Scheduling an Aircraft Repair Shop

Paper # 13

Abstract

We address a scheduling problem in the context of aircraft maintenance where the goal is to maximize the expected coverage of a flight program. Missions within a flight program are undertaken even if they do not have a full complement of aircraft and coverage is the extent to which the aircraft requirements are met. The assignment of aircraft to flights must consider the probability of aircraft failure and capacity of the repair shop that maintains the aircraft. Therefore, we must both assign aircraft to flight programs and schedule the repair shop in order to meet the assignments. We propose a dispatching heuristic algorithm; three complete approaches based on mixed integer programming, constraint programming, and logic-based Benders decomposition; and a hybrid heuristic-complete approach. Experiments demonstrate that the logic-based Benders variation combining mixed integer programming and constraint programming outperforms the other approaches, that the dispatching heuristic can feasibly schedule the repair shop in a very short time, and that using the dispatching solution as a bound marginally improves the complete approaches.

Introduction

For a growing number of industries, especially those possessing expensive assets, repairing a failed asset is significantly more economical than replacing it. Furthermore, there is economic pressure to reduce the number of such expensive assets so as to not waste their capacity: companies often cannot afford an inventory of back-up parts. Under such conditions, high-quality schedules which integrate maintenance activities and "regular" activities can have a substantial effect on the performance of the overall system. For scheduling research, such problems raise the dual challenges of dealing with the uncertainty of breakdown and with simultaneously reasoning about maintenance and traditional scheduling metrics like makespan, through-put, and on-time performance.

Following the case study presented in Safaei et al. (Safaei, Banjevic, and Jardine 2010), we study the scheduling of an aircraft maintenance facility. In the problem, there exists a flight program consisting of a number of "waves" (flights or missions), each with a requirement for a specific number of aircraft of different types. At the beginning of the time horizon, an aircraft is either ready for a pre-flight check or is waiting to be repaired in the repair shop. The high-level goal is to assign aircraft to waves under the constraints that:

- A given aircraft can participate in multiple waves, provided the waves do not overlap in time and the aircraft does not breakdown.
- Aircraft are subject to breakdown which can be detected during systematic pre- and post-flight checks. We assume in this paper that once an aircraft fails, it leaves the system (i.e., to be repaired during the next scheduling horizon).
- Aircraft in the repair shop must be repaired before being sent on a wave.
- The repair shop has limited capacity.

Given the possibility of failure and the repair shop capacity, a solution will determine an assignment of aircraft to waves and a schedule of repair jobs which will maximize the expected wave coverage, considering the known failure probabilities and the characteristics of the repair activities such as processing times and resource requirements. The flow of an aircraft during a single scheduling horizon is illustrated in Figure 1.

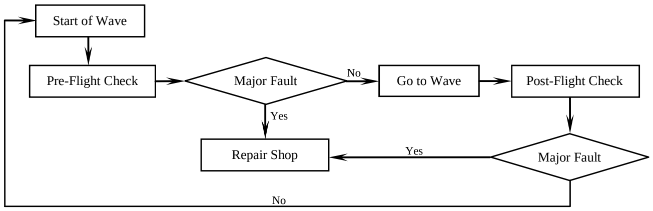

Figure 1: A Flow Chart Representing the Aircraft Flow among Waves, Checks, and the Repair Shop.

We explore several techniques to solve this problem: a mixed integer programming (MIP) model; three logic-based Benders decomposition (LBBD) models using either MIP or constraint programming (CP) for the due date assignment and CP for the maintenance scheduling; 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).

Our empirical studies indicate that the LBBD variation that combines mixed-integer programming and constraint programming outperforms the other complete approaches. The dispatching heuristic can provide the repair shop with a feasible schedule in a very short time, however the quality of the solution is likely to be insufficient. Finally, the integration of the dispatching heuristic and complete approaches results in a small reduction in the mean time to optimally schedule the repair shop.

The contributions of this paper are:

- Developing a logic-based Benders decomposition for scheduling an aircraft repair shop.
- Designing a dispatching heuristic algorithm guaranteeing a feasible schedule for the repair shop in a short time.
- Estimating aircraft availability and therefore wave coverage through incorporating known statistics on their probability of failure.

The following section defines logic-based Benders decomposition and our problem. We then describe the proposed solution approaches and subsequently present our experiments and results before presenting related work on maintenance scheduling, concluding, and discussing directions for the future work.

Background

In this section, the literature on logic-based Benders decomposition is reviewed and the formal definition of the problem is given.

Logic-based Benders Decomposition

After logic-based Benders decomposition (LBBD) was introduced in the context of circuit verification (Hooker and Yan 1995), it was further developed and applied to integer programming by Hooker & Ottosson (Hooker and Ottosson 2003). The LBBD approach 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 (Hooker 2005) 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 (Beck 2010).

To our knowledge, this is the first work using logic-based Benders decomposition for a maintenance scheduling problem.

Problem Definition

The problem is to assign aircraft to waves to maximize expected coverage while at the same time creating a feasible maintenance schedule. The problem definition is due to Safaei et al. (Safaei, Banjevic, and Jardine 2010).

Figure 2 is a snapshot of the problem at time 0, where circles represent aircraft. Three waves and their corresponding pre- and post-flight checks are already scheduled. A number of aircraft (three in the diagram) are ready for the pre-flight check while others are currently in the repair shop awaiting maintenance before they can proceed to a pre-flight check.

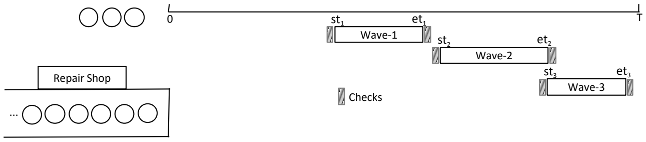

Figure 2: Snapshot of the Problem at Time 0.

We adopt the following notation to represent the problem.

- N is the set of aircraft. Failure is only detected during a check and we assume that a check will always correctly assess the status of an aircraft.
- K is the set of aircraft types. For each aircraft type $k \in K$, there are A_k available aircraft at the beginning of the time horizon and λ_k is the mean failure rate over all aircraft of type k.
- W is the set of waves and D is the set of wave start-times plus a big value, B. Each wave, $w \in W$ has a start-time, $st_w \in D$, and an end-time et_w. Each wave requires a_{kw} aircraft of type k.
- R is the set of repair resources (called "trades"). The maximum capacity of trade $r \in R$ is C_r.
- J is the set of jobs (i.e., aircraft to be repaired). Each job is associated with a specific aircraft and I_k denotes the set of repair jobs for aircraft type k. M_r is the set of jobs requiring trade r. Each job might require more than one trade to be processed. 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.
- T is the scheduling horizon (i.e., maximum time).

To find the probability of failure for 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 possibilities: 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 possibilities. 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 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.

To break this loop and avoid the growth of the state space, we distinguish aircraft based on their type and use a recursive equation (see Eqn (4)) to estimate aircraft availability and therefore wave coverage. The failure rate, λ_k, is used to calculate the probability of failure associated with each aircraft type during pre- and post-flight check, respectively: $\xi_k^{pre} = (1 - e^{-\alpha\lambda_k})$ and $\xi_k^{post} = (1 - e^{-\beta\lambda_k})$, where we require $\alpha < \beta$ to reflect deterioration of the aircraft through use. It is assumed that the probability of failure in pre- and post-flight check is a univariate function of failure rate as given above. Note that Safaei et al. used the same probability of failure for all aircraft type both in pre- and post-flight checks in their problem definition (Safaei, Banjevic, and Jardine 2010).

Solution Approaches

In this section, the details of different solution techniques are provided.

Mixed Integer Programming

We propose a novel mixed integer programming (MIP) model based on the common *time-indexed* approach (Queyranne and Schulz 1994). This model is different from and significantly faster than that of Safaei et al. The variables are defined in Table 1. The MIP model is defined in Figure 3.

Var.	Definition
Z_{kw}	The number of aircraft of type k that 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 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 (Eqn (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. Equation (7) calculates the end-time of the jobs. The end-time of each job is guaranteed to be less than the assigned due date by constraint (8). Constraint (9) enforces the capacity limit of each trade, where t denotes the discrete time during which job j is under way.

$$\text{max.} \sum_{w=1}^{W} \sum_{k=1}^{K} Z_{kw} \tag{1}$$

$$\text{s.t. } U_{kw} = \sum_{j \in I_k} x_{ij}, \text{ if } d_i = st_w \tag{2}$$

$$E_{k1} = (A_k + U_{k1})(1 - \xi_k^{pre}), \; \forall k \tag{3}$$

$$\begin{aligned} E_{kw} &= (E_{k(w-1)} - Z_{k(w-1)} + U_{kw})(1 - \xi_k^{pre}) + \\ &\sum_{v=1}^{w-1} Z_{kv}(1 - \xi_k^{post})(1 - \xi_k^{pre}), \\ &\text{if } st_{w-1} < et_v \le st_w, \forall w \neq 1, k \end{aligned} \tag{4}$$

$$Z_{kw} \le \min(E_{kw}, a_{kw}), \; \forall k, w \tag{5}$$

$$\sum_{i=1}^{|D|} x_{ij} = 1, \; \forall j \tag{6}$$

$$st_{jr} + p_{jr} = et_{jr}, \forall j, r \tag{7}$$

$$et_{jr} \le \sum_{i=1}^{|D|} x_{ij} d_i, \; \forall j, r \tag{8}$$

$$\sum_{j \in M_r} c_{jr} \le C_r, \text{if } st_{jr} \le t < et_{jr}, \; \forall t, r \tag{9}$$

$$x_{ij} \in \{0, 1\}, \; \forall i, j \tag{10}$$

$$0 \le E_{kw} \le N, \; \forall k, w \tag{11}$$

$$st_{jr}, et_{jr} \in \mathbb{Z}^+ \cup \{0\}, \; \forall j, r \tag{12}$$

$$Z_{kw} \in \mathbb{Z}^+ \cup \{0\}, Z_{kw} \le N, \; \forall k, w \tag{13}$$

Figure 3: The Global MIP Model for the Aircraft Maintenance Scheduling Problem.

Note that Z_{kw}, the number of aircraft of type k that 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}).

We refer to this model in our experimental results as *MIP*.

Logic-based Benders Decomposition

As the 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 subproblems check the feasibility of 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 and the latter has a tighter sub-problem relaxation; and *Benders-CP*, with a constraint programming-based master problem. All models use CP for the scheduling sub-problems.

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:

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

The master problem incorporates a number of the constraint in the global MIP model. However, 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 “Benders cuts” and a relaxation of the sub-problems.

The Benders Cuts Before explaining the cut formally, we demonstrate the intuition with an example. Consider that the due date set is equal to $D = \{14, 17, 20, 100\}$ and, for a given trade with five jobs, the current master solution is: $x_{21} = 1, x_{12} = 1, x_{43} = 1, x_{14} = 1$, 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 looser due date. We can constrain the sum of all the consecutive x_{ij} up to and including the ones assigned to 1 to be one less than the number of jobs: at least one of the x_{ij}'s not in the sum must be equal to 1. In our example, the cut would be:

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

Formally, assume that in iteration h, the solution of the DAMP assigns a set, Q, of due dates to the jobs in trade r. Assume further that there is no feasible solution on trade r with the assignments in Q. The cut after iteration h is:

$$\sum_{j \in M_r} \sum_{i \in I^r_{jh}} x_{ij} \le |M_r| - 1, \quad \forall r \qquad (15)$$

where $I^r_{jh} = \{i' | i' \le i, \text{ and } x^h_{ij} = 1\}$ 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 correctness of this cut is proved in the following section.

The Sub-problem Relaxation Constraint (14) is the relaxation of the capacity of a trade, expressing a limit on the area of jobs processed by the same trade. 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 (Sadykov and Wolsey 2006).

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 $[0, B]$ where B is the end of the time horizon. For each interval, the limit on the area is defined for the jobs whose assigned due date is less than or equal to the end-time of the interval. Formally, the tighter relaxation replaces Constraint (14) with:

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

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.

$$\begin{aligned} &\text{max. Objective (1)} \\ &\quad \text{s.t. Constraints (3) to (5), (11), (13)} \\ &\qquad \text{GCC}([U_{kw}], [st_1, st_2, ..., st_W], [d_{j \in I_k}]),\ \forall k, w \qquad (16) \\ &\qquad \sum_{j \in M_r} c_{jr} p_{jr} \le C_r \max_{j \in M_r} (d_j),\ \forall r \qquad (17) \\ &\qquad d_j \in \{st_1, st_2, ..., st_W, B\} \qquad (18) \\ &\qquad \text{CP cuts} \end{aligned}$$

Constraint (16) 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 (17) guarantees that the sum of processing areas for the set of jobs on the same trade does not exceed the maximum available area.

The form of the cut (19) for the CP model is based on the same reasoning as with the MIP model. 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 will be presented as follows:

$$\bigvee d_j > d^h_j, \quad \forall j \in J_r \qquad (19)$$

where d^h_j is the due date assigned to job j in iteration h.

Job Scheduling Sub-Problem Given a set of due dates assigned to the jobs on each trade, the goal of the job scheduling sub-problem (JSSP) is to assign start-times to the jobs on a trade while satisfying the due dates and ensuring that the resource consumption at any given time never exceeds the trade capacity. The JSSP for each trade can be modeled using cumulative constraints (Hooker 2005). We use a CP formulation:

$$\begin{aligned} &\text{Cumulative}([t_j | d^h_j], [p_{jr}], [c_{jr} | d^h_j], C_r),\ \forall r \\ &0 \le t_j \le d^h_j - p_{jr},\ \forall j, r \qquad (20) \end{aligned}$$

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 variables p_{jr}, c_{jr}, C_r are as defined above. Constraint (20) enforces the time windows: the job cannot be started later than $d_j^h - p_{jr}$.

A Dispatching Heuristic

Since the problem modeled in this paper is NP-hard, solving it to optimality may be prohibitive. We therefore investigate a heuristic approach, inspired by the Apparent Tardiness Cost (ATC) heuristic (Pinedo 2005) that guarantees a feasible schedule in a short time. ATC is a composite dispatching rule combining two simpler dispatching heuristics and is typically applied to a single machine with the sum of weighted tardiness objective (Pinedo 2005). 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 higher the job will be ranked in the list. The exponential function is used to place more weight on the start-time.

Hybrid Heuristic-Complete Approaches

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

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

Theoretical Results

To guarantee the finite convergence of an LBBD model to a globally optimal solution, the Benders cut 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 (Chu and Xia 2004). The former guarantees the finite convergence and the later guarantees the optimality. As the decision variables in DAMP have a finite domain, showing the satisfaction of two conditions proves the validity of the proposed Benders cut.

Theorem 1. *Cut* (15) *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 (15) excludes the current assignment of master problem.

For condition (2), consider a global optimal solution S that does not satisfy cut (15) 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 area 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 (19) in the CP model of DAMP.

Experimental Results

The next sub-section describes the problem instances and the experimental details. We then compare logic-based Benders decomposition, mixed integer programming, and heuristic algorithm experimentally and present insights into each algorithm's performance through a deeper analysis of the results.

Experimental Setup

For our problem instances, the number of aircraft, the number of trades, the number of waves are chosen from the uniform integer distributions $[10, 30]$, $[3, 4]$, and $[3, 4]$, respectively.

Aircraft The number of aircraft types is equal to $\frac{|A|}{5}$, where $|A|$ is the number of aircraft. The aircraft are randomly assigned to different types and the failure rate for each aircraft is randomly chosen from the uniform distribution of $[0, 0.1]$. The failure rate for aircraft of type k, λ_k, is the mean failure rate over all aircraft of type k. The values of α and β are 1 and 3, respectively.

Waves The plane requirement for each wave is randomly generated considering the number of aircraft of each type. The length of each wave was 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 time horizon 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 the scheduling horizon to $T = 1.2 \times LB$. The end-time for each wave, et_w, is generated as $et_W = T - rand[0,3]$ for the final wave, W, and $et_w = et_{w+1} - rand[0,3]$ for $w < W$.

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

Repair Jobs Eighty percent of the aircraft in the repair shop at the beginning of time horizon, resulting in $J = 0.8|A|$ 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 with three or four trades, some jobs 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 $[r, 10r]$. Therefore, jobs on trades with lower indices have shorter expected 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.

Computational Results

Figures 4 show 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 not equal only if one is at least 10% less than the other.

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 for all algorithm, sorted in ascending percentage of problems not solved to optimality: the mean run-time, the mean number of iterations, the mean time spent on the master problem and on the sub-problems per iteration, and the percentage of unsolved problems.

Method	Mean Time	# Iter.	MP Time per Iter.	SP Time 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 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 time-index model. The shorter the time horizon, the smaller the MIP model because it has variables for every time-point and job.

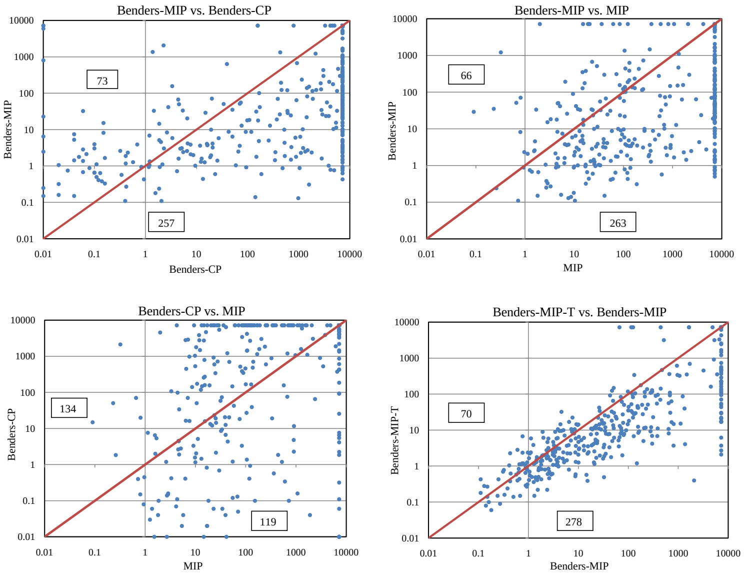

Figure 4: Run-times of the Four Complete Models

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%. We observe that 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 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 substantially improve Benders-CP, perhaps, to the point of being better than Benders-MIP.

Benders-MIP-T vs. Benders-MIP The tighter relaxation in Benders-MIP-T clearly speeds up LBBD as 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 over Benders-MIP by 40%,

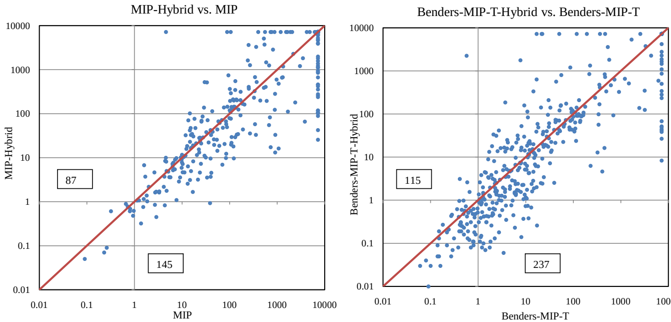

Figure 5: Run-time of Heuristic-Complete Approaches vs. Complete Approaches.

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 can be understood from the fact that the tighter relaxation in the master problem results in the fewer “easy” sub-problems. 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-instances increasing the mean run-time per sub-problem.

Incomplete and Hybrid Approaches The dispatching heuristic is very 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. The time for Benders-MIP-T to find and prove optimality for these problem instances has a mean of 30.4 seconds (with the two longest running instances in this set taking 2430 seconds and 361 seconds each). Therefore, it seems that the heuristic can find the optimal solution only when the problem instance is relatively easy. The mean percent relative error and standard deviation are 4.98 and 6.33, respectively. In the context studied here, industries with highly expensive assets, an almost five percent reduction wave coverage can result in significant lost revenue.

Finally, to evaluate the effect of bounding the cost function in the complete approaches by the heuristic, the performance of two complete approaches, Benders-MIP-T and MIP, are studied 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 as a lower bound on the objective. However, solving the master problem in LBBD requires searching in a relaxed feasible space and therefore the lower bound may 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.

Figures 5 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 (Cohen 1995) indicate that there is no significant difference in mean run-time at $p \leq 0.01$ for either hybrid.

Summary The following observations are supported by the empirical results.

- The LBBD approach combining mixed integer programming and constraint programming outperforms the mixed integer programming model. Benders-MIP is over 300 times faster than MIP, on average, with a mean time ratio of 356.69. 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 50 times faster than Benders-MIP. (The mean time ratio is 51.02).
- 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 bounding the cost function using the dispatching heuristic. However, neither benefit is statistically significant.

Related Work

While there is a significant literature on maintenance scheduling for a deteriorating system, the majority of the articles assume an infinite horizon and provide tactical-level results, finding, for example, long-term maintenance frequencies for groups of related machines or control-limits for carrying out maintenance depending on the state of components (Nicolai and Dekker 2008).

There are a number of pieces of work that examine operational level maintenance scheduling. Grigoriev et al. (Grigoriev, van de Klundert, and Spieksma 2006) took a deterministic approach to deal with the problem of cyclically scheduling maintenance activities assuming a fixed cycle length. Several formulations based on integer programming, flow formulation, and set-partitioning are proposed and their computational behaviour is investigated.

Budai et al. (Budai, Huisman, and Dekker 2006) considered the problem of scheduling preventive railway maintenance activities through a mathematical programming formulation. Maintenance jobs are assigned to different time periods to minimize the track down-time as well as the maintenance cost. The objective, unlike common objective functions in scheduling, tends to schedule the jobs as close together as possible.

Frost & Dechter (Frost and Dechter 1998) addressed the problem of optimally scheduling preventive maintenance of power generating units within a power plant. The goal was to find the duration and sequence of outages of power generating units while minimizing operating and maintenance costs. Weekly power demand and consecutive maintenance periods are examples of the constraints considered. The problem is cast as a continuous satisfaction problem and an "iterative learning" algorithm is used to solve the problem.

Haghani & Shafahi (Haghani and Shafahi 2002) studied the problem of scheduling bus maintenance activities. A mathematical programming approach for the problem is proposed to find the daily inspection and maintenance schedule in order to minimize the interruption in the daily bus operating schedule and to maximize the reliability of the system.

Safaei et al. (Safaei, Banjevic, and Jardine 2010) modeled the problem studied here as a mixed integer programming model that includes an assignment problem and two network structure-based problems. The former assigned the aircraft to the waves, while the latter were used to calculate the expected number of available aircraft for the waves as well as the expected number of available workers for the repair jobs. The waves were considered as the nodes with the flow of the aircraft and workers in the two network structure-based problems. The distinguishing feature of the problem is that the objective function (wave 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 explored here.

Conclusion

In this paper, several algorithms are proposed to schedule an aircraft repair shop. The scheduling system is responsible for determining the start-time of maintenance activities to maximize the expected wave coverage considering the aircraft failure probabilities and maintenance capacity. The known statistics of probability of failure for each aircraft type is used to estimate the expected aircraft availability and therefore expected wave coverage.

The proposed algorithms include a mixed integer programming model, three logic-based Benders decomposition variants, a dispatching heuristic, and a hybrid heuristic-complete approach.

The computational results showed that: the MIP/CP LBBD technique results in a three orders-of-magnitude speed up over the pure MIP model. A tighter relaxation in the master problem speeds up the LBBD model another 50 times. The dispatching heuristic can provide a feasible schedule in a short time with five percent relative error on average and bounding the cost function in complete approaches by heuristic algorithm speeds up the complete approaches, although the speed up is not statistically significant.

A focus of our future work is to consider the repair system in a dynamic environment. Above we assumed that the failed aircraft cannot be scheduled within the current time horizon. We are interested in relaxing this assumption such that the scheduling system can be modified as soon as failure is detected and are investigating the impact of such a change on overall system performance.

References

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

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

Chu, Y., and Xia, Q. 2004. 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)*, 127–136.

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

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

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

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

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

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

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

Nicolai, R. P., and Dekker, R. 2008. *Complex System Maintenance Handbook*. Springer Series in Reliability Engineering. chapter 11.

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

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

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

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