The recoverable robust tail assignment problem

Gary Froyland, Stephen J Maher
School of Mathematics and Statistics, University of New South Wales, Sydney NSW 2052, Australia.

Cheng-Lung Wu
School of Aviation, University of New South Wales, Sydney NSW 2052, Australia.

Abstract

Schedule disruptions are commonplace in the airline industry with many flight-delaying events occurring each day. Recently there has been a focus on introducing robustness into airline planning stages to reduce the effect of these disruptions. We propose a recoverable robustness technique as an alternative to robust optimisation to reduce the effect of disruptions and the cost of recovery. We solve the recoverable robust tail assignment problem (RRTAP) using column generation and Benders decomposition, simultaneously solving the optimal planning and recovery problems. The recovery stages of the RRTAP incorporate policies of flight cancellations, delays, and aircraft swapping. To highlight the benefits of the RRTAP we compare our tail assignment solution with the tail assignment generated using a connection cost function presented in Grönkvist. Using airline data we demonstrate that by using a tail assignment developed via the RRTAP framework, one can reduce recovery costs in the event of a disruption.

Key words: robust airline optimisation, recovery, Benders decomposition

1 Introduction

During the airline planning process, various stages are solved in the expectation that operations will be executed as planned. Generally this is not the case, and delays and cancellation of flights are commonplace every day. In August 2011 the on-time performance of the airline industry of Australia averaged 83.7% for departures and 81.7% for arrivals, where on-time is defined as flights departing or arriving within 15 minutes of the scheduled time [7]. For the same period, Europe experienced an on-time performance of 82.48% [8] and the US had 79.34% [25] of all flights on time from reporting carriers. The operations of an airline are susceptible to outside influences such as weather, and in the long run a significant percentage of flights will be disrupted.
It is becoming more accepted by the airline industry that airline operations will be affected by some level of disruption and that this should be accounted for in the planning process. The considerable additional cost associated with any delay has spurred an interest in robust airline planning. These robust plans focus on incorporating features to improve operational performance in the presence of unforeseen disruptions. There are inherent challenges with developing robust plans, many of which stem from the ability (or inability) to identify beneficial plan characteristics to reduce the impact of a disruption. As a result there have been a number of different techniques that have been investigated, each with their own methods of improving an airline’s operational performance.

The complete airline planning process is a large, intractable problem, which is usually broken down into a number of smaller stages. These stages typically consist of, but are not limited to, the schedule design, fleet assignment, aircraft routing, crew planning and passenger routing. Each of the individual stages are difficult problems to solve on their own, which has led to a sequential solution process. Once one stage is completed, the solution is fixed and used in the subsequent stages. Fixing the solution at each step has the benefit of reducing the complexity of the planning process. The disadvantage of this process is that there is limited feedback between the stages, so the optimal solution from one stage may not allow for the globally optimal solution to be found in following stages.

There has been increasing interest in finding an integrated solution for two or more stages in this planning process. Integrating multiple stages in the planning process allows for feedback between these stages to find the most optimal solution. This has the effect of greatly reducing the planning costs and avoiding many infeasibilities. Further there is a belief that integration will provide more robust solutions by directly or indirectly minimising the potential for propagated delay. Cordeau et al. [11] integrates the crew scheduling and aircraft routing problems using Benders decomposition. This integration permits the use of short connections only when the two connecting flights are assigned to the same aircraft and crew. Ensuring that the short connections are used by the same aircraft and crew improves the robustness by reducing the possibility of delays spreading throughout the network, commonly called delay propagation. The Benders decomposition approach to integrated planning is extended by Mercier et al. [22], introducing the concept of restricted connections, which are penalised if the two flights are serviced by the same crew and not the same aircraft. Further integration using Benders decomposition is carried out by Papadakos [24], with the integration of fleet assignment, maintenance routing and crew pairing.

While the above papers use Benders decomposition to integrate the aircraft routing and crew scheduling problems, an iterative approach has been employed by Weide et al. [30] and Dunbar et al. [12]. This approach is applied to handle the combined delay interactions from the routing and crew connection networks with a focus on propagated delay. The iterative approach allows for a robustness measure to be calculated while solving the crew pairing (aircraft routing) problem using a fixed aircraft routing (crew pairing) solution from the previous iteration. Weide et al. [30] attempts to increase the amount of slack,
excess connection time, between the restricted connections to mitigate delay propagation. Dunbar et al. [12] explicitly measures the delay that would be propagated along assigned flight routes, or strings, in the combined routing and crewing network, and minimises this. Both Weide et al. [30] and Dunbar et al. [12] determine that the length of the connections is a contributing factor to the performance of an airline planning solution.

Integration of two or more airline planning processes has been shown to improve robustness, however there have been many alternate methods proposed for individual planning stages. In the case of the aircraft routing problem, Lan et al. [19] propose the use of flight re-timing to reduce delay propagation. The objective of this model is to find an optimal aircraft routing in an attempt to reduce the amount of delay experienced by passengers and missed connections. Borndörfer et al. [6] present an alternative model for the aircraft routing problem, more specifically the tail assignment problem, which aims to find the aircraft routing with the lowest potential propagated delay. In this paper the probability of the length of delay is explicitly modelled, and the expected delay is included in the objective function. A general method of introducing robustness into the tail assignment problem is via key performance indicators, examples of such methods are explained in Wu [31]. Using a set of scenarios, developed from airline data, [6] compare the amount of propagated delay resulting from their model with a tail assignment developed from a traditional key performance indicator method.

Within airline planning problems there are a number of different characteristics identified that can be exploited to improve operational performance. Operational performance can be improved through planning to avoid disruptions or by embedding recovery options within the operational plan. In either case, improved operational performance is measurable through a reduction in recovery costs in the event of a disruption. In the development of a robust airline schedule, Ageeva [2] presents the idea of incorporating aircraft swaps to avoid the propagation of delay. Swapping occurs at points when two aircraft are planned to be on the ground at the same airport at the same time. In the event that one of the aircraft is disrupted, a swap can be made to allow the higher valued route to continue on time. Eggenberg [13] presents both robust and recoverable aircraft routing problems by optimising favourable characteristics of the model to improve recoverability. The aircraft swapping technique from Ageeva [2] is incorporated into robust and recoverable models, in addition to increases in idle time and passenger connection times in an attempt to find a robust aircraft routing through the use of uncertainty feature optimisation. To determine the potential recoverability of the planned solution, Eggenberg [13] solves a recovery problem given a number of different disruption scenarios. Kang [17] proposes a unique method for robust planning involving decomposing the airline schedule into a number of different layers, partitioning the flights by their expected yield. The different layers provide a priority in which flights are protected in a disruption, the highest layer is rerouted first and the lowest layer has a higher delay and cancellation priority. This proposed method is an attempt to improve the recoverability of the planned solution, which is evaluated by testing the solution against disruptions using simulation software. These
papers are attempting to provide options for the airline in the event of disruption, reducing the potential recovery costs.

Some US airline networks are designed with a hub and spoke structure, with the majority of the activity occurring at the hubs. Rosenberger et al. [27] exploits this particular network structure by introducing the concept of hub isolation and short cycles. By limiting the number of aircraft that service each hub in the network, it is possible to isolate a disruption to a particular hub, protecting all other flights. These concepts focus on the possible recovery decisions that could be made, providing the operations controller with a number of low cost recovery options, which is tested with an airline operations simulator. The ideas of Rosenberger et al. [27] are extended by Smith and Johnson [28] to include the concept of “station purity”, which involves limiting the number of fleets that can service each station. When each aircraft within a fleet is interchangeable, the introduction of these features can provide more aircraft swapping opportunities. The integration of this work with crew planning is demonstrated in Gao et al. [15], providing station purity restrictions on each crew base. This station purity on crew bases helps to ensure that each crew can return to base in a disruption since there are compatible aircraft at that station, avoiding costly overnight stays and deadheads.

Each of the above papers present robust models that identify a particular aspect of the airline planning process that is a proxy for robustness. We call this class of robust planning models proxy robust. The performance of these approaches depends on how efficacious the aspect is at capturing the desired robustness goals and can vary across datasets. Furthermore, additions to the model (eg. enlarging the set of possible decisions, or introducing additional constraints) may render a particular proxy robust approach less effective. There is, by definition of this class, no feedback between the planning stage and the operations stage that could improve the robust solution potentially leading to an overly conservative planned solution.

Conversely, other approaches incorporate this feedback via second stage (recovery) decisions, and we call this class feedback robust. Feedback robust approaches are a superior method for reducing the weighted recovery costs since it is possible to improve the planned solution with outcomes from simulated recovery scenarios. The recovery decisions that are made in a simulation stage provide the feedback that will improve the expected operational performance, and as a result reduce recovery costs. Yen and Birge [32] present an example of such an approach solving a robust crew scheduling problem using stochastic programming. In this stochastic programming model each subproblem describes a disruption scenario. The authors evaluate the effect of particular disruption scenarios on the propagated delay caused by crew pairings. The second stage outcomes are used to improve the first stage decisions, while the benefits from using stochastic programming is expressed by the value of the stochastic solution.

Outside the airline literature, Liebchen et al. [20] have developed a concept called recoverable robustness with an application to railway transportation. This technique focuses on finding an optimal planning solution that is recoverable in a limited number of steps. The authors contrast recoverable
robustness with robust planning, indicating that strict robustness can often be overly conservative, requiring a solution to perform under all disruptions. In this work we expand upon the recoverable robust technique in an application to the tail assignment problem. A key feature that we have adopted from Liebchen et al. [20] is the development of a planning solution that is recoverable with limited effort, which we define for the recoverable robust tail assignment problem (RRTAP) as the lowest recovery cost. In addition, the recovery robust timetabling attempts to minimise the changes that are made to the planned timetable in the recovery stages. In our airline application of recoverable robustness we view the objective of minimal deviation as an important aspect of this technique. Since this is a feedback robust approach, the quality of the planned solution is dependent on the feedback provided from the recovery scenarios. In this respect we have extended the recoverable robust technique by including the recovery options of flight cancellations and delays and also allowing for aircraft rerouting. The inclusion of these recovery policies provides an accurate simulation of an airline operations control centre to produce quality feedback for the planning problem. Further, extending the recovery techniques for the RRTAP results in a difficult mixed integer program, which requires a number of enhancement techniques to solve. Recoverable robustness recognises that in a disruption the planned solution will need to be changed in operations, so our goal is to reduce the expected recovery costs and the number of routing changes required.

The RRTAP attempts to improve the planned solution based on information from recovery scenarios. This type of problem can be classified as feedback robust since there is a feedback mechanism between the planning and recovery problems. The proxy robust models incorporate robustness through focusing on perceived beneficial characteristics to the planning problem. There is no guarantee that the selected characteristic will improve robustness, thus can only be evaluated through experiments. The RRTAP focuses on the recoverability of the planned solution, and attempts to reduce the expected cost of recovery for a large range of scenarios. By explicitly solving the recovery problem, the RRTAP is not limited to specific actions and will find the lowest cost recovered solution to a given planned tail assignment. The motivation for using a feedback robust formulation is to allow the model to select the best tail assignment without the restriction of enforcing specific planning characteristics. This approach is similar to the stochastic programming approach applied to the crew pairing problem in Yen and Birge [32]. While Yen and Birge [32] use limited recovery options in the second stage, we have implemented a more complex recovery subproblem in the RRTAP which uses a full set of recovery options, aircraft rerouting and delaying or cancelling flights. Using the full set of options properly simulates the actions of the operations controller, and these superior decisions shape and improve the recoverability of the planning variables in the first stage. To the best of the authors’ knowledge this two-stage approach incorporating a full set of allowable recovery decisions has not been applied to airline optimisation. The resulting planned tail assignment solution will assign flights to aircraft which will reduce the expected cost of cancellation, delays and aircraft swaps.
The RRTAP is a large scale optimisation problem solving the optimal planning and recovery tail assignment problems simultaneously. The aim is to develop a tail assignment that is recoverable across a variety of disruptions, so we use a large number of scenarios to provide feedback to the optimal planning problem. As a result the naïve formulation of the RRTAP is a very large and intractable problem. While it is important to develop a more recoverable tail assignment, it is also critical that this model can be solved within a reasonable time frame. To address this concern we have implemented a number of decomposition and enhancement techniques to improve the running time. Above we have presented robust planning problems that have integrated the decomposition techniques of column generation and Benders decomposition [11, 22, 24]. We have incorporated these ideas including the use of the Magnanti-Wong method [21] and modified them for use in the RRTAP. A common technique used to solve integer programs is branch-and-bound and is commonly performed within the column generation framework in the form of branch-and-price. We identify specific structures in our problem which we have exploited to enhance the branch-and-price process with the development of problem specific branching rules.

Section 2 will describe the concept of recoverable robustness and how we apply it to the tail assignment problem. This section includes a brief review of recovery literature and a description of the policies that have been applied in our model. We also present the mathematical model for the recoverable robust tail assignment problem in Section 2. A description of the techniques used to solve the RRTAP is provided in Section 3. In Section 4 we present our numerical results based on airline data and in Section 5 we report our conclusions.

2 The recoverable robust tail assignment problem - RRTAP

2.1 Recoverable robustness and recovery

We define the tail assignment problem as the task of assigning routes to individual aircraft, ensuring that all flights are serviced in a network while maintaining operational constraints. The RRTAP solves the planning and recovery tail assignment problems simultaneously in a stochastic programming framework to improve the recoverability of the planned solution. In general the tail assignment problem is a feasibility problem with a large number of feasible solutions. Given this, the RRTAP attempts to improve the expected recovery costs of the planned solution without any additional planning costs.

Liebchen et al. [20] describe recoverable robustness as a methodology for producing a solution that is able to be recovered from a set of disruption scenarios, using a restricted number of recovery algorithms, in limited effort. For our model, we take the definition of limited effort to be the lowest cost recovery solution. The costs of recovery include flight delay and cancellation costs and passenger rerouting costs. We also attempt to minimise the deviation of the recovery solution from the optimal planning solution.

There are a number of different techniques that are available to the airline operations control centre to mitigate the effects of the disruption; these are generally in the form of flight delays, cancellations
and aircraft swapping [18]. Due to the high cost of recovery, in the event of a disruption the objective is generally to minimise the use of any recovery techniques. The minimisation problem is constructed by using either real costs, some of which can be very difficult to evaluate, or artificial costs, which allow the user to develop recovery solutions with preferred characteristics [4]. In addition to minimising the use of these recovery techniques there is a focus on returning operations back to plan either within a specified time period or as quickly as possible. A very good, recent review of the recovery literature can be found in Clausen et al. [9].

The feedback decisions for the planning tail assignment problem relies on the recovery techniques that are included in the subproblems. The RRTAP improves upon current feedback robust techniques by fully modelling the tail assignment recovery process. The recovery problems aim to accurately simulate the decisions that are made by the operations control centre by allowing aircraft rerouting and flight cancellations and delays. Through this full tail assignment recovery problem the feedback is able to greatly improve the recoverability of the planned tail assignment problem.

There is an interesting discussion in the recovery literature on what costs to use in the objective function. The majority of the literature uses actual costs in the objective function when valuing delays and cancellations in the model. Thengvall et al. [29] and Andersson [4] suggest using weights for costs given the difficulty in determining the value of the actual costs. The use of weights can allow an operations controller greater freedom in finding the best solution with particular characteristics. For our model we will be using the actual costs for the recovery process since we are attempting to simulate the actions taken in a recovery scenario.

2.2 The mathematical model

We have developed a tail assignment model using a flight string formulation introduced by Barnhart et al. [5]. The notation presented in Table 1 is used to describe the planning and recovery stages in the RRTAP. A superscript \( s \) denotes the components of the model that relate to the recovery stages and the disruption scenario which they belong to from the set of all scenarios \( S \). A flight string, or flight route, \( p \) is defined as a sequence of connected flights to be operated by one aircraft \( r \). The decision variables \( y^p_r \) and \( y^{sr}_r \) equal 1 to indicate the flight string \( p \) that will be operated by aircraft \( r \) for the optimal planning and recovery scenarios respectively. The cost of using flight route \( p \) for aircraft \( r \) is given by \( c^p_r \) and \( c^{sr}_p \) for the optimal planning and recovery scenarios. The cost of a flight route for the optimal planning variables is dependant on the length of the connections contained in the flight route. In the recovery scenarios, the cost of a flight route is defined by the delay on flights contained in the string which is incurred during the recovery process. In the model constraints the parameters \( a^p_{jp} \) and \( a^{sr}_{jp} \) are the coefficients of the decision variables, \( y^p_p \) and \( y^{sr}_p \) respectively, that capture whether flight \( j \) is included in string \( p \). All feasible connections are contained in the set \( C \) and the rules that govern these connections include: i) the destination of the incoming flight must be the same as the origin of the outgoing flight;
and ii) there exists a minimum connection time, called the minimum turn time, between the arrival and departure of the two connecting flights. In addition to describing a set of connected flights, the flight string also indicates end-of-day locations. All end-of-day locations $b$, described as aircraft bases or overnight airports, used in the model are contained in the set $B$. The parameters $o_{bp}$ and $o_{bp}^o$ equal 1 if flight string $p$ terminates at base $b$ for the optimal planning and recovery scenarios respectively. The RRTAP is a single day problem, so to maintain feasibility for the following days’ schedule we enforce a minimum number of aircraft to terminate at each end-of-day location $b$ through the parameter $M_b$. The sequence of flights and the end-of-day location described in the flight string represents a column in the constraint matrix.

The tail assignment problem is the task of assigning flight routes or strings to individual aircraft. Treating each aircraft individually in this problem requires an explicit definition of all aircraft $r$ contained in the set $R$. The set $R$ contains all aircraft used in the model, and this is the same set used for the optimal planning and all recovery stages. We generate individual strings for each aircraft $r$, and additionally we generate strings for the optimal planning and each recovery stage contained in the sets $P_r$ and $P_{sr}$ respectively. As a result we treat the sets $P_r$ and $P_{sr}$ as disjoint.

To solve the recovery tail assignment problem we implement the recovery techniques of flight delays and cancellations while also allowing aircraft rerouting. Flight delays are represented by flight copies, with each copy representing a different departure time of the same flight. All departure times for the flight copies are later than the planned departure. Flight cancellations are included in the model through the additional variables $z_s^j$, which contribute a cost of $d_j$ to the objective, allowing the decision of an aircraft operating a flight or cancelling it.

There are two main objectives for this problem: minimising the cost of recovery and minimising the deviation from the planned solution. Since we are attempting to simulate the recovery process while finding the planned solution, it is important to enforce non-anticipativity. All of the flights $j$ in this model are contained in the set $N$, and to model non-anticipativity we define two partitions of this set, $N^{s-pre}$ and $N^{s-post}$, $N = N^{s-pre} \cup N^{s-post}$. The sets $N^{s-pre}$ and $N^{s-post}$ include all of the flights that depart before and after the first disrupted flight in scenario $s$ respectively. To reduce the number of deviations in the recovery solution we penalise any difference in the flight routes assigned to an individual aircraft through the use of the variables $\epsilon_{jr}^{s+}$ and $\epsilon_{jr}^{s-}$. In the objective function the penalty $g_s^r$ is applied for every flight $j$ that is added to the planned route for aircraft $r$ in the recovered solution for scenario $s$, indicated by $\epsilon_{jr}^{s+} = 0$ and $\epsilon_{jr}^{s-} = 1$. It is possible to add and remove flights from an aircraft’s planned route, however removing a flight is penalised through either cancellation or adding it to another route. The second objective of the RRTAP attempts to find a set of recovered solutions with limited effort; in our case this is defined as the lowest cost recovery solution. As mentioned in Section 2.1, we will be using actual costs for the delay and cancellation of flights in the recovery scenarios.

The recoverable robust tail assignment problem is formed by simultaneously solving the optimal
\[ S \] is the set of all scenarios \( s \),
\[ R \] is the set of all aircraft \( r \),
\[ P^r \] is the set of all strings \( p \) for aircraft \( r \), the optimal planning variables,
\[ P^{sr} \] is the set of all strings \( p \) for aircraft \( r \) in scenario \( s \), the recovery variables,
\[ N \] is the set of all flights \( j \),
\[ N^{s-pre} \] is the set of all flights \( j \) that depart before the first disrupted flight in scenario \( s \),
\[ N^{s-post} \] is the set of all flights \( j \) that depart after and including the first disrupted flight in scenario \( s \),
\[ C \] is the set of all feasible connections in the network, \( C = \{(i, j) | i, j \in N\} \),
\[ B \] is the set of all airports \( b \) an aircraft can terminate its flight route,
\[ M_b \] is the minimum number of aircraft required to start the following days flight from base \( b \),
\[ y^r_p \] = 1 if aircraft \( r \) uses string \( p \), 0 otherwise,
\[ c^r_p \] = the cost of flying aircraft \( r \) on string \( p \),
\[ a^s_{jp} \] = 1 if flight \( j \) is contained in string \( p \), 0 otherwise,
\[ o^s_{bp} \] = 1 if string \( p \) terminates at airport \( b \), 0 otherwise,
\[ y^{sr}_p \] = 1 if aircraft \( r \) uses string \( p \) in scenario \( s \), 0 otherwise,
\[ c^{sr}_p \] = the cost of flying aircraft \( r \) on string \( p \) in scenario \( s \), this includes the cost of any delayed flight on that string,
\[ a^s_{jp} \] = 1 if flight \( j \) is contained in string \( p \) in scenario \( s \), 0 otherwise,
\[ o^s_{bp} \] = 1 if string \( p \) terminates at airport \( b \) in scenario \( s \), 0 otherwise,
\[ d_j \] = the cost of cancelling flight \( j \),
\[ z^s_j \] = 1 if the flight \( j \) is cancelled in scenario \( s \), 0 otherwise,
\[ \epsilon_j^{+} \]
\[ \epsilon_j^{-} \]
\[ \begin{align*}
\epsilon_j^{s+} &= 1, \epsilon_j^{s-} = 0 & \text{if flight } j \text{ is assigned to aircraft } r \text{ for optimal planning} \\
& \text{but not for recovery in scenario } s, \\
\epsilon_j^{s+}, \epsilon_j^{s-} &= 1 & \text{if flight } j \text{ is assigned to aircraft } r \text{ for recovery in scenario } s \\
& \text{but not for optimal planning,} \\
\epsilon_j^{s+} &= 0, \epsilon_j^{s-} = 1 & \text{if flight } j \text{ is not assigned to aircraft } r \text{ for both optimal planning} \\
& \text{and recovery in scenario } s, \\
\end{align*} \]
\[ g^s \] weight applied to \( \epsilon_j^{s-} \) in the objective function, the aircraft swap cost.
\[ w^s \] weight for each scenario \( s \) in the objective function.

Table 1: Notation used in the model.
planning and recovery problems within the one model. We describe the RR TAP as follows,

$$
\min \sum_{r \in R} \sum_{p \in P^r} c_{rp}^p y_p^r + \sum_{s \in S} w^s \left\{ \sum_{r \in R} \sum_{p \in P^r} c_{rp}^s y_p^r + \sum_{j \in N} d_{rj} z_j^s + \sum_{r \in R} \sum_{j \in N} g_j^s z_j^s \right\},
$$

s.t.

$$
\sum_{r \in R} \sum_{p \in P^r} a_{jp} y_p^r = 1 \quad \forall j \in N, \tag{2}
$$

$$
\sum_{p \in P^r} y_p^r \leq 1 \quad \forall r \in R, \tag{3}
$$

$$
\sum_{r \in R} \sum_{p \in P^r} o_{bp}^r y_p^r \geq M_b \quad \forall b \in B, \tag{4}
$$

$$
\sum_{r \in R} \sum_{p \in P^r} a_{jp}^s y_p^r + z_j^s = 1 \quad \forall s \in S, \forall j \in N, \tag{5}
$$

$$
\sum_{p \in P^r} y_p^r \leq 1 \quad \forall s \in S, \forall r \in R, \tag{6}
$$

$$
\sum_{r \in R} \sum_{p \in P^r} o_{bp}^s y_p^r \geq M_b \quad \forall s \in S, \forall b \in B, \tag{7}
$$

$$
\sum_{r \in R} \sum_{p \in P^r^s} a_{jp}^s y_p^r - \sum_{p \in P^r^s} a_{jp}^s y_p^r = 0 \quad \forall s \in S, \forall r \in R, \forall j \in N^{s-pr}, \tag{8}
$$

$$
\sum_{r \in R} \sum_{p \in P^r^s} a_{jp}^s y_p^r - \sum_{p \in P^r^s} a_{jp}^s y_p^r = c_{jr}^{s+} - c_{jr}^{s-} \quad \forall s \in S, \forall r \in R, \forall j \in N^{s-po}, \tag{9}
$$

$$
y_p^r \in \{0, 1\} \quad \forall r \in R, \forall p \in P^r, \tag{10}
$$

$$
y_p^s \in \{0, 1\} \quad \forall s \in S, \forall r \in R, \forall p \in P^r, \tag{11}
$$

$$
z_j^s \in \{0, 1\} \quad \forall s \in S, \forall j \in N, \tag{12}
$$

$$
c_{jr}^{s+} \geq 0, c_{jr}^{s-} \geq 0 \quad \forall s \in S, \forall r \in R, \forall j \in N. \tag{13}
$$

The objective function (1) minimises the cost of the optimal planning tail assignment and the weighted cost of the recovery solutions from all scenarios, weighted by $w^s$, with a penalty for each flight change, $g^s$. Flight coverage in the optimal planning model is enforced through constraints (2) and in the recovery model through constraints (5) with an additional variable, $z_j^s$, to allow for flight cancellations. The restriction on the number of available aircraft is described in constraints (3) and (6) for the optimal planning and recovery scenarios. Also, constraints (4) and (7) ensure the required number of aircraft are positioned at each airport at the end of the day to begin the next days flying in the optimal planning and recovery scenarios. The set of constraints (8) are the non-anticipativity constraints ensuring that each aircraft $r$ is assigned to the same flights in both the planning and recovery stages up to the first disrupted flight for each scenario $s$. Since all scenarios are known ahead of time we require these constraints to reflect decisions that would be made by the airline operations control centre in the event of a disruption. After and including the first disrupted flight the constraints (9) are used to count any deviation in the flight strings assigned to each aircraft for planning and recovery variables in absolute terms. In the objective function we have only included the variable $c_{jr}^{s-}$, which represents whether flight $j$ is added to
the recovered flight route of aircraft $r$.

3 Solution Methodology

There are two main characteristics of this problem that introduce complexity into the solution process for our model. This problem contains a large number of variables representing feasible aircraft routings and a large number of constraints that track the difference between the optimal planning and recovery problems. For airline specific problems with a large variable set, it is very common to apply column generation to generate only a subset of all feasible routings. Column generation is very efficient in this case since the subproblems generally decompose into network flow or shortest path problems, for which there are a variety of solution algorithms. Benders decomposition is useful for decomposable problems with many constraints and is a common technique that is utilised for two-stage stochastic programs, with which this model shares some common characteristics. The recovery scenarios are synonymous with the random scenarios of a stochastic program, each providing feedback to the deterministic master problem. Both decomposition techniques have been implemented for this problem, with the Benders master and subproblems being solved by column generation. A similar method of solution has been implemented by Cordeau et al. [11], Mercier et al. [22] and Papadakos [24] to solve an integrated crew scheduling and aircraft routing problem.

Our particular model is in line with a stochastic programming approach with all of the recovery scenarios making up the Benders subproblems. The Benders master problem is solved without any uncertainty information, which is then revealed through each of the subproblems. By using Benders decomposition we are able to move the difficult constraints, equations (8)-(9), to the subproblem, and by fixing the optimal planning variables, the individual recovery problems are solved more efficiently.

In the RRTAP the planning and recovery tail assignment problems share a similar structure and are solved in comparable ways. By decomposing the model by Benders decomposition the Benders master problem and subproblem need not be similar in any way. This is a great strength in our solution methodology since this allows for the use of any form of planning and recovery problems. The only requirement for each problem description is that the flights allocated to each aircraft are easily identifiable and can be constrained through equations (8)-(9).

In the next sections we will briefly describe how we applied the techniques of Benders decomposition and column generation to the RRTAP. A more in depth description of the solution methodology is provided by Froyland et al. [14].

3.1 Benders Decomposition

The decomposition for this problem is clear given the distinct separation of variables, $y_p^r$, $z_j^s$, $e_j^s$ and $e_j^s_r$, between each of the recovery scenarios, $s \in S$. The Benders decomposition master problem (BMP)
consists only of the optimal planning variables, \( y_{rp} \), \( \forall r \in R, \forall p \in P^r \) and the constraints (2)-(4), with the disruption scenarios \( s \in S \) making up the subproblems (PBSP-\( s \)). At each iteration of the Benders decomposition algorithm a Benders cut will be found from the optimal dual solutions for every scenario \( s \) and introduced to the BMP. The feasibility Benders cuts ensure that PBSP-\( s \) is feasible for the solution of the optimal planning variables, and the optimality cuts provide a lower bound for the objective function of PBSP-\( s \) in the BMP. Each subproblem, PBSP-\( s \), finds an optimal recovery strategy for a given set of optimal planning variables \( \bar{y} \) and a particular disruption scenario \( s \). Since each subproblem represents a specific disruption scenario \( s \), PBSP-\( s \) only includes the recovery string, cancellation and comparison variables, \( y_{srp} \), \( \forall r \in R, \forall p \in P^r \), \( z_{sj} \), \( \forall j \in N \) and \( \epsilon_{jr}^+, \epsilon_{jr}^- \), \( \forall r \in R, \forall j \in N^{s-post} \) respectively, with the constraints (5)-(9).

The primal Benders decomposition subproblem for scenario \( s \) is described by:

\[
\begin{align*}
\min & \quad \mu^s(\bar{y}) = \sum_{r \in R} \sum_{p \in P^r} c_{rp}^s y_{rp}^s + \sum_{j \in N} d_j z_{sj}^s + \sum_{r \in R} \sum_{j \in N} g_j^s \epsilon_{jr}^s, \\
\text{s.t.} & \quad \sum_{r \in R} \sum_{p \in P^r} a_{jp}^s y_{rp}^s + z_{j}^s = 1 \quad \forall j \in N, \\
& \quad \sum_{p \in P^r} y_{rp}^s \leq 1 \quad \forall r \in R, \\
& \quad \sum_{p \in P^r} a_{bp}^s y_{rp}^s \geq M_b \quad \forall b \in B, \\
& \quad \sum_{p \in P^r} a_{jp}^s y_{rp}^s = \sum_{p \in P^r} a_{jp} \bar{y}_{rp}^r \quad \forall r \in R, \forall j \in N^{s-pre}, \\
& \quad \sum_{p \in P^r} a_{jp}^s y_{rp}^s + \epsilon_{jr}^+ - \epsilon_{jr}^- = \sum_{p \in P^r} a_{jp} \bar{y}_{rp}^r \quad \forall r \in R, \forall j \in N^{s-post}, \\
& \quad y_{rp}^s \geq 0 \quad \forall r \in R, \forall p \in P^r, \\
& \quad z_{j}^s \geq 0 \quad \forall j \in N, \\
& \quad \epsilon_{jr}^+ \geq 0, \epsilon_{jr}^- \geq 0 \quad \forall r \in R, \forall j \in N.
\end{align*}
\]

The optimal solution for BMP is given by the set of variables \( \bar{y} \), which are fixed and passed to

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Phi )</td>
<td>the objective function value of the Benders master problem (BMP),</td>
</tr>
<tr>
<td>( \varphi^s )</td>
<td>the decision variable added to the objective function of BMP and included in the optimality cuts from each of the scenario subproblems (PBSP-( s )), ( \forall s \in S ),</td>
</tr>
<tr>
<td>( \bar{y} )</td>
<td>is the solution for the optimal planning variables in BMP,</td>
</tr>
<tr>
<td>( \mu^s(\bar{y}) )</td>
<td>is the objective function value of PBSP-( s ) for a fixed optimal planning solution ( \bar{y} ), ( \forall s \in S ),</td>
</tr>
<tr>
<td>( \Omega^s )</td>
<td>the set of all Benders optimality cuts ( \omega ) for scenario ( s ) added to BMP.</td>
</tr>
</tbody>
</table>

Table 2: Additional notation for the Benders decomposition model.
PBSP-s for each scenario \( s \). To ensure that PBSP-s is always feasible, an initial set of strings, \( p^{0r} \in P^{sr} \), are generated by replicating the routes from the optimal master problem variables, \( \bar{y} \), for all flights \( j \) contained in \( N^{s-pre} \), \( a_{jp}^{0r} \bar{y}_{pr} = a_{jp}^{s} y_{pr}^{0s} \forall j \in N^{s-pre}, \forall r \in R, \forall p \in P^{r} \). This satisfies the cover constraints (15) since we are able to set \( z_{j}^{*} = 1, \forall j \in N^{s-post} \) and the non-anticipativity constraints (18). These initial strings provide a starting basis in the column generation algorithm for solving the PBSP-s. We define the dual variables as \( \beta^{s} = \{ \beta_{j}^{s} | \forall j \in N \}, \gamma^{s} = \{ \gamma_{\bar{r}}^{s} | \forall r \in R \}, \lambda^{s} = \{ \lambda_{b}^{s} | \forall b \in B \}, \) and \( \delta^{s} = \{ \delta_{j}^{s} | \forall r \in R, \forall j \in N \} \) for the constraints (15), (16), (17), and (18)-(19) respectively. For each scenario \( s \), after solving PBSP-s the Benders cuts are generated from the dual solutions of (15)-(19). The resulting Benders optimality cut for an iteration of the Benders decomposition algorithm are defined as,

\[
\varphi^{s} \geq \sum_{j \in N} \beta_{j}^{s} + \sum_{r \in R} \gamma_{\bar{r}}^{s} + \sum_{b \in B} \lambda_{b}^{s} M_{b} + \sum_{r \in R} \sum_{j \in N} \sum_{p \in P^{r}} \delta_{j}^{s} \alpha_{jp} \bar{y}_{pr} \tag{23}
\]

For a fixed \( \bar{y} \), the right hand side of the Benders optimality cut, equation (23), is the objective function value for the dual of PBSP-s. The dual solutions of (15)-(19) express an extreme point of the dual problem of PBSP-s. Since the PBSP-s is always feasible for any \( \bar{y} \) the cuts that will be added to BMP will only be optimality cuts, given by equation (23).

The quality of the cuts generated for the PBSP-s has a significant effect on the efficiency of the Benders decomposition algorithm. Given the structure of the PBSP-s, it is common for the final primal solution to be degenerate which results in multiple dual solutions. In the case of a degenerate primal solution we have implemented the Magnanti-Wong method [21] to find the Pareto optimal cuts from the set of multiple dual solutions. This method involves solving an auxiliary problem (DMWAP-s) for each scenario \( s \) when the primal solution of PBSP-s is degenerate. We define the dual Magnanti-Wong auxiliary problem as,

\[
\begin{align*}
\text{max} & \quad \sum_{j \in N} \beta_{j}^{s} + \sum_{r \in R} \gamma_{\bar{r}}^{s} + \sum_{b \in B} \lambda_{b}^{s} M_{b} + \sum_{r \in R} \sum_{j \in N} \sum_{p \in P^{r}} \delta_{j}^{s} \alpha_{jp} \bar{y}_{pr} \\
\text{s.t.} & \quad \sum_{j \in N} \beta_{j}^{s} + \sum_{r \in R} \gamma_{\bar{r}}^{s} + \sum_{b \in B} \lambda_{b}^{s} M_{b} + \sum_{r \in R} \sum_{j \in N} \sum_{p \in P^{r}} \delta_{j}^{s} \alpha_{jp} \bar{y}_{pr} = \mu^{s}(\bar{y}), \tag{24}
\end{align*}
\]

\[
(\beta^{s}, \gamma^{s}, \lambda^{s}, \delta^{s}) \in \Delta^{s}, \tag{25}
\]

where \( \Delta^{s} \) is the dual solution feasibility space from the PBSP-s and its objective function value is given by \( \mu^{s}(\bar{y}) \). To find the Pareto optimal cut we define a core point \( y^{0} \) that is within the relative interior of the LP relaxation of (28) - (33), \( y^{0} \in \text{ri}(y^{LP}) \).

To apply the Benders cuts from PBSP-s to BMP an additional decision variable \( \varphi^{s} \) must be added to the master problem objective function. The value of \( \varphi^{s} \) in the solution of BMP provides the current lower bound of the objective function for PBSP-s in the master problem, constrained through the added cuts. In the solution process of the Benders decomposition scheme we also introduce a new set \( \Omega^{s} \), which is the set of all Benders cuts \( \omega \) for an individual scenario \( s \). Each Benders cut \( \omega \) is defined by the dual variables \( \beta^{\omega s}, \gamma^{\omega s}, \lambda^{\omega s} \) and \( \delta^{\omega s} \) from PBSP-s for disruption scenario \( s \). The Benders decomposition
master problem is described as follows,

$$\begin{align*}
\min & \quad \Phi = \sum_{r \in R} \sum_{p \in P^r} c_p^r y_p^r + \sum_{s \in S} w^s \varphi^s, \\
\text{s.t.} & \quad \sum_{r \in R} \sum_{p \in P^r} a_{jp}^r y_p^r = 1 \quad \forall j \in N, \\
& \quad \sum_{p \in P^r} y_p^r \leq 1 \quad \forall r \in R, \\
& \quad \sum_{r \in R} \sum_{p \in P^r} o_{bp}^r y_p^r \geq M^b \quad \forall b \in B, \\
& \quad \varphi^s - \sum_{r \in R} \sum_{j \in N} \sum_{p \in P^r} \delta_{jpr} y_p^r \geq \sum_{j \in N} \beta_{js}^s + \sum_{r \in R} \gamma_{sr}^s + \sum_{b \in B} \lambda_{bs}^s M^b, \\
& \quad \forall s \in S, \forall \omega \in \Omega^s, \\
& \quad y_p^r \in \mathbb{Z}^+ \quad \forall r \in R, \forall p \in P^r, \\
& \quad \varphi^s \geq 0 \quad \forall s \in S.
\end{align*}$$

(BMP)

The Benders decomposition algorithm terminates when there are no more cuts that can be added to the BMP from the PBSP-s. The criteria to determine whether a cut will be added to the BMP is given by the difference between the upper and lower bounds for the PBSP-s relative to the master problem objective function $\Phi$. The lower bound for each scenario $s$ is provided by the value of $\varphi^s$ in the solution to BMP and the upper bound is the objective function value of PBSP-s for a given $\bar{y}, \mu^s(\bar{y})$, at each iteration. A cut will be added to the BMP from the solution of PBSP-s if the condition,

$$\frac{\mu^s(\bar{y}) - \varphi^s}{\Phi} < \varepsilon \quad \forall s \in S,$$

is violated. This condition is similar to the condition proposed in Papadakos [24] as the stopping criteria. We define $\varepsilon$ as the stopping condition tolerance for our model, set at $\varepsilon = 10^{-4}$. The Benders master problem is solved when no improvement can be made with further cuts from the PBSP-s; this is equivalent to condition (34) being satisfied for all $s \in S$.

### 3.2 Column Generation

Given the exponentially large number of variables in the Benders master (BMP) and subproblems (PBSP-s), both are solved using column generation. Each of these problems share a similar structure and the column generation subproblems are solved using the same algorithm, so for conciseness we will only describe in detail the implementation of column generation for the PBSP-s.

The PBSP-s is formulated as a LP and can be efficiently solved using column generation. Each iteration of the column generation method improves the master problem by introducing negative reduced cost columns generated from the subproblem using the current LP dual solutions of the PBSP-s. Using the dual variables defined in Section 3.1 for the PBSP-s, the reduced cost of the recovery variable $p$ for
aircraft \( r \) in scenario \( s \) is given by,

\[
\bar{c}_{sr}^p = c_{sr}^p - \sum_{j \in N} a_{jp}^s \beta_j^s - \gamma_{sr} - \sum_{b \in B} \alpha_{sb}^p \lambda_b^s - \sum_{j \in N} a_{js}^r \delta_{sr}^j, \quad \forall s \in S, \forall r \in R.
\] (35)

The purpose of the column generation subproblem is to find a flight route, or string, \( p \) for aircraft \( r \) with a negative reduced cost as defined by equation (35). As a result the structure of the column generation subproblem is in the form of a shortest path problem. We have implemented a pulling algorithm, described in Algorithm 1, to find the shortest path for each aircraft using equation (35) as the objective function. Since the network we are using for the shortest path problem is a directed acyclic graph it is possible to construct a topological ordering, which improves the efficiency of the algorithm. A topological ordering is defined as a list where node \( i \) is ordered before node \( j \) if \( \exists (i, j) \in C \) [3].

In Table 3 we introduce the parameters and variables that are used in Algorithm 1. We identify each of the nodes within the network by the indices \( i \) and \( j \), with the arc connecting the two nodes labelled as \( (i, j) \). In the network there are multiple source and sink nodes representing the airports which aircraft overnight at. Algorithm 1 finds the shortest path for a single aircraft so we have included only one source node, which is labelled \( -1 \) for convenience, and multiple sink nodes. In processing each node \( i \) we store the distance of the shortest path to a node \( j \) with the variable \( \text{dist}(j) \). We also record the shortest path to node \( j \) by storing only the node \( i \) immediately preceding \( j \) in that path, by setting \( \text{prev}(j) \leftarrow i \). The shortest path from the source node to a sink node is given by a set of nodes \( j \) and a set of connections \( (i, j) \). Using the cost for each connection \( (i, j) \), \( \text{cost}(i, j) \), the distance of the shortest path is calculated as the sum of \( \text{cost}(i, j) \) for all connections \( (i, j) \) that exist in the path.

The main feature of Algorithm 1 is to process each node in the network and update the shortest path at each connected node if required. The processing of each node involves calculating the shortest path from the current node \( i \) to each connected node \( j \) given by \( \text{dist}(i) + \text{cost}(i, j) \). If the cost of the shortest path to node \( j \) is greater than the cost of the path to node \( i \) with the addition of the connection cost \( \text{cost}(i, j) \), \( \text{dist}(j) > \text{dist}(i) + \text{cost}(i, j) \), then the shortest path at node \( j \) is updated, \( \text{dist}(j) \leftarrow \text{dist}(i) + \text{cost}(i, j) \). The algorithm terminates when all nodes are processed.

There is little difference in the column generation subproblem for BMP and PBSP-s since Algorithm 1 is identical for both. For the BMP we define the optimal dual solutions as \( u = \{u_j | \forall j \in N \} \),

| \( \text{dist}(j) \) | is the distance stored at node \( j \), |
| \( \text{dist}(-1) \) | is the distance stored at the source node, set to 0, |
| \( \text{prev}(j) \) | is the node, \( i \), previous to node \( j \) in the shortest path to \( j \), \( (i, j) \in C \), |
| \( \text{prev}(-1) \) | is the previous node to the source node, set to \( -1 \), |
| \( \text{cost}(i, j) \) | is the cost of using the connection \( (i, j) \in C \). |

Table 3: Definitions for variables used in Algorithm 1.
Algorithm 1 Shortest path on an acyclic network

Set $\text{dist}(-1) \leftarrow 0$ and $\text{prev}(-1) \leftarrow -1$ for the source node,
set $\text{dist}(i) \leftarrow \infty$ and $\text{prev}(i) \leftarrow -2$ for all nodes $i \in N$.

for all nodes, $i$, in the topologically sorted list do

for all nodes, $j$, such that $(i, j) \in C$ do

Set $\text{tempDistance} \leftarrow \text{dist}(i) + \text{cost}(i, j)$.
if $\text{tempDistance} < \text{dist}(j)$ then

Set $\text{dist}(j) \leftarrow \text{tempDistance},$
set $\text{prev}(j) \leftarrow i$.
end if
end for
end for

Let $i$ be the sink node.

while $\text{prev}(i)$ is not the source node do

Add $\text{prev}(i)$ to the shortest path,
set $i \leftarrow \text{prev}(i)$.
end while

$v = \{v^r | \forall r \in R\}, w = \{w_b | \forall b \in B\},$ and $\rho = \{\rho^{s\omega} | \forall s \in S, \forall \omega \in \Omega^s\}$ for the constraints (28)-(31) respectively. As a result the reduced cost of a variable $p$ for aircraft $r$ in the BMP is given by,

$$\bar{c}^r_p = c^r_p - \sum_{j \in N} a^r_j p_j - \sum_{b \in B} a^r_b w_b - \sum_{s \in S} \sum_{\omega \in \Omega^s} \left\{ \sum_{j \in N} \delta^r_{s\omega j} a^r_j \right\} \rho^{s\omega}, \quad \forall r \in R. \quad (36)$$

The only difference in the column generation algorithm between the BMP and PBSP-s is the objective function used in Algorithm 1, equations (36) and (35) respectively.

3.3 The two-phase algorithm

Given the size of the Benders master problem, it is computationally difficult to solve to integral optimality for every iteration. To overcome this complication we have implemented a two-phase algorithm which is based off the three-phase algorithm developed to solve the integrated crew scheduling and aircraft routing problem with Benders decomposition [11, 22, 24]. The two-phase algorithm is a heuristic that initially solves the linear relaxation of the RRTAP, and re-introduces the integrality requirements to the BMP after the first phase is completed, as described in Algorithm 2. In Cordeau et. al. [11], Mercier et. al. [22] and Papadakos [24] the third phase is used to check the feasibility of adding integrality to the Benders subproblem after solving the integral Benders master problem. This is not necessary for our model since for all scenarios $s$ the PBSP-s is always feasible for any solution to the master problem, as explained in Section 3.1. Thus, it is only necessary to implement the two-phase algorithm for our model.
Algorithm 2 The two-phase algorithm

**PHASE 1**

Relax the integrality requirements for the BMP and PBSP-s \( \forall s \in S \).

Set \( \varphi^* \leftarrow 0, \forall s \in S \).

repeat

Solve the BMP using column generation, (27)-(33).

for all scenarios \( s \in S \) do

Solve the PBSP-s using column generation, (14)-(22).

if condition (34) is not satisfied then

if solution to PBSP-s is degenerate then

Use the Magnanti-Wong method to find the Pareto optimal Benders cut.

end if

Add cut to the BMP, of type (23).

end if

end for

until condition (34) is satisfied, \( \forall s \in S \).

**PHASE 2**

Re-introduce the integrality requirements for the BMP.

Retain all cuts that have been added in **PHASE 1**.

repeat

Solve the BMP using column generation, (27)-(33).

for all scenarios \( s \in S \) do

Solve the PBSP-s using column generation, (14)-(22).

if condition (34) is not satisfied then

if solution to PBSP-s is degenerate then

Use the Magnanti-Wong method to find the Pareto optimal Benders cut.

end if

Add cut to the BMP, of type (23).

end if

end for

until condition (34) is satisfied, \( \forall s \in S \).
4 Computational Experiments

To evaluate the benefit of the RRTAP we take the integral BMP solution and evaluate the recovery costs by solving the PBSP-s to integrality. Given the structure of the two-phase algorithm it is possible to evaluate the solution at the completion of both phases. Using the solution to the BMP at the completion of Phase 1 can provide a good upper bound on the recovery solutions, and since it is solved purely as an LP the solution times are very fast. To find this upper bound, the BMP is solved once to integrality using all the cuts added to the BMP by the completion Phase 1. This provides an optimal planning solution to evaluate by solving for all scenarios using the PBSP-s. For all scenarios s, the PBSP-s is then solved to integrality to find the current best recovery solution to the RRTAP. The cuts which are added in Phase 2 tighten this upper bound, however in Section 4 we demonstrate the Phase 1 bound is very close to the optimal solution.

It is not necessary to include the Magnanti-Wong method [21] in Algorithm 2, however we have found great computational benefit from its use each time the PBSP-s is solved as demonstrated in Section 4.3. Another possible technique to improve the computational performance of the Benders decomposition algorithm is to implement the independent Maganati-Wong method [23], which is solved to find cuts independent of the subproblem solution. In Papadakos [23], the benefit of using the independent Magnanti-Wong method is to generate initial cuts that provide a good lower bound from the master problem without having to solve the Benders subproblems. Since our problem requires a large number of cuts to find the optimal solution, the addition of one initial cut does not create a significant enough improvement in the BMP. As a result we have chosen to not include the independent Magnanti-Wong method [23] and only implement the original Magnanti-Wong method [21].

4 Computational Experiments

To test the effectiveness of using the recoverable robustness technique we compare the difference in cost for a simulated recovery scenario between our solution and a representative proxy robust algorithm. We develop a proxy robust algorithm using the connection cost function of Grönkvist [16] in defining the cost of a flight string $c_p$. The representative proxy robust model consists of the objective function $\sum_{r \in R} \sum_{p \in P^r} c_p y^r_p$ and the constraints (2)-(4) and (10) which is solved to find an optimal tail assignment.

The connection cost function presented in the PhD thesis of Grönkvist [16] is a simple but effective method to improve the robustness of the tail assignment by assigning costs to flight connections based on their length. We illustrate the connection cost function implemented for our representative proxy robust model and in the first stage of our recoverable robust model in Figure 1. It is common in industry to increase the utilisation of the aircraft to generate as much revenue as possible. Grönkvist argues that very short connections, $t_{\min} \leq t \leq t_{\text{start}}$, while ideal in regards to aircraft utilisation, are more prone to propagating delay, and suggests that a compromise ideal connection length of $t = 120$. The medium length connections, $t_{\text{lower}} \leq t \leq t_{\text{upper}}$, are penalised heavily since they do not provide high
enough utilisation and are too short to service extra flights in a recovery situation. The long connections, $t \geq t_{end}$, are also favoured since in a recovery situation the aircraft can be used to service additional flights within that connection time.

![Connection cost function](image)

Figure 1: Connection cost function presented in Grönkvist [16]. Time parameters (minutes): $t_{min} = 40, t_{start} = 120, t_{lower} = 180, t_{upper} = 300, t_{end} = 360$. Cost Parameters ($\$$): $c_{start} = 500, c_{lower} = 100, c_{upper} = 5000$.

A key indicator of the operational performance of a planned tail assignment is the amount of additional cost that is incurred through disruption management. We evaluate the effectiveness of the Grönkvist connection cost function by individually optimising for each disruption scenario its recovery decisions using the PBSP-s. The resulting cost indicates the recovery performance of the Grönkvist planning solution. We expect the results to already be very good, given the intelligent choice of objective function and exact optimisation of recovery. Indeed, such an exact quantification of the performance of the Grönkvist solution by explicitly determining the optimal recovery strategies and evaluating the recovery costs, has to our knowledge not been carried out. It is against this representative proxy robust model that we review the performance of the RRTAP. We compare the weighted recovery costs and the constituent costs for aircraft swaps, cancellations and delay minutes in our experiments.

A major benefit to the RRTAP is that the weighted recovery cost of the final solution will be no worse than that of the solution to the model used in the BMP. So there is always potential benefits in applying the recoverable robustness technique to any model.

We implemented this model in C++ and called SCIP 2.0.1 [1] to solve the integer program using CPLEX 12.2 as the linear programming solver.

### 4.1 Description of scenarios and model parameters

The test data for this model consists of 53 flights with 341 feasible connections in a domestic network operating with 3 major airports, serviced by 10 aircraft. Two different types of disruption scenarios have been implemented, which are airport closures and aircraft grounding, resulting in 102 scenarios. We have
selected these types of scenarios given their impact on the operational performance of the tail assignment problem. The airport closures present the model with a large disruption scenario where returning to plan as quickly as possible is a high priority. Further the aircraft grounding scenarios directly affect the aircraft operations, as such are an important scenario type. The specifics of the disruption scenarios are presented in Table 4.

<table>
<thead>
<tr>
<th>Type</th>
<th>Affected</th>
<th>Start Time</th>
<th>Duration</th>
<th>Weight $w_s$</th>
<th>Scenarios</th>
</tr>
</thead>
<tbody>
<tr>
<td>Airport Closure</td>
<td>One scenario for each major airport</td>
<td>6am</td>
<td>180min</td>
<td>0.07%</td>
<td>0-2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12pm</td>
<td>300min</td>
<td>0.03%</td>
<td>3-5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>180min</td>
<td>1.4%</td>
<td>6-8</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>300min</td>
<td>0.6%</td>
<td>9-11</td>
</tr>
<tr>
<td>Aircraft Grounding</td>
<td>One scenario for each aircraft</td>
<td>6am</td>
<td>60min</td>
<td>3.5%</td>
<td>12-21</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>120min</td>
<td>0.7%</td>
<td>22-31</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>240min</td>
<td>0.14%</td>
<td>32-41</td>
</tr>
<tr>
<td>Aircraft Grounding</td>
<td>One scenario for each aircraft</td>
<td>12pm</td>
<td>60min</td>
<td>3.5%</td>
<td>42-51</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>120min</td>
<td>0.7%</td>
<td>52-61</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>240min</td>
<td>0.14%</td>
<td>62-71</td>
</tr>
<tr>
<td>Aircraft Grounding</td>
<td>One scenario for each aircraft</td>
<td>5pm</td>
<td>60min</td>
<td>3.5%</td>
<td>72-81</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>120min</td>
<td>0.7%</td>
<td>82-91</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>240min</td>
<td>0.14%</td>
<td>92-101</td>
</tr>
</tbody>
</table>

Table 4: Disruption scenarios used.

We estimate the relative probability of each of the above scenarios occurring in a single day and encode these probabilities as the weights, $w^*$, in equations (1) and (27). There are a number of different disruptions that could affect the operations of an airline, and within our model we only include a subset of them. Given that each of these scenarios are not mutually exclusive the sum of the probabilities assigned to each scenario do not equate to 1. To determine the probability of an airport closure we assume that this schedule is a summer schedule, so it is more likely for an afternoon closure to occur than a morning closure. In the summer there is very little chance of fog, which is the main contributor to morning airport closures. Also, in summer, severe storms characterised by high winds and regular lightning strikes generally occur in the afternoon. We estimate that in a season a single airport may experience an afternoon closure approximately 3-4 times, so we assign a daily probability of 2% for a closure of any length. Similarly, we expect that there is little chance of a morning closure during the summer season so we assign a probability of 0.1% for a closure of any length. Further, we have estimated that in the case of an airport closure there is a 70% chance that it will last for 180min, and a 30% chance
that it will last for 300min. For example we estimate that an afternoon airport closure for 300min will occur with a probability of $2\% \times 30\% = 0.6\%$.

An aircraft grounding could be attributed to a number of different factors, which include technical issues, delays in the cleaning of an aircraft or baggage loading/unloading issues. The scenarios represent the situation when an aircraft is not ready for a scheduled departure. Using data for US airlines published by the Bureau of Transportation Statistics [26], in August 2011 approximately 18% of all flights were delayed and out of all flight delays approximately 26% are caused by factors in the airlines control, so $18\% \times 26\% = 4.68\%$ of all flights are delayed due to airline factors. We assume that an aircraft grounding causes flight delays due to airline factors, so for a single aircraft grounded for 60, 120 and 240min we assign the probabilities of 3.5%, 0.7% and 0.14% respectively. Now, $3.5\% + 0.7\% + 0.14\% = 4.34\%$ which is less than 4.68%, the percentage of all flight delayed due to airline factors, and this difference occurs since we are approximating the rate of delay which can vary from month to month.

To determine the lowest cost recovery solution we have assigned costs for each minute delayed and for flight cancellations. As mentioned in Section 2.1, we aim to use actual costs in evaluating our model. It is very difficult to determine the actual costs for both delays and flight cancellations since there is an unknown component of lost revenues. The delay costs are set at $100 AUD per minute for a full aircraft, which is based off the EUROCONTROL report by Cook et al. [10] which estimates the cost of delays at €74 per minute. The cancellation cost per passenger for all flights in the network is estimated using an average ticket price of $350 multiplied by a lost revenue parameter or ‘loss rate’. The loss rate would be an airline specific value indicating the expected amount of passenger recapture after a cancellation. A loss rate less than 1 indicates that a percentage of passengers are recaptured by rebooking themselves on another provided flight, whereas a value greater than 1 represents the loss of all passengers and possible future bookings with the airline. The inclusion of a loss rate is an attempt to capture the direct and indirect costs, such as lost revenues and loss of goodwill respectively, associated with the cancellation of a flight. Since the loss rate is very difficult to estimate we have presented our results using a set of values ranging from 0.01 to 3 to provide a broad test of our model. Both the average ticket price and loss rate could be flight specific and the implementation of this is trivial.

In determining the cost of flight delays and cancellations we assume that the aircraft are at 75% capacity. Since the aircraft is not booked to capacity we have developed a simple method for calculating the cost of cancellations for each individual flight. We assume that one third of the passengers on the cancelled flight will be rebooked to the next available flight with the same O-D pair at a cost of 25% of $100 AUD per minute to the next departure. This cost of rebooking passengers onto the next flight is simply the cost of the delay experienced in waiting for the next departure. The rest of the passengers on the cancelled flight do not get rebooked and as a result the revenue is lost. The revenue that is lost from the cancelled flight is calculated from this proportion of passengers. Since the passengers are not accommodated on the next available flight, there is the option of rebooking themselves with this airline
or another. This rebooking process is captured in the lost revenue parameter, the ‘loss rate’. Also, the delay cost per flight given in the EUROCONTROL report [10] is based off a full aircraft and given that we are assuming a 75% capacity the delay cost of an aircraft per minute in our model is $75.

As mentioned in Section 2.2 we handle the flight delays by including a set of flight copies in the recovery network. For our experiments we have used a maximum allowable delay of 180min, with flight copies for every 30min. Given that the delay increment is 30min, we will be over estimating the delay costs, since many shorter feasible connections exist in the delay window for each flight. Greater granularity is possible by decreasing the delay copy increment, however this degrades the computational performance with the addition of more flight copy arcs to the connection network.

4.2 Comparison of recoverable robust solutions and Grönkvist solution

The RRTAP attempts to reduced the weighted recovery cost of the planned tail assignment while reducing the number of changes required. To represent the difficulty faced by operations controllers to reroute aircraft we use an aircraft swap penalty parameter $g^s$. The aircraft swap penalty is also used to reduce the number of changes that are performed in a disruption scenario, a key aspect of the RRTAP. This penalty parameter can be likened to a cap on the number of allowable changes during a disruption, however a cap is more restrictive than a penalty. In our model, low swap penalties result in more changes made in the recovered solution, which provides a lower recovery cost. We have found that the lower recovery cost occurs because greater flexibility is allowed in the model, so it is possible to reroute more aircraft to avoid costly delays and cancellations. As a review of the different trade-offs and outcomes we present our results with aircraft swap penalties in the range $10 \leq g^s \leq 10000$. An aircraft swap penalty of 10 illustrates the case where there is virtually no penalty for swapping aircraft. It is also trivial to implement this parameter with different values for each flight or aircraft in the model.

We are using the Grönkvist connection cost function in the BMP of the recoverable robust model to investigate whether our algorithm can provide superior recovered solutions when compared to the proxy robust model. The costs presented in Figure 2 are the weighted sum of the recovery costs over all scenarios. This is defined as

$$\text{WeightedCost} = \sum_{s \in S} w^s \left\{ \sum_{r \in R, p \in P^r} c^w_{rp} y_{wr} + \sum_{j \in N} d_j z^s_j + \sum_{r \in R, j \in N} g^s \epsilon_{jr}^s \right\},$$

which is the second term in the objective function (1). We use equation (37) to calculate the weighted simulated recovery cost using the solution from the proxy robust model and the solution at the completion of Phase 1 and 2 of the two-phase algorithm, Algorithm 2. We compare the weighted recovery costs calculated with different cancellation loss rates and penalty weights. For all parameter sets, the weighted recovery cost of the recoverable robust solution at the completion of Phase 2 either equals or better the proxy robust solution.
A very important feature of the RRTAP is the ability to reduce the weighted recovery costs when compared to the proxy robust model, and in general to any planning tail assignment model. Since the tail assignment can be formulated as a feasibility problem, any improvement in the recoverability of the planned solution is made at no extra cost. In Table 5 we present the percentage improvement in the weighted recovery costs at the completion of Phase 1 and 2. The largest improvement in the weighted recovery costs achieved at the completion of Phase 2 for a specific penalty weight occurs with \( g^* = 1000 \) and a loss rate \( \geq 0.25 \), a 14.83% cost reduction is achieved with a loss rate equal to 1. There is an apparent nonlinear relationship between recoverable robustness improvement and penalty weight and the greatest improvement occurs with weights in the range \( 1000 < g^* < 5000 \). In that range and with a loss rate \( \geq 0.25 \) the RRTAP achieves an average improvement over the proxy robust solution of 8.77%, with a minimum improvement of 1.21%.

The results in Figure 2 and Table 5 demonstrate that the RRTAP improves upon the weighted recovery costs of the proxy robust solution in most cases. To illustrate the performance of the RRTAP solution when compared to the proxy robust solution for each individual recovery scenario we have selected 4 representative cases; i) Penalty = 500, Loss Rate = 0.5; ii) Penalty = 1000, Loss Rate = 1.5; iii) Penalty = 2500, Loss Rate = 1; and iv) Penalty = 2500, Loss Rate = 2.5. Figure 3 presents the individual recovery costs in each scenario for the proxy robust, Phase 1 and Phase 2 solutions. Comparing the results of the RRTAP and the proxy robust solution we find much more improvement variability in the individual recovery costs across the scenario set. The individual recovery costs for each
Table 5: Percentage difference between the Grönkvist solution ($x$) and the recoverable robust solution, Phase 2 ($y$) and Phase 1 ($z$), $((y - x)/x \%)$, $((z - x)/x \%)$. For conciseness the results for penalty 10 are omitted. Percentage difference greater than 5% is highlighted.

\[
\text{Scenario} \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \quad 11
\]
\[
\text{Flights Affected} \quad 2 \quad 8 \quad 8 \quad 16 \quad 5 \quad 10 \quad 8 \quad 13 \quad 4 \quad 5 \quad 1 \quad 1
\]

Table 6: Number of flight affected in the airport closure scenarios. Scenarios 0-11.
Figure 3: Individual recovery costs for all 102 scenarios - comparison between Grönkvist, Phase 1, and Phase 2 results. The order for the scenarios is presented in Table 4
While the RR-TAP solution may perform better across the weighted sum of the individual recovery costs than the proxy robust model, there are a number of scenarios where it performs worse. The largest relative improvements occur for the aircraft grounding scenarios. This demonstrates that for the larger disruption scenarios, which involve more flights and aircraft, the planned routing does not have much impact on the recovery costs. The results in Figure 3 show that the improvement in the sum of the recovery costs is attributable to an improvement in the cost for a large proportion of the individual scenarios. This is quite important since we are attempting to improve the recoverability of the tail assignment for a broad range of disruptions, so if the improvements were restricted only to a few scenarios the efficacy of this technique would be reduced.

4.3 Behaviour of solution running times

![Figure 4: Running time for the recoverable robust model. The dots represent the total running time for each parameter set.](image)

The running times for the RR-TAP is very dependant on the parameters that are used for the aircraft swap penalty and loss rate. Figure 4 illustrates the running times required to calculate the results presented in Figure 2 for the RR-TAP. In our experiments we limited the running times to 3 hours. In the cases where the running times exceeded 3 hours, the model was terminated after the first run of the Benders decomposition algorithm which completes after the 3 hour time limit. Given that the Benders subproblems’ solutions, the upper bounds, are not strictly nonincreasing, the best solution found during the run time is presented in Figure 2. In Figure 2 we see that there is little difference between the optimal solution at the completion of Phase 2 and the upper bound calculated at the completion of Phase 1. Figure 4 shows that in some cases the time spent in Phase 2 attempting to find the optimal
solution, which is not always found, can be quite significant. In order to achieve a fast solution that is close to optimal one could simply complete just Phase 1 of the two-phase algorithm.

### 4.4 Investigation of individual recovery costs

The recovery costs that were presented in Figure 2 can be broken down into their constituent costs for aircraft swaps, delays and cancellations. Figure 5 presents this breakdown, demonstrating the trade-offs that can occur when setting the model parameters. In Section 4.2 we have demonstrated that the recoverable robust solution either equals or improves on the weighted recovery cost of the proxy robust solution. Given that the model optimises the weighted recovery costs, it is possible for the proxy robust solution to outperform the recoverable robust solution for individual recovery policy costs. One such example is the aircraft swap costs with a penalty weight of $g^s = 5000$, where the number of aircraft swaps in the proxy robust solution is lower than the recoverable robust solution. However, for the same penalty weight of $g^s = 5000$, the recoverable robust solution significantly outperforms the proxy robust result for the delay costs. For other penalty weights, such as $g^s = 1000$ and $g^s = 2500$, the results are quite varied with the improvement being attributed to a decrease in aircraft swaps and delays respectively. This is attributable to the minimum length of delay for an individual flight and its associated cost. Since the flight delays are discretised to be every 30min from the original departure, the minimum cost of delay is $75 \times 30\text{min} = $2250. For the proxy robust solution, the tendency is to allow more aircraft swaps than to delay flights for penalty weights less than 2250, and for weights greater than 2250 to allow more flight delays than aircraft swaps. So in the case of the penalty weight of $g^s = 5000$ we find that the preferred recovery policy for the proxy robust solution is to delay flights, so any improvement from the RRTAP will be found through reducing the number of delayed flights.

The implication for airline disruption management is that given the non-linear nature of total recovery costs, airline operations control needs to have access to a comprehensive decision-making tool to evaluate recovery options and implementation costs. For certain recovery policies, specific recovery options may be preferred, such as more delayed flights or more aircraft swaps instead. It is noted that any recovery option to aircraft routing would eventually cause further disturbance to passenger itineraries and crewing. For instance, delaying 10 flights each by 30 mins is a recovery option that would upset about 2000 passengers (10 narrow-body aircraft), while has limited impact on aircraft crewing (possibly overtime pays and some late crew connections beyond the 10 flights). For solving the same situation, swapping three pairs of aircraft may upset minimum passengers, but would cause larger disturbance to crewing beyond the six flights, and possibly passenger itineraries as well (due to product availability from different aircraft cabin configurations). Nevertheless, at the planning stage of airline scheduling it is essential to exploit recovery scenarios so as to generate corresponding preferred schedule recovery policies for disruption management. Hence, the nature extension of our model is to integrate with crew pairing and passenger connections in future work, so we can further improve the recoverable robustness of airline schedule.
In Figure 2 it is possible to see that the greatest total improvement in costs occurs when $g^s = 2500$ and $g^s = 5000$. We have presented above the case where $g^s = 5000$, the benefit is attributable to a decrease in the total delay costs. This is combined with an increase in the total number of aircraft swaps performed in the recovery process. The largest weighted recovery cost improvement occurs when $g^s = 2500$ and we see that this benefit is largely due to a reduction in the delay costs. While the greatest improvement is in the delay costs, there is still an improvement in the aircraft swap costs. As mentioned above, we find that the improvement in both costs is due to the minimum cost of delay being close to the penalty weight of $g^s = 2500$. The result of decreased flight delays is particularly important since this directly reduces the number of delay minutes experienced by passengers, which has a real effect on the on-time performance of an airline.
5 Conclusions

In this paper we have presented a novel recoverable robustness model for the tail assignment problem. We showed that the solutions to this model guaranteed reduced recovery costs and have no increase in planning costs. Further, our model has a full set of recovery decisions, including flight cancellations and delays and aircraft rerouting. The sophisticated recovery subproblems create a complex mixed integer program, which we are able to solve in a reasonable time frame using enhancement techniques. We have formulated the RRTAP as a stochastic program drawing on solution methods such as Benders decomposition. By solving this problem as a two-stage stochastic program, it is possible to separate the planning and recovery tail assignment problems. This structure allows for the use of different recovery algorithms and a wide range of different planning algorithms and methods which can be improved upon using the RRTAP.

Through the use of Benders decomposition and column generation we have demonstrated that RRTAP is able to be solved in an efficient way. Acceleration techniques of the Magnanti-Wong method [21] have been applied to the Benders decomposition formulation, which have provided a significant improvement in the computational performance. Further, we have improved the column generation process for the Benders decomposition master problem by introducing a branching technique to eliminate symmetry, branching on a aircraft pair and starting flight.

We have compared the results of the RRTAP against a proxy robust solution developed using a connection cost function presented in Grönkvist [16]. We found this connection cost function provided a simple way to introduce robustness into the tail assignment problem. Through the use of our recovery algorithm, we have evaluated this connection cost function to demonstrate the recoverability of the optimal planning tail assignment solution. Further, using this connection cost function in the planning stage of our recoverable robustness algorithm we have been able to develop an even lower cost recoverable planned solution compared to the Grönkvist solution.

With the use of the two-phase algorithm for solving the Bender decomposition problem we have found that the integral master problem solution using the cuts added by the completion of Phase 1 provides a good upper bound on the optimal solution. A large proportion of the running times are spent in Phase 2 adding cuts to improve the lower bound towards the Phase 1 upper bound. Terminating the algorithm at the completion of Phase 1 results in near optimal solutions with very fast computational times.

Our results have been presented with a range of values for airline specific parameters, the aircraft swap penalties and lost revenue rates. The method used to assign costs to aircraft swaps, delays and cancellations can have varied effects on the cost benefits from the RRTAP. We demonstrated that due to the value of the minimum delay cost and the penalty weight that there can be a tendency towards greater aircraft swaps or delays in the final solution. By presenting a range of values we have demonstrated that a trade-off between different recovery costs can be achieved. This allows each airline to value the delays, cancellations and aircraft swaps differently depending on their individual situations.
In future work, we will be looking to extend the recoverable robustness technique for airline problems. With our current formulation it is possible to integrate one or more airline planning stages with the tail assignment problem. Given the high cost of crew to the airline, it appears that this would be an important aspect to add to this problem. Also, in a recovery situation it is important to monitor passenger flow in the network. Explicit modelling of passengers in the recovery models could provide a set of more robust results.

References


REFERENCES


