

# An Optimization Approach to Airline Integrated Recovery

Jon D. Petersen\*, Gustaf Sölveling, Ellis L. Johnson, John-Paul Clarke, Sergey Shebalov

May 31, 2010

## Abstract

While the airline industry has benefited from advancements made in advanced analytical OR methods, most products used in operation stem from the frictionless environment of the planning stage. With 22% of all flights being delayed and 3% being canceled in the U.S. since 2001, schedule perturbations are inevitable. The complexity of the operational environment is exacerbated by the need for obtaining a solution in as close to real-time as possible. Given some time horizon, the recovery process seeks to repair the flight schedule, aircraft rotations, crew schedule, and passenger itineraries in a tractable manner. Each component individually can be difficult to solve, so early research on irregular operations has studied these problems in isolation leading to a sequential process by which the recovery process is conducted. Recent work has integrated a subset of these four components, usually abstracting from crew recovery. We present an optimization-based approach to solve the airline integrated recovery problem. After our solution methodology is presented, it is tested using data from an actual U.S. carrier with a dense flight network. It is shown that in several instances an integrated solution is delivered in a reasonable runtime. Moreover we present results showing the integrated approach can substantially improve the solution quality over the incumbent sequential approach. To the best of our knowledge, are the first to present computational results on the fully integrated airline recovery problem.

## 1 Introduction

The airline industry has been one of the biggest beneficiaries of the advancements made in the application of advanced optimization methodologies. Fleet assignment, aircraft scheduling, crew scheduling, dynamic pricing and revenue management, and other paradigms have received considerable attention in both industry and academia throughout the past few decades. Such decisions are made well in advance of the day of operations under the assumption that the actual schedule is flown during the day of operation. However in practice operations are rife with frictions caused by disturbances such as weather or mechanical failure. In spite of all the advances made at the planning level, there has been relatively little work done at the operational level.

Even though problems at the operational phase are similar to that of the planning phase, the former's problems are exacerbated by two things. First is additional operational complexities that arise. For example, suppose an aircraft is approaching its destination but is unable to land due to convective weather. The aircraft may be placed into a holding pattern requiring additional flying time for the cockpit crew. By the time the aircraft does land, the crew may not be legal to fly their subsequent scheduled leg due to the exceeding their allowed flying time within a 24-hour period. Thus, the second leg may be disrupted if no other crew are readily available. The second problem is that of timing. Most airlines utilize an operations control center (OCC) in which provide a centralized decision making environment. OCC coordinators naturally wish to make decisions in as close to a real-time environment as possible, unlike the planning phase which are sometimes made over one year in advance of operations. Because decisions involving repairing the schedule, aircraft, crew, and passengers are combinatorial in nature, using an optimization-based approach may not be plausible due to the computing time necessary to solve any one of these operational problems. As a result

---

\*corresponding author: H. Milton Stewart School of Industrial and System Engineering, Georgia Institute of Technology, [petersen@gatech.edu](mailto:petersen@gatech.edu)

unlike the planning environment, airlines do not generally rely on the use of mathematical programming in the presence of a disruption to their operations.

Given a disruption to the existing schedule, the airline is said to be in a *recovery operation*. Developing an optimization model is naturally of interest to the O.R. practitioner given the challenges posed. The immense nominal costs also make it of interest to an airline. While estimates vary, these are generally considered to be tens of billions of dollars annually in the U.S. alone - see Bonnefoy [11]. Airline passengers also have an obvious interest in the problem as passenger delays have become more problematic as the growth in air transportation has outpaced that of capacity at major airports. In some rare instances passenger delays have drawn global attention as passengers have been subjected to exceedingly long tarmac delays. These occurrences have prompted U.S. Congress to consider drafting a passengers' bill of rights. Effective April 2010 the U.S. Department of Transportation implemented a fine of up to \$27,500 per passenger who exceed a three-hour delay on the tarmac. While there have been some advancements made in applying mathematical programming to the operational phase of airline scheduling, little advancements have been made in practice. One possible explanation is that the literature has considered only a proper subset of decisions required during a recovery period in order to deliver a solution in a computationally tractable manner. Such a solution scheme may not be implementable by an OCC depending on the airline's objective.

The principle goal of this paper is to define, formulate, and solve a fully integrated recovery problem that is amenable to the constraints imposed by an OCC coordinator. We formally discuss the various components of the integrated recovery problem, discuss the method by which we are reducing the scope of the problem so as to make the problem solvable within a suitable time frame, and using data from a real U.S.-based airline validate our method by providing computational results. To the best of our knowledge, we are the first to provide such computational results to the fully-integrated problem. In the context of solving this problem we also introduce some results that can extend to other related problems within the industry.

The remainder of the paper is organized as follows. Section 2 provides a review of relevant work done within irregular airline operations. The problem and model are formally defined in Section 3. Section 4 discusses how the scope of the recovery operation is limited to make the problem solvable. Our decomposition scheme is outlined in Section 5. Computational results are shown in Section 6 that validates our approach. Here we observe the improvement the integrated approach yields relative to several key performance metrics. Section 7 summarizes our work and suggests related future work.

## 2 Literature Review

While there has been relatively little work previously done for studying and solving the airline integrated recovery problem, various components within the problem have been studied. We review some of the seminar earlier work done. This is by no means a complete survey of irregular operations. Filar et al. [21] provides an exceptional survey of previous work. Clausen et al. [18] give a recent state-of-the-art overview of disruption management of schedule, aircraft, crew, passenger, and integrated recovery.

### 2.1 Schedule Recovery

Teodorovic and Guberinic [45] consider the problem of reassigning aircraft rotation when one or more aircraft are taken out of operation that minimizes total passenger delay. A flight network is formed and the schedule is repaired with the reduced set of aircraft. The solution is obtained by the branch-and-bound method for which an efficient two-step branching rule is implemented.

Using a lexicographic dynamic programming heuristic, Teodorovic and Stojkovic [46] introduce a model that seeks to minimize total flight cancelations while minimizing passenger delay. This is the first model that considers restoring the schedule and aircraft rotations in tandem.

The first work to integrate crew rotations with aircraft rotations was studied in Teodorovic and Stojkovic [47]. A heuristic model is introduced in which both aircraft and crew rotations are repaired through a first-in, first-out (FIFO) rule and a dynamic programming algorithm that incorporates re-timing decisions.

Jarrah et al. [25] introduce two network models that form the basis for irregular operations control at United Airlines. They allow the possibility of equipment swapping and allow the use of spare aircraft. The first model seeks to output a flight delay plan until the shortage of aircraft is resolved by minimizing total disutility. The second model achieves the same objective but considers flight cancellations instead of delays. Computational results are presented for each model showing considerable improvement relative to an unoptimized schedule.

Yan and Yang [53] provide the first study that allows for delays and cancellations simultaneously. A network flow model with side constraints was introduced that are solved by Lagrangian relaxation with the subgradient method. By obtaining efficient bounds on the optimal objective, computations were tractable and their model was readily seen to deliver efficient solutions.

Yan and Tu [52] consider schedule re-optimization in the presence of multiple fleets. A multicommodity network flow model is introduced that is efficiently solved by a modified Lagrangian relaxation scheme using the subgradient method. A case study is presented in which their framework improved profits in each scenario. See Yan and Lin [51] for a similar study.

Clarke [16] introduces the Airline Schedule Recovery Problem (ASRP) that is strongly related to our model below. The comprehensive framework that is proposed considers flight delays and cancellations in tandem, as well the management of air traffic control (ATC). He also imposes constraints on crew availability so as to make the schedule compatible with respect to the initial positions of each crew. Two greedy heuristic procedures and an optimization-based solution procedure are considered and the results are evaluated under different scenarios.

Argüello et al. [7] use metaheuristic approach by presenting a greedy randomized adaptive search procedure (GRASP) to restore aircraft routings in the presence of a ground delay program. Their algorithm is polynomial in the number of flights and aircraft that has found near-optimal solutions to minimize delay and cancellation costs under a wide range of scenarios. Over 90% of the GRASP solutions were within 10% of optimality that were generally obtained within a few seconds.

A binary quadratic programming approach is introduced by Cao and Kanafani [14] and [15] that integrates delays and cancellations. Their model maximizes profit while penalizing undesirable outcomes.

An overview of the decision-making environment at OCCs is given in Clarke [17]. This paper discusses the primary causes of irregularities, reviews the information systems and decision-support systems utilized, and proposes a new decision framework. A more recent, but similar exposition is given by Kohl et al. [27].

Three multicommodity network flow models are presented in Thengvall et al. [48] for schedule recovery that follows a hub closure. Each model considers flight cancellations, delays, ferrying, and swaps. The first two models - a pure network with side constraints and a generalized network - seek to maximize profit that attempts to keep as much of the original schedule preserved as possible. The third model, which is a pure network with side constraints with a discretized time horizon, seeks to minimize the cost incurred from flight cancellations and delays. Their results show that swapping opportunities have a substantial impact in the solution quality.

Stojković et al. [42] proposed a model that allows for not only the delaying of flights, but altering the duration of service as well to preserve maintenance schedules, ground service, crew connections, and passen-

ger connections. The dual to their proposed model is a network model which allows for computation in near real-time.

Rosenberger et al. [33] develop a set packing model that seeks to assign routes to aircraft by minimizing an objective that is comprised of both the assignment cost as well as cancelation cost. Maintenance feasibility is preserved by enumerating all routings involving a maintenance activity a priori. Their model is considered in the presence of both aircraft disruptions as well as station disruptions in a ground delay program. They present an efficient heuristic that is used to identify the subset of aircraft that are to be rerouted, and their model is validated by simulation. They also extend their model to consider crew and passenger connections.

Eggenberg et al. [20] repair the schedule through an efficient column generation scheme in which new columns are quickly generated through solving a resource constrained shortest path problem.

## 2.2 Crew Recovery

To our knowledge, the first to study crew recovery were Wei et al. [49]. The authors propose a comprehensive multicommodity network flow network. A heuristic-based search algorithm is used within the context of a depth-first search branch-and-bound algorithm that seeks to repair the original crew pairings. Song et al [41] consider a similar structure.

Stojković et al. [44] propose a model that, given a fixed flight schedule, seeks to output a set of modified crew pairings at minimum cost through a set partitioning problem that uses column generation throughout the branch-and-bound tree in a suitable runtime between a few seconds and about 20 minutes.

Our work is strongly related to Lettovsky et al. [29]. Given the set of canceled flights they also assign crew to modified pairings at minimum cost. They allow crews to deadhead either within the modified pairing or back to their crew base. They present efficient preprocessing techniques to identify the subset of the schedule to be disruptable. The model is solved by the primal-dual method on the LP relaxation of the model. Three branching techniques are studied, and they show that branching on follow-ons (where consecutive flight legs either are or are not present in a pairing) tends to be an efficient procedure for obtaining integer solutions.

Stojković and Soumis [43] consider a one day crew recovery model that allows for scheduling changes that keep aircraft routings fixed. Their problem is formulated as an integer nonlinear multicommodity network flow problem that is solved by Dantzig-Wolfe decomposition with branch-and-bound. Three problem instances are run showing that even in the largest instance, quality solutions were obtained in under 15 seconds.

Nissan and Haase [31] present a new methodology that is particularly appropriate to European carriers as their model assumes a fixed-cost structure of crew as oppose to pay-and-credit that is prevalent among North American carriers. Their objective is therefore to adhere as close as possible to the old schedule. By not explicitly repairing broken crew pairings, the problem size is diminished considerably in that they solve a disruption for every duty period. A set-covering model is solved using branch-and-price with new columns being added from a residual network by solving a shortest path problem. Their approach is shown to solve in a runtime that is acceptable in operations.

## 2.3 Passenger Recovery

For the most part, airlines abstract passenger disruption within the context of their decision-making process. Finding an optimal tradeoff in the disruption of the schedule and its passengers, Bratu and Barnhart [12] suggest a framework that can reduce passenger disruptions while holding down other scheduling costs in irregular operations. Their model allows flight delays and cancelations that assigns reserve crew and spare

aircraft to accommodate the new schedule. Two models are presented: the disrupted passenger metric (DPM) model and the passenger delay metric (PDM) model. The former model assigns only disrupted passengers and is only a proxy of actual delay costs, whereas the latter model assigns all passengers and provides a more accurate description of the true costs of delay. Their model is validated by a simulated OCC. While the DPM model is shown to not solve in sufficient time so as to implement in an actual OCC, the PDM model suggests that it might be amenable to a real-time decision making environment.

Zhang and Hansen [55] propose integrating other means of transportation to accommodate disrupted passengers. Such intermodal connections are often preferred particularly when the destination is relatively nearby the disrupted station within a hub-and-spoke network. By incorporating ground transportation into passenger recovery, they propose a mixed integer nonlinear programming model that is solved heuristically by first relaxing integrality and then fixing variables. Runtimes were shown to be under 20 minutes. Moreover their experiments show that the number of disrupted passengers may be greatly reduced by allowing intermodal substitution; one experiment showed this number was reduced by more than 84%.

## 2.4 Robust Scheduling

An area closely related to recovery is schedule robustness. The central idea is to design a schedule that is able to be recovered from more efficiently in the presence of irregularity. Robust scheduling was studied extensively in Ageeva [3], Smith [39], Rosenberger et al. [34], Smith and Johnson [40], AhmadBeygi et al. [4], and Burke et al. [13]. Crew robustness was studied in Klabjan et al. [26], Yen and Birge [54], Shebalov and Klabjan [38], Ball et al. [8], Gao et al. [24], Weide et al. [50], and Dunbar et al. [19]. The impact of schedule robustness to passenger recovery can be seen in Lan et al. [36].

## 2.5 Partially and Fully Integrated Recovery

There have been a number of studies whose aim is to partially integrate operations under irregularity. Abdelghany et al. [1] presented a decision support tool in which combines a schedule simulation with a resource optimization model that minimizes cancellations and disruptions while incorporating important crew considerations of both pilots and flight attendants. Given the anticipated severity of disruption the flight simulation model predicts a list of disrupted flights. Given this disruption the resource assignment optimization model assigns an efficient plan that is to delay and cancel flights that consider crew and aircraft swaps and utilization of reserve resources. A drawback of their approach is they do not allow crews to deadhead. After 177 potential flight disruptions are simulated, their iterative process saves 661 minutes of delay - 8.7% of the observed delay in the actual scenario which is found in just over 3 seconds.

The 2009 ROADEF challenge [32] introduced a competition that sought to deliver a recovery solution that was to integrate the schedule, aircraft, and passengers. Gabteni [22] presents an overview of the different proposed methodologies. The winning team, seen in Bisailon et al. [10] employ a large-scale neighborhood search heuristic that iteratively constructs, repairs, and improves solutions that incorporates randomness to diversify the search procedure. Feasibility was quickly achieved in the first phase, while the third phase was shown to be significant as cost reductions we shown to be apparent in several instances.

The third place entry is shown in Acuna-Agost et al. [2]. They define a MIP model to achieve their objective by first solving the problem on a very limited set with many variables fixed a priori. The novel feature of their framework is the introduction of a Statistical Analysis of Propagation of Incidents (SAPI). Using a logistic regression, the probability of each flight being disrupted are estimated. If these probabilities exceed a certain threshold flight cancellation variables are fixed, and if the probabilities are sufficiently low, the previous MIP solution is fixed. Neighboring solutions are then explored by local branching and fed back into the MIP. Because the search space is limited, the MIP computation is tangible.

## 2.6 Fully-Integrated Recovery

Handling aircraft and crew in concert is an arduous task which explains why previous computational studies have ignored crew considerations. There have been some studies that include a fully integrated airline recovery framework, although these tend to be only formulations.

Two such proposals for integrated recovery are seen in Ph.D. dissertations by Lettovsky [28] and Gao [23]. The formulation given by the former is closely related to our work. He presented a fully integrated model that decomposes into a structure suitable for Benders decomposition. The linking variables are fleeting decisions to flight legs in which are passed to subproblems represented by repairing aircraft rotations, crew pairings, and passenger itineraries. While a formulation was provided, no computations were performed.

## 3 The Airline Integrated Recovery Problem

We formally define the *airline recovery problem* to be comprised of the following four problems:

- The *schedule recovery problem* seeks to fly, delay, cancel, or divert flights from their original schedule. We call the solution to this problem the *repaired schedule*.
- The *aircraft recovery problem* assigns individual aircraft to accommodate the repaired schedule that are feasible for the constraints imposed by maintenance requirements.
- The *crew recovery problem* assigns individual (cockpit) crew members to flights according to the repaired schedule that satisfy the complex legality requirements. We ignore assigning cabin crew members to new schedules.
- The *passenger recovery problem* re-assigns disrupted passengers to a new itinerary.

Given a disruption, we define the *time window* to be an exogenous interval  $\mathcal{T} := [\underline{t}, \bar{t}]$  in which flights, aircraft rotations, crew schedules, and passenger itineraries are allowed to be disrupted. The requirement is for all the components to be back on schedule by the end of the time window  $\bar{t}$ . While one could define different time windows for each component, our analysis includes the same time window for all problems.

There is an inherent tradeoff between solution quality and runtime. A possible method might be to develop a recovery scheme in a two-phased approach that first seeks to recover the schedule, then to recover the other three components taking the repaired schedule as given. However tractable as this method seems, there are a number of problems associated with this scheme. Conflicting objectives almost certainly exist between the schedule, crew costs, and passenger delays. Passing a single feasible schedule is too restrictive with respect to each of the second-stage problems. We argue that if this were a plausible recovery method in practice, virtually every airline OCC would have already implemented a variation of such a solution strategy. Instead, airlines often try to find a single feasible schedule manually. The other extreme would be to deliver a fully integrated solution that is globally optimal with respect to each of the four components. And while an integrated recovery framework is naturally desirable, the size and complexity may preclude such a mechanism to be implemented in practice. Therefore a balance between these two extremes must be reached with the goal of delivering an integrated solution.

Formally the airline integrated recovery (AIR) problem is defined to be the union of the following four problems.

### 3.1 Schedule Recovery

The Schedule Recovery Model (SRM) returns re-timing and flight cancellation decisions. Our model is closely related to Clarke [16] in that we consider additional constraints imposed by air traffic control management systems.

Instead of a leg-based model, we utilize flight strings which was introduced by Barnhart et al. [9]. A *flight string* (which, from now, we refer to as *string*) is a sequence of flights, with timing decision, to be operated by the same aircraft. There could be several copies made of the same sequence of flight with different re-timing decisions. A string-based model has a number of advantages. While the number of strings naturally grows significantly with respect to the number of flights, efficient column generations techniques can be employed. Strings are also able to capture network effects that single flight decisions that are not possible with individual flight assignments. Also, ground arcs need not formally be defined in the underlying time-space network. The biggest advantage is in the ease of obtaining integer solutions to the routing problem (which is discussed formally below).

### 3.1.1 Sets

$F$ : set of all flight legs

$E$ : set of equipment types (fleets)

$S$ : set of flight strings

$A$ : set of all airports

$A^{\text{arr}}(a, \underline{t}^a, \bar{t}^a)$ : arrival capacity of station  $a$  between  $\underline{t}^a$  and  $\bar{t}^a$

$A^{\text{dep}}(a, \underline{t}^a, \bar{t}^a)$ : departure capacity of station  $a$  between  $\underline{t}^a$  and  $\bar{t}^a$

$G(a, \underline{t}^a, \bar{t}^a)$ : number of gates available at station  $a$  between  $\underline{t}^a$  and  $\bar{t}^a$

$I(a, \underline{t}^a, \bar{t}^a)$ : set of strings that are inbound to station  $a$  between  $\underline{t}^a$  and  $\bar{t}^a$

$O(a, \underline{t}^a, \bar{t}^a)$ : set of strings that are outbound from station  $a$  between  $\underline{t}^a$  and  $\bar{t}^a$

$F^{\text{strategic}}$ : set of strategic flights that are prohibited from being canceled

$F^{\text{market}}$ : set of flights requiring a minimum number of seats

### 3.1.2 Data

$c_{e,s}^{\text{assign}}$ : cost of assigning equipment type  $e \in E$  to string  $s \in S$

$c_f^{\text{cancel}}$ : cost of canceling flight  $f \in F$

$\beta_e$ : number of seats available on equipment type  $e \in E$

$n_{seats}(f)$ : minimum number of seats required by flight  $f \in F^{\text{market}}$

$\mathbf{1}_{f,s}$ : indicator variable; 1 if flight  $f \in F$  is in string  $s \in S$

### 3.1.3 Decision Variables

$$x_{e,s} = \begin{cases} 1 & \text{if equipment type } e \in E \text{ is assigned to string } s \in S \\ 0 & \text{otherwise} \end{cases}$$

$$\kappa_f = \begin{cases} 1 & \text{if flight } f \in F \text{ is canceled} \\ 0 & \text{otherwise} \end{cases}$$

### 3.1.4 SRM Formulation

The SRM seeks to minimize the total costs incurred by assigning strings and canceling flights. The SRM formulation is given as follows:

$$\min \sum_{e \in E} \sum_{s \in S} c_{e,s}^{\text{assign}} x_{e,s} + \sum_{f \in F} c_f^{\text{cancel}} \kappa_f \quad (3.1)$$

$$\text{s.t.} \quad \sum_{e \in E} \sum_{s \in S} \mathbf{1}_{f,s} x_{e,s} + \kappa_f = 1 \quad \forall f \in F \quad (3.2)$$

$$\sum_{e \in E} \sum_{s \in S} \mathbf{1}_{f,s} x_{e,s} = 1 \quad \forall f \in F^{\text{strategic}} \quad (3.3)$$

$$\sum_{e \in E} \sum_{s \in I(a, \underline{t}^a, \bar{t}^a)} x_{e,s} \leq a^{\text{arr}} \quad \forall a^{\text{arr}} \in A^{\text{arr}}(a, \underline{t}^a, \bar{t}^a) \quad (3.4)$$

$$\sum_{e \in E} \sum_{s \in O(a, \underline{t}^a, \bar{t}^a)} x_{e,s} \leq a^{\text{dep}} \quad \forall a^{\text{dep}} \in A^{\text{dep}}(a, \underline{t}^a, \bar{t}^a) \quad (3.5)$$

$$\sum_{e \in E} \sum_{s \in O(a, \underline{t}^a, \bar{t}^a)} x_{e,s} - \sum_{e \in E} \sum_{s \in I(a, \underline{t}^a, \bar{t}^a)} x_{e,s} \leq g \quad \forall g \in G(a, \underline{t}^a, \bar{t}^a) \quad (3.6)$$

$$\sum_{e \in E} \sum_{s \in S} \mathbf{1}_{f,s} x_{e,s} \beta_e \geq n_{\text{seats}}(f) \quad \forall f \in F^{\text{market}} \quad (3.7)$$

$$x_{e,s} \in \{0, 1\} \quad \forall e \in E, \forall s \in S$$

$$\kappa_f \in \{0, 1\} \quad \forall f \in F$$

The objective (3.1) is to minimize the aggregate cost comprised of assigning strings (including re-timing decisions) and flight cancelations. Either a flight must be contained in exactly one string or canceled, as seen in (3.2). To prohibit strategic flights from being canceled constraints of the form (3.3) are added. Arrival and departure capacities at certain airports at given time intervals are not to be exceeded as captured in (3.4) and (3.5), respectively. (3.6) ensures the number of aircraft on the ground does not exceed the number of gates available at certain station and times. Market requirements are captured in (3.7); they ensure that a minimal number of seats are operated on certain flights. There are also other constraints that prohibit certain resources from being assigned to certain flights. For instance, a curfew constraint ensures no flight arrives or departs within a curfew period. Other such constraints include weather restrictions, and constraints prohibiting certain fleet types from operating at specific stations that cannot accommodate that type of aircraft.

Recall that re-timing decisions are incorporated in the flight strings. That is, multiple strings may have the same sequence of flights, although the strings differ by the departure times of flights within each string.

## 3.2 Aircraft Recovery

The Aircraft Recovery Model seeks to assign specific tail numbers to strings while meeting maintenance and other aircraft requirements. The ARM is solved for each equipment type  $e \in E$ .

### 3.2.1 Sets

$AC(e)$ : set of aircraft of equipment type  $e \in E$

$H(e)$ : set of aircraft of equipment type  $e \in E$  requiring maintenance within  $\mathcal{T}$

$A^{\text{maint}}(e)$ : set of stations that are capable of performing schedule maintenance of equipment type  $e \in E$

### 3.2.2 Data

$c_{e,s}^n$ : cost of assigning tail  $n \in AC(e)$  to string  $s \in S$

$\mathbf{1}_{m,s}$ : indicator variable; 1 if an eligible maintenance station  $m \in A^{\text{maint}}(e)$  is visited by string  $s \in S$

### 3.2.3 Decision Variables

$$x_{e,s}^n = \begin{cases} 1 & \text{if aircraft } n \in AC(e) \text{ is assigned to string } s \\ 0 & \text{otherwise} \end{cases}$$

### 3.2.4 ARM Formulation

The costs structure of the ARM generally will be to award bonuses for preserving the same tail assignments to flight numbers from the original schedule, or penalizing undesirable aircraft rotations. Unlike its leg-based counterpart, a complex routing problem need not be solved for with flight strings. Rather, the problem amounts to a maximum-cardinality bipartite matching problem for which an integral solution is delivered by solving the LP-Relaxation.

Given equipment type  $e \in E$ , the Aircraft Recovery Model, or ARM( $e$ ) is

$$\min \sum_{n \in AC(e)} \sum_{s \in S} c_{e,s}^n x_{e,s}^n \quad (3.8)$$

$$\text{s.t.} \quad \sum_{n \in AC(e)} x_{e,s}^n = x_{e,s} \quad \forall s \in S \quad (3.9)$$

$$\sum_{s \in S} x_{e,s}^n = 1 \quad \forall n \in AC(e) \quad (3.10)$$

$$\sum_{s \in S} \sum_{m \in A^{\text{maint}}(e)} \mathbf{1}_{m,s} x_{e,s}^n = 1 \quad \forall n \in H(e) \quad (3.11)$$

$$x_{e,s}^n \in \{0, 1\} \quad \forall s \in S, \forall n \in AC(e) \quad (3.12)$$

The objective (3.8) seeks to minimize the cost associated with aircraft assignment. The cost can be thought of penalties/bonuses. For instance, a penalty may be imposed for any deviation from the original routing. The string cover constraints (3.9) assure each string that is chosen from the SRM is assigned to some eligible aircraft. (3.10) ensure each aircraft is assigned to no more than one string. In the event that the initial and end stations coincide for a particular aircraft, we define a *null* string to be one with no flights so the aircraft stays on the ground. Thus for any aircraft whose initial station and end station differ must be assigned to some string. Maintenance cover constraints are seen in (3.11). This simply ensures a maintenance opportunity is built in, and the specific maintenance planning can be done post-optimization. Other constraints we include but do not explicitly formulate are user-dependant constraints prohibiting certain aircraft from operating at some airports, and similar operational restrictions.

## 3.3 Crew Recovery

Crew members are assigned to *pairings* which are flight assignments over the course of a number of *duties* that contain specific flight assignments over a period of time that allow for sufficient rest. A duty is usually a day's work of flying, and the pairing usually spans between 2 and 4 duties. A *roster* period is comprised of a number of pairings for a period of time, usually about one month. If a specific crew has a pairing that becomes disrupted, the pairing is said to be *broken*. A broken pairing may be augmented during the period overlapping with the time window  $\mathcal{T}$  so as to deliver the crew to the station they are required to be at immediately outside of  $\mathcal{T}$ . All other components within the crew schedule outside of  $\mathcal{T}$  are to be preserved.

We ensure the repaired pairing is legal for the entire duration of the crew's pairing, although it may be not be the case for the roster period in which case this would have to be fixed between the end of the pairing and end of the roster.

The Crew Recovery Model (CRM) seeks to repair disruptable crew pairings at minimal cost. Like the ARM, the CRM is solved for each equipment type corresponding to crew rating. For brevity within the context of CRM, a *pairing* is really meant by ‘the broken part of the original crew pairing’.

### 3.3.1 Sets

$K$ : set of all crew members not on reserve

$P_k$ : set of eligible pairings for crew  $k \in K$

$P$ : set of all pairings, i.e.  $P = \bigcup_{k \in K} P_k$

A pairing  $p \in P_k$  is *eligible for crew  $k$*  if:

- $p$  begins at the station where crew  $k$  is at the beginning of the time window  $\underline{t}$
- $p$  ends at the station where crew  $k$  are required to be at the end of the time window  $\bar{t}$
- all flight, duty, and pairing legality requirements are satisfied

### 3.3.2 Data

$c_{k,p}^{\text{assign}}$ : cost of assigning crew  $k \in K$  to pairing  $p \in P_k$

$d_f^{\text{pairing}}$ : cost of deadheading a crew on flight  $f \in F$

$d_k^{\text{base}}$ : cost of deadheading crew  $k$  back to base

$\mathbf{1}_{f,p}$ : indicator variable; 1 if flight  $f \in F$  is found in pairing  $p \in P$

### 3.3.3 Decision Variables

$$y_{k,p} = \begin{cases} 1 & \text{if crew } k \in K \text{ is assigned to pairing } p \in P_k \\ 0 & \text{otherwise} \end{cases}$$

$$\nu_k = \begin{cases} 1 & \text{if crew } k \in K \text{ is to deadhead back to base} \\ 0 & \text{otherwise} \end{cases}$$

$s_f$  = the number of surplus crew (deadheads while on pairing) on flight  $f \in F$

### 3.3.4 CRM Formulation

Let  $c_f^{\text{pairing}}$  denote the penalty associated with having a crew deadhead on flight  $f \in F$  and  $c_k^{\text{base}}$  denote the penalty associated with deadheading a crew back to base. The CRM model we consider is

$$\min \sum_{k \in K} \sum_{p \in P_k} c_{k,p}^{\text{assign}} y_{k,p} + \sum_{f \in F} d_f^{\text{pairing}} s_f + \sum_{k \in K} d_k^{\text{base}} \nu_k \quad (3.13)$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{p \in P_k} \mathbf{1}_{f,p} y_{k,p} - s_f = 1 - \kappa_f \quad \forall f \in F \quad (3.14)$$

$$\sum_{p \in P_k} y_{k,p} + \nu_k = 1 \quad \forall k \in K \quad (3.15)$$

$$y_{k,p} \in \{0, 1\} \quad \forall k \in K, \forall p \in P_k$$

$$\nu_k \in \{0, 1\} \quad \forall k \in K$$

$$s_f \in \mathbb{Z}_+ \quad \forall f \in F$$

The objective (3.13) seeks to minimize total crew cost. (3.14) ensures that some crew operates each flight that is not canceled. (3.15) assigns each crew to either some eligible pairing or deadheads home.

## 3.4 Passenger Recovery

The passenger recovery problem is a multicommodity network flow problem whose arcs correspond to flights, whose capacities correspond to the number of available seats on that flight leg, and whose commodities are passengers with a defined 4-tuple consisting of origin, departure time at origin, destination, and scheduled arrival at destination.

### 3.4.1 Sets

$PAX$ : set of O-D passengers with a unique itinerary

$\Gamma$ : set of all itineraries given the repaired schedule

### 3.4.2 Decision Variables

$z_{i,\gamma}$  = number of passengers  $i \in PAX$  who are assigned to itinerary  $\gamma \in \Gamma$

$s_i$  = number of passengers  $i \in PAX$  who cannot be assigned to an itinerary

### 3.4.3 Data

$\delta_{i,\gamma}$ : delay (in minutes) associated with assigning passenger  $i \in PAX$  to itinerary  $\gamma \in \Gamma$

$c^{\text{delay}}$ : cost per minute of passenger delay

$c_i^{\text{unassign}}$ : cost of being unable to assign passenger  $i$  to an itinerary

$\xi_f^{\text{planned}}$ : number of nondisrupted passengers on flight  $f \in F$

$\eta_e$ : capacity of equipment type  $e \in E$

### 3.4.4 PRM Formulation

$$\min \sum_{i \in PAX} \sum_{\gamma \in \Gamma} c^{\text{delay}} \delta_{i,\gamma} z_{i,\gamma} + \sum_{i \in PAX} c_i^{\text{unassign}} s_i \quad (3.16)$$

$$\text{s.t.} \quad \sum_{i \in PAX} \sum_{\gamma \in \Gamma} \mathbf{1}_{f,\gamma} z_{i,\gamma} \leq \sum_{e \in E} \sum_{s \in S} \mathbf{1}_{f,s} x_{e,s} \eta_e - \xi_f^{\text{planned}} \quad \forall f \in F \quad (3.17)$$

$$\sum_{\gamma \in \Gamma} z_{i,\gamma} + s_i = 1 \quad \forall i \in PAX \quad (3.18)$$

$$z_{i,\gamma} \in \mathbb{Z}_+ \quad \forall i \in PAX, \forall \gamma \in \Gamma$$

The objective (3.16) seeks to minimize the sum of aggregate delay and cost associated with being unable to assign a passenger to an itinerary within  $\mathcal{T}$ . (3.17) prohibits spilling any passenger and (3.18) assigns a passenger to precisely one itinerary. If a passenger cannot be assigned to an itinerary, they may overnight at a connection point, be placed on another airline, or have their itinerary delayed outside of  $\mathcal{T}$ .

## 4 Limiting the Scope of Recovery Operation

The size and complexity of the integrated recovery problem outlined above most likely precludes a global optimum from being delivered. Therefore any practical method that is to solve the problem must carefully consider how to limit the size of the problem and how to construct a computationally tractable solution method.

For most realistic disruption scenarios not all resources are to be affected throughout a given network. Filtering out resources that are unaffected is said to be *limiting the scope* of a recovery operation. The inherent tradeoff in limiting the scope is between reducing the problem size versus solution quality. We now discuss the procedure by which flights, aircraft, and crews are limited in context of the integrated recovery module.

### 4.1 Limiting Flights

For a given disruption, a subset of flights that are unaffected by their resources are not considered in the model and are to operate as planned. All other flights are said to be *disruptable* as they can be delayed, canceled, or possibly flown as planned. Disruptable flights are included through disruptions to aircraft, crew, and passengers.

**Flights from disrupted routings** A flight is said to be *disrupted* if a disruption occurs to a resource used by that flight. A routing is disrupted if it contains a disrupted flight within its scheduled routing. Because of propagation effects of a disruption, all subsequent flight legs within that routing are candidates to be disrupted. Figure 1 illustrates an example of this concept with a disruption at ATL in a given time horizon.

**Flights from disrupted crew** We denote another set of disrupted flights that are generated through crew schedules. For each crew that has a broken duty, new flights may be added outside the previous disruptable set of flights. Like the case with routings, each subsequent flight within that duty may be perturbed from delay propagation and all subsequent flights within the duty are then added. Figure 2 illustrates how disruptable flights from crew schedules are added using the disrupted routings shown in Figure 1.

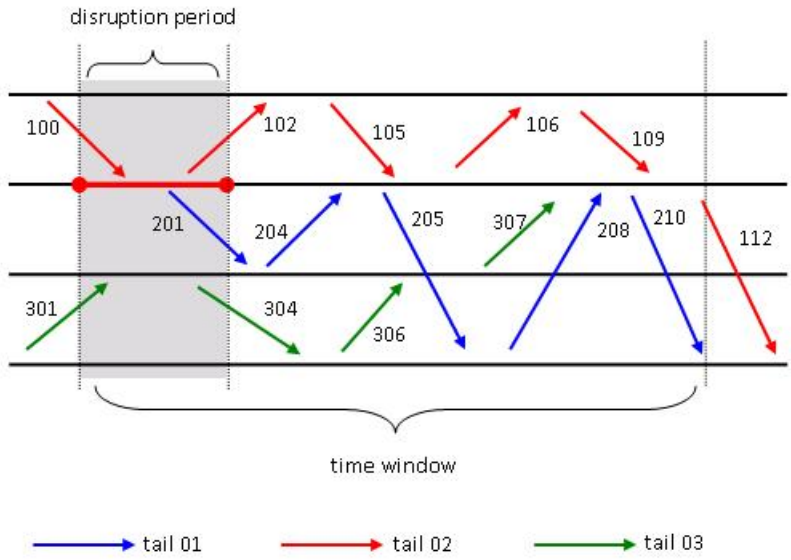


Figure 1: Disruptable flights: 102, 105, 106, 109, 201, 204, 205, 208, 210

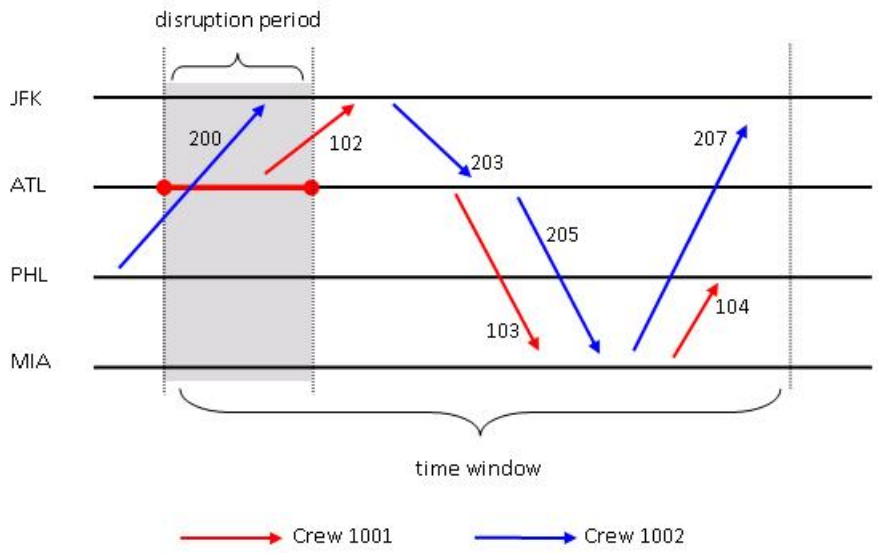


Figure 2: New disruptable flights: 103, 104, 207

**Flights from tight passengers connections** Because adding all scheduled flights are not possible, some potential swapping opportunities are missed. Because we take a passenger-centric approach to the recovery problem, we consider some additional candidate flights to assign passengers through preprocessing. A *disruptable passenger* is a passenger whose itinerary contains a disruptable flight. Through preprocessing we identify a new set of flights that are added if the absolute value of the connection time is below a certain threshold. Figure 3 shows this concept for a passenger originating in New York (JFK) whose destination is Miami (MIA) connecting through Atlanta (ATL). The bold arcs indicate the original itinerary for which it is assumed both flights are disruptable. There are four outbound flights from ATL to MIA. Because flights 102 and 210 are both disruptable, there is a chance of a misconnection in ATL. Flights 300, 302, and 107 are candidates to include in the disruptable flight set. For a given tolerance parameter of 30 minutes, we would include all (initially) nondisruptable flight legs and include if the absolute value of the connection time were within this bound. In this example, the departure of flight 302 is sufficiently close to the arrival of flight 102 that makes 302 disruptable.

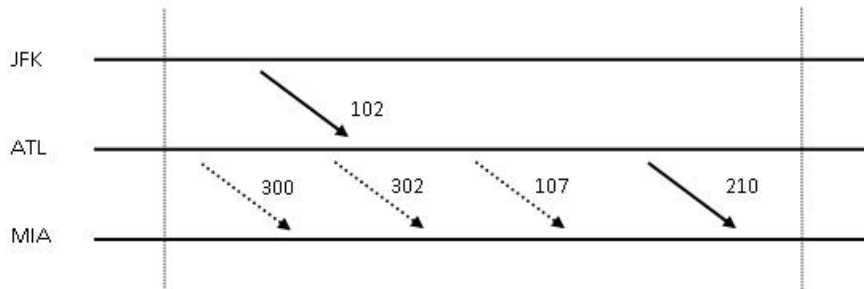


Figure 3: Disruptable flight: 302

#### 4.1.1 Limiting Other Resources

The preceding section has shown how we limit the scope of our recovery operation with respect to flight resources. To keep the problem size reasonable, we consider aircraft and crew swaps only among the set of disruptable resources - and not with those resources that are nondisruptable.

We allow the use of reserve crew  $K^{\text{reserve}} \subset \mathcal{K}$  that are defined by a base, and time interval in which they are allowed to operate. A penalty term is added for every reserve crew used in the model.

#### 4.1.2 Retiming Flights

Initial work on recovery modeled flight delays as making  $k$  copies of each flight arc that departed at uniform intervals. Events where crew and passengers connect are also relevant for considering delay alternatives (see Clarke [16] and Gao [23]). While the uniform flight copy approach is a simple and intuitive approach, it is

not attractive for our flight string model considering some strings will be dominated by ‘better’ strings.

Some constraints in the SRM are a function of time - referred to as *time-dependent constraints*. The number of delay options for a given flight is a function of the number of time dependent constraints that particular flight contains within a maximum delay period. The actual departure time (delay) of a flight is then determined in the string generation process. As an example, suppose flight  $f_i$  arrives to the same station before the departure of flight  $f_j$ . Suppose  $f_j$  is found in three time-dependent constraints corresponding to time intervals  $I_1, I_2, I_3$  that partition the maximum delay period. If the arrival time of flight  $f_1$  plus a given turn time is contained in the interval  $I_2$ , then there will be two delay options for flight  $f_j$  in the given string: one departing at the earliest ready time within  $I_2$  and the other corresponding to the earlier boundary point of  $I_3$ .

Figure 4 illustrates an example of how strings are generated from four flights. Each box corresponds to a flight with the tail of the outbound arc corresponding to departure time. The length of each rectangle represents the maximum delay time. Departing at the leftmost boundary of each box indicates the flight is not disrupted. The number of partitions for each flight corresponds to the number of time-dependent constraints that flight is in (e.g., airport flow restrictions, gates, curfew restrictions). The time-dependent constraints for which forbid a flight from operating (i.e., where flow restrictions are set to zero) are the shaded regions. The first flight in each string will consider the minimal delay within that sub-interval. Delay propagation may cause excessive delays, and strings only consider a valid flight connection given each delay. In the following example, when looking at the four-flight strings, there are 4 strings that can be generated with all valid connections. Table 1 shows all strings generated from these four flights.

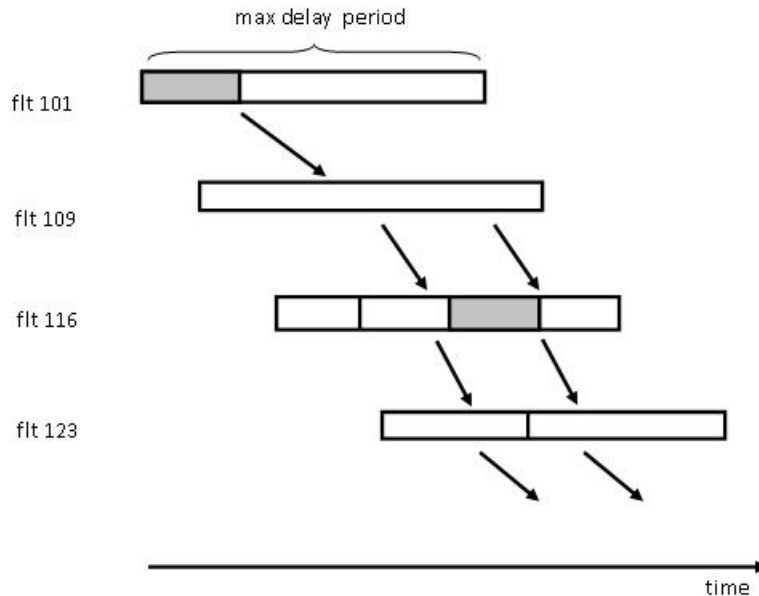


Figure 4: String Generation Process

Table 1

Four-flight strings from Figure 4				
String	Copy of Flight			
	101	109	116	123
1	1	1	1	1
2	1	1	1	2
3	1	1	2	2
4	1	2	2	2

## 5 Solution Methodology

Even by limiting the scope of the problem to make computational tractable, the problem is likely too large and complex to return a globally optimal solution for most reasonable disruption scenarios. Additional restrictions are further considered that we show in Section 6 that delivers an improved solution within a reasonable runtime.

Our approach is to return with a solution that is globally optimal with respect to aggregate passenger delay meaning passenger assignments are optimal over both all itineraries *and* all flight strings and cancellations. While this is clearly desirable for crew scheduling decisions as well, the crew recovery component is the bottleneck of the process and the number of repaired pairings is so large that optimizing over all possible (repaired) crew pairings *and* flight strings is unlikely to solve in an efficient manner. Therefore the crew scheduling decisions are *locally optimal*. Since total crew recovery costs are driven by the problem for pilots, we only solve the CRM over the cockpit crew. We do not solve the problem for cabin crew. Since the latter are not subject to as rigid of legality requirements, one could assume the cabin crew stay with the cockpit crew at the time of the disruption to obtain an upper bound on the true cost of the crew recovery component. This approach is just our approach; the model is flexible enough whereby the user could employ a more aggressive crew solution procedure.

Other than being computationally tractable, returning a globally optimal passenger solution has another advantage: it is more satisfying to passengers whose aggregate delay is at a minimum. Recent news headlines have read about excessive passenger delays inducing a sort of ‘passenger revolt’ and a number of variants for a passenger bill of rights have been proposed among Congress. Effective April 2010 the U.S. Department of Transportation has enacted a rule whereby airlines would be forced to pay up to \$27,500 for *each* passenger experiencing a tarmac delay in excess of three hours (U.S. Department of Transportation 49 U.S.C. 40113).

It should be noted that the PRM as formulated above seeks to assign the number of passengers to each itinerary to minimize aggregate delay. That is, we consider homogeneous passengers. In practice, attributes like fare class or frequent flyer status comprise of a more granular problem known as passenger reaccommodation that seeks which *individual* passengers are to be assigned to which flights.

### 5.1 Decomposition

Because scheduling decisions affect repaired aircraft rotations, crew schedules, and passenger itineraries, employing a Benders’ decomposition scheme would be natural to decompose the problem. Therefore the master problem will be the SRM model with linking variables  $\{x_{e,s}\}$ ,  $\{\kappa_f\}$  to be passed in to the subsequent subproblems: ARM, CRM, and PRM.

The master problem is therefore the SRM (Section 3.1.4) with five classes of Benders cuts: ARM feasi-

bility cuts, CRM feasibility cuts, CRM optimality cuts, PRM feasibility cuts, and PRM optimality cuts. We model the ARM is merely a feasibility problem (i.e.  $c_{e,s}^n = 0$  for all  $e \in E, s \in S, n \in AC(e)$ ) so ARM optimality cuts are not necessary. Only the relaxation of each of the three subproblems are solved so as to obtain coefficients of the Benders cuts. The master problem is first solved as an LP-Relaxation, and new strings are generated based on the corresponding dual extreme ray if the relaxed SRM is infeasible. Otherwise the MIP is solved. Obtaining integer solutions for the three subproblems is further discussed in Section 5.4.

The five families of Benders cuts that are included in the master are

$$\sum_{e \in E} \sum_{s \in S} \pi_{e,s}^{\text{ARM}} x_{e,s} \leq \pi_0^{\text{ARM}} \quad (5.1)$$

$$\sum_{f \in F} (1 - \kappa_f) \pi_f^{\text{CRM}} + \sum_{k \in K} \rho_k^{\text{CRM}} \leq 0 \quad (5.2)$$

$$\sum_{f \in F} (1 - \kappa_f) \pi_f^{\text{CRM}} + \sum_{k \in K} \rho_k^{\text{CRM}} \leq \eta^{\text{CRM}} \quad (5.3)$$

$$\sum_{e \in E} \sum_{s \in S} \sum_{f \in F} \mathbf{1}_{f,s} \pi_f^{\text{PRM}} x_{e,s} + \sum_{i \in PAX} \rho_i^{\text{PRM}} \leq 0 \quad (5.4)$$

$$\sum_{e \in E} \sum_{s \in S} \sum_{f \in F} \mathbf{1}_{f,s} \pi_f^{\text{PRM}} x_{e,s} + \sum_{i \in PAX} \rho_i^{\text{PRM}} \leq \eta^{\text{PRM}} \quad (5.5)$$

where the superscripts denote the given subproblem,  $\pi_0^{\text{ARM}}$  is a constant that depends on the dual variables from constraints (3.11) and (3.10),  $\eta^{\text{CRM}}$  and  $\eta^{\text{PRM}}$  are new decision variables in the master corresponding to the optimal objectives in the CRM and PRM, respectively. The cuts are ARM feasibility, CRM feasibility, CRM optimality, PRM feasibility, and PRM optimality, respectively.

## 5.2 Column Generation

Given the large number of flight strings and repaired crew pairings, only a subset of columns are generated through each of these problems. Many columns are generated through a residual network which is built from the flight network for flight strings and the crew network for repaired crew pairings. Given a directed network  $G = (V, A)$ , a dummy source and sink node are added in which a variable (flight string or repaired crew pairing) corresponds to an  $s-t$  path. Paths are constructed by computing the reduced cost for every arc  $a \in A$ . Arcs with a sufficiently high reduced cost are eliminated and resulting paths (columns) are generated. In order to generate many columns at once, a tolerance parameter  $\epsilon > 0$  is defined and all columns whose path  $p$  prices out less than  $\epsilon$  is then added. This method is sometimes known as path generation through an  $\epsilon$ -residual network (see [5] for a general description; Shaw [37] gives an example pertinent to a traditional crew pairing problem). A summary of this method is shown in Algorithm 1.

## 5.3 Simultaneous Row and Column Generation

The preceding section illustrates how we are employing both Benders cuts as well as column generation. While these two classical large-scale optimization methods are widely known, they are in isolation of one another. Given an infeasible or suboptimal subproblem a Benders cut  $f(x_{e,s}, \kappa_f) \leq f_0$  is added to the master problem. But this cut generated is valid only over the subset of strings  $S' \subseteq S$  that have been generated. Moreover in the case of the CRM where repaired crew pairings are also being generated, the given cut is valid only over those subset of pairings  $P' \subseteq P$  that have been generated.

We discuss two cases how these methods are used together. The first way deals with linking variables that are done in a brute force way. The second way shows how this is done with local variables present in only the CRM subproblem. A result is presented showing a certificate that proves the validity of the cut being returned to the master problem.

---

**Algorithm 1** Path Generation Through  $\epsilon$ -Residual Network

---

**Given:** List of resources  $R$ , general resource network  $G = (V, A)$ , dual information  $\pi_v \quad \forall v \in V$ , and tolerance parameter  $\epsilon > 0$   
**Initialize:** Generated variables  $X = \emptyset$   
**for**  $i = 1$  **to**  $|R|$  **do**  
    create augmented network for resource  $i$ ,  $G^i = (V, A)$   
    add source node  $s$  and sink node  $t$   
    construct all arcs from  $s$  to eligible initial nodes and arcs to  $t$  from eligible end nodes  
    **for all**  $a \in A$  **do**  
        compute reduced cost  $\bar{c}_a$   
        **if**  $\bar{c}_a > \epsilon$  **then**  
            **delete** arc  $a$ :  $A \leftarrow A \setminus \{a\}$   
        **end if**  
    **end for**  
    Let  $X^i$  denote all  $s - t$  paths in the residual network whose reduced cost is less than  $\epsilon$   
     $X \leftarrow X^i$   
**end for**  
**return** new columns  $X$

---

### 5.3.1 Flight Strings

Given a general Benders cut  $f(x_{e,s}, \kappa_f) \leq f_0$  it is valid over only those generated flight strings  $S' \subseteq S$ . As new strings are added, the Benders cut may be invalid for some  $s \in S \setminus S'$ . While to the best of our knowledge, there does not exist a way to overcome this barrier<sup>1</sup>, we simply remove the Benders cuts anytime new strings are added. Because cycling may occur once the cuts are deleted, we do not generate new strings within every iteration. Rather, they are generated every  $k > 1$  iterations from the LP-Relaxation of the master problem.

### 5.3.2 Repaired Crew Pairings

If at a given iteration the CRM is infeasible or suboptimal, a Benders cut is valid over the subset  $P'$  of all pairings  $P$ . A method is developed below that is checked before the cut is added to the master problem that is a certificate of infeasibility over all  $P$ , and thus the cut being valid over all  $P$ . If the certificate is not found, then the cut is not added to the master. In both cases, new columns are being generated while the certificate is obtained.

As discussed in section 5.2, columns (repaired crew pairings) are generated through a resource network called the *master crew duty network*  $G = (\mathcal{D}, A)$  where  $\mathcal{D}$  denotes the set of all duties that have been enumerated a priori, and  $A$  is the set of arcs that can legally connect two duties. Recall that a duty is a sequence of flights scheduled to be flown by a crew in a period of time that usually corresponds to one day. Given the stations and times at which crew  $k$  is at the time of the disruption, and where they need to be at the end of the disruption, the individual crew duty network  $G^k = (\mathcal{D}^k, A^k)$  is constructed. A source and sink node are added that connect all eligible initial and end duties, respectively. A path from source to sink is a repaired pairing such that the crew is at the station where originally scheduled at the end of the time window.

Recall the CRM formulation given in section 3.3. Let  $\pi_f$  and  $\rho_k$  denote dual variables for constraints 3.14 and 3.15, respectively. The reduced cost of a pairing  $p$  for crew  $k$  is given by  $\bar{c}_{k,p} = c_{k,p}^{\text{assign}} - \sum_{f \in p} \pi_f - \rho_k$ . As a special case of the CRM we consider a zero objective on the crew pairing assignments, i.e.  $c_{k,p}^{\text{assign}} = 0 \quad \forall k \in K, \forall p \in P$ . The primary reason for this is the crew recovery is quite different from the well-known crew pairing problem. In the traditional crew pairing problem, the objective is to minimize the sum of crew

---

<sup>1</sup>a related problem introduced by T. Van Roy is that of cross decomposition

costs that is known as *pay-and-credit*, which is a complex objective which factors in the total time the crew is away from their base, flying hours, and the number of duties. However by minimizing the number of deadheads, pay-and-credit can be reduced over the time window. Therefore the reduced cost of our special case becomes

$$\bar{c}_{k,p} = - \sum_{f \in p} \pi_f - \rho_k \quad (5.6)$$

Let  $P_k$  denote the set of all pairings for some crew  $k \in K$  where a subset  $P'_k \subseteq P'$  have been generated. The following result gives a certificate for which the CRM feasibility cuts remain optimal over all  $P$ .

**Theorem 5.1.** (Extending CRM feasibility cuts over new pairings) *Suppose the CRM is infeasible over a subset of pairings  $P' \subset P$ . Let  $\{\pi_f\}_{f \in F}$ ,  $\{\rho_k\}_{k \in K}$  denote a dual extreme ray. Denote*

$$\begin{aligned} K^{\geq} &:= \{k \in K : \rho_k \geq 0\} \\ F^{\leq} &:= \{f \in F : \pi_f \leq 0\} \\ P_k^{\leq} &:= \{p \in P'_k : f \in F^{\leq} \text{ for all flights } f \text{ in } p\} \end{aligned}$$

For crew  $k$  let  $P_k^{\text{new}} \subseteq P \setminus P'$  denote the new set of pairings generated for which  $\bar{c}_{k,p} \leq 0$ .

If  $\sum_{f \in F} (1 - \kappa_f) \pi_f + \sum_{k \in K} \rho_k < 0$  for every  $k \in K^{\geq}$  and  $p \in P_k^{\leq}$  then the CRM remains infeasible over the extended set of pairings  $P' \cup (\bigcup_{k \in K^{\geq}} P_k^{\text{new}})$

*Proof.* By Farkas' Lemma the CRM is infeasible over  $P'$  if and only if there exists some  $\{\sigma_f\}_{f \in F}$ ,  $\{\chi_k\}_{k \in K}$  such that the following system has a solution:

$$\begin{aligned} \sum_{f \in F} (1 - \kappa_f) \sigma_f + \sum_{k \in K} \chi_k &< 0 \\ \sum_{f \in p} \sigma_f + \chi_k &\geq 0 \quad \forall k \in K, \forall p \in P' \\ \sigma_f &\leq 0 \quad \forall f \in F \\ \chi_k &\geq 0 \quad \forall k \in K \end{aligned} \quad (\clubsuit)$$

As the CRM is infeasible let  $\{\pi_f\}_{f \in F}$ ,  $\{\rho_k\}_{k \in K}$  denote a dual extreme ray for the dual to the LP relaxation of the CRM. In order to construct a solution for  $(\clubsuit)$  over the newly generated pairings let

$$\sigma_f = \min \{\pi_f, 0\} \quad \forall f \in F, \quad \chi_k = \max \{\rho_k, 0\} \quad \forall k \in K$$

The first condition in  $(\clubsuit)$  holds by assumption, the latter two by construction, and the second condition holds as it is equivalent to the reduced cost condition (5.6) found by pricing out over a subgraph of  $G^k (D^k, A^k)$ .  $\square$

While Theorem 5.1 shows that the candidate Benders feasibility cut  $\sum_{f \in F} (1 - \kappa_f) \pi_f^{\text{CRM}} + \sum_{k \in K} \pi_k^{\text{CRM}} \leq 0$  is valid over  $P' \cup \bigcup_{k \in K} P_k^{\text{new}}$ , it remains to be seen that the cut is valid over all of  $P$ . The following result shows this is indeed the case.

**Theorem 5.2.** (Extending CRM feasibility cuts over all pairings) *Suppose the relaxed CRM is infeasible over a subset of pairings  $P' \subseteq P$ . If there exists a new crew pairing assignment such that the conditions of Theorem 5.1 hold, then the CRM is infeasible over all pairings  $P$ .*

*Proof.* It suffices to show that  $\sum_{f \in p} \sigma_f + \chi_k \geq 0 \quad \forall k \in K, \forall p \in P$ . Consider an arbitrary new pairing  $p \in P \setminus P'$  assigned to crew  $k$ . Summing over all previously eligible pairings for crew  $k$  yields

$$\sum_{f \in p} \tau'_f \sigma_f + \sum_{f \notin p} \tau''_f \sigma_f + n_k \chi_k \geq 0$$

where  $n_k$  represents the number of times previously assigned to crew  $k$  and  $0 \leq \tau'_f \leq n_k$  denotes the number of times flight  $f$  is in pairing  $p$  for crew  $k$  (and  $\tau''_f$  represents the number of inequalities  $f$  is not present). Note that  $\sigma_f \leq 0$ , so  $-\sum_{f \notin p} \tau''_f \sigma_f \geq 0$  is valid. Summing this inequality along with the preceding one gives

$$\sum_{f \in p} \sigma_f + \chi_k \geq \sum_{f \in p} \frac{\tau_f}{n_k} \sigma_f + \chi_k \geq 0$$

where the first inequality is valid as  $0 \leq \frac{\tau_f}{n_k} \leq 1$ . Therefore  $\bar{c}_{k,p} \leq 0$  for all  $p \in P$  and the candidate CRM feasibility cut is valid to add to the master problem.  $\square$

**Corollary 5.3.** *The same results from above hold to the case where the CRM is feasible, but not necessarily optimal over all  $P$ . The analogous condition to that given in Theorem 5.1 differs by only a constant on the right hand side.*

We do not include the proof for brevity, but is nearly identical to the preceding proofs. The detailed proof is available upon request to the authors.

Algorithm 2 summarizes the implementation of the two preceding theorems in the context of our solution strategy for the case of an infeasible CRM.

---

**Algorithm 2** Handling Column Generation and Constraint Generation Together in CRM

---

```

solve LP-Relaxation for CRM
initialize validCut = true
if CRM is infeasible over  $P'$  then
  Extract dual extreme ray  $\{\pi_f\}_{f \in F}, \{\rho_k\}_{k \in K}$ .
  Let  $\sum_{f \in F} (1 - \kappa_f) \pi_f^{\text{CRM}} + \sum_{k \in K} \pi_k^{\text{CRM}} \leq 0$  denote the candidate Benders feasibility cut
  for all crew  $k \in K : \rho_k \geq 0$  do
    Construct subgraph  $\tilde{G}^k(D, A)$  of crew duty network  $G^k$  where  $\pi_f \leq 0$  for all  $D$  in  $\tilde{G}^k$ 
    Generate new columns  $P_k^{\text{new}}$  over the  $\epsilon$ -residual network over  $\tilde{G}^k(D, A)$ 
    update columns  $P' \leftarrow P_k^{\text{new}}$ 
    if  $\sum_f (1 - \kappa_f) \pi_f + \sum_{k \in K} \rho_k \geq 0$  then
      set validCut = false
    end if
  end for
  if validCut = true then
    add candidate Benders cut to master problem
  end if
end if

```

---

## 5.4 Integrality

Our optimization module solves only the master problem (SRM) to integrality and solves the subsequent three subproblems in their respective LP relaxations. Once the iterative algorithm has terminated, then branching is done to find a nearby solution. If no feasible solution is found by branching, the algorithm begins again starting with the previous solution obtained. We discuss how integrality is obtained in each of the three subproblems.

**ARM Integrality** One of the advantages of the flight string models is it makes the routing problem nearly trivial to solve, as shown in Theorem 5.4.

**Theorem 5.4.** (ARM Integrality) *The polyhedron associated with the LP-Relaxation of the ARM is integral*

*Proof.* This problem reduces to a maximum cardinality bipartite matching problem for node sets aircraft-string assignments  $\{x_{e,s}^n\}$  and strings from the master problem  $\{x_{e,s}^*\}$ . This class of problems is well-known to be integral - see Nemhauser and Wolsey [30] for a good description.  $\square$

**CRM Integrality** Solving the LP-relaxation of the CRM induces integer solutions in many scenarios. However the polytope is itself not integral and we use the idea first proposed by Ryan and Falkner [35]. A *follow-on* is a sequence of consecutive flights that appear in multiple fractional pairings containing the same sequence. Branching on follow-ons has shown to be a successful branching strategy (see Anbil et al. [6], Lettovsky et al. [29], and Shaw [37]).

**PRM Integrality** Solving the PRM could be done through a multi-commodity network flow algorithm yielding integer solutions. However the associated polyhedra is highly integral and branching is done only in the presence of a fractional solution.

## 5.5 Overview

Figure 5 summarizes our approach to solving the AIR model.

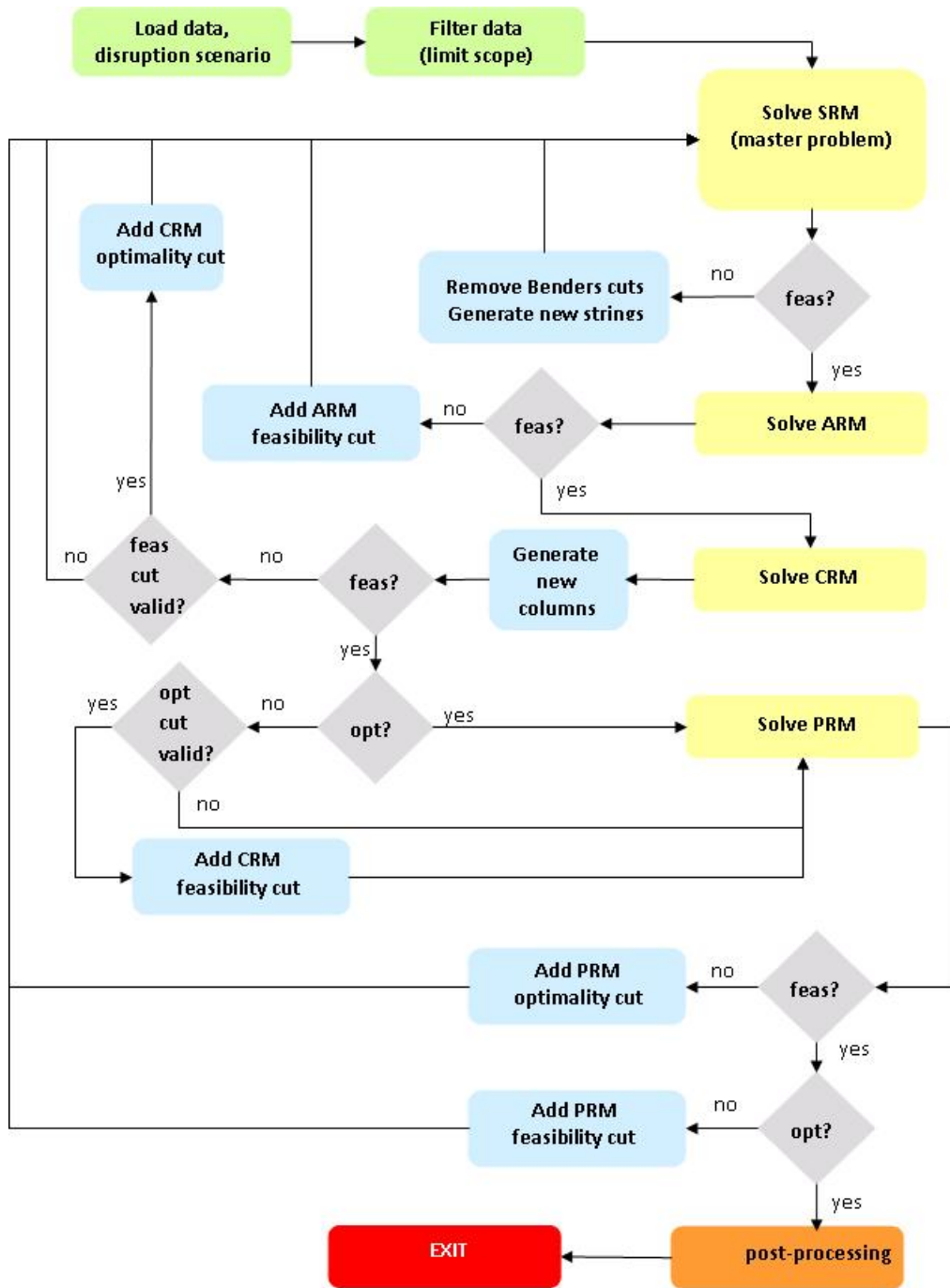


Figure 5: AIR Optimization Module

## 6 Computational Results

Our model is tested from actual 2007 data from a hub-and-spoke airline based in the U.S. with approximately 800 daily flights and two fleet types. The main disruption of interest is a flow rate reduction into the hub, and possibly other stations. We consider a reduction in terms of a certain percentage of scheduled operations as well as a full hub closure for some period of time. Table 2 summarizes the benchmark parameters used in the results obtained. Most figures have come from a priori knowledge about the given network and airline under consideration.

As shown in the table, the cost objective is only to minimize the cost associated with canceling flights, whilst ignoring the cost of assigning equipment to flight strings. An obvious alternative is to penalize all flights whose equipment type deviates from the schedule. The same could be said for assigning individual tails to flight strings. We use a warm start to the initial master problem corresponding to the sequence of scheduled routings. While the same sequence may be found in several strings, we use the initial solution whose delay is minimal across all flights in the string. Some instances have terminated with the warm start solution, and for other instances computation time was often significantly less than without the warm start.

As discussed above, overall crew costs are driven by pay-and-credit, which is the cost over the entire pairing and not the broken part of it. Deadhead costs within the broken pairing can proxy the total cost of the complete pairing, and therefore we abstract from individual assignment costs. Typically a large penalty would be assigned to the use of a reserve crew, but because of incomplete data, we do not explicitly model reserve crew (all crew presently on duty can be used, so reserve crew would not be needed).

Table 2

Benchmark Parameters Used in Computations		
parameter	description	value
$c_{e,s}^{\text{assign}}$	cost of assigning equipment $e \in E$ to string $s \in S$	\$0
$c_f^{\text{cancel}}$	cost of canceling flight $f \in F$	\$25,000
$c_{e,s}^n$	cost of assigning tail $n \in AC(e)$ to string $s \in S$	\$0
$c_{k,p}^{\text{assign}}$	cost of assigning crew $k$ pairing $p$	\$0
$d_f^{\text{pairing}}$	cost of deadheading on flight $f$ within a pairing	\$1,000
$d_k^{\text{base}}$	cost of crew $k$ deadheading to crew base	\$2,000
$c^{\text{delay}}$	cost in passenger goodwill per minute of delay	\$10
$c_i^{\text{unassign}}$	cost of unassigned itinerary for passenger $i$	\$2,500

Our model has been implemented in C++ using Concert/CPLEX 12.1 on a quad-core 2.66 GHz Xeon X5355 processor.

**Problem Size and Length of Disruption** Section 4 discussed how the scope of the recovery operation was limited so as not to affect all flights in the problem. Figure 6 shows how the number of disruptable flights grows with respect to the duration of closure at the hub beginning at 8:00 local time. While a one-hour

disruption affects nearly half the flights, every flight is disruptable when the length of the disruption reaches 105 minutes. This is partially due to the fact that the data set comes from a regional carrier whose flight legs are typically short relative to major carriers whose networks span a larger geographical region. This is readily seen as that every tail number has some activity at the hub between 8:00 and 9:45 AM local time.

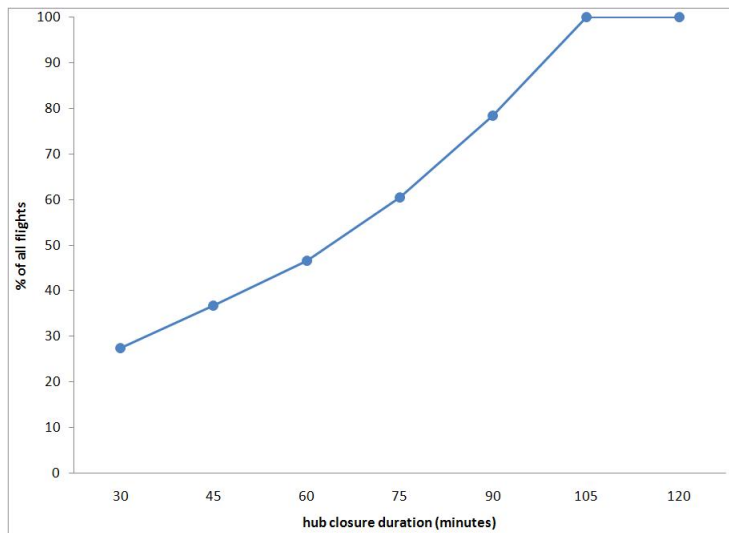


Figure 6: Disruptable Flights and Length of Hub Closure

**Build versus Repair of Crew Duty Network** One of the major bottlenecks in the solution process outlined above is the construction of, and generating paths through the crew duty network (which can also be expensive in terms of memory consumption). Because this network is apt to change for each new scheduling decision made in the master problem, there are two approaches how to manage the crew duty network. The first is to build it once before the iterative process begins, then heuristically repair broken duties and missed duty connections and repair the original network based on the current scheduling decisions. The second is to construct a new network entirely after each master solution. The obvious tradeoff is computational resources spent constructing the crew duty network and information of the true network. For disruptions terminating before 15:00, we define the end of the time window to be midnight of that day. If the time window includes more than one day, the number of connecting duties increases substantially thereby making the CRM even more complex. As a first attempt to study the AIR problem, we begin by keeping the time window to be one day so that the crew duty network can be rebuilt within each iteration. It may be naturally of interest to take the other approach for larger problems.

## 6.1 Disruption Scenarios

We model three classes of disruption scenarios:

1. 50 % reduction in flow rate (arrivals & departures)
2. 75 % reduction in flow rate (arrivals & departures)
3. 100 % reduction in flow rate (arrivals & departures)

Each scenario will examine four different disruption events characterized by a disruption time, disruption location, and time window shown in Table 3. The fourth column represents the time window  $\mathcal{T}$ . Events 1, 2, and 4 consider different durations of the disruption period that are to be returned to the schedule by the

end of the day. Scenario 3 considers two disruptions: one at the hub and the other at one of the largest spokes used in the network. Given the growth of problem size on the length of hub closure (see Figure 6), we consider a maximum hub disruption to be 75 minutes.

Table 3

Disruption Events				
event	disruption time	disruption location	time window $\mathcal{T}$	max delay time (minutes)
1	08:00 - 08:30	hub	08:00 - 23:59	90
2	08:00 - 09:00	hub	08:00 - 23:59	120
3	08:00 - 09:00 09:00 - 14:00	hub spoke	08:00 - 23:59	120
4	08:00 - 09:15	hub	08:00 - 23:59	150

The final column represents the maximal delay considered which has a profound affect on the number of strings being generated. For a two hour hub disruption, the total number of flight strings (that contain no more than 7 flights) increase from under 200,000 using a one hour maximum delay period to more than 2.6 million using a three hour maximum delay period.

## 6.2 Integrated versus Sequential Recovery

We report costs for all subproblems and important metrics that determine quality of solution. We do not report costs for the integrated model since it amounts to a feasibility problem, and is always feasible under the integrated approach. All delay times reported are all conditional on being delayed, and are presented in HH:MM format.

### 6.2.1 Disruption Scenario 1: 50% Flow Rate Capacity Reduction

Tables 4 and 5 show the first set of results for a 50% flow rate reduction into and out of the hub for the sequential process and integrated process, respectively.

Table 4

Sequential Recovery Summary (50% flow rate reduction)					
		Event			
		1	2	3	4
subproblem costs (\$)	SRM	0	0	0	0
	ARM	0	0	INFEAS	INFEAS
	CRM	0	0	INFEAS	INFEAS
	PRM	15,590	28,590	INFEAS	INFEAS
solution metrics	mean flt delay	13:51	22:20	37:47	36:10
	no. canceled flts	0	0	0	0
	total deadheads	0	0	INFEAS	INFEAS
	mean PAX delay	23:30	19:57	INFEAS	INFEAS
	unassigned PAX	0	0	INFEAS	INFEAS
	CPU time (MM:SS)	1:04	13:31	2:09	2:35

Table 5

Integrated Recovery Summary (50% flow rate reduction)					
		Event			
		1	2	3	4
subproblem costs (\$)	SRM	0	0	75,000	0
	CRM	0	0	6,000	0
	PRM	5,820	19,440	76,520	40,370
solution metrics	mean flt delay	12:32	13:55	41:29	30:53
	no. canceled flts	0	0	3	0
	total deadheads	0	0	3	0
	mean PAX delay	14:13	13:11	34:41	26:40
	unassigned PAX	0	0	6	0
	CPU time (MM:SS)	7:29	17:21	28:54	40:07

### 6.2.2 Disruption Scenario 2: 75% Flow Rate Capacity Reduction

The following two tables show the results from reducing capacity by 75%.

Table 6

Sequential Recovery Summary (75% flow rate reduction)					
		Event			
		1	2	3	4
subproblem costs (\$)	SRM	0	0	150,000	0
	ARM	0	0	INFEAS	INFEAS
	CRM	0	0	INFEAS	INFEAS
	PRM	16,890	31,440	INFEAS	INFEAS
solution metrics	mean flt delay	17:49	24:24	44:01	30:24
	no. canceled flts	0	0	6	0
	total deadheads	0	0	INFEAS	INFEAS
	mean PAX delay	24:04	18:08	INFEAS	INFEAS
	unassigned PAX	0	0	INFEAS	INFEAS
	CPU time (MM:SS)	1:30	10:02	2:39	3:15

Table 7

Integrated Recovery Summary (75% flow rate reduction)					
		Event			
		1	2	3	4
subproblem costs (\$)	SRM	0	0	50,000	0
	CRM	0	0	4,000	0
	PRM	6,960	29,640	74,630	49,230
solution metrics	mean flt delay	14:32	23:51	45:13	31:33
	no. canceled flts	0	0	2	0
	total deadheads	0	0	2	0
	mean PAX delay	11:13	15:34	31:57	20:46
	unassigned PAX	0	0	3	0
	CPU time (MM:SS)	9:09	15:59	29:30	44:23

### 6.2.3 Disruption Scenario 3: Hub Closure

Table 8

Sequential Recovery Summary (hub closure)					
		Event			
		1	2	3	4
subproblem costs (\$)	SRM	0	0	225,000	0
	ARM	0	0	INFEAS	0
	CRM	0	0	INFEAS	0
	PRM	6,680	47,730	INFEAS	66,730
solution metrics	mean flt delay	28:10	25:28	49:47	27:51
	no. canceled flts	0	0	9	0
	total deadheads	0	0	INFEAS	0
	mean PAX delay	17:56	22:23	INFEAS	31:58
	unassigned PAX	0	0	INFEAS	0
	CPU time (MM:SS)	0:35	17:50	24:41	31:01

Table 9

Integrated Recovery Summary (hub closure)					
		Event			
		1	2	3	4
subproblem costs (\$)	SRM	0	0	75,000	0
	CRM	0	0	6,000	0
	PRM	6,680	47,730	87,420	57,480
solution metrics	mean flt delay	28:10	25:28	25:13	26:53
	no. canceled flts	0	0	3	0
	total deadheads	0	0	4	0
	mean PAX delay	17:56	22:23	27:44	31:04
	unassigned PAX	0	0	6	0
	CPU time (MM:SS)	01:57	18:25	34:02	38:42

### 6.2.4 Analysis

In the cases where the solution for the integrated and sequential approaches coincide in the hub closure scenario, it is because the MIP start (original schedule with minimal re-timing decisions) is the optimal master solution. We note that the 75 minute disruption seems to prohibit meeting the goal of obtaining a solution in a 30 minute runtime. This is primarily because a larger share of all flights become disruptable, and that the number of strings is vastly higher due to a longer maximum flight delay period. However the first three instances show that the integrated module does deliver a solution within this goal (with the sole exception of a multi-airport hub closure). Moreover we note the improvement in solution quality the integrated approach delivers over the sequential one. In the scenarios in which a feasible solution exists, we have shown that both mean passenger delay and mean flight delay can be reduced by at least 30%. There are other scenarios that do not even deliver a feasible recovery solution under the sequential approach where one is delivered in an integrated framework.

Any subset of problems can be solved as well. Some studies have examined this problem ignoring the crew recovery component. By turning off the CRM module and focusing on the other three components, solutions can be obtained quickly and can handle much larger disruption scenarios.

## 7 Conclusion and Future Work

This paper sought to solve the airline integrated recovery problem by mathematical programming techniques yielding a passenger-friendly solution with crew considerations. Unless the disruption period affects only a small measure of flights, delivering a globally optimal solution is unlikely to be achieved within a reasonable runtime. Therefore schemes that limit the problem size and are able to decompose the problem efficiently are essential in the construction of the solution procedure. With these strategies implemented as we have discussed, we have shown that the AIR problem is solvable under several reasonable sized disruptions.

This paper was a first attempt to solve the fully integrated problem. Our integrated model has shown to be effective when no more than 65% of the flights are disruptable, and the time horizon is one day for this particular airline. When either of these criteria is violated, the model tends to grow too rapidly for the current implementation to handle. We reiterate that the airline under consideration operates a dense network and it may well handle longer time horizons on other networks. An efficient procedure has introduced that allows us to simultaneously consider constraint generation and column generation by working on a subnetwork of the original crew duty network when generating repaired crew duties.

There are a number of interesting questions that arise from our study. The main difficulty in solving such instances involving longer disruptions stems from large overhead costs in terms of building the crew duty network. Recall that in our procedure, this network is built after each solution from the master problem is found. Building the network, and more so generating paths over the network can be time consuming even for a one-day problem when the number of duties that can connect to other duties is relatively small. Extending this to a multi-day framework where the number of connecting duties grows rapidly makes this process unlikely to yield a satisfactory result in a 30 minute time frame. To handle such larger problems, it would likely be advantageous to build the network once before the optimization module is called based of the original flight schedule, and locally repair the network within each iteration as oppose to building it from scratch every iteration. Another relevant question is to seek whether the Benders cuts can be strengthened to obtain a tighter LP-Relaxation of the master problem. Convergence could be accelerated if this were possible by perhaps lifting in a subset of variables whose dual was zero, or by perhaps using polyhedral cuts in conjunction with the Benders framework. Finally, while we have shown a procedure to handle column generation and constraint generation simultaneously in the context of the CRM, no such procedure has been found in the SRM for which linking variables are being added. Results could likely be strengthened by considering an analogous method for the master problem.

## Acknowledgements

We are grateful that this work has been supported in large part by Sabre Holdings. We have also benefited from fruitful discussions with Michael Clarke, Barry Smith, George Nemhauser, Shabbir Ahmed, Tina Shaw, and Alex Kalyta.

## A Appendix

Algorithm 3 explicitly describes how disruptable resources are added. The algorithm calls individual procedures for obtaining disruptable resources from aircraft rotations, crew schedules, and passenger itineraries.

---

### Algorithm 3 Limiting Scope Algorithm

---

Given:

$\mathcal{T} := [\underline{t}, \bar{t}]$ : exogenous time window

$\mathcal{T}^d(a) \subseteq \mathcal{T}$ : disruption period at station  $a \in A$

Let  $\mathcal{F}$ ,  $\mathcal{AC}$ ,  $\mathcal{K}$ ,  $\mathcal{I}$  denote the set of *all* flights, aircraft, crew, and itineraries within  $\mathcal{T}$ , respectively.

Let  $F$ ,  $AC$ ,  $K$ ,  $I$  be the set of *disruptable* flights, aircrafts, crew, and passengers, respectively.

**Step 1:** Given  $\mathcal{T}^d(a)$ :

extract the initial disruptable flights  $F^{\text{rotations}}$  and aircraft  $AC^{\text{rotation}}$  from **Algorithm 4**

**Step 2:** Given  $F^{\text{rotations}}$  and  $AC^{\text{rotations}}$ :

extract the set of disrupted crew  $K \subseteq \mathcal{K}$ , and disruptable flights from crew  $F^{\text{crew}}$  along with the new disruptable aircraft  $AC^{\text{crew}}$  from **Algorithm 5**

**Step 3:** Given  $F^{\text{rotations}} \cup F^{\text{crew}}$  and  $AC^{\text{rotations}} \cup AC^{\text{crew}}$ :

extract the disruptable passengers  $I \subseteq \mathcal{I}$ , candidate move up flights  $F^{\text{pax}}$  along with the new set of aircraft  $AC^{\text{pax}}$  from **Algorithm 6**

**Step 4:** Define disruptable data as follows:

$$\begin{aligned} \text{(set of disruptable flights)} \quad F &= F^{\text{rotations}} \cup F^{\text{pax}} \cup F^{\text{crew}} \\ \text{(set of disruptable aircraft)} \quad AC &= AC^{\text{rotations}} \cup AC^{\text{pax}} \cup AC^{\text{crew}} \end{aligned}$$


---

---

### Algorithm 4 Extract Disruptable Flights from Routings

---

Given: disruption scenario  $\mathcal{T}^d(a)$ , scheduled flights  $\mathcal{F}$ , and aircraft  $\mathcal{AC}$

**Initialize:**

set of disruptable flights from rotations  $F^{\text{rotations}} = \emptyset$

set of disruptable aircraft from rotations  $AC^{\text{rotations}} = \emptyset$

**for all** aircraft  $n = 1$  to  $|\mathcal{AC}|$  **do**

extract all  $m_n$  flights in the routing  $\mathcal{F}^n = f_1^n, f_2^n, \dots, f_{m_n}^n$

**if** routing is disrupted **then**

extract first flight  $f_j^n$  whose resource is disrupted

$F^{\text{rotations}} \leftarrow F^{\text{rotations}} \cup \{f_j^n, f_{j+1}^n, \dots, f_{m_n}^n\}$

$AC^{\text{rotations}} \leftarrow AC^{\text{rotations}} \cup \{n\}$

**end if**

**end for**

**return**  $F^{\text{rotations}}, AC^{\text{rotations}}$

---

---

**Algorithm 5** Extract Disruptable Flights from Crew

---

```
for all crew  $k \in \mathcal{K}$  do
  extract all duties  $D^k$  that overlap with  $\mathcal{T}$ 
  for all duties  $d^k \in D^k$  do
    extract all flights in duty  $d^k$ ,  $\mathcal{F}^{d^k} = f_{1,k}^{a_1}, f_{2,k}^{a_2}, \dots, f_{m_k,k}^{a_{m_k}}$  ( $f_{j,k}^{a_j}$  : flight  $j$  in duty  $d_k$  flown by aircraft  $a_j$ )
    for all  $j = 1$  to  $|\mathcal{F}^{d^k}|$  do
      if  $f_{j,k}^{a_j} \in F$  then
        duty  $d^k$  is disrupted
         $F \leftarrow F \cup \{f_{j,k}^{a_j}, f_{j+1,k}^{a_{j+1}}, \dots, f_{m_k,k}^{a_{m_k}}\}$ 
         $K \leftarrow K \cup \{k\}$ 
      end if
      if  $a_j \notin A$  then
         $A \leftarrow A \cup \{a_j\}$  and all subsequent flights flown in the duty
      end if
    end for
  end for
end for
return disruptable crew  $K$ , disruptable flights  $F^{\text{crew}}$ , and disruptable aircraft  $AC^{\text{crew}}$ 
```

---

---

**Algorithm 6** Extract Disruptable Flights from Passenger Itineraries

---

```
Given: all passengers  $\mathcal{I}$ , disrupted flights  $F' \subseteq \mathcal{F}$ 
Initialize:
  set of disruptable passengers           $I$           =  $\emptyset$ 
  set of disruptable flights from move-ups  $F^{\text{pax}}$  =  $\emptyset$ 
  set of disruptable aircraft from move-ups  $AC^{\text{pax}}$  =  $\emptyset$ 
for all passengers  $i \in \mathcal{I}$  do
  if itinerary for pax  $i$  contains a flight from  $F$  then
     $I \leftarrow I \cup \{i\}$ 
    extract itinerary  $\gamma$  with flights  $f_1^\gamma, f_2^\gamma, \dots, f_{m_\gamma}^\gamma$ 
    if  $\gamma$  is a multi-flight itinerary ( $m_\gamma \geq 2$ ) then
      search for flight  $f \in \mathcal{F} \setminus F'$  such that  $f$  is an eligible move-up flight
      if move-up flight  $f$  with aircraft  $n$  exists then
         $F^{\text{pax}} \leftarrow F^{\text{pax}} \cup \{f\}$ 
         $AC^{\text{pax}} \leftarrow AC^{\text{pax}} \cup \{n\}$ 
      end if
    end if
  end if
end for
return disruptable passengers  $I$ , eligible move-up flights  $F^{\text{pax}}$ , and their associated aircraft  $AC^{\text{pax}}$ 
```

---

## References

- [1] K.F. Abdelghany, A.F. Abdelghany, and G. Ekollu. An integrated decision support tool for airlines schedule recovery during irregular operations. *European Journal of Operational Research*, 185(2):825–848, 2008.
- [2] R. Acuna-Agost, D. Feillet, P. Michelon, and S. Gueye. Rescheduling flights, aircraft, and passengers simultaneously under disrupted operations - a mathematical programming approach based on statistical analysis. submitted for Anna Valicek Medal, AGIFORS, 2009.
- [3] Y. Ageeva. Approaches to incorporating robustness into airline scheduling. MS thesis, Massachusetts Institute of Technology, Aug 2000.
- [4] S. AhmadBeygi, A. Cohn, Y. Guan, and P. Belobaba. Analysis of the potential for delay propagation in passenger airline networks. *Journal of Air Transport Management*, 14(5):221–236, 2008.
- [5] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Upper Saddle River, New Jersey, 1993.
- [6] R. Anbil, R. Tanga, and E.L. Johnson. A global approach to crew pairing optimization. *IBM Systems Journal*, 31(1):71–78, 1992.
- [7] M.F. Argüello, J.F. Bard, and G. Yu. A grasp for aircraft routing in response to groundings and delays. *Journal of Combinatorial Optimization*, 5(1):211–228, 1997.
- [8] M. Ball, C. Barnhart, G.L. Nemhauser, and A. Odoni. Air transportation: Irregular operations and control. In C. Barnhart and G. Laporte, editors, *Transportation Volume 14: Handbook of Operations Research and Management Science*. Elsevier, Amsterdam, 2007.
- [9] C. Barnhart, N.L. Boland, L.W. Clarke, E.L. Johnson, and G.L. Nemhauser. Flight string models for aircraft fleetting and routing. *Transportation Science*, 32(3):208–220, 1998.
- [10] S. Bisailon, J.-F. Cordeau, G. Laporte, and F. Pasin. A large neighbourhood search heuristic for the aircraft and passenger recovery problem. <http://neumann.hec.ca/chairelogistique/jfc.shtml>, April 2010.
- [11] P.A. Bonnefoy. *Development of Multi-Airport Systems: A Worldwide Approach*. PhD dissertation, MIT, Engineering Systems Division, Jun 2008.
- [12] S. Bratu and C. Barnhart. Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, 9:279–298, 2006.
- [13] E.K. Burke, P. De Causmaecker, G. De Maere, J. Mulder, M. Paelinck, and G. Vanden Berghe. A multi-objective approach for robust airline scheduling. *Computers and Operations Research*, 37(5):822–832, 2010.
- [14] J.M. Cao and A. Kanafani. Real-time decision support for integration of airline flight cancellations and delays, part i: Mathematical formulation. *Transportation Planning and Technology*, 20:183–199, 1997.
- [15] J.M. Cao and A. Kanafani. Real-time decision support for integration of airline flight cancellations and delays, part ii: Algorithm and computational experiments. *Transportation Planning and Technology*, 20:201–217, 1997.
- [16] M. D.D. Clarke. *Development of Heuristic Procedures for Flight Rescheduling in the Aftermath of Irregular Airline Operations*. PhD dissertation, MIT, International Center for Air Transportation, Feb 1998.
- [17] M. D.D. Clarke. Irregular airline operations: A review of the state-of-the-art practice in airline operations control center. *Journal of Air Transport Management*, 4(2):67–76, 1998.

- [18] J. Clausen, A. Larsen, J. Larsen, and N.J. Rezanova. Disruption management in the airline industry - concepts, models and methods. *Computers and Operations Research*, 37:809–821, 2010.
- [19] M. Dunbar, G. Froyland, and C.-L. Wu. Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. *Optimization online*, 2010.
- [20] N. Eggenberg, M. Bierlaire, and M. Salani. A column generation algorithm for disrupted airline schedules. Technical report, Ecole Polytechnique Federale de Lausanne, 2007.
- [21] J. Filar, P. Manyem, and K. White. How airlines and airports recover from schedule perturbations: A survey. *Annals of Operations Research*, 108:315–333, 2001.
- [22] S. Gabteni. Roadeff competition winners methods review, May 2009. Agifors Airline Operations Meeting, Montreal.
- [23] C. Gao. *Airline Integrated Planning and Operations*. PhD dissertation, Georgia Institute of Technology, Aug 2007.
- [24] C. Gao, E. Johnson, and B. Smith. Integrated airline fleet and crew robust planning. *Transportation Science*, 43(1):2–16, 2009.
- [25] A.I.Z. Jarrah, G. Yu, N. Krishnamurthy, and A. Rakshit. A decision support framework for airline flight cancellations and delays. *Transportation Science*, 27(3):266–280, 1993.
- [26] D. Klabjan, A.J. Shaefer, E.L. Johnson, and G.L. Nemhauser. Robust airlines crew scheduling. Technical report, The Logistics Institute, Georgia Institute of Technology, 2002.
- [27] N. Kohl, A. Larsen, J. Larsen, A. Ross, and S. Tiourine. Airline disruption management - perspectives, experiences and outlook. *Journal of Air Transport Management*, 13(3):149–162, 2007.
- [28] L. Lettovsky. *Airline Operations Recover: An Optimization Approach*. PhD dissertation, Georgia Institute of Technology, Oct 1997.
- [29] L. Lettovsky, E.L. Johnson, and G. Nemhauser. Airline crew recovery. *Transportation Science*, 34(4):337–348, 2000.
- [30] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, New York, New York, 1999.
- [31] R. Nissen and K. Haase. Duty-period-based network model for crew rescheduling in european airlines. *Journal of Scheduling*, 9(3):255–278, 2006.
- [32] M. Palpant, M. Boudia, C.-A. Robelin, S. Gabteni, and F. Laburthe. Roadeff 2009 challenge: Disruption management for commercial aviation, 2009. <http://challenge.roadeff.org/2009>.
- [33] J. Rosenberger, E.L. Johnson, and G.L. Nemhauser. Rerouting aircraft for airline recovery. *Transportation Science*, 37(4):408–421, 2003.
- [34] J.M. Rosenberger, E.L. Johnson, and G.L. Nemhauser. A robust fleet-assignment model with hub isolation and short cycles. *Transportation Science*, 38(3):357–368, 2004.
- [35] D.M. Ryan and J.C. Falkner. A bus crew scheduling system using a set partitioning model. *Asia-Pacific Journal of Operational Research*, 4:39–56, 1997.
- [36] J.-P. Clarke S. Lan and C. Barnhart. Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science*, 40(1):15–28, 2006.
- [37] T. Shaw. *Hybrid Column Generation for Large Networks Routing Problems: with Implementations in Airline Crew Scheduling*. PhD dissertation, Georgia Institute of Technology, April 2003.

- [38] S. Shebalov and D. Klabjan. Robust airline crew pairing: Move-up crews. *Transportation Science*, 40(3):300–312, 2006.
- [39] B. Smith. *Robust Airline Fleet Assignment*. PhD dissertation, Georgia Institute of Technology, Aug 2004.
- [40] B.C. Smith and E.L. Johnson. Robust airline fleet assignment: Imposing station purity using station decomposition. *Transportation Science*, 40(4):497–516, 2006.
- [41] M. Song, G. Wei, and G. Yu. A decision support framework for crew management during airline irregular operations. In G. Yu, editor, *Operations Research in the Airline Industry*. Kluwer Academic Publishers, Boston, 1998.
- [42] G. Stojković, F. Soumis, J. Desrosiers, and M. Solomon. An optimization model for a real-time flight scheduling problem. *Transportation Research, Part A*, 36:779–788, 2002.
- [43] M. Stojkovic and F. Soumis. An optimization model for the simultaneous operational flight and pilot scheduling problem. *Management Science*, 47(9):1290–1305, 2001.
- [44] M. Stojković, F. Soumis, and J. Desrosiers. The operational airline crew scheduling problem. *Transportation Science*, 32(3):232–245, 1998.
- [45] D. Teodorovic and G. Guberinic. Optimal dispatching strategy on an airline network after a schedule perturbation. *European Journal of Operational Research*, 15:178–182, 1984.
- [46] D. Teodorovic and G. Stojkovic. Model for operational daily airline scheduling. *Transportation Planning and Technology*, 14(4):273–285, 1990.
- [47] D. Teodorovic and G. Stojkovic. Model to reduce airline schedule disturbances. *Journal of Transportation Engineering*, 121(4):324–331, 1995.
- [48] B.G. Thengvall, Y. Gang, and J.F. Bard. Multiple fleet aircraft schedule recovery following hub closures. *Transportation Research, Part A*, 35:289–308, 2001.
- [49] G. Wei, G. Yu, and M. Song. Optimization model and algorithm for crew management during airline irregular operations. *Journal of Combinatorial Optimization*, 1:305–321, 1997.
- [50] O. Weide, D. Ryan, and M. Ehrgott. An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers and Operations Research*, 37(5):833–844, 2010.
- [51] S. Yan and C.-G. Lin. Airline scheduling for the temporary closure of airports. *Transportation Science*, 31(1):72–82, 1997.
- [52] S. Yan and Y-P. Tu. Multifleet routing and multistop flight scheduling for schedule perturbation. *European Journal of Operations Research*, 103(1):155–169, 1997.
- [53] S. Yan and D-H. Yang. A decision support framework for handling schedule perturbation. *Transportation Research B*, 30(6):405–419, 1996.
- [54] J.W. Yen and J.R. Birge. A stochastic programming approach to the airline crew scheduling problem. *Transportation Science*, 40(1):3–14, 2006.
- [55] Y. Zhang and M. Hansen. Real-time intermodal substitution: Strategy for airline recovery from schedule perturbation and for mitigation of airport congestion. *Transportation Research Record*, 2052:90–99, 2008.