A Comparison of Stochastic Programming and Bi-Objective Optimization Approaches to Robust Airline Crew Scheduling

Bassy Tam[1], Matthias Ehrgott[1,2], David Ryan[1], Golbon Zakeri[1]
[1]Department of Engineering Science, The University of Auckland
Private Bag 92019, Auckland, New Zealand.
email: bassytam@gmail.com, {m.ehrgott,d.ryan,g.zakeri}@auckland.ac.nz
[2] Laboratoire d'Informatique de Nantes Atlantique, Université de Nantes
2 rue de la Houssinière, BP 9208 44322 Nantes, France.
email: matthias.ehrgott@univ-nantes.fr

August 10, 2007

Abstract

A prominent problem in *airline crew scheduling* is the pairings or Tour-of-Duty planning problem. The objective is to determine a set of *pairings* (or *Tours-of-Duty*) for a crew group to minimize the planned cost of operating a schedule of flights. However, due to unforeseen events the performance in operation can differ considerably from planning, sometimes causing significant additional recovery costs.

In recent years there has been a growing interest in robust crew scheduling. Here, the aim is to find solutions that are "cheap" in terms of planned cost as well as being robust, meaning that they are less likely to be disrupted in case of delays. Taking the stochastic nature of delays into account, Yen and Birge (2006) formulate the problem as a two-stage stochastic integer programme and develop an algorithm to solve this problem. Based on the contradictory nature of the goals, Ehrgott and Ryan (2002) formulate a bi-objective set partitioning model and employ elastic constraint scalarization to enable the solution by set partitioning algorithms commercially used in crew scheduling software.

In this paper we compare the two solution approaches. We improve the algorithm of Yen and Birge (2006) and implement both methods with a commercial crew scheduling software. The results of both methods are compared with respect to characteristics of robust solutions, such as the number of aircraft changes for crew. We also conduct experiments to simulate the performance of the obtained solutions. All experiments are performed using actual schedule data for a New Zealand domestic airline.

1 Airline Crew Scheduling

Over the last three decades, airlines have devoted a great effort to solve *airline scheduling problems*. One of these is the Tour-of-Duty (ToD) planning problem, which consists in constructing sequences of flights to crew the flight schedule. All flights in the given schedule within the planning horizon are partitioned into sequences of flights. Each sequence of flights that a crew member can fly is referred to as a *Tour-of-Duty* (ToD) or *pairing*.

Traditionally, a crew schedule is constructed in terms of minimizing the planned cost. However, it is perceived that in operation such a schedule might result in high realized cost. Since crew schedules with minimal planned cost usually contain ToDs which are not flexible to accommodate minor flight time changes, and are therefore sensitive to disruptions, extra cost is incurred to recover from those disruptions.

Today, airlines are not only interested in a crew schedule with minimal planned cost but also in a robust crew schedule which minimizes the expected cost in operation. A robust schedule is one under which effects of disruptions are less likely to be propagated into the future.

A ToD is a sequence of duty periods, normally starting and ending at a specified airport, the crew's home base (*crew base*). Any overnight connection between two duty periods is called *rest period* or *layover*. A ToD can also be called a *pairing*, trip or a rotation.

A *duty period* is a crew's working day which consists of a sequence of flights with *ground times* that cannot be shorter than a certain time period, e.g. 30 minutes, between consecutive flights, meal times and maybe *passengering flights* (PAX), in which a crew member travels as a passenger to get to a particular airport for a subsequent flight or back to the crew base. Passengering is also referred to as *dead heading*.

The cost minimization problem is modelled as a generalized set partitioning problem (GSPP) (Barnhart *et al.*, 2003):

$$\begin{array}{lrcl} \text{minimize} & c^T x & & \\ \text{subject to} & \boldsymbol{A}x & = & e \\ & \boldsymbol{M}x & \left\{\begin{matrix} \geq \\ = \\ \leq \end{matrix}\right\} & b \\ & x & \in & \{0,1\}^n, \end{array} \tag{1}$$

where $\boldsymbol{A}$ is a binary matrix and e is a vector of ones.

In the ToD planning model (1), each column or variable corresponds to one feasible ToD that can be flown by some crew member. The value of c_j, the cost of variable j, reflects the dollar cost of operating the j^{th} ToD. The decision variable x_j is equal to 1 if pairing j is included in the solution and 0 otherwise. The first set of constraints in (1) is referred to as flight constraints, and the second set contains the crew base balancing constraints.

Each *flight constraint* corresponds to a particular flight sector and ensures that the sector is included in exactly one ToD, where a *flight sector* is a non-stop connection from an origin to a destination. The elements of the $\boldsymbol{A}$ matrix are

$$a_{ij} = \begin{cases} 1, & \text{if pairing } j \text{ contains flight } i \text{ as operating sector,} \\ 0, & \text{otherwise.} \end{cases}$$

Note that if the j^{th} pairing includes the i^{th} flight as a passengering flight, this will result in a column in which $a_{ij} = 0$.

The ToD planning model is usually augmented with additional constraints referred to as *crew base balancing constraints*. They ensure the distribution of work over the set of crew bases matches the crew resources by permitting restrictions to be imposed on the number of crew resources included from each crew base: The number of crews contained in the chosen pairings which originate at a given crew base must be between specified lower and upper bounds.

Each crew base balancing constraint represents a crew base restriction for the respective crew base. In this case, b_i is the maximal/minimal available resource, and m_{ij} is the resource attributed to the crew base balancing constraint i if pairing j is used. An example is to limit the number of ToDs to be operated from a crew base.

One of the main difficulties with the ToD planning problem is the complicated set of rules and regulations that must be satisfied by each ToD. Because those rules and regulations often cannot be easily expressed in mathematical terms, a *column generation* technique is often used to generate pairings while solving the ToD planning problem (Anbil *et al.*, 1998). The column generation problem is a resource constrained shortest path problem, where the resource constraints ensure that only legal pairings that satisfy all rules are generated.

Another difficulty with the ToD planning problem is that the number of feasible pairings is extremely large even for problems with relatively few flights, so generating all possible ToDs for the optimization problem is often impossible. To some extent, this problem can be overcome by using a *dynamic column generation* technique to generate columns during the optimization process (Barnhart *et al.*, 1998). Thus generation of pairings and solving the generalized set partitioning problem is done iteratively during the optimization process. In this way, the number of feasible pairings increases dynamically, but most of the feasible pairings will never be considered.

To obtain an integer solution, a *branch and price* approach with *constraint branching* is used (Ryan and Foster, 1981). The branch and price procedure is similar to the branch and bound technique, but dynamic column generation is used at each node of the branch and bound tree. *Follow-on branching* (Ryan and Falkner, 1988), which is a constraint branching strategy commonly used for this type of problem, is to force or ban two flights to be operated as a subsequence in a pairing. The flight pair (F_r, F_s) is operated as subsequence if a crew assigned to operate flight F_s after the operation of the flight F_r, with no other operating sector in between flights F_r and F_s. On the one branch, all pairings that operate only one of the two flights are eliminated. On the zero branch, all pairings that operate both flights are eliminated.

2 Operational Robustness

The ToD planning problem is solved well before the flight schedule becomes operational. In this planning stage, all flights are assumed to have departure times that are both fixed and known. This assumption is often proven wrong when the crew schedule is actually implemented.

ToDs are usually less expensive if crews spend less time on the ground between arrival and departure of two consecutive flights, hence the total working or operating hours are minimized. Such crew schedules happen to become "de-optimized" in actual operation, as they are easily disrupted and chain impacts are usually found as a result. Thus, if the airline provides connection times between consecutive flights to both aircraft and crew members, which just satisfy the minimum time legally required, a late arriving flight will cause the following flight to depart late. Not only will the downstream flight which operates on the delayed aircraft depart late, but also the late arriving crew members who are changing aircraft will board late for their outgoing flights. After a few aircraft changes, many flights may be delayed by the initially minor delay.

Furthermore, disruptions may require the use of reserve crews to get back on schedule and originally scheduled crew might not be able to continue on their duty because of rule violations. As a result, substantial unplanned costs, such as overtime, fuel costs and compensations for parking and passengers with delayed or cancelled flights, can be incurred.

So airlines not only require minimum cost solutions, but are also very interested in robust solutions. A *robust ToD planning problem* is the problem of obtaining aircrew schedules in planning that are not necessarily optimal in terms of the planned crew cost but that yield low crew cost in operation. Approaches to robust aircrew scheduling have been developed only recently, but all of the approaches have different measures of operational robustness.

Rosenberger *et al.* (2000, 2002) and Schaefer *et al.* (2005)) solve a problem very similar to the original ToD planning model. However, they replace the objective coefficients c_j in the model with the expected cost of the j^{th} ToD. Ehrgott and Ryan (2002) solve the robust ToD planning problem using a bi-criteria approach with an additional objective to maximize the operational robustness of the crew schedule with the planned crew cost to be minimized. Yen and Birge

(2006) formulate the robust ToD planning problem as a two-stage stochastic binary programming model with recourse. Shebalov and Klabjan (2006) solve the robust ToD planning problem using a bi-criteria approach with an additional objective to maximize the number of opportunities for crew swapping.

In this paper, we will focus on two robust ToD planning problem approaches with similar operational robustness measures, the bi-criteria optimization approach introduced by Ehrgott and Ryan (2002) and the stochastic programming approach introduced by Yen and Birge (2006).

Ehrgott and Ryan (2002) point out that "a robust solution is one in which crew changing aircraft is discouraged if insufficient ground time occurs to compensate late arrivals". In other words, a robust solution would have the property that if an upstream flight is likely to be delayed, crew should not be scheduled to change aircraft for a successive flight, which leaves after only minimal ground time. Thus, crews change aircraft between operating flight sectors less frequently in a robust ToD solution. They develop an objective function to penalize ToDs which are not robust.

There is a trade-off between minimizing the crew cost and minimizing the non-robustness penalty. A schedule that minimizes the non-robustness measure will have high crew cost. The traditional ToD planning problem is solved first, giving a minimum planned crew cost. Ehrgott and Ryan solve the LP relaxation of the bi-criteria problem using the *ε-constraint scalarization*, i.e. the non-robustness objective is minimized with an added constraint to control the crew cost with an upper bound, so that the planned crew cost objective is not too far from the minimum crew cost objective. To solve the IP the ε-constraint is transformed to an *elastic constraint* by an additional surplus variable to allow a small violation of the cost constraint if robustness can be improved in the branch and price process.

Yen and Birge (2006) solve the robust ToD planning problem as a two-stage stochastic binary programming model with recourse with a similar robustness measure as Ehrgott and Ryan but assuming the flight operation time is a random variable.

Given a crew schedule, the recourse problem is a large-scale LP to measure the cost of delays, with the first stage problem being the traditional ToD planning problem in GSPP formulation (1). They develop a method based on follow-on branching to solve the model. They sample 100 disruption scenarios and evaluate the solution of the second stage LP for each scenario to determine the "switching cost" associated with aircraft changes. The "switching cost" is then passed back to the first stage problem to remove any "expensive" aircraft changes, by branching on the sector pair with the highest "switching cost".

The main drawback of this approach is that it is very computationally expensive because the set partitioning problem needs to be solved often. Yen and Birge only show computational results on a problem with a maximum of 79 flight sectors, which is rather small.

Another drawback of this approach is the aggregation of planned crew cost and expected delay. Firstly, airlines need a good estimate on the recovery cost for each delay minute of a flight to obtain good robust crew schedule. But the respective measures are incommensurate. Secondly, their model assumes that flight delay has a positive linear relationship to the cost of delays. This might not be true in real operation.

3 Description of the Methods

We are interested in the performance of crew schedules generated by the two different robustness approaches. To remove all other possible factors that might influence the outcome of crew schedules, we have to implement the two robustness approaches using the same ToD optimizer.

For a given flight schedule, enumerating all feasible pairings is very computationally expensive due to their enormous number. So we need to reinvestigate the stochastic programming approach. We have successfully integrated dynamic column generation with the stochastic programming approach to reduce the number of variables in the ToD planning problem while being able to apply it to a real schedule.

3.1 Delay Analysis

The approaches of Ehrgott and Ryan (2002) and Yen and Birge (2006) both require flight delay time as part of their robustness measures. To ensure both approaches use the same parameter for the flight delay time, there is a need to investigate some historical delay data. We have studied 82 weeks of flight delay data which included over 40,000 flights of domestic operations. Due to the problem of dependence between observations, a direct statistical model cannot be drawn from the given delay data.

Flights might be delayed under different circumstances. Some are circumstances within the airline's control, such as delayed crew, maintenance, baggage loading or other schedule problems. Some are uncontrollable and unavoidable. Examples are delayed passengers, bad weather, air traffic control or airport operation. Some are due to chain impacts from the initial delay, e.g. if only minimum ground time is available and a previous flight on the same aircraft arrived late, the following flight will depart late. (According to US Department of Transportation (2007), more than 30% of flights that arrive 15 minutes later than scheduled were due to late arriving aircraft.) Furthermore, if a crew is scheduled to spend minimum time on ground, when the crew are changing aircraft after a delayed flight, subsequent flights operated by the crew will be delayed.

To develop a good statistical model for flight delays, it is necessary to remove the delays due to chain impacts, otherwise proportion and duration of delay associated with each flight might be overestimated. We deducted all delays due to late arriving aircraft, but delays due to late arriving crew could not be eliminated due to missing information on crew schedules for the period for which we had delay data. Although the resulting flight delay times are not completely independent of the delay from other flights, the degree of dependence has been significantly reduced. In our analysis, we found that 50% of flights that arrive 15 minutes later than scheduled were due to late arriving aircraft.

Removing delays due to late arriving aircraft might result in underestimating the proportion of delayed flights as well as the delay time associated with each flight, as we are assuming that the flight delays are additive. This might not be true in real life, e.g. if a flight has a delay of 20 minutes due to the late arrival of the aircraft by 20 minutes, it does not mean the flight would have been able to depart on time had the aircraft arrived on time.

Removing delays due to late arriving aircraft results in underestimating the delay associated with each flight, while not removing delays due to the late arriving aircraft results in overestimating the delay associated with each flight. We choose the former. We have observed that if a serious disruption happens at the beginning of the day of operation, delays are more frequent and delay durations are longer on this day than on any other day. Although some of the delay measures might be underestimated in our analysis, our sample size is large enough to smooth out those underestimated delays. That is, we still have a large proportion of flights the delays of which are not associated with the aircraft and this allows an accurate model of those underestimated delay times to be found.

Table 1 gives a brief summary of the proportions of delayed flight and total delay minutes by day of the week.

		$\mathbf{D}$		$\tilde{\mathbf{D}}$			$\hat{\mathbf{D}}$	
Weekday	**P(D)**	**Mean**	**Median**	**Mean**	**Median**	$\mathbf{P(\hat{D})}$	**Mean**	**Median**
Monday	0.54	649.24	534.50	376.24	285.50	0.40	298.37	275.00
Tuesday	0.47	460.94	314.00	270.27	176.00	0.34	232.77	180.50
Wednesday	0.51	537.01	372.00	317.47	199.00	0.38	257.56	218.00
Thursday	0.57	730.73	511.00	439.70	309.00	0.43	321.74	265.00
Friday	0.65	1013.12	735.00	640.99	455.00	0.47	387.10	327.00
Saturday	0.43	380.76	243.50	194.67	96.00	0.33	184.54	139.50
Sunday	0.53	579.07	416.00	348.33	248.50	0.38	237.76	198.00

Table 1: Proportions of delay and total delay minutes by day of the week.

The $\mathbf{P(D)}$ column of Table 1 indicates the proportions of delayed flights (i.e. actual arrival time is later than the scheduled arrival time), $\mathbf{D}$ is the total delay (in minutes) per day (actual arrival time minus scheduled arrival time if the flight is delayed) and $\tilde{\mathbf{D}}$ is the total delay (in minutes) caused by late arriving planes per day. The $\mathbf{P(\hat{D})}$ column of the table shows the proportion of flights that are delayed after removing the delay caused by the late arriving aircraft and $\hat{\mathbf{D}}$ is the total delay per day of the flights without the impact of late arriving aircraft.

As we can see, more than half of the delay minutes are caused by plane connections. For example, 51% of the flights were delayed on Wednesday and the total delay minutes per day averaged 537.01 of which 317.47 minutes have been caused by late arriving aircraft.

We also see that flight delays are associated with the day of week on which the flight operated. The fact that the flight schedule is tighter, there are more passengers and airports are busier on Friday than on other weekdays and weekends, might cause more and longer delays. For the same reasons, we believe that the flight delay is also related to the departure time of the flight, the arrival time of the flight and the departure and arrival airport of the flight.

We have found the probability of flight delay and the distribution of delay time based on these components using multi-variable regression. Considering the large sample size, we partitioned our sample according to the departure weekday of the flight for analysis, i.e. we have seven flight delay probabilities and seven delay time distribution models.

We found that flight delays are more frequent and more serious during the peak time than the off-peak time. This means the departure and arrival times of the flight are not linearly correlated to the delay time, which makes the regression model difficult. To overcome this problem, we separated the time component into two parts. The first, the hour of the departure/arrival time, is treated as a category variable. The second is the minute component of the departure/arrival time and this is treated as a continuous variable.

The probability (π^k) of a flight delay on the k^{th} weekday is modelled by a logistic regression:

$$\log\left(\frac{\pi^k}{1-\pi^k}\right) = \beta_0^k + \sum_{i=0}^{23}\left(\beta^k_{\text{DH}_i}x^k_{\text{DH}_i} + \beta^k_{\text{AH}_i}x^k_{\text{AH}_i}\right) + \beta^k_{\text{DM}}x^k_{\text{DM}} + \beta^k_{\text{AM}}x^k_{\text{AM}} + \sum_{i\in P}\left(\beta^k_{\text{D}_i}x^k_{\text{D}_i} + \beta^k_{\text{A}_i}x^k_{\text{A}_i}\right),$$

where

- $x_{\text{DH}_i} \in \{0,1\}$ for $i \in \{0,1,2,...,23\}$ indicates the scheduled departure hour for the flight. If a flight is scheduled to depart within hour j then $x_{\text{DH}_j} = 1$ and $x_{\text{DH}_i} = 0$ for all $i \neq j$,
- $x_{\text{AH}_i} \in \{0,1\}$ for $i \in \{0,1,2,...,23\}$ indicates the scheduled arrival hour for the flight. If a flight is scheduled to arrive within hour j then $x_{\text{AH}_j} = 1$ and $x_{\text{AH}_i} = 0$ for all $i \neq j$,
- $x_{\text{DM}} \in [0,60)$ is the minute of the scheduled departure time for the flight,
- $x_{\text{AM}} \in [0,60)$ is the minute of the scheduled arrival time for the flight,
- $x_{\text{D}_i} \in \{0,1\}$ for $i \in P$ indicates the origin of the flight, where P is a collection of airports. If a flight departs from airport j then $x_{\text{D}_j} = 1$ and $x_{\text{D}_i} = 0$ for all $i \neq j$,
- $x_{\text{A}_i} \in \{0,1\}$ for $i \in P$ indicates the destination of the flight. If a flight arrives at airport j then $x_{\text{A}_j} = 1$ and $x_{\text{A}_i} = 0$ for all $i \neq j$,

the βs are the coefficients of the associated components and β_0 is a constant.

To model the flight delay time, we built a model on the observations suffering from delay. For flights departing on the k^{th} weekday, the flight delay time (DT^k) is modelled by multi-variable regression

$$\log\left(\text{DT}^k\right) = \alpha_0^k + \sum_{i=0}^{23}\left(\alpha^k_{\text{DH}_i}x^k_{\text{DH}_i} + \alpha^k_{\text{AH}_i}x^k_{\text{AH}_i}\right) + \alpha^k_{\text{DM}}x^k_{\text{DM}} + \alpha^k_{\text{AM}}x^k_{\text{AM}} + \sum_{i\in P}\left(\alpha^k_{\text{D}_i}x^k_{\text{D}_i} + \alpha^k_{\text{A}_i}x^k_{\text{A}_i}\right),$$

where the αs are the coefficients of the associated components and α_0 is a constant.

We found that after removing the effects of the components, flights were delayed randomly. However, the distribution of flight delay time was found to be multi-modal. This is possibly because some proportion of the flight delay has been underestimated and/or due to some factors we cannot capture, such as weather conditions or problems during the transit period when there is a change in the flight schedule. However, the right tail of the distribution was well explained by an exponential distribution.

The 95% quantile of the delay time model has been used as the delay measure in the robustness measure of the bi-criteria approach from Ehrgott and Ryan (2002). The probability of flight delay and the distribution of delay time were used to generate delay scenarios for the stochastic programming approach from Yen and Birge (2006).

3.2 The Bi-Criteria Approach

Ehrgott and Ryan (2002) develop a robustness measure by estimating the propagation of delays through the flight schedule. They form an objective function to penalize pairings which are not robust and then try to minimize this objective while at the same time maintaining a cost effective solution.

This non-robustness measure for each ToD is obtained by considering each consecutive sector pair in any given pairing. A penalty will be incurred on a connection (F_r, F_s), if the scheduled ground time minus the required minimum ground time is less than the delay measure of the first flight, F_r. If consecutive sectors are on the same aircraft no penalty is incurred, since the delay only affects flights on this same aircraft and these are inevitable. In addition, if the last sector of a ToD is a passengering flight, no penalty is added, as the crew member can usually take a later flight to the crew base. That is, if we consider the j^{th} ToD consisting of S sectors, the non-robustness measure of this ToD, r_j is calculated by:

$$r_j = \sum_{i=1}^{S-1} p_{F_{(i+1)}}^{F_{(i)}}$$

where $F_{(i)}$ is the i^{th} sector in the pairing and

$$p_{F_{(i+1)}}^{F_{(i)}} = \begin{cases} 0, & \text{if} \begin{cases} \text{plane}(F_{(i)}) = \text{plane}(F_{(i+1)}), \\ \text{or} \\ i = S-1 \text{ and } F_{(i+1)} \text{ is PAX} \end{cases} \\ \max\left\{0, \text{GDT}_{F_{(i+1)}}^{F_{(i)}} + \text{DM}_{F_{(i)}} - \text{SGT}_{F_{(i+1)}}^{F_{(i)}}\right\}, & \text{otherwise.} \end{cases}$$

where

- $\text{GDT}_{F_{(i+1)}}^{F_{(i)}}$ is the minimum required ground duty time between the flights $F_{(i)}$ and $F_{(i+1)}$,
- $\text{SGT}_{F_{(i+1)}}^{F_{(i)}}$ is the scheduled ground time between the flights $F_{(i)}$ and $F_{(i+1)}$ in the tour of duty,
- $\text{DM}_{F_{(i)}}$ is the measure for the delay of the incoming flight $F_{(i)}$.

The delay measure can be chosen as the expected delay of the flight or expected delay plus the standard deviation, or some other measures. Ehrgott and Ryan (2002) use 2 standard deviations above the mean as the delay measure. In this paper we use the 95% quantile as the delay measure.

The consideration of both a cost objective function and a robustness objective function leads to a bi-criteria problem with the following formulation, which is obtained by including the second objective in the original ToD planning model.

$$\begin{array}{lrcl} \text{minimize} & r^T x & & \\ \text{minimize} & c^T x & & \\ \text{subject to} & \boldsymbol{A}x & = & e \\ & \boldsymbol{M}x & \left\{\begin{array}{c} \geq \\ = \\ \leq \end{array}\right\} & b \\ & x & \in & \{0,1\}^n. \end{array} \tag{2}$$

This problem is solved using the elastic constraint scalarization technique. In the elastic constraint method, the cost objective is formulated as an elastic constraint resulting in the following model.

$$
\begin{array}{llllll}
\text{minimize} & r^T x & + & p s_c & & \\
\text{subject to} & c^T x & - & s_c & \leq & \varepsilon \\
& \boldsymbol{A} x & & & = & e \\
& \boldsymbol{M} x & & & \left\{ \begin{array}{c} \geq \\ = \\ \leq \end{array} \right\} & b \\
& x & & & \in & \{0,1\}^n \\
& & & s_c & \geq & 0.
\end{array}
\tag{3}
$$

The right hand side value for the first constraint (ε) is the planned crew cost that the airline is willing to pay. The new surplus variable for this cost constraint, s_c, is introduced when the branch and price process begins, aimed to reduce computational difficulties arising from adding this constraint in the set partitioning integer programme, see Ehrgott and Ryan (2003). The cost coefficient p for s_c is a penalty for violating the cost constraint. To obtain an integer solution, the original branch and price process is used.

3.3 The Stochastic Programming Approach

Yen and Birge (2006) use a similar robustness measure as Ehrgott and Ryan (2002), but instead of using a deterministic delay parameter, they use disruption scenarios to evaluate crew schedules and use a branch and bound strategy to ban non-robust flight connections. They formulate the robust ToD planning model as a two-stage stochastic binary programming model.

In the bi-criteria approach, flight delays are considered independently. That is, Ehrgott and Ryan implicitly assume that the plane predecessor flight of F_{i+1} is not delayed when calculating the penalty $p^{F_{(i)}}_{F_{(i+1)}}$ for a sector pair, (F_i, F_{i+1}), given that F_i and F_{i+1} are not on the same aircraft. This assumption does not hold for the stochastic programming model.

Yen and Birge enumerate all feasible pairings before the optimization. This is possible if there are only a few flights in the schedule (they show results for a schedule with 79 flights). For a flight schedule of one week which may contain a few hundred flights, it is costly or even impossible to enumerate all feasible pairings. In our test problem, the flight schedule contains 442 flights. Instead of enumerating all feasible pairings we used the dynamic column generation technique.

After the ToD planning problem is solved in terms of minimal planned crew cost, the crew schedule x is evaluated under some disruption scenarios. Let a disruption scenario ω be a random element of some space Ω, that occurs with probability $\mathcal{P}(\omega)$. The crew schedule x will incur a recovery cost $\mathcal{Q}(x, \omega)$ under disruption scenario ω. The expected value of future action to operate the crew schedule x is denoted by $\mathcal{Q}(x)$ and it is defined as

$$\mathcal{Q}(x) = \sum_{\omega \in \Omega} \mathcal{P}(\omega) \mathcal{Q}(x, \omega).$$

Therefore, the stochastic programming formulation of the robust ToD planning problem is

$$
\begin{array}{llllllll}
\text{minimize} & z & = & c^T x & + & \mathcal{Q}(x) & & \\
\text{subject to} & & & \boldsymbol{A} x & & & = & e \\
& & & \boldsymbol{M} x & & & \left\{ \begin{array}{c} \geq \\ = \\ \leq \end{array} \right\} & b \\
& & & x & & & \in & \{0,1\}^n.
\end{array}
\tag{4}
$$

To evaluate a crew schedule under a disruption scenario ω, the pushback recovery procedure is used. *Pushback recovery* means that a flight is delayed until all resources (crew members and aircraft) are available. A summary of the notation to evaluate a crew schedule is given now. Let

- SDT_{F_i} be the scheduled departure time of the flight F_i,

- SAT_{F_i} be the scheduled arrival time of the flight F_i,
- FT_{F_i} be the scheduled flight time of F_i, i.e. $\mathrm{FT}_{F_i} = \mathrm{SAT}_{F_i} - \mathrm{SDT}_{F_i}$,
- F_i^p be the plane predecessor flight of F_i,
- $F_i^{c_n}$ be the crew predecessor flight of F_i under pairing n,
- $\mathrm{PGT}_{F_i}^{F_i^p}$ be the minimum required plane ground time between the flights F_i^p and F_i,
- $\mathrm{GDT}_{F_i}^{F_i^{c_n}}$ be the minimum required ground duty time between the flights $F_i^{c_n}$ and F_i,
- $\mathrm{DT}_{F_i}^{\omega}$ be the delay time associated with the flight F_i under scenario ω, $\mathrm{DT}_{F_i}^{\omega}$ is a random variable,
- $\mathrm{ADT}_{F_i}^{\omega}$ be the actual departure time of the flight F_i under scenario ω,
- $\mathrm{AAT}_{F_i}^{\omega}$ be the actual arrival time of the flight F_i under scenario ω,
- $\widehat{\mathrm{ADT}}_{F_i}^{\omega}$ be the actual departure time of the flight F_i under scenario ω without crew interactions,
- $\widehat{\mathrm{AAT}}_{F_i}^{\omega}$ be the actual arrival time of the flight F_i under scenario ω without crew interactions,
- $\mathrm{TD}_{F_i}^{\omega}$ be the total delay to the flight F_i under scenario ω,
- $\mathrm{ND}_{F_i}^{\omega}$ be the non-crew induced delay to the flight F_i under scenario ω.

If we consider a crew schedule x consisting of N pairings, the actual departure time of a flight F_i under a disruption scenario ω is

$$\mathrm{ADT}_{F_i}^{\omega} = \max\left\{\mathrm{SDT}_{F_i}, \mathrm{AAT}_{F_i^p}^{\omega} + \mathrm{PGT}_{F_i}^{F_i^p}, \mathrm{AAT}_{F_i^{c_n}}^{\omega} + \mathrm{GDT}_{F_i}^{F_i^{c_n}}\right\}$$

while the actual departure time without crew interactions is

$$\widehat{\mathrm{ADT}}_{F_i}^{\omega} = \max\left\{\mathrm{SDT}_{F_i}, \widehat{\mathrm{AAT}}_{F_i^p}^{\omega} + \mathrm{PGT}_{F_i}^{F_i^p}\right\}.$$

The actual arrival time of a flight F_i under a disruption scenario ω is

$$\mathrm{AAT}_{F_i}^{\omega} = \mathrm{ADT}_{F_i}^{\omega} + \mathrm{FT}_{F_i} + \mathrm{DT}_{F_i}^{\omega}$$

while the actual arrival time without crew interactions is

$$\widehat{\mathrm{AAT}}_{F_i}^{\omega} = \widehat{\mathrm{ADT}}_{F_i}^{\omega} + \mathrm{FT}_{F_i} + \mathrm{DT}_{F_i}^{\omega}.$$

The total delay, including plane induced delay and crew induced delay, to the flight F_i under scenario ω is defined by the actual arrival time of the flight F_i under scenario ω minus the scheduled arrival time of the flight F_i, that is

$$\mathrm{TD}_{F_i}^{\omega} = \mathrm{AAT}_{F_i}^{\omega} - \mathrm{SAT}_{F_i}$$

and the non-crew induced delay to the flight F_i under scenario ω is

$$\mathrm{ND}_{F_i}^{\omega} = \widehat{\mathrm{AAT}}_{F_i}^{\omega} - \mathrm{SAT}_{F_i}.$$

The recovery cost of operating crew schedule x for a flight schedule consisting of S flights under the disruption scenario ω is thus

$$\mathcal{Q}(x, \omega) = \sum_{i=1}^{S} p_{F_i}(\mathrm{TD}_{F_i}^{\omega} - \mathrm{ND}_{F_i}^{\omega}),$$

where p_{F_i} is the penalty cost for each minute of delay minute of flight F_i.

After the crew schedule is evaluated, flight pairs are priced by switch delay. *Switch delay* is a delay due to aircraft change, its definition is similar to the non-robustness penalty in the bi-criteria approach.

Given that the flights F_i and F_j are not on the same aircraft and crew were assigned to perform the connection (F_i, F_j) in the ToD, switch delay for connection (F_i, F_j) under scenario ω is the total delay of flight F_j under scenario ω minus the delay incurred if the crew were assigned to perform the connection (F_j^p, F_j) under scenario ω. Hence, if we consider a crew schedule x consisting of N pairings, the switch delay for a flight connection (F_i, F_j) over all scenarios in Ω is

$$\mathrm{SD}_{F_j}^{F_i} = \sum_{n=1}^{N} \delta_{nij} \sum_{\omega \in \Omega} \mathcal{P}(\omega) \max \left\{ 0, \mathrm{TD}_{F_j}^{\omega} - \left(\mathrm{AAT}_{F_j^p}^{\omega} + \max \left\{ \mathrm{PGT}_{F_j}^{F_j^p}, \mathrm{GDT}_{F_j}^{F_j^p} \right\} - \mathrm{SDT}_{F_j} \right) \right\},$$

where

$$\delta_{nij} = \begin{cases} 1, & \text{if} \begin{cases} x_n &= 1, \\ a_{in} &= 1, \\ a_{jn} &= 1, \\ F_j^{c_n} &= F_i, \\ \text{plane}(F_i) &\neq \text{plane}(F_j) \end{cases} \\ 0, & \text{otherwise.} \end{cases}$$

Once the switch delay for each flight pair is calculated for a crew schedule x, the flight pair with the highest switch delay cost will be banned to appear from any pairing selected in the next cost minimal GSPP solution. This is equivalent to imposing a 0-branch on the flight pair with highest switch delay, and is hence called *flight-pair branching*. If the total cost (crew cost and recovery cost) of the new crew schedule is worse than that of the previous one, a 1-branch is imposed on that highest switch delay flight pair, and a 0-branch is imposed on the next highest switch delay flight pair.

The processes of obtaining a cost minimal crew schedule (with imposed/forbidden flight pairs), evaluating the cost minimal crew schedule, calculating switch cost and flight-pair branching are repeated until no flight pair with positive switch delay cost is found or the increase in planned crew cost is larger than the decrease of the expected recovery cost.

4 Potential Problems and Enhancements

The crew schedules obtained from solving the test problem with the stochastic programming approach do not perform as well as we expected. We found some potential problems in the stochastic programming approach, some of which have been solved in an enhanced method, but some of which remain unsolved.

4.1 Optimality

Yen and Birge (2006) state that the algorithm must terminate with an optimal solution. We found that in some circumstances the true optimal solution will not be found. On Tam *et al.* (2007) we have given an example for which the flight-pair branching algorithm does not find the optimal solution of problem (4).

The failure of the algorithm occurs because one of the stopping criteria of the flight-pair branching algorithm is invalid, namely that the increase of planned crew cost is larger than the decrease of the recovery cost. The planned crew cost is an increasing function along a branch of the tree constructed by the delay branching algorithm, but since the recovery cost is not a decreasing function, the upper bound of the algorithm is not valid.

In our implementation, we only impose the 0-branch on the flight pair with the highest switching cost in the crew schedule. The stopping criterion is that no flight pair with positive switch delay is found or the ToD planning problem becomes infeasible.

If $p_{F_i} = p$ is the same for all flights, just imposing the 0-branch on a flight pair with highest switching cost implies that the same sequence of solutions will be generated during the flight-pair branching algorithm for every value of p. This removes the need to determine the penalty cost of delay minutes for every flight, and allows the airline to choose a robust crew schedule with reasonable crew cost.

4.2 Switch Delay

We also found that the switch delay of a flight pair might be underestimated in the current calculation, mainly because the current formula only estimates the delay introduced to the outgoing flight, but not the chain impact for the delay. We will demonstrate this by an example.

Suppose a solution x contains a duty period consisting of three flights (F_1, F_2 and F_3) and they are all on different planes (P_1, P_2 and P_3, respectively). Their schedule and delay details are as follows.

Flight	**Plane**	**Origin**	**Destination**	**SDT** (hh:mm)	**SAT** (hh:mm)	**DT** (min)
1	1	A	B	12:00	13:00	15
2	2	B	C	13:30	15:00	0
3	3	C	B	15:30	16:30	0

A crew is assigned to operate F_1 followed by F_2 followed by F_3. Suppose the predecessor plane of F_2 experiences a delay of 5 minutes and there is no delay on the predecessor planes of flights F_1 and F_3. Assuming the minimum plane ground time (PGT) and ground duty time (GDT) are both 30 minutes, the on-time performance of this partial solution is as follow.

Flight	**ADT** (hh:mm)	$\widehat{\textbf{ADT}}$ (hh:mm)	**AAT** (hh:mm)	$\widehat{\textbf{AAT}}$ (hh:mm)	**TD** (min)	**ND** (min)
1	12:00	12:00	13:15	13:15	15	15
2	13:45	13:35	15:15	15:05	15	5
3	15:45	15:30	16:45	16:30	15	0

With the Yen and Birge (2006) definition, the switch delay for the flight pair (F_1, F_2), $\mathrm{SD}_{F_2}^{F_1}$, is 10 minutes and the switch delay for the flight pair (F_2, F_3), $\mathrm{SD}_{F_3}^{F_2}$, is 15 minutes.

From our point of view, the value of $\mathrm{SD}_{F_2}^{F_1}$ has been underestimated. This is because the delay from F_1 resulted in an extra 10 minutes delay of F_2 and hence leads to an extra 10 minutes delay of F_3. Thus, if a crew is assigned to service F_2 after he/she has served on the F_2 predecessor plane instead of F_1, the total delay of F_2 will be 5 minutes and the total delay of F_3 will be 5 minutes. In the Yen and Birge definition of switch delay, the extra 10 minutes delay of F_3 caused by the connection (F_1, F_2) are not included in the calculation of $\mathrm{SD}_{F_2}^{F_1}$.

We have a new definition of switch delay which can overcome this underestimation. Consider a crew schedule x consisting of N pairings. Let $\mathbf{C}$ be the set of flight pair connections in the crew schedule x, i.e.

$$(F_i, F_j) \in \mathbf{C}, \quad \text{if} \begin{cases} x_n &= 1, \\ a_{in} &= 1, \\ a_{jn} &= 1, \\ F_j^{c_n} &= F_i \end{cases}$$

for some $n \in \{1, \ldots, N\}$.

The recovery cost $\mathcal{Q}(x, \omega)$ of operating crew schedule x under disruption scenario ω is equivalent to the recovery cost $\tilde{\mathcal{Q}}(\mathbf{C}, \omega)$ of operating the set of crew connections $\mathbf{C}$ under the same disruption scenario ω, i.e.

$$\mathcal{Q}(x, \omega) = \tilde{\mathcal{Q}}(\mathbf{C}, \omega) \text{ for all } \omega \in \Omega.$$

We define the switch delay $\mathrm{SD}_{F_j}^{F_i}$ of the flight pair (F_i, F_j) to be the recovery cost of operating the set of crew connections $\mathbf{C}$ minus the recovery cost of operating this set of crew connections except the connection $(F_i,\ F_j)$, but including connection $(F_j^p,\ F_j)$, over all scenarios in Ω. That is,

$$\mathrm{SD}_{F_j}^{F_i} = \sum_{\omega \in \Omega} \mathcal{P}(\omega) \max \left\{0, \tilde{\mathcal{Q}}(\mathbf{C}, \omega) - \tilde{\mathcal{Q}}(\widehat{\mathbf{C}_{F_j}^{F_i}}, \omega)\right\},$$

where

$$\widehat{\mathbf{C}_{F_j}^{F_i}} = \mathbf{C} \cup \{(F_j^p, F_j)\} \backslash \{(F_i, F_j)\}.$$

With this new definition, the switch delay for the flight pair $(F_1,\ F_2)$ in our example becomes 20 minutes and the switch delay for the flight pair $(F_2,\ F_3)$ is 15 minutes.

4.3 Passengering Flights

Another problem we have encountered is associated with the passengering flights. We observed that the number of passengering flights increases significantly with the number of flight-pair branches imposed, while the ground time before or after passengering decreases.

The increase in passengering flights is due to an increase in the number of duty periods and hence higher crew cost, but the short passengering ground time limits the robustness quality. This is because the calculation of recovery cost does not consider any passengering flight, and switch delay is not evaluated on the connections associated with passengering flights. Hence a flight-pair branch cannot be imposed on a connection that includes a passengering flight. We redefine the set of flight pair connections as:

$$(F_i, F_j) \in \mathbf{C}, \quad \text{if} \begin{cases} x_n & = & 1, \\ a_{in} & = & 1 \text{ or } F_i \text{ is PAX}, \\ a_{jn} & = & 1 \text{ or } F_j \text{ is PAX with } F_j \neq \text{last sector of ToD}, \\ F_j^{c_n} & = & F_i \end{cases}$$

for some $n \in \{1, \ldots, N\}$.

5 Computational Results and Comparisons

Next we report the implementation results for the test problem based on a domestic flight schedule. The test problem is a 7 day flight schedule consisting of 442 flights. This schedule services seven cities (Auckland, Christchurch, Dunedin, Hamilton, Rotorua, Queenstown and Wellington), with Auckland, Christchurch and Wellington as the crew bases. We will show the results obtained from the bi-criteria approach, followed by the results obtained from the stochastic programming approach. We then compare some robustness indicators of the solutions between the two approaches.

Both problems are solved using the same GSPP optimizer. The solutions are optimized with an optimiality gap of 2% and node limit of 1,000 for each GSPP. All settings are identical for both methods.

5.1 Bi-Criteria Approach

For the bi-criteria approach, thirteen crew schedules are obtained from using a right hand side value of the cost constraint between 0.0% and 1.2% above the cost objective from the optimal LP relaxation, in increments of 0.1%.

It can be shown that the penalty value for cost violation has to be greater than or equal to the trade-off between cost and robustness, see Ehrgott and Ryan (2003). To estimate this value we solve the LP relaxation of the problem with a strict cost constraint with a right hand side value between 0.0% and 1.6% above the cost objective, in increments of 0.05%. Figure 1 shows the trade-off between the two objectives for the LP relaxation. The percentage change in cost and robustness is with respect to the optimal value of the LP relaxation with the cost objective alone. If the trade-off for the LP is less than 0.1 we set $p = 0.1$.

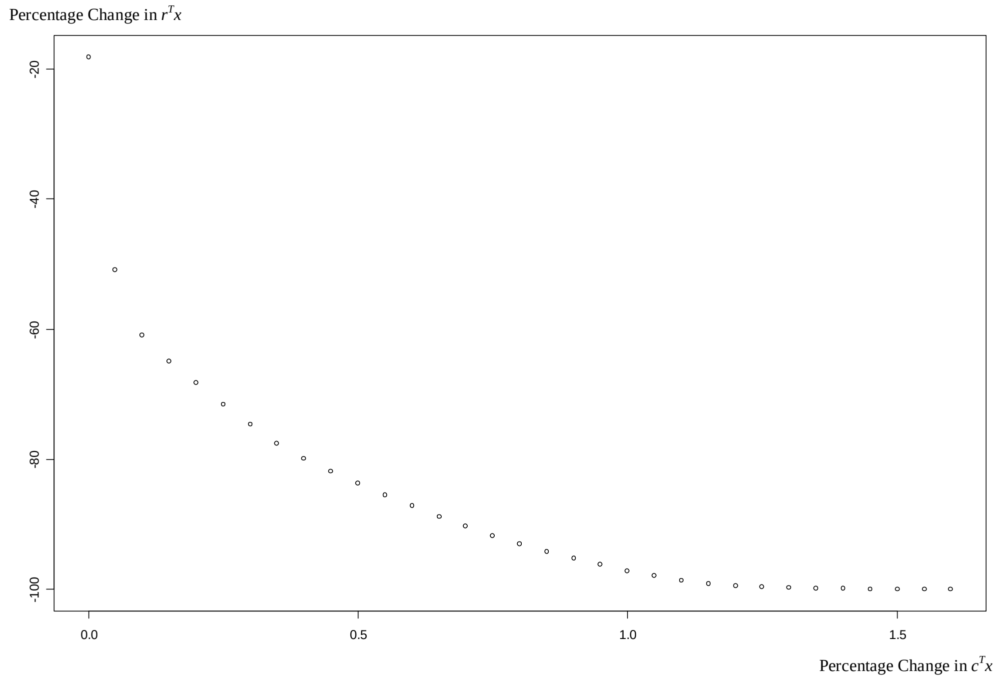

Figure 1: $r^T x$ versus $c^T x$ for the LP relaxation.

We expect the trade-off curve for the integer solutions to be similar to the one shown in Figure 1. Without increase in crew cost, we expect around 20% improvement in robustness of the crew schedule. With only 0.1% increase in crew cost, we get a decrease of approximately 60% in the non-robustness measure of the crew schedule. But it is necessary to spend another 0.3% to get a further 20% improvement in robustness.

Figure 2 is a plot of the two objectives $c^T x$ and $r^T x$ for the thirteen crew schedules obtained. The cost minimal solution without consideration of robustness is shown as a solid diamond. Numbers in Figure 2 are the allowable percentage increases in crew cost compared to the cost optimal LP solution. All Pareto optimal solutions are circled.

% Δ in $c^T x$ (L)	**$c^T x$ (L)**	**$r^T x$ (L)**	**Penalty**	**$c^T x$ (I)**	**$r^T x$ (I)**
Cost Optimal	38947.3	564.28	NA	39982.3	720.26
0.00	38947.3	461.99	9.50	39256.8	464.47
0.10	38986.2	220.54	2.04	39176.7	227.03
0.20	39025.2	179.15	0.96	39231.1	248.94
0.30	39064.1	143.01	0.86	39431.7	209.07
0.40	39103.1	113.22	0.63	39318.4	186.73
0.50	39142.0	91.80	0.52	40065.4	163.69
0.60	39181.0	72.37	0.48	39751.8	136.97
0.70	39219.9	54.60	0.44	39876.4	45.88
0.80	39258.9	39.16	0.35	40423.2	49.50
0.90	39297.8	27.02	0.29	39798.8	41.08
1.00	39336.8	16.01	0.25	40020.5	11.09
1.10	39375.7	7.93	0.18	39960.9	3.39
1.20	39414.7	2.54	0.10	40025.9	3.39

Table 2: Results of the bi-criteria approach.

Table 2 gives information of those thirteen crew schedules as well as the result from the cost

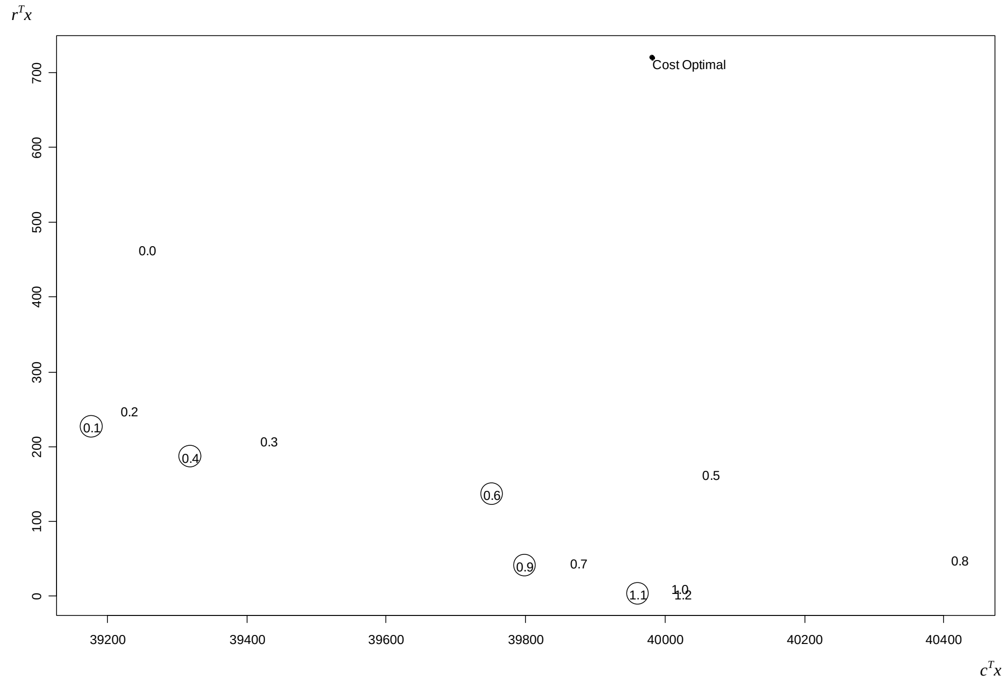

Figure 2: $r^T x$ versus $c^T x$ for solutions from the bi-criteria approach.

optimal crew schedule. The first column is the desired percentage increase in crew cost, the second column is the crew cost of the LP relaxation, i.e. the right hand side value of the cost constraint. The third column is the robustness objective in the optimal solution of the LP relaxation. The fourth column is the penalty used for each unit of crew cost violation in the IP. The fifth and sixth columns are the crew cost and robustness objective for the integer solutions, respectively.

Compared to the cost optimal solution we obtain a crew schedule with non-robustness objective of 3.39 without any increase in crew cost, i.e. a 99.5% improvement of the robustness of the crew schedule. Note that the branch and price processes were terminated at the node limit of 1,000 in all GSPPs – better solutions may exist.

5.2 Stochastic Programming Approach

For the stochastic programming approach, we randomly sample 100 scenarios from the delay distribution we modelled earlier, assuming that each disruption scenario is equally likely.

To examine the consistency of our delay model with the historical data, we evaluate the on-time performance of the optimal crew cost solution under recovery using the disruption scenarios. A brief summary of the proportions of delayed flights and average delay time from the actual data and the scenarios are given in Table 3.

The proportions of delayed flights are a bit lower than the actual values from historical data because of the underestimation we mentioned earlier. The average delay times are, however, higher than the historical delay times because of the inconsistency of the recovery procedure we are using in evaluation with the recovery procedures used in real life.

For example, if a flight suffers from a delay of 3 hours, the airline might decide to cancel the subsequent flight for the aircraft operating this flight. Since cancelled flights are not recorded in our data, this does not contribute to the delay minutes when we build our model. Under the pushback recovery, however, any subsequent flight is always delayed and not cancelled, causing this increase of total delay minutes. However, we believe that our delay model closely captures the disruptions in real life.

Using the flight-pair branching strategy, 100 GSPPs were solved before the optimization process

	Actual		Simulation	
Weekday	**P(D)**	**Mean(D)**	**P(D)**	**Mean(D)**
Monday	0.54	649.24	0.47	879.27
Tuesday	0.47	460.94	0.40	646.64
Wednesday	0.51	537.01	0.44	643.37
Thursday	0.57	730.73	0.48	1020.96
Friday	0.65	1013.12	0.56	1663.18
Saturday	0.43	380.76	0.34	551.70
Sunday	0.53	579.07	0.44	728.81

Table 3: Comparison between actual and evaluated on-time performance.

terminates. Figure 3 shows the value of $c^T x + \mathcal{Q}(x)$ at each flight-pair branching node with $p_{F_i} = 100$ for all flights.

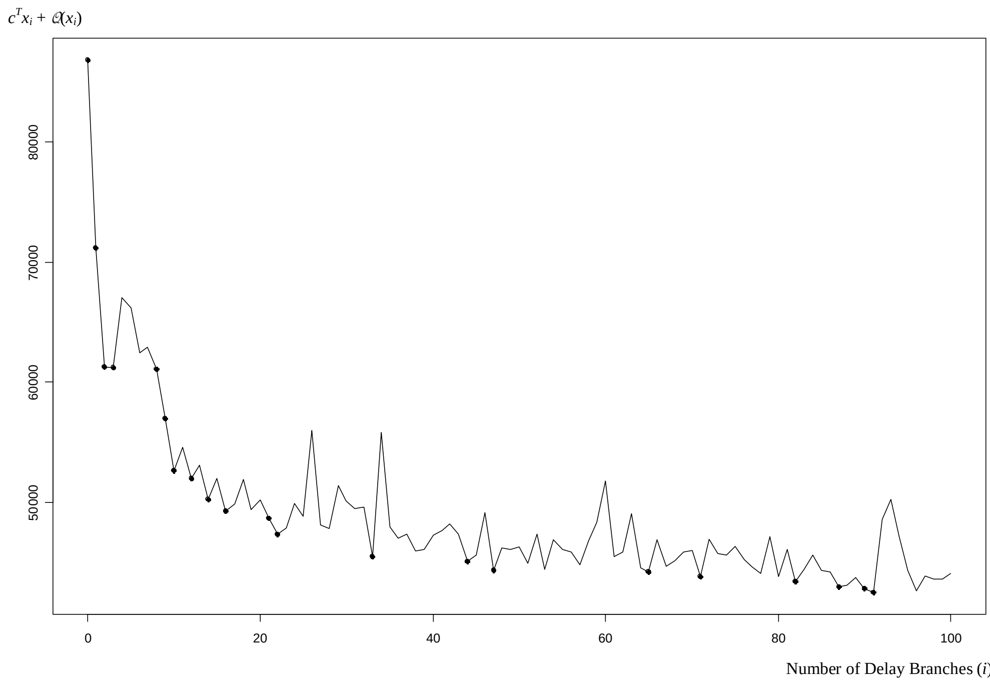

Figure 3: $c^T x_i + \mathcal{Q}(x_i)$ vs. number of flight-pair branches.

The value of $c^T x + \mathcal{Q}(x)$ fluctuates along branches of the tree constructed by the delay branching algorithm. This fluctuation comes from the recovery cost. This is due to the recovery cost not being a strictly decreasing function of the number of branches of the tree in the flight-pair branching algorithm. It is important to note that every iteration of the flight-pair branching algorithm finds a feasible solution of the crew scheduling problem. We cannot guarantee that any one of those solutions is an optimal solution of (4). All solutions differ, however, in their planned crew cost $c^T x$ and recovery cost $\mathcal{Q}(x)$. Therefore we consider all solutions which improve $c^T x + \mathcal{Q}(x)$ (the objective of (4)) during the flight-pair branching algorithm in our comparisons.

Table 4 gives the objective function value progression of improving solutions (those marked by dots in Figure 3). The first column identifies the number of flight-pair branches imposed. The second column is the optimal value of the LP relaxation in terms of crew cost, and the third column is the value of the optimal IP solution in terms of crew cost. The fourth column is the expected recourse cost of the optimal IP solution and the fifth column is the overall objective value ($c^T x + \mathcal{Q}(x)$) of the optimal IP solution. Columns 6, 7 and 8 are the percentage change in crew cost compared to the initial solution x_0. Column 9 is the percentage change in the expected recourse cost compared to the initial solution and column 10 is the percentage change in the overall objective cost compared to the initial solution.

	$c^T x_i$				**%Δ in** $c^T x_i$				
i	(L)	(I)	$\mathcal{Q}(x_i)$	$z(x_i)$	$L_i : L_0$	$I_i : L_0$	$I_i : I_0$	$\%\Delta\mathcal{Q}(x_i)$	$\%\Delta z(x_i)$
0	38947.3	39982.3	46869	86851.3	0.00	2.66	0.00	0.00	0.00
1	38947.3	39835.4	31348	71183.4	0.00	2.28	-0.37	-33.12	-18.04
2	39124.8	39548.8	21724	61272.8	0.46	1.54	-1.08	-53.65	-29.45
3	39135.5	39626.8	21590	61216.8	0.48	1.74	-0.89	-53.94	-29.52
8	39147.8	39611.1	21474	61085.1	0.51	1.70	-0.93	-54.18	-29.67
9	39151.3	39712.8	17241	56953.8	0.52	1.97	-0.67	-63.21	-34.42
10	39163.5	39760.8	12898	52658.8	0.56	2.09	-0.55	-72.48	-39.37
12	39174.1	40311.4	11653	51964.4	0.58	3.50	0.82	-75.14	-40.17
14	39177.4	39689.8	10542	50231.8	0.59	1.91	-0.73	-77.51	-42.16
16	39180.4	39714.3	9521	49235.3	0.60	1.97	-0.67	-79.69	-43.31
21	39232.8	39921.0	8724	48645.0	0.73	2.50	-0.15	-81.39	-43.99
22	39234.7	39864.7	7472	47336.7	0.74	2.36	-0.29	-84.06	-45.50
33	39253.8	39782.9	5694	45476.9	0.79	2.15	-0.50	-87.85	-47.64
44	39296.2	39921.9	5150	45071.9	0.90	2.50	-0.15	-89.85	-48.10
47	39325.3	39783.4	4531	44314.4	0.97	2.15	-0.50	-90.33	-48.98
65	39427.5	40166.0	4038	44204.0	1.23	3.13	0.46	-91.38	-49.10
71	39428.8	40289.2	3474	43763.2	1.24	3.45	0.77	-92.59	-49.61
82	39441.1	40051.0	3321	43372.0	1.27	2.83	0.17	-92.91	-50.06
87	39455.1	40099.5	2828	42927.5	1.30	2.96	0.29	-93.97	-50.57
90	39460.5	40115.0	2672	42787.0	1.32	3.00	0.33	-94.30	-50.74
91	39486.7	40254.2	2212	42466.2	1.38	3.36	0.68	-95.28	-51.10

Table 4: Improving solutions from stochastic programming approach.

Looking at the best solution, which is found after 91 flight-pair branches, the average delay due to crew connections is only 22 minutes, while the cost optimal solution (the solution with no delay branch imposed) has an average delay due to crew connections of 469 minutes. That shows a 95% decrease in delay minutes due to crew connections and a 51% decrease in total cost (crew cost and expected recovery cost) with only 0.7% increase in crew cost.

Although a substantial number of GSPPs need to be solved in the flight-pair branching algorithm both overall objective cost and recourse cost improve fast during the first few iterations. With only one flight-pair branch, the recourse objective decreases by 33% and the overall objective decreases by 18%. Another iteration reduce the recourse cost by a further 20% and the overall objective by another 11%. After 20 flight-pair branches were imposed the convergence of both objectives becomes very slow.

An important aspect of a weighted sum objective as in $cTx + \mathcal{Q}(x)$ is determining the trade-off between the two objectives, here the planned crew cost and the uncertain future recovery cost. The penalty value p_F for each delay minute of flight F controls this trade-off. In the improved flight-pair branching algorithm we use the same value for each flight. Hence, instead of looking for an optimal solution for different penalty values, we can now interpret the stochastic programming approach as looking for a set of Pareto optimal (efficient) solutions with objectives $c^T x$ and $\mathcal{Q}(x)$.

Figure 4 shows the crew induced average delay versus the planned crew cost for solutions obtained after each flight-pair branch. The numbers in the figure are the number of flight-pair branches imposed, and the solutions in circles are the Pareto optimal solutions.

This interpretation is more general than using different penalty values for delay minutes: Not every Pareto optimal solution is an optimal solution for some value p_F. Table 5 gives details of Pareto optimal solutions and a range of penalty values for which the solution would be optimal for minimizing $c^T x + \mathcal{Q}(x)$. The first column of the table identifies the number of flight-pair branches imposed. The second column is the optimal IP solution at that node in terms of crew cost. The

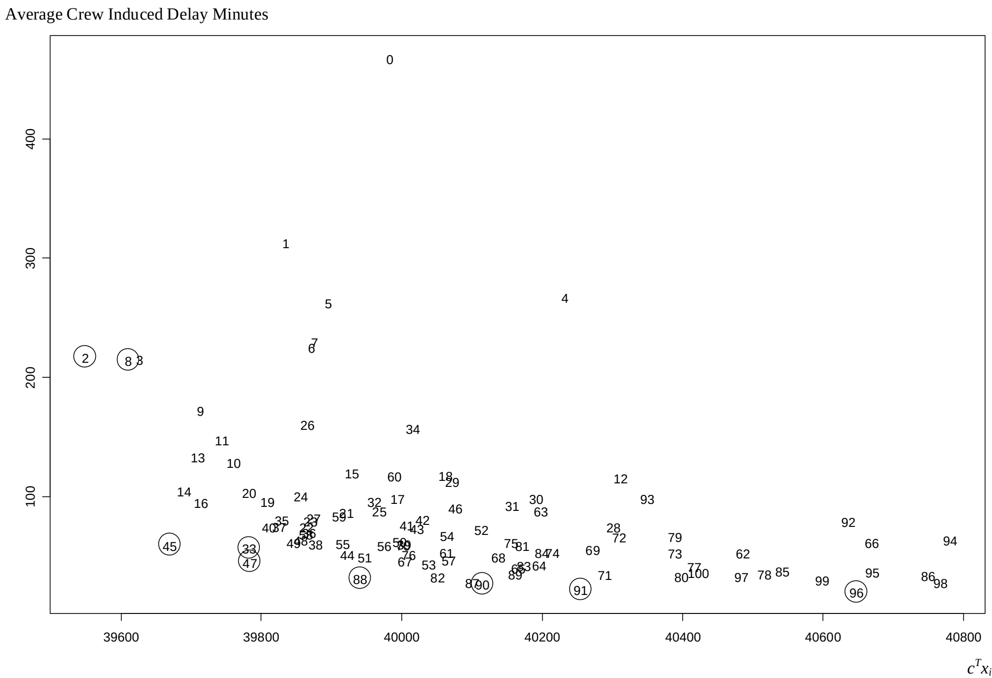

Figure 4: Trade-off between crew cost and delay minutes.

third column is the average number of delay minutes due to crew connections for that solution and the fourth column is the range of penalty values so that the solution would be optimal for minimizing $x^T x + \mathcal{Q}(x)$ in a weighted sum problem.

i	$c^T x_i$	**Mean(D)**	**Penalty Range**
2	39548.8	217.24	[0.00, 0.77]
8	39611.1	214.74	NA
45	39669.9	59.14	[0.77, 8.21]
33	39782.9	56.94	NA
47	39783.4	45.31	[8.21, 11.24]
88	39940.4	31.34	[11.24, 34.03]
90	40115.0	26.72	NA
91	40254.2	22.12	[34.03, 170.35]
96	40647.7	19.81	[170.35, ∞]

Table 5: Pareto optimal solutions from the stochastic programming approach.

5.3 Comparisons

In this section, we compare the solutions obtained by the bi-criteria approach and the improving solutions of stochastic programming approach found during the flight-pair branching algorithm (note again that none of these solutions is guaranteed to be an optimal solution of the stochastic programme). First we will look at the relationship between the average delay minutes due to crew connections and the non-robustness objective $r^T x$. In order to make an independent comparison of the robustness performance of the solutions obtained by both approaches, we sampled another 100 disruption scenarios. These 100 disruption scenarios are used in conjunction with pushback recovery to evaluate the average delay minutes due to crew connections.

Figure 5 shows the average delay minutes due to crew connections versus the value of $r^T x$.

The solid circles are solutions from the stochastic programming approach and the open squares are solutions from the bi-criteria approach.

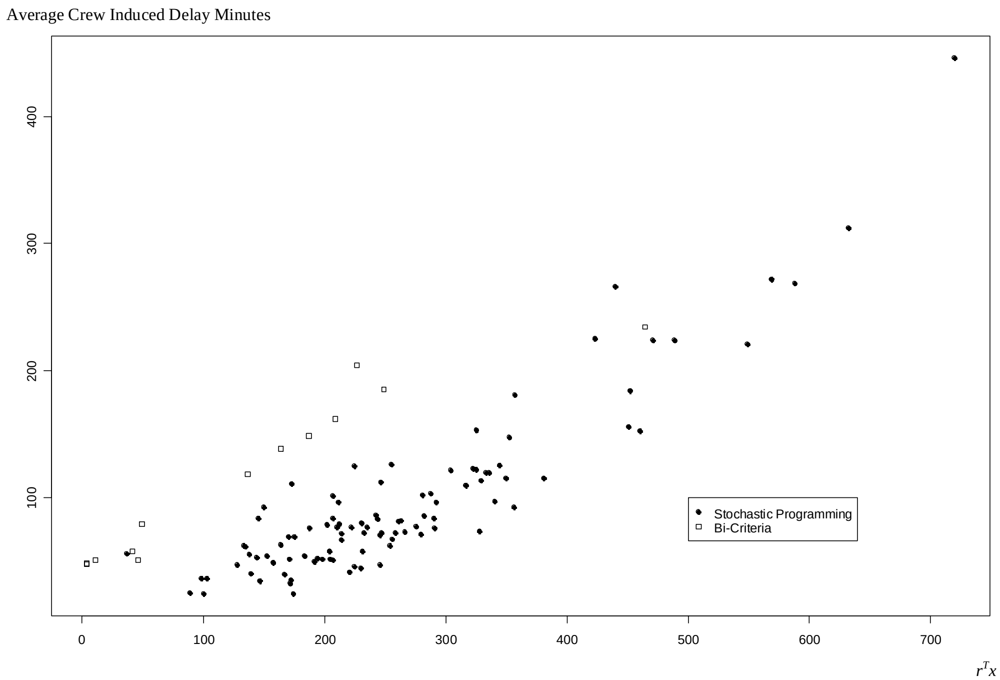

Figure 5: Relation between average crew-induced delay and non-robustness measure.

Apparently there is a positive linear relationship between the average delay minutes due to crew connections and the non-robustess measure. Statistical tests confirmed this relationship ($p < 10^{-7}$). However, for the same value of $r^T x$ solutions from the stochastic programming approach show lower delay minutes. Furthermore, with the same value of average crew-induced delay minutes, solutions from the bi-criteria approach show better values of $r^T x$. These differences are also statistically significant. This result can of course be expected because the two approaches optimize delay minutes and $r^T x$, respectively. The important conclusion is that the positive linear relationship between the two objectives indicates that the non-robustness measure of the bi-criteria approach is a good estimate of the delay caused by crew connections.

Next, we look at the on-time performance of the crew schedules from the two approaches. Figure 6 shows the average delay minutes versus planned crew cost $c^T x$. Again, the solid circles are solutions from the stochastic programming approach and the open squares are solutions from the bi-criteria approach.

This suggests that the bi-criteria approach gives better solutions when the desired cost increase to improve the robustness of the crew schedule is small, and the stochastic programming approach is better when the airline aims to further improve the robustness for the price of higher planned cost. Note that the advantage of the stochastic programming approach is only apparent after 40 flight-pair branches are imposed.

To further compare the degree of robustness between the bi-criteria and stochastic programming approaches, we look at some robustness indicators. The first indicator is the number of connections that are following the aircraft (i.e. crew and aircraft operate the same pair of subsequent flights). Crew induced delay is only caused by crew changing aircraft, hence the more crew connections follow the aircraft, the more robust the solution. The solutions generated by the bi-criteria approach show an increase in the number of aircraft following connections compared to the cost optimal solution, but no significant increase as the non-robustness measure $r^T x$ decreases. The solutions obtained from the stochastic programming approach show a slight increase of aircraft following connections as their non-robustness measure decreases. Comparing the results

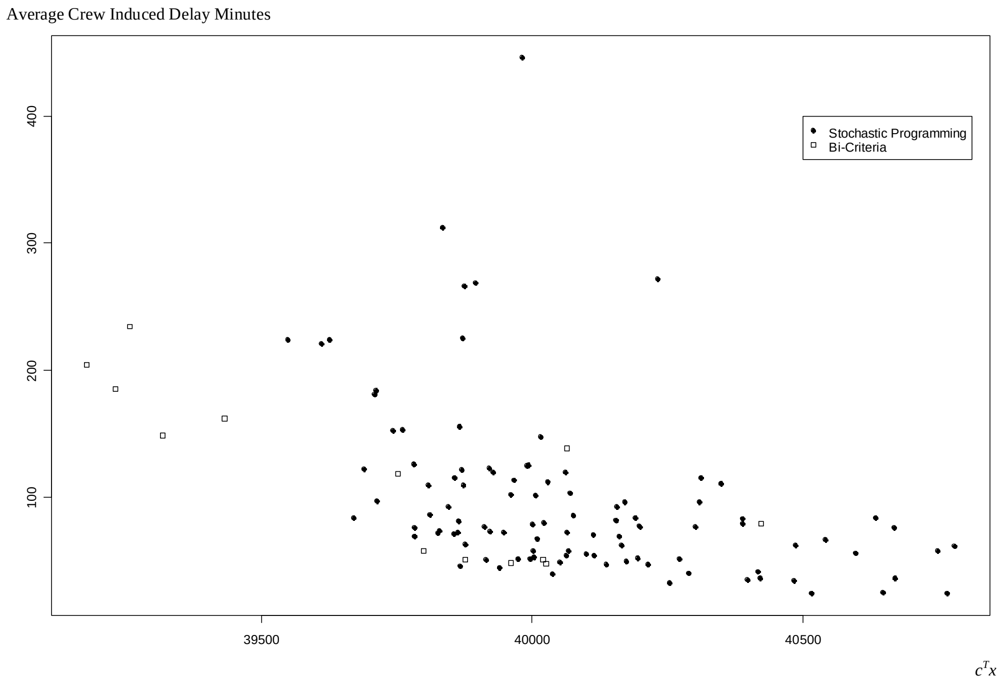

Figure 6: Relation between average crew-induced delay and planned crew cost.

from both approaches, solutions generated by the bi-criteria approach show a higher number of aircraft following connections.

Another robustness indicator is the average ground time. Longer ground time is required when crew need to switch aircraft to obtain a more robust solution. The additional ground time allows compensation of delays of incoming flights. The average ground time increases significantly for solutions generated by the bi-criteria approach compared to the cost optimal solution, and increases further as the non-robustness measure decreases. Solutions obtained in the first few iterations of stochastic programming approach show no difference in average ground time compared to the cost optimal solution and a small increase after more flight-pair branches.

Table 6 gives a summary of the number of aircraft following connections and average ground time. Column 2 lists the number of flight-pair branches imposed for solutions from the stochastic programming approach (SP) and the desired percentage increase in crew cost for solutions from the bi-criteria approach (Bi). The third and fourth columns are the planned crew cost and average crew-induced delay minutes, respectively, while the fifth column is the non-robustness measure. The sixth and seventh columns show the number of aircraft following connections and average ground time.

Another comparison that can be made between the solutions is the *subsequence count*. The *subsequences* of an incoming flight are all possible successor flights that can be operated by the crew up to the scheduled one. The subsequence count of a flight is the number of its subseqences up to the selected one. A schedule is more robust if crew members take later subsequences in case there is a need to switch aircraft. Columns eight to twelve of Table 6 show the subsequence counts for non-aircraft following connections. If a crew is assigned to the 3^{rd} subsequence, this means that the crew is taking the 3^{rd} available outgoing flight after minimum legal ground time.

Solutions from the bi-criteria and stochastic programming approaches both show a decrease of the first subsequence, while the decrease is more significant for the bi-criteria approach. The first subsequence often is the flight that departs immediately or very shortly after ground duty time. In solutions from the bi-criteria approach more second and third subsequences are operated. In solutions from the stochastic programming approach crews are taking fourth or the fifth subsequences in order to improve the robustness.

		$c^T x$	**D**	$r^T x$	**FA**	**GDT**	**Subsequence Count** (in %) 1	2	3	4	5
Cost Optimal		39982.3	446.36	720.26	209	82.36	58.2	27.3	10.9	1.8	1.8
SP	2	39548.8	223.50	471.34	224	83.00	67.5	12.5	12.5	7.5	0.0
	8	39611.1	220.93	549.37	227	80.75	60.0	20.0	12.5	5.0	2.5
	45	39669.9	83.25	207.07	223	108.52	54.5	18.2	13.6	13.6	0.0
	33	39782.9	75.60	290.84	224	110.73	63.4	14.6	9.8	9.8	2.4
	47	39783.4	68.96	174.86	215	117.87	59.6	14.9	14.9	8.5	2.1
	36	39867.4	45.27	224.51	235	102.21	55.9	17.6	8.8	14.7	2.9
	88	39940.4	44.32	229.78	229	113.47	52.8	22.2	8.3	13.9	2.8
	53	40038.7	39.35	167.10	228	120.75	47.5	20.0	15.0	10.0	7.5
	91	40254.2	32.19	172.05	223	111.34	53.7	26.8	7.3	9.8	2.4
	78	40516.1	24.24	174.51	234	113.13	56.3	12.5	9.4	15.6	6.3
	98	40766.3	23.96	99.99	232	115.76	51.5	21.2	15.2	6.1	6.1
Bi	0.10	39176.7	204.45	227.03	234	96.86	54.3	20.0	17.1	0.0	8.6
	0.20	39231.1	185.48	248.94	229	100.00	64.1	20.5	7.7	0.0	7.7
	0.40	39318.4	148.56	186.73	235	92.58	51.5	27.3	12.1	3.0	6.1
	0.60	39751.8	118.14	136.97	239	105.15	45.5	33.3	12.1	6.1	3.0
	0.90	39798.8	57.46	41.08	230	112.98	47.6	33.3	11.9	2.4	4.8
	0.70	39876.4	50.46	45.88	230	110.92	55.3	26.3	13.2	2.6	2.6
	1.10	39960.9	48.01	3.39	231	117.84	40.5	29.7	10.8	13.5	5.4
	1.20	40025.9	47.08	3.39	228	122.26	38.1	38.1	9.5	11.9	2.4

Table 6: Robustness indicators comparison.

We conclude that both the bi-criteria and stochastic programming approaches construct robust solutions by extending ground time when aircraft changes occur and by increasing the number of aircraft following connections. However, the solutions differ in detail, such as the subsequnces taken. These differences in detail account for the differences observed when plotting $c^T x$ versus $\mathcal{Q}(x)$. Together with the linear relationship between $r^T x$ and $\mathcal{Q}(x)$, it is clear that both approaches are valid models for robust crew scheduling. Computational performance, however, favours the bi-criteria approach for finding a range of solutions representing available trade-offs between planned cost and recovery cost with reasonable effort.

6 Conclusion

In this paper we have compared two approaches to the robust airline crew scheduling problem, namely the bi-criteria model of Ehrgott and Ryan (2002) and the stochastic programming model of Yen and Birge (2006). In this comparison we have used real world crew schedules and delay data from a New Zealand domestic airline. We have also used the same ToD optimizer in both solution algorithms for a fair comparison.

While the ideas behind both models differ, we have verified that crew schedules resulting from both models are considerably more robust than those obtained from the traditional model while only slightly increasing planned crew cost. We have confirmed that the deterministic robustness measure used in Ehrgott and Ryan (2002) captures the essential disruption information well. The major drawback of the stochastic programming approach appears to be the larger number of GSPP problems that have to be solved during the flight-pair branching algorithm compared to the bi-criteria approach, so that the latter seems to be the more promising choice for application in practice.

References

Anbil, R., Forrest, J., and Pulleyblank, W. (1998). Column generation and the airline crew pairing problem. *Documenta Mathematica*, **Extra volume ICM III**, 677 – 686.

Barnhart, C., Johnson, E. L., Savelsbergh, M. W. P., and Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, **46**, 316 – 329.

Barnhart, C., Cohn, A., Johnson, E., Klabjan, D., Nemhauser, G., and Vance, P. (2003). Airline crew scheduling. In R. Hall, editor, *Handbook of Transportation Science*, pages 493 – 521. Kluwer Scientific Publishers.

Ehrgott, M. and Ryan, D. (2002). Constructing robust crew schedules with bicriteria optimization. *Journal of Multi-Criteria Decision Analysis*, **11**, 139 – 150.

Ehrgott, M. and Ryan, D. (2003). The method of elastic constraints for multiobjective combinatorial optimization and its application in airline crew scheduling. In T. Tanino, T. Tanaka, and M. Inuiguchi, editors, *Multi-Objective Programming and Goal-Programming*, pages 117 – 122. Springer Verlag, Berlin.

Rosenberger, J. M., Schaefer, A. J., Goldsman, D., Johnson, E. L., Kleywegt, A., and Nemhauser, G. L. (2000). Simair: A stochastic model of airline operations. In J. Joines, P. Barton, P. Fishwick, and K. Kang, editors, *Proceedings of the 2000 Winter Simulation Conference*, pages 1118 – 1122.

Rosenberger, J. M., Schaefer, A. J., Goldsman, D., Johnson, E. L., Kleywegt, A., and Nemhauser, G. L. (2002). A stochastic model of airline operations. *Transportation Science*, **36**, 357 – 377.

Ryan, D. M. and Falkner, J. (1988). On the integer properties of scheduling set partitioning models. *European Journal of Operational Research*, **35**, 442 – 456.

Ryan, D. M. and Foster, B. A. (1981). An integer programming approach to scheduling. In A. Wren, editor, *Computer Scheduling of Public Transport*, pages 269–280. North-Holland, Amsterdam.

Schaefer, A. J., Johnson, E. L., Kelywegt, A. J., and Nemhauser, G. L. (2005). Airline crew scheduling under uncertainty. *Transportation Science*, **39**, 340 – 348.

Shebalov, S. and Klabjan, D. (2006). Robust airline crew pairing: Move-up crews. *Transportation Science*, **40**, 300 – 312.

Tam, B., Ehrgott, M., Ryan, D., and Zakeri, G. (2007). A note on "a stochastic programming approach to the airline crew scheduling problem" by j.w. yen and j.r. birge, transportation science, 40:3-14, 2006. Technical report, Department of Engineering Science, The University of Auckland.

US Department of Transportation (2007). Airline on-time performance data. Bureau of Transportation Statistics `http://www.transtats.bts.gov`.

Yen, J. W. and Birge, J. R. (2006). A stochastic programming approach to the airline crew scheduling problem. *Transportation Science*, **40**, 3 – 14.