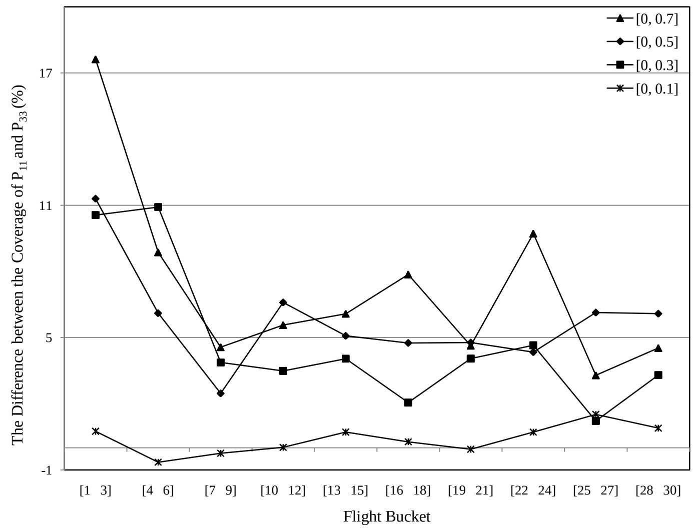

Figure 12: The difference between P_{11} and P_{33} in different uncertainty conditions.

system will change in the future. If the future does not behave as we expect, reasoning about a longer horizon does not help and clearly increases the computational effort. Furthermore, if the frequency of rescheduling exceeds the frequency of information change, the rescheduling strategy does not improve the schedule quality.

Taking a broader perspective, we decomposed our problem into stochastic long-term planning and deterministic short-term scheduling problems in DAMP and RSSP, respectively. In the long-term plan, we deal with the uncertainty using the known information on the probability of aircraft failures to trade-off the coverage between the flights. In the short-term schedule, we construct a feasible repair schedule to achieve the coverage as decided in the long-term plan. We then designed different rescheduling policies to investigate how the information revealed over time can be effectively used to adjust the repair schedule and change the long-term plan. We conclude that changing the long-term plan based on short-term information which cannot be completely incorporated in the long-term plan from the beginning significantly increases the utilization of the airplanes.

A typical hierarchical optimization approach does not deal with the interdependency between different levels of decision makings. However a communication technique as designed in this paper, capable of utilizing the short-term deterministic scheduling in the long-term stochastic planning is more likely to lead to a higher system performance. In the problem addressed here, the assignment of the aircraft to the flights and the scheduling of the repair activities in the repair shop represent the long-term and short-term reasoning, respectively. As a wide range of operational decisions can be viewed as integrated optimization problems, the pattern of the algorithms designed here might be applicable. The maintenance or inventory planning with job production scheduling is an example of the integrated operational problem where at the higher level of decision-making, the maintenance or inventory policy is determined, whereas at the lower level, the jobs are scheduled. If the maintenance or inventory policy leads to an infeasible scheduling problem at the lower level, then the higher level needs to be informed with this information so that the inventory or maintenance decisions are adjusted.

6 Conclusion

In this paper, we address the problem of scheduling a dynamic repair shop in the context of aircraft fleet management. The goal is to maximize the flight coverage over a long-term by considering the repair capacity and the aircraft failures. The number of failed aircraft dynamically changes because of aircraft breakdowns. Our proposed solution solves the dynamic problem as successive static scheduling problems over shorter time periods. Several scheduling algorithms and different rescheduling policies are proposed to schedule the repair activities online with dynamic reaction to the aircraft failures. The length of the scheduling horizon and the frequency of rescheduling are the features defining our three policies.

The computational results show that an optimization approach using logic-based Benders decomposition, scheduling over a longer horizon, incorporating the known information on aircraft failures, and adjusting the repair schedule as soon as new jobs enter the repair shop yield higher mean coverage. The results also provide evidence that when the future is more uncertain, a quicker reaction to the aircraft failures is more important than scheduling over a longer horizon.

Establishing a formal framework to exactly determine how long into the future we should plan ahead and how quickly we should change our schedule based on the new information in a dynamic system is a very interesting direction for the future work. The answer of these two questions seems to be highly dependent on the arrival rate of the new information and the impact of the new information on the overall system performance. How to quantify these two values can be the starting point to provide solid responses to these two questions.

References

[1] M. Aramon Bajestani and J. C. Beck. Scheduling a dynamic aircraft repair shop. In *Proceedings of the Scheduling and Planning Applications Workshop (SPARK)*, 2011.

[2] M. Aramon Bajestani and J. C. Beck. Scheduling an aircraft repair shop. In *Proceedings of the twenty-first International Conference on Automated Planning and Scheduling (ICAPS'11)*, pages 10–17, 2011.

[3] H. Aytug, M Lawley, K. McKay, S. Mohan, and R. Uzsoy. Executing production schedules in the face of uncertainties: A review and future directions. *European Journal of Operational Research*, 161:86–110, 2005.

[4] J. C. Beck. Checking-up on branch-and-check. In *Proceedings of the Sixteenth International Conference on Principles and Practice of Constraint Programming (CP2010)*, pages 84–98, 2010.

[5] J. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[6] J. Bidot, T. Vidal, P. Laborie, and J.C. Beck. A theoretical and practical framework for scheduling in a stochastic environment. *Journal of Scheduling*, 12:315–344, 2009.

[7] G. Budai, D. Huisman, and R. Dekker. Scheduling preventive railway maintenance activities. *Journal of the Operational Research Society*, 57:1035–1044, 2006.

[8] A. Caprara and M. Monaci. On the two-dimensional knapsack problem. *Operations Research Letters*, 32:5–14, 2004.

[9] Y. Chu and Q. Xia. Generating Benders cuts for a general class of integer programming problems. In *Proceedings of the First International Conference on the Integration of AI and OR Techniques in Constraint Programming (CPAIOR04)*, pages 127–136, 2004.

[10] P. R. Cohen. *Empirical Methods for Artificial Intelligence*. The MIT Press, Cambridge, Mass., 1995.

[11] P. Cowling and M. Johansson. Using real time information for effective dynamic scheduling. *European Journal of Operational Research*, 139:230–244, 2002.

[12] C. Derman, Lieberman G. J., and S. M. Ross. On the optimal assignment of servers and a repairman. *Journal of Applied Probabilities*, 19:577–581, 1980.

[13] M. M. Fazel-Zarandi and J. C. Beck. Using logic-based benders decomposition to solve the capacity- and distance-constrained plant location problem. *INFORMS Journal on Computing*, 2011. in press.

[14] D. Frost and R. Dechter. Optimizing with constraints: a case study in scheduling maintenance of electric power units. *Lecture Notes in Computer Science*, 1520:469–488, 1998.

[15] A. M. Geoffrion and G. W. Graves. Multicommodity distribution system design by benders decomposition. *Management Science*, 20(5):822–844, 1974.

[16] A. Grigoriev, J. van de Klundert, and F. C. R. Spieksma. Modeling and solving the periodic maintenance problem. *European Journal of Operational Research*, 172:783–797, 2006.

[17] A. Haghani and Y. Shafahi. Bus maintenance systems and maintenance scheduling: model formulations and solutions. *Transportation Research Part A*, 36:453–482, 2002.

[18] L. Haque and Armstrong M. J. A survey of the machine interference problem. *European Journal of Operational Research*, 179:469–482, 2007.

[19] J. Hooker. A hybrid method for planning and scheduling. *Constraints*, 10:385–401, 2005.

[20] J. Hooker. Planning and scheduling by logic-based benders decomposition. *Operations Research*, 55:588–602, 2007.

[21] J. Hooker and G. Ottosson. Logic-based Benders decomposition. *Mathematical Programming*, 96:33–60, 2003.

[22] J. Hooker and H. Yan. Logic circuit verification by Benders decomposition. In V. Saraswat and P. Van Hentenryck, editors, *Principles and Practice of Constraint Programming: The Newport Papers*, chapter 15, pages 267–288. MIT Press, 1995.

[23] Seyed M. R. Iravani, V. Krishnamurthy, and G.H. Chao. Optimal server scheduling in nonpreemptive finite-population queueing systems. *Queueing System*, 55:95–105, 2007.

[24] S. Q. Liu, H. L. Ong, and K. M. Ng. Metaheuristics for minimizing the makespan of the dynamic shop scheduling problem. *Advances in Engineering Software*, 36:199–205, 2005.

[25] R. O'Donovan, R. Uzsoy, and K. N. McKay. Predictable scheduling of a single machine with breakdowns and sensitive jobs. *International Journal of Production Research*, 37:4217–4233, 1999.

[26] I. M. Ovacik and R. Uzsoy. Rolling horizon algorithms for a single-machine dynamic scheduling problem with sequence-dependent setup times. *International Journal of Production Research*, 32:1243–1263, 1994.

[27] I. M. Ovacik and R. Uzsoy. Rolling horizon procedures for dynamic parallel machine scheduling with sequence-dependent setup times. *International Journal of Production Research*, 33:3173–3192, 1995.

[28] M. L. Pinedo. *Planning and Scheduling in Manufacturing and Services*. Springer Series in Operations Research. Springer, 2005.

[29] M. Queyranne and A. Schulz. Polyhedral approaches to machine scheduling problems. Technical Report 408/1994, Department of Mathematics, Technische Universitat Berlin, Germany, 1994. revised in 1996.

[30] I. Sabuncuoglu and M. Bayiz. Analysis of reactive scheduling problems in a job shop environment. *European Journal of Operational Research*, 126:567–586, 2000.

[31] R. Sadykov and L. A. Wolsey. Integer programming and constraint programming in solving a multimachine assignment scheduling problem with deadlines and release dates. *INFORMS Journal on Computing*, 18:209–217, 2006.

[32] N. Safaei, D. Banjevic, and A. K. S. Jardine. Workforce constrained maintenance scheduling for aircraft fleet: A case study. In *Proceedings of Sixteenth ISSAT International Conference on Reliability and Quality in Design*, pages 291–297, 2010.

[33] N. Safaei, D. Banjevic, and A. K. S. Jardine. Workforce-constrained maintenance scheduling for military aircraft fleet: a case study. *Annals of Operations Research*, 186:295–316, 2011.

[34] K. E. Stecke. *Machine Interference: Assignment of Machines to Operators. Handbook of Industrial Engineering*. John Wiley & Sons, 2 edition, 1992.

[35] D. Terekhov, J. C. Beck, and K. N. Brown. A constraint programming approach for solving a queueing design and control problem. *INFORMS Journal on Computing*, 21(4):546–561, 2009.

[36] G. E. Vieira, J. W. Hermann, and E. Lin. Predicting the performance of rescheduling strategies for parallel machine systems. *Journal of Manufacturing Systems*, 19:256–266, 2000.

[37] G.E. Vieira, J.W. Hermann, and E. Lin. Rescheduling manufacturing systems: a framework of strategies, policies and methods. *Journal of Scheduling*, 6:36–92, 2003.

[38] V. Vinod and R. Sridharan. Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. *International Journal of Production Economics*, 129:127–146, 2011.

[39] H. M. Wagner, R. J. Giglio, and R. G. Glaser. Preventive maintenance scheduling by mathematical programming. *Management Science*, 10(2):316–334, 1964.

[40] H. Wang. A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research*, 139:469–489, 2002.