Transportation Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

An Optimization Approach to Airline Integrated Recovery

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

TRANSPORTATION SCIENCE
Vol. 46, No. 4, November 2012, pp. 482–500
ISSN 0041-1655 (print) | ISSN 1526-5447 (online)

http://dx.doi.org/10.1287/trsc.1120.0414

An Optimization Approach to Airline Integrated Recovery

Jon D. Petersen, Gustaf Sölveling
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332
{petersen@gatech.edu, gustaf.solveling@gatech.edu}

John-Paul Clarke
The Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332,
johnpaul@gatech.edu

Ellis L. Johnson
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia 30332,
ellis.johnson@gatech.edu

Sergey Shebalov
Sabre Holdings Research, Southlake, Texas 76092, sergey.shebalov@sabre.com

Although the airline industry has benefited from advancements made in computational and operational research methods, most implementations arise from the frictionless environment of the planning stage. Because 22% of all flights have been delayed and 3% have been cancelled in the United States 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 fully integrated airline recovery problem. After our solution methodology is presented, it is tested using data from an actual U.S. carrier with a dense hub-and-spoke network using a single-day horizon. It is shown that in several instances an integrated solution is delivered in a reasonable runtime. Moreover, we show the integrated approach can substantially improve the solution quality over the incumbent sequential approach. To the best of our knowledge, we are the first to present computational results on the fully integrated problem.

Key words: irregular operations; airline recovery; integrated recovery
History: Received: September 2010; revisions received: May 2011, September 2011; accepted: January 2012. Published online in *Articles in Advance* June 21, 2012.

Introduction

The airline industry has been one of the biggest beneficiaries of 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 in an environment ignoring disruptions. However in practice, operations are rife with frictions caused by disturbances such as inclimate 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 those of the planning phase, the former's problems are exacerbated by two things. The first are additional operational complexities that arise. For example, suppose an aircraft is approaching its destination but is unable to land because of convective weather. The aircraft may be placed into a holding pattern, requiring additional flying time for the cockpit crew. By the time the aircraft lands, the crew may not be legal to fly their subsequent leg because they have exceeded their allowed flying time within a 24-hour period, rendering a disruption to the subsequent legs. The second problem is that of timing. Most airlines utilize an operations control center (OCC) that provides a centralized decision making environment. Unlike the planning phase in which problems are sometimes made more than a year in advance of operations, OCC coordinators are constrained to making decisions in as close to real-time as possible. Because decisions involving repairing the schedule, aircraft, crew, and passengers are combinatorial in nature, using an optimization-based approach may not be tractable because of the complexity of solving each of these operational problems. As a result, 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 operations research (OR) practitioner, given the challenges posed. The immense nominal costs also make it of interest to an airline. Although estimates vary, these are generally considered to be tens of billions of dollars annually in the United States alone (see Bonnefoy 2008). Airline passengers also have a vested interest in the problem because passenger delays have become more problematic as the growth in air transportation has outpaced that of capacity at major airports. In some instances passenger delays have drawn global attention as passengers have been subjected to excessively long tarmac delays. These occurrences have, in part, prompted the U.S. Congress to draft a passengers' bill of rights. Effective April 2010 the U.S. Department of Transportation implemented a fine of up to \$27,500 per passenger for an airline that exceeds a tarmac delay of three hours. Although there have been some advancements made in applying mathematical programming to the operational phase of airline scheduling, little advancements have been implemented 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 timely manner. Such a solution scheme may not be of use to an OCC—for example, the recovered flight schedule may not be feasible for existing crew schedules.

The principle goal of this paper is to define, formulate, solve, and analyze a fully integrated recovery problem in a manner that is amenable to the constraints imposed by an OCC. By heuristically reducing the set of disruptable resources that are to be rescheduled, we propose an optimization module that is to reassign the schedule, aircraft, crews, and passengers within some time horizon. We validate our method by providing computational results using data from a real U.S.-based airline. To the best of our knowledge, we are the first to provide such 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 1 provides a review of relevant work within irregular airline operations. The problem and model are formally defined in §2. Section 3 discusses how the scope of the recovery operation is limited to make the problem solvable. Our decomposition scheme is outlined in §4. Computational results are shown in §5 that validate our approach. Here we observe the improvement the integrated approach yields relative to several key performance metrics. Section 6 summarizes our work and suggests related future work.

1. Literature Review

Although there has been relatively little prior work for studying the airline integrated recovery problem, various components within the problem have been studied. We review some of the seminal earlier work. Filar, Manyem, and White (2001) provide an exceptional survey of previous work. Clausen et al. (2010) give a recent state-of-the-art overview of disruption management of schedule, aircraft, crew, passenger, and integrated recovery. An overview of the decision-making environment at OCCs is given by Clarke (1998b) and Kohl et al. (2007).

Early work on irregular operations focused on repairing a schedule when one or more aircraft are taken out of operation as studied by Teodorovic and Guberinic (1984). Teodorovic and Stojković (1990, 1995) developed a model that is to repair scheduling decisions while minimizing the impact on passengers and crew, respectively. Jarrah et al. (1993) 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. Clarke (1998a) 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 together. The model considers additional constraints posed by air traffic control (ATC). Three multicommodity network flow models are presented in Thengvall, Yu, and Bard (2001) for schedule recovery that follows a hub closure. Each model considers flight cancellations, delays, ferrying, and swaps. Their results show that swapping opportunities have a substantial impact in the solution quality.

Aircraft recovery seeks to schedule individual aircraft in a way that must be maintenance feasible. In addition to preserving maintenance schedules, a model proposed by Stojković et al. (2002) also preserves ground service, crew connections, and passenger connections. The dual to their proposed model is a network model that allows for computation in near real-time. Rosenberger, Johnson, and Nemhauser (2003) develop a set packing model that assigns routes to aircraft by minimizing an objective that comprises both the assignment and cancellation costs. Maintenance feasibility is preserved by enumerating all routings involving a maintenance activity a priori.

To our knowledge Wei, Yu, and Song (1997) were the first to study crew recovery through a comprehensive multicommodity network flow framework. 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. Stojković, Soumis, and Desrosiers (1998) 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. Our work is strongly related to Lettovsky, Johnson, and Nemhauser (2000). Given the set of cancelled 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. Stojković and Soumis (2001) 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.

Passenger recovery was studied in Bratu and Barnhart (2006), who suggest a framework that can reduce passenger disruptions while holding down other scheduling costs in the presence of a disruption. Their model allows flight delays and cancellations and assigns reserve crew and spare aircraft to accommodate the new schedule. Zhang and Hansen (2008) propose integrating other modes of transportation to accommodate disrupted passengers. Such intermodal connections are often preferred particularly when the destination is nearby the disrupted station within a hub-and-spoke network.

An area closely related to recovery is schedule robustness. The central idea is to design a schedule that is able to be recovered more efficiently in the presence of irregularity. Robust scheduling was studied extensively in Smith (2004), Rosenberger, Johnson, and Nemhauser (2004), and Smith and Johnson (2006). Crew robustness was also studied by Klabjan et al. (2002), Shebalov and Klabjan (2006), Ball et al. (2007), and Gao, Johnson, and Smith (2009). The impact of schedule robustness to passenger recovery can be seen in Lan, Clarke, and Barnhart (2006).

A number of studies aim to partially integrate operations under irregularity. The 2009 ROADEF challenge (see Palpant et al. 2009) introduced a competition that sought to deliver a recovery solution that was to integrate the schedule, aircraft, and passengers. Gabteni (2009) presents an overview of the proposed methodologies. The winning team, seen in Bisaillon et al. (2010), employs a large-scale neighborhood search heuristic that iteratively constructs, repairs, and improves solutions and that incorporates randomness to diversify the search procedure.

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 doctoral dissertations by Lettovsky (1997) and Gao (2007). 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.

2. The Airline Integrated Recovery Problem

We formally define the *airline recovery problem* to comprise 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 routings to accommodate the repaired schedule that are feasible for the constraints imposed by maintenance requirements.
- The *crew recovery problem* assigns individual crew members to flights according to the repaired schedule, to satisfy the complex legality requirements.
- The *passenger recovery problem* reassigns disrupted passengers to new itineraries that deliver them to their destination.

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 modified. Each component may have a different interval, although we restrict our analysis to the same horizon. The requirement is that all components be back on their original (undisrupted) schedule by the end of the time window $\bar{t}$.

2.1. Schedule Recovery

The schedule recovery model (SRM) returns retiming and flight cancellation decisions. Our model is closely related to Clarke (1998a) in that we consider additional constraints imposed by air traffic control systems.

Instead of a leg-based model, we utilize flight strings, a concept introduced by Barnhart et al. (1998). A *flight string* (which we refer to as *string*) is a sequence of flights, with timing decisions, to be operated by the same aircraft. The same sequence of flights might be present in multiple strings, although each sequence must have a unique set of retiming decisions. A string-based model has a number of advantages. Although the number of strings naturally grows significantly with respect to the number of flights, efficient column generation techniques can be employed. Strings are also able to capture network effects that individual flight decisions do not. Also, ground arcs need not formally be defined in the underlying time-space network. The biggest advantage is that integer solutions to the aircraft recovery problem (discussed in §2.2) are immediately obtained from the LP-relaxation.

2.1.1. Sets.

F: set of all flight legs;

E: set of equipment types (fleets);

S: set of flight strings;

A: set of all airports;

A^{arr}: set of arrival slot capacities specified by an inbound station, arrival limit, and time interval;

A^{dep}: set of departure slot capacities specified by an outbound station, departure limit, and time interval;

G: set of gate restrictions specified by a station, gate limit, and time interval;

$I(a, \underline{t}^a, \bar{t}^a)$: 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 $\bar{t}^a$;

$W(a, \underline{t}^a, \bar{t}^a)$: set of strings that occupy a gate at station a between $\underline{t}^a$ and $\bar{t}^a$;

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

F^{market}: set of flights that have exogenous market requirements set by the airline that require a minimum number of flights or seats to be offered in a given segment.

2.1.2. Data.

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

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

CAP_e: capacity of equipment type $e \in E$;

n_f^{seats}: minimum number of seats required by flight $f \in F^{\text{market}}$.

2.1.3. Decision Variables.

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

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

2.1.4. SRM Formulation. The SRM formulation is given as follows:

$$\min \sum_{e\in E}\sum_{s\in S} c_{e,s}^{\text{assign}} x_{e,s} + \sum_{f\in F} c_f^{\text{cancel}} \kappa_f \tag{1}$$

$$\text{s.t.} \sum_{e\in E}\sum_{\substack{s\in S:\\ s\ni f}} x_{e,s} + \kappa_f = 1 \quad \forall f \in F, \tag{2}$$

$$\sum_{e\in E}\sum_{\substack{s\in S:\\ s\ni f}} x_{e,s} = 1 \quad \forall f \in F^{\text{strategic}}, \tag{3}$$

$$\sum_{e\in E}\sum_{s\in I(a,\underline{t}^a,\bar{t}^a)} x_{e,s} \le n_a^{\text{arr}} \quad \forall (a, n_a^{\text{arr}}, \underline{t}^a, \bar{t}^a) \in A^{\text{arr}}, \tag{4}$$

$$\sum_{e\in E}\sum_{s\in O(a,\underline{t}^a,\bar{t}^a)} x_{e,s} \le n_a^{\text{dep}} \quad \forall (a, n^{\text{dep}}, \underline{t}^a, \bar{t}^a) \in A^{\text{dep}}, \tag{5}$$

$$\sum_{e\in E}\sum_{s\in W(a,\underline{t}^a,\bar{t}^a)} x_{e,s} \le n_a^{\text{gates}} \quad \forall (a, n_a^{\text{gates}}, \underline{t}^a, \bar{t}^a) \in G, \tag{6}$$

$$\sum_{e\in E}\sum_{s: s\ni f} \text{CAP}_e x_{e,s} \ge n_f^{\text{seats}} \quad \forall f \in F^{\text{market}}, \tag{7}$$

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

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

The objective (1) is to minimize the aggregate cost of string assignment (including retiming decisions) and flight cancellations. Flight assignment constraints, as seen in (2), either require a flight to be contained in exactly one string or cancelled. To prohibit strategic flights from being cancelled, constraints of the form (3) are added. Arrival and departure capacities at certain airports at given time intervals are not to be exceeded, as captured in (4) and (5), respectively. Constraints (6) ensures the number of aircraft on the ground does not exceed the number of gates available at certain stations and times. Market requirements are captured in (7); they ensure that a minimum number of seats is operated on certain flights. There are also other constraints that prohibit certain resources from being assigned to certain flights that we do not explicitly include for brevity. 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.

2.2. Aircraft Recovery

The aircraft recovery model (ARM) assigns individual tail numbers to strings while meeting maintenance and other aircraft requirements. The ARM is solved for each equipment type $e \in E$.

2.2.1. Sets.

$\mathrm{AC}(e)$: set of aircraft of equipment type $e \in E$;

$A^{\mathrm{maint}}(e)$: set of maintenance stations capable of maintenance of equipment type $e \in E$;

$H(e)$: set of aircraft of type $e \in E$ that requires maintenance activity within the time window $\mathcal{T}$;

$S_n(a, t_{\min}, T)$: set of eligible strings to be flown by aircraft $n \in \mathrm{AC}(e)$ that visit station $a \in A^{\mathrm{maint}}(e)$ for at least $t_{\min}$ units of time within subinterval $T \subset \mathcal{T}$.

2.2.2. Data.

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

2.2.3. Decision Variables.

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

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

$$\min \sum_{n \in AC(e)} \sum_{s \in S} c^n_{e,s} x^n_{e,s} \tag{8}$$

$$\text{s.t.} \sum_{n \in AC(e)} x^n_{e,s} = x_{e,s} \quad \forall s \in S, \tag{9}$$

$$\sum_{s \in S} x^n_{e,s} = 1 \quad \forall n \in \mathrm{AC}(e), \tag{10}$$

$$\sum_{s \in S_n(a, t_{\min}, T)} x^n_{e,s} \geq 1 \quad \forall (n, a, t_{\min}, T) \in H(e), \tag{11}$$

$$x^n_{e,s} \in \{0, 1\} \quad \forall s \in S, \forall n \in \mathrm{AC}(e). \tag{12}$$

The objective (8) minimizes the cost associated with aircraft assignment. The cost can be thought of penalties or bonuses. For instance, a penalty may be imposed for any deviation from the original routing. The string cover constraints (9) assure that each string chosen from the SRM is assigned to some eligible aircraft. Constraints (10) ensure each aircraft is assigned to precisely one string. In the event that the required 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. Maintenance cover constraints are seen in (11). This simply ensures that at least one maintenance opportunity is built in for all tail numbers requiring maintenance. The inputs to this class of constraints includes the eligible station(s), latest possible time for service, and minimum time duration necessary to perform the maintenance event. Different types of maintenance checks can be incorporated into these constraints with the given parameters required. The specific maintenance planning of choosing which event opportunities are to be utilized can be done post-optimization. Other constraints we include but do not explicitly formulate are user-dependent constraints prohibiting certain aircraft from operating at some airports and similar operational restrictions.

2.3. Crew Recovery

Crew members are assigned to *pairings*, which comprise *duties* that contain specific flight assignments over a period of time. Each consecutive duty assignment must observe a rigid set of legality rules as mandated by the Federal Aviation Administration (FAA) and possible additional airline and union requirements. A duty typically represents a single day of flying, and the pairing usually spans between two and four duties. A *roster* period consists of a number of pairings over a period of time, typically 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 members 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 original pairing for the crew, although it may be not be the case for the roster period; in this case, the roster 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 minimum 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* really means "the broken component of the original crew pairing."

Crew deadheading is an important component to the crew recovery process. Formally, a *deadhead* occurs when a crew member is transported on a flight but does not operate the aircraft. Deadheading occurs during the recovery process when a schedule imbalance creates a shortage or surplus of crew members at a given station. There are two classes of deadheads. The first is deadheading within a pairing, i.e., when a crew member deadheads to some station to then operate a subsequent flight. The second class of deadheading is when crew members deadhead home to their crew base, ending their current pairing. This is common when stringent legality requirements are nearly exhausted for a crew and no pairing can be assigned during the time window. Airlines typically have vastly different policies on deadheading crew members. Our module requires penalties for each class of deadheads that occurs.

2.3.1. Sets.

K: set of all available crew members;

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* $k \in K$ if

(i) p begins at the station where crew k is at the beginning $\underline{t}$ of the time window $\mathcal{T}$;

(ii) p ends at the station where crew k is required to be at the end $\bar{t}$ of the time window $\mathcal{T}$; and

(iii) all flight, duty, and pairing legality requirements are satisfied.

2.3.2. Data.

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

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

d_k^{base}: cost of deadheading crew $k \in K$ back to base.

2.3.3. 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 on flight, $f \in F$ (deadheads within pairing).

2.3.4. CRM Formulation. The CRM model we consider for equipment type $e \in E$, CRM(e), is as follows:

$$\min \sum_{k\in K}\sum_{p\in P_k} c_{k,p}^{\text{assign}} y_{k,p} + \sum_{f\in F} d_f^{\text{pairing}} s_f + \sum_{k\in K} d_k^{\text{base}} \nu_k \tag{13}$$

$$\text{s.t.} \sum_{k\in K}\sum_{\substack{p\in P_k:\\ p\ni f}} y_{k,p} - s_f = 1-\kappa_f \quad \forall f \in F, \tag{14}$$

$$\sum_{p\in P_k} y_{k,p} + \nu_k = 1 \quad \forall k \in K, \tag{15}$$

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

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

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

The objective (13) seeks to minimize total crew cost; (14) ensures that some crew operates each flight that is not cancelled. If $s_f > 0$, the flight is to contain at least one crew that is to deadhead on a pairing. Constraints (15) assigns each crew to either some eligible pairing or to their home crew base.

2.4. Passenger Recovery

There are two components to the passenger recovery process. The first is an iterative module by which the costs from aggregate itinerary delays are minimized by integration with the SRM, ARM, and CRM. The second problem takes the eligible set of itineraries from the first problem and assigns itineraries to passenger groups to minimize the actual cost associated with passenger delay.

Each passenger is defined as a four-tuple consisting of origin, departure time at origin, destination, and scheduled time of arrival at destination. All possible eligible itineraries are generated a priori from the original flight schedule. Some itineraries are constructed even though they may be infeasible from the initial schedule because they may become feasible with delays. For example, consider a passenger scheduled to depart at 8:00. If there is a flight between the same origin and destination scheduled to depart at 7:00, then that flight might be able to be used in the recovery solution if it experiences a delay of at least one hour. If it does not, then constraints will prohibit the use of that itinerary. Our model reassigns disrupted passengers to new itineraries assuming homogeneous passengers. In practice a more granular version of this is employed that distinguishes each individual passenger based on certain attributes like fare class or frequent flier status. Our framework chooses the specific itineraries that are to be used determining the flow of passengers to be assigned to each itinerary only and not which specific passengers are to be assigned (this could be done post-processing).

2.4.1. Sets.

OD: set of disrupted passengers classified by an origin-destination (OD) pair;

Γ: set of all passenger itineraries;

$\Gamma_i \subseteq \Gamma$: set of all itineraries eligible to assign passenger $i \in \text{OD}$;

$\Gamma_i^{\text{multiflt}} \subseteq \Gamma_i$: set of multiflight itineraries available to passenger $i \in \text{OD}$.

2.4.2. Decision Variables.

$z_{i,\gamma}$: number of passengers from $i \in \text{OD}$ to assign to itinerary $\gamma \in \Gamma_i$;

s_i: number of passengers from $i \in \text{OD}$ that are not assigned to an itinerary;

$\delta_{i,\gamma}$: hourly delay if passenger $i \in \text{OD}$ is assigned to itinerary $\gamma \in \Gamma_i$.

2.4.3. Data.

$c_{i,\gamma}^{\text{delay}}$: hourly cost of passenger delay associated with assigning $i \in \text{OD}$ to itinerary $\gamma \in \Gamma_i$;

c_i^{unassign}: cost of being unable to assign a passenger to an itinerary;

$\omega_{i,\gamma}$: weight of assigning $i \in \text{OD}$ to itinerary $\gamma \in \Gamma_i$ in the aggregate delay cost;

n_i^{PAX}: number of passengers for $i \in \text{OD}$;

CAP_e: capacity of equipment type $e \in E$;

$\underline{f}(\gamma)$: initial flight in itinerary $\gamma \in \Gamma$;

$\bar{f}(\gamma)$: final flight in itinerary $\gamma \in \Gamma$;

t_f^{arr}: actual time of arrival for flight $f \in F$;

t_f^{dep}: actual time of departure for flight $f \in F$;

t_i^{STD}: scheduled time of departure at origin for $i \in \text{OD}$;

t_i^{STA}: scheduled time of arrival at destination for $i \in \text{OD}$;

$t_{\min}^{\text{connect}}$: minimum passenger connection time for multiflight itineraries.

2.4.4. Itinerary Recovery Model. As previously discussed, all eligible itineraries are initially constructed given the original flight schedule. Several of the itineraries will be ineligible with different solutions provided by the SRM. The itinerary recovery model (IRM) seeks to output the set of eligible itineraries for each OD such that the *aggregate delay costs* are minimized subject to ensuring a feasible set of passenger itinerary assignments. The IRM is formulated as follows.

$$\min \sum_{i\in \mathrm{OD}}\sum_{\gamma\in\Gamma_i} c^{\mathrm{delay}}_{i,\gamma}\omega_{i,\gamma}\delta_{i,\gamma}+\sum_{i\in OD} c^{\mathrm{unassign}}_i s_i \tag{16}$$

$$\text{s.t.} \sum_{i\in OD}\sum_{\substack{\gamma\in\Gamma_i:\\ \gamma\ni f}} z_{i,\gamma}\le \sum_{e\in E}\sum_{\substack{s\in S:\\ s\ni f}} x_{e,s}\mathrm{CAP}_e \quad \forall f\in F, \tag{17}$$

$$\sum_{\gamma\in\Gamma_i} z_{i,\gamma}+s_i=n^{\mathrm{OD}}_i \quad \forall i\in \mathrm{OD}, \tag{18}$$

$$\delta_{i,\gamma}\ge \sum_{e\in E}\sum_{\substack{s\in S:\\ s\ni \bar f(\gamma)}} x_{e,s}t^{\mathrm{arr}}_{\bar f(\gamma)}-t^{\mathrm{STA}}_i \quad \forall i\in \mathrm{OD}, \forall \gamma\in\Gamma_i, \tag{19}$$

$$z_{i,\gamma}\le n_i(1-\kappa_f) \quad \forall f\in F, \forall (i,\gamma)\in \mathrm{OD}\times\Gamma_i\colon\ \gamma\ni f, \tag{20}$$

$$\sum_{e\in E}\sum_{\substack{s\in S:\\ s\ni \underline{f}(\gamma)}} t^{\mathrm{dep}}_{\underline{f}(\gamma)}x_{e,s}\ge t^{\mathrm{STD}}_i - M_{i,\gamma}(1-v_{i,\gamma}) \quad \forall i\in \mathrm{OD}, \forall\gamma\in\Gamma_i, \tag{21}$$

$$z_{i,\gamma}\le M_{i,\gamma}v_{i,\gamma} \quad \forall i\in \mathrm{OD}, \forall \gamma\in\Gamma_i, \tag{22}$$

$$\sum_{e\in E}\sum_{\substack{s\in S:\\ s\ni f_j}} t^{\mathrm{dep}}_{f_j}x_{e,s}-\sum_{e\in E}\sum_{\substack{s\in S:\\ s\ni f_i}} t^{\mathrm{arr}}_{f_i}x_{e,s} \ge t^{\mathrm{connect}}_{\min}-M'_{i,\gamma}(1-w_{i,\gamma}) \quad \forall i\in \mathrm{OD}, \forall (f_i,f_j)\in \Gamma^{\mathrm{multiflt}}_i, \tag{23}$$

$$z_{i,\gamma}\le M'_{i,\gamma}w_{i,\gamma} \quad \forall i\in \mathrm{OD}, \forall\gamma\in\Gamma^{\mathrm{multiflt}}_i, \tag{24}$$

$$(z_{i,\gamma},\delta_{i,\gamma},v_{i,\gamma},w_{i,\gamma})\in \mathbb{Z}\times\mathbb{R}\times\{0,1\}\times\{0,1\}, \quad \forall i\in \mathrm{OD}, \forall\gamma\in\Gamma_i,$$

$$s_i\in\mathbb{Z} \quad \forall i\in \mathrm{OD}.$$

The objective (16) seeks to minimize the total weighted nominal delay cost of all itineraries and unassigned passengers. The weights can be either unit-valued or reflect the share of OD passengers present in the disruption. Constraints (17) prohibits the spilling of passengers. For each OD, (18) either assigns passengers to a feasible itinerary or strands them with no itinerary being assigned. If a passenger cannot be assigned to an itinerary, he or she may overnight at a connection point, be placed on another airline, or have the itinerary delayed outside of $\mathcal{T}$. Constraints (19) tracks the delay of each passenger-itinerary pair where the itinerary delay is the difference between the actual arrival time of the last flight in the itinerary and the scheduled time of arrival to the passenger's destination. Constraint (20) ensures no passenger is assigned to an itinerary that contains a cancelled flight (where n_i is an upper bound for $z_{i,\gamma}$). Recall that all eligible itineraries are overbuilt a priori in which some itineraries are infeasible with respect to the original schedule but may become eligible through delays. The next four constraints are logical constraints that ensure only legal itineraries are considered given the solution from the SRM. Inequalities (21) and (22) prohibit assigning any itineraries to passengers in which the initial flight in the itinerary departs prior to the passenger ready time. For all $i\in \mathrm{OD}$ and $\gamma\in\Gamma_i$, $M_{i,\gamma}=t^{\mathrm{STD}}_i$ is chosen as a valid upper bound. Given the solution from the SRM, passenger connection times are observed. If the connection time does not exceed the minimum necessary connection time $t^{\mathrm{connect}}_{\min}$, then no passengers can be assigned to that itinerary. This is reflected in (23) and (24) where $M'_{i,\gamma}>0$ is appropriately chosen (for example, maximum possible connection time).

2.4.5. Passenger Reaccommodation Model. Once the set of flight strings has been found that induces the minimal aggregate passenger delay, the passenger reaccommodation model (PRM) is solved. The PRM allocates passengers to the given set of itineraries to minimize the total *assignment cost*.

For all $i\in \mathrm{OD}$ let Γ^*_i denote the set of eligible itineraries for the given OD induced by the optimal SRM solution. The PRM is formulated as

$$\min \sum_{i\in \mathrm{OD}}\sum_{\gamma\in\Gamma^*_i} c^{\mathrm{delay}}_{i,\gamma}\delta^*_{i,\gamma}z_{i,\gamma}+\sum_{i\in OD} c^{\mathrm{unassign}}_i s_i \tag{25}$$

$$\text{s.t.} \sum_{i\in \mathrm{OD}}\sum_{\substack{\gamma\in\Gamma^*_i:\\ \gamma\ni f}} z_{i,\gamma}\le \sum_{e\in E}\sum_{\substack{s\in S:\\ s\ni f}} x_{e,s}\mathrm{CAP}_e \quad \forall f\in F, \tag{26}$$

$$\sum_{\gamma\in\Gamma^*_i} z_{i,\gamma}=n^{\mathrm{OD}}_i \quad \forall i\in \mathrm{OD}, \tag{27}$$

$$z_{i,\gamma}\in\mathbb{Z} \quad \forall i\in \mathrm{OD}, \forall\gamma\in\Gamma^*_i,$$

$$s_i\in\mathbb{Z} \quad \forall i\in \mathrm{OD}.$$

Note the summations in (26) and (27) differ from (17) and (18) in that the former are taken over the index sets Γ^*_i. Although the objective function of the IRM does not depend on $z_{i,\gamma}$, constraints (17) and (18) are included in the IRM to ensure a feasible solution in the PRM. Moreover, the cost coefficients $c^{\mathrm{delay}}_{i,\gamma}$ are chosen to be identical for both the IRM and PRM to measure the cost associated with passenger delay.

The two-stage approach to passenger recovery can be combined into a single step in which reaccommodation is done explicitly. However, our approach is advantageous in two ways. Considerable computational effort is required to model each passenger individually; the number of cut coefficients generated by the Benders cut has introduced a vast complexity to the master problem, which is solved as a mixed-integer programming problem. Secondly, our approach only requires a single call to the itinerary generator a priori as opposed to building new itineraries every time the master problem is solved.

3. Limiting the Scope of Recovery

The size and complexity of the integrated recovery problem outlined above most likely precludes the delivery of a globally optimal solution. In order to tractably solve the problem for reasonably large scenarios, careful consideration must be placed on how to limit the size or scope of the problem.

A flight is said to be *disrupted* if one of its resources precludes the flight from operating as scheduled. Such resources include the arrival or departure airport, aircraft, or assigned crew members. Flight disruptions may be exogenous or endogenous. An example of an exogenous disruption is the closure of an airport for a specific period of time, in which all flight activity to or from the airport within that time interval must be altered. However, system-wide disruptions can be mitigated by *endogenous* flight disruptions. An example of an endogenous flight disruption is shown in Figure 1 on a simple flight network consisting of three flights: 101 from MIA to ATL, 102 from ATL to ORD, and 114 from CLT to ATL. The thick black segment at ATL represents a closure that forces the (exogenous) disruption to flight 101. Although flight 102 is unaffected by the disruption, it may be advantageous to (endogenously) delay the flight in order to accommodate connecting passengers. Of course this illustration is simplistic, but it shows the combinatorial nature of the problem.

Flights that are candidates for disruptions are said to be *disruptable*. For example, consider flight 114 from Figure 1 that is unaffected by the disruption directly. It would be plausible to not consider that flight as a candidate for disruption. Although simple to identify

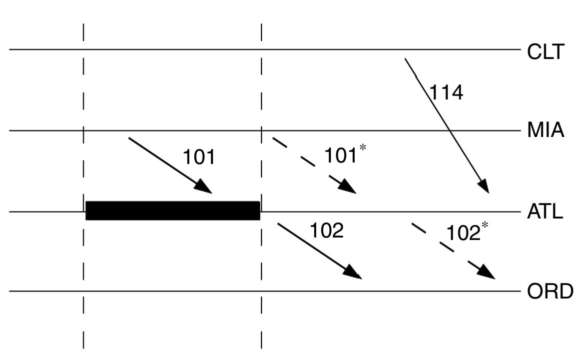

Figure 1 **Exogenous vs. Endogenous Flight Disruptions**

on a three-flight example, the process of identifying which subset of flights to be considered disruptable poses a considerable challenge.

We now discuss the procedure by which we identify all disruptable flights. Initially the disruptable flight set includes those flights that are directly affected by a resource at the airport. The set is then expanded to consider aircraft, crews, and passengers.

3.1. Limiting Flights

The disruptable flight set is instantiated with all exogenous flight disruptions that contain a resource that forces a delay or cancellation.

Flights from Disrupted Routings. A *disruptable aircraft* exists if its scheduled routing contains a disruptable flight. Suppose k_n flights are scheduled for disruptable aircraft n within the time window $\mathcal{T}$ denoted by $f_1, f_2, \ldots, f_{k_n}$. Let f_i denote the earliest flight from the disruptable routing present in the disruptable flight set. Denote $F_n \equiv \{f_i, f_{i+1}, \ldots, f_{k_n}\}$ as all subsequent flights within $\mathcal{T}$ that were scheduled to be operated by aircraft n. Because of delay propagation, a disruption to flight f_i may cause disruption to the subsequent flights from F_n. Thus the disruptable flight set is appended with *all* flights from F_n. Repeating this procedure for all disruptable routings gives the updated disruptable flight set.

Flights from Disrupted Crew. Similar to that of aircraft, a *disruptable crew* exists if a crew is scheduled to fly a disruptable flight within its pairing. The disruptable flight set is appended in a similar fashion to that of aircraft. A list of flights is extracted that each crew member is scheduled to fly in the disruption period. If a disruptable flight is present, then that flight and all subsequent flights within the scheduled pairing within the disruption period are added to the flight set.

The new flights that have been added from the crew schedules might be operated by aircraft not previously identified as disruptable. In this case, the new aircraft are appended to the disruptable set of aircraft.

Flights from Tight Passenger Connections. We take a passenger-centric approach to integrated recovery, and thus minimizing passenger delay is central to our study. We further modify the disruptable flight set by considering additional candidate flights that are identified for abating passenger delay through preprocessing. Consider a passenger originating in MIA whose destination is ORD, seen in Figure 2. Note that the connection between flights 101 and 102 appears to be tight. Even a moderate disruption in flight 101 is likely to break the connection for such passengers. Additional flight candidates are introduced for such tight connections through a simple rule. If a nondisruptable flight has the same origin and destination

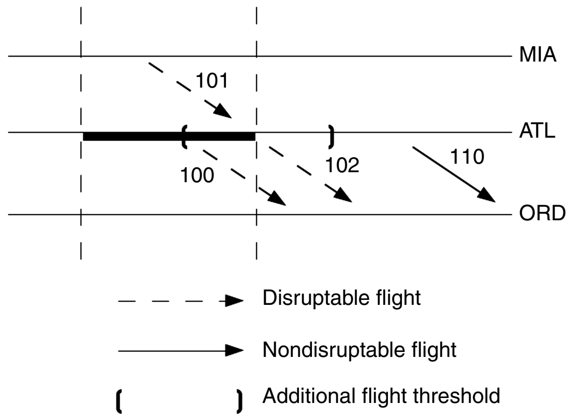

Figure 2 Identifying Passenger-Friendly Flights

from a flight contained in a tight connecting itinerary, then that flight is introduced as disruptable if the departure times are within some tolerance threshold specified by the airline. Figure 2 illustrates this concept of augmenting the disruptable flight set to mitigate passenger delay. There are two other nondisruptable flights from ATL to ORD. Flight 100 departs from ATL relatively near that of flight 102 and is added to the disruptable flight set assuming the difference is within the threshold. Naturally all passengers on flight 100 are then considered in our model because the flight becomes disruptable. If the departure of flight 110 is too late (i.e., outside the threshold), it remains nondisruptable.

These new flights will have new aircraft and new crew members associated with them. As was done with adding new flights from crew schedules, we consider the single-flight entities only and ensure both the aircraft and crew members are eligible to operate the next flight in their respective schedules.

3.2. Retiming Flights

Initial work on airline recovery modeled flight delays by making copies of each flight arc that departed at uniform intervals (see Clarke 1998a; Gao 2007). Although the uniform flight copy approach is simple and intuitive, generating strings over copies of flights becomes extraordinarily large and complex. We instead model delays through an event-driven approach. The idea is that events like arrivals and times associated with constraints from the SRM give more relevant delay decisions than arbitrary departure times from uniform flight copies.

Given a maximum allowable delay period $d_{\max}$, a timeline is created for each flight from 0 to $d_{\max}$ representing the given flight delay. Note that in the SRM some constraints are a function of time (see, for example, constraints (4) through (6)). Formally these are referred to as *time-dependent constraints*. Table 1 gives an example of a set of time-dependent constraints present in the flight network from Figure 2.

Table 1 An Example of Time-Dependent Constraints

Event	Time	Station	Constraint	Directly affected by disruption?
1	0930–1030	ATL	Flow rate reduction	Yes
2	0930–1000	MIA	Gate restriction	No
3	1130–1200	ATL	Slot restriction	No
4	1200–1245	MIA	Gate restriction	No

The flight departure interval is partitioned into $k \geq 1$ disjoint subintervals from the set of time-dependent constraints that give a maximum of $k+1$ departure options. If a flight f is present in any of the time-dependent constraints, then a new subinterval is created, representing a new candidate departure time. Each string must then have no more than one departure from each subinterval. Strings are generated through the *augmented flight network*, defined to be the original flight network whose number of copies (i.e., delay options) corresponds to the number of subintervals from the delay interval.

Figure 3 shows a simple two-flight example of how delay options are generated from these events using a maximum allowable delay ($d_{\max}$) of two hours. The shaded regions in Figure 3(a) represent the time-dependent constraints as given in Table 1. Figure 3(b) shows how the flight network is augmented to accommodate different departure times. Both flights are partitioned into three subintervals giving a maximum of four departure options for each flight.

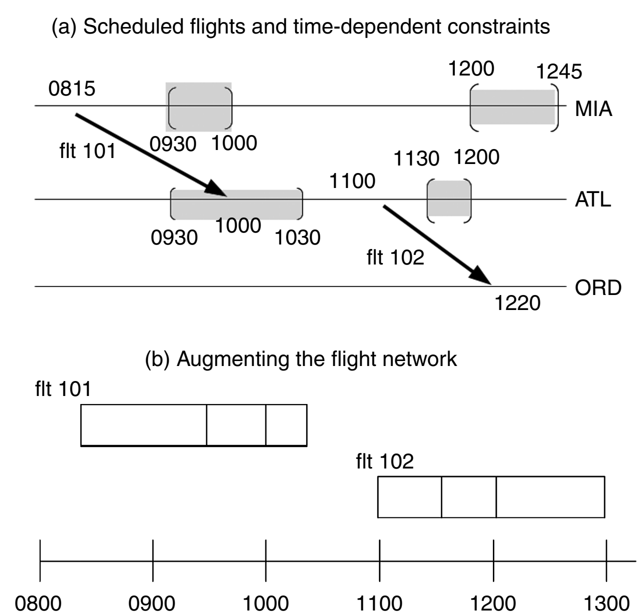

Figure 3 Modeling Event-Driven Flight Delays

The idea of event-driven delays is that the strings present in the augmented flight network are likely to dominate most strings created from uniform flight copies while generating fewer flight strings. From Figure 3 there are a maximum of 4^2 possible strings from this approach. If uniform flight copies were instead employed at a coarse discretization of 15 minutes, eight delay options would arise in addition to the original flight departure time. Thus, nine copies of the same flight are represented for two flights, giving a maximum of 9^2 possible strings for just this trivial two-flight illustration. Another problem with uniform flight copies is that several strings are likely to be present in the same set of time-dependent constraints and therefore exhibit duplicate columns in the SRM formulation.

4. Solution Methodology

Even by limiting the scope of the problem to make it computationally tractable, the problem is likely too large and complex to return a globally optimal solution for most reasonable disruption scenarios. 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. There are a number of problems associated with this scheme, however tractable it seems. 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 although 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.

Our approach is to return a solution that is globally optimal with respect to aggregate passenger delay, meaning passenger assignments are globally optimal over all itineraries *and* all flight strings. We emphasize that optimality is in accordance to our model over the reduced problem whose scope has been limited as discussed in the preceding section. Although 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 can be so large that optimizing over all pairings *and* strings is unlikely to solve in an efficient manner. Two tactics are employed to ameliorate the large cost associated with crew recovery:

1. We do not require the delivered solution to be globally optimal over all strings and pairings. New pairings are priced out until the master solution $(x^*_{e,s}, \kappa^*_f)$ is globally optimal for the IRM and feasible for the ARM and CRM. When this termination criterion is reached, no further pairings are priced out (see Figure 4). Thus our approach is considered to be passenger-friendly with crew considerations.
2. Multiple cockpit crew members are required for each flight, usually two including a captain and a first officer. Even though the crew members may have different pairings, we assume the pair of crew members assigned at the beginning of the time window stay fixed through the time window. We solve the CRM only for the captain and check the legality of the first officer in the post-processing stage. If the assigned pairing violates some legality restriction, a swap is conducted or a reserve crew is assigned if possible.

Other than being computationally tractable for a single-day horizon, 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 reported about excessive passenger delays inducing a "passenger revolt" and a number of variants for a passenger bill of rights have been proposed in Congress. Effective April 2010 the U.S. Department of Transportation 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).

4.1. 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. The master problem is the SRM with linking variables $\{x_{e,s}\}$, $\{\kappa_f\}$ that are passed into the subsequent subproblems: ARM, CRM, and IRM.

Although the three subproblems are independent of each other, they are solved sequentially. First, the SRM and IRM iterate until the aggregate passenger delay cost is minimal. The ARM is then solved. If the ARM is infeasible, a Benders feasibility cut is added to the SRM. Otherwise, the CRM is then solved. Again, a feasibility cut is added if the CRM is infeasible. Otherwise, a tentative solution is found. If the optimality gap between the current CRM iterate is within some tolerance level specified by the user, a solution to the iterative scheme is given. Otherwise, new

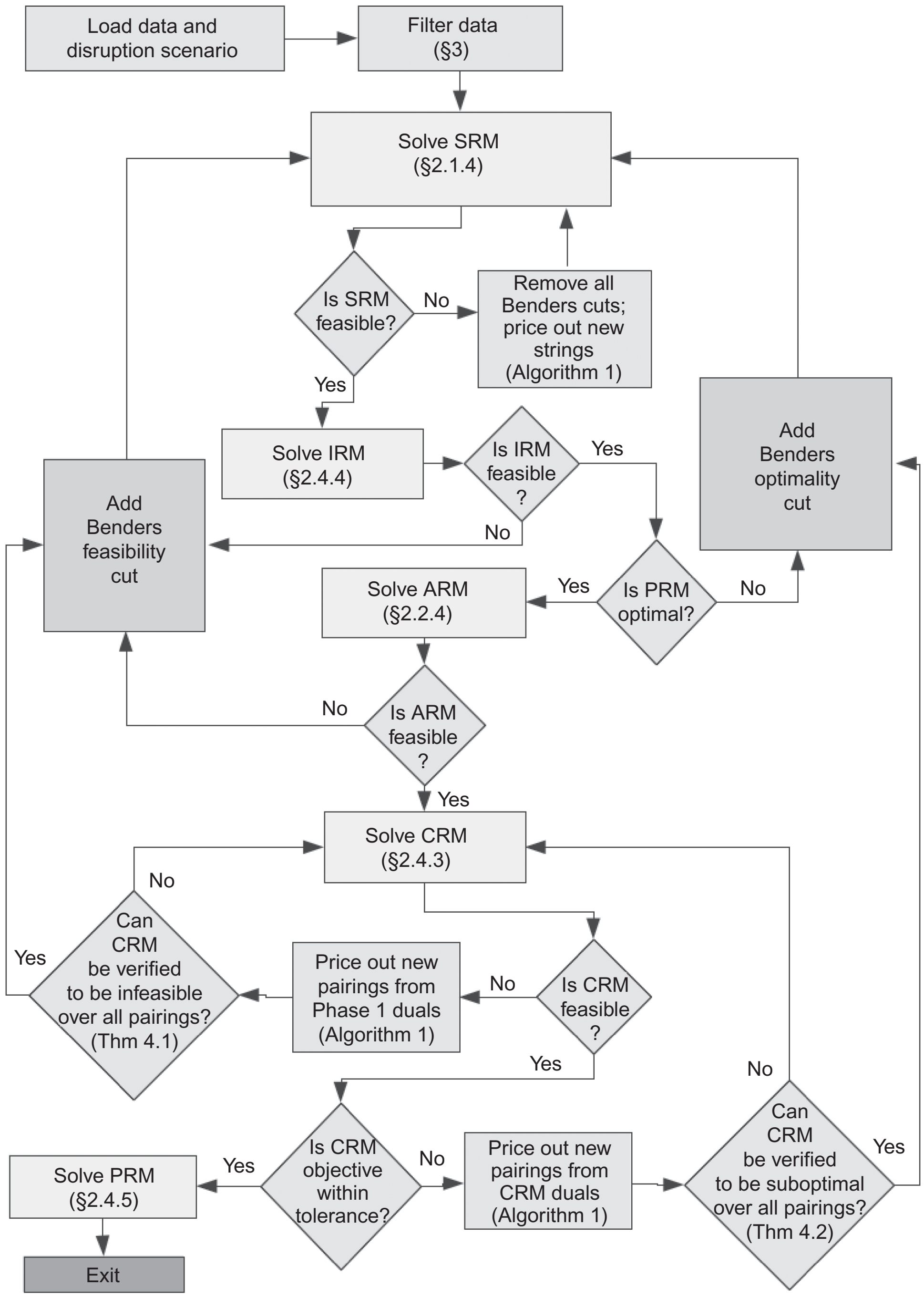

Figure 4 **AIR Optimization Module**

columns are generated and returned to the CRM, or a Benders optimality cut is returned to the SRM. The problem structure is amenable to parallelization, but we employ the sequential implementation.

There are five classes of Benders cuts that are passed into the master problem. Only the relaxation of each of the three subproblems is 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 until feasibility is attained. Obtaining integer solutions for the three subproblems is further discussed in §4.4.

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

$$\sum_{e\in E}\sum_{s\in S}\pi_{e,s}^{\mathrm{ARM}}x_{e,s}\le \pi_0^{\mathrm{ARM}}, \tag{28}$$

$$\sum_{f\in F}(1-\kappa_f)\pi_f^{\mathrm{CRM}}+\sum_{k\in K}\rho_k^{\mathrm{CRM}}\le 0, \tag{29}$$

$$\sum_{f\in F}(1-\kappa_f)\pi_f^{\text{CRM}}+\sum_{k\in K}\rho_k^{\text{CRM}}\le \eta^{\text{CRM}}, \tag{30}$$

$$\begin{aligned}&\sum_{e\in E}\sum_{f\in F}\sum_{s\ni f}\pi_f^{\text{IRM}}\text{CAP}_e x_{e,s}+\sum_{i\in PAX}\sum_{\gamma\in\Gamma_i}\sum_{e\in E}\sum_{s\ni\bar{f}(\gamma)}\sigma_{i,\gamma}^{\text{IRM}}t_{\bar{f}(\gamma)}^{\text{arr}}x_{e,s}\\&\quad+\sum_{i\in\text{OD}}\sum_{\gamma\in\Gamma_i}\sum_{f\in\gamma}\nu_f^{\text{IRM}}n_i(1-\kappa_f)\\&\quad+\sum_{i\in\text{OD}}\sum_{\gamma\in\Gamma_i}\theta_{i,\gamma}^{\text{IRM}}\Bigg(\sum_{e\in E}\sum_{\substack{s\in S:\\ s\ni \underline{f}(\gamma)}}t_{\underline{f}(\gamma)}^{\text{dep}}x_{e,s}\Bigg)\\&\quad+\sum_{i\in PAX}\sum_{\gamma\in\Gamma_i^{\text{multiflt}}}\tau_{i,\gamma}^{\text{IRM}}\Bigg(\sum_{e\in E}\sum_{s\ni f_j}t_{f_j}^{\text{dep}}x_{e,s}-\sum_{e\in E}\sum_{s\ni f_i}t_{f_i}^{\text{arr}}x_{e,s}\Bigg)\\&\le\pi_0^{\text{IRM}},\end{aligned}\tag{31}$$

$$\begin{aligned}&\sum_{e\in E}\sum_{f\in F}\sum_{s\ni f}\pi_f^{\text{IRM}}\text{CAP}_e x_{e,s}+\sum_{i\in PAX}\sum_{\gamma\in\Gamma_i}\sum_{e\in E}\sum_{s\ni\bar{f}(\gamma)}\sigma_{i,\gamma}^{\text{IRM}}t_{\bar{f}(\gamma)}^{\text{arr}}x_{e,s}\\&\quad+\sum_{i\in\text{OD}}\sum_{\gamma\in\Gamma_i}\sum_{f\in\gamma}\nu_f^{\text{IRM}}n_i(1-\kappa_f)\\&\quad+\sum_{i\in\text{OD}}\sum_{\gamma\in\Gamma_i}\theta_{i,\gamma}^{\text{IRM}}\Bigg(\sum_{e\in E}\sum_{\substack{s\in S:\\ s\ni \underline{f}(\gamma)}}t_{\underline{f}(\gamma)}^{\text{dep}}x_{e,s}\Bigg)\\&\quad+\sum_{i\in PAX}\sum_{\gamma\in\Gamma_i^{\text{multiflt}}}\tau_{i,\gamma}^{\text{IRM}}\Bigg(\sum_{e\in E}\sum_{s\ni f_j}t_{f_j}^{\text{dep}}x_{e,s}-\sum_{e\in E}\sum_{s\ni f_i}t_{f_i}^{\text{arr}}x_{e,s}\Bigg)\\&\le\eta^{\text{IRM}}+\pi_0^{\text{IRM}}.\end{aligned}\tag{32}$$

The superscripts in (28)–(32) denote the given subproblem, π_0^{ARM} and π_0^{IRM} are constants that depends on the dual variables from the right-hand side of constraints that do not depend on master variables from the ARM and IRM, respectively. In addition, η^{CRM} and η^{IRM} are new decision variables in the master problem corresponding to the optimal objectives in the CRM and IRM, respectively. The cuts are ARM feasibility, CRM feasibility, CRM optimality, IRM feasibility, and IRM optimality, respectively. We model the ARM as a feasibility problem so that ARM optimality cuts are unnecessary.

4.2. Column Generation

Given the large number of flight strings and repaired crew pairings, only a subset of columns is generated through each of these problems. Multiple columns are generated through a residual network, which is built from the flight network for flight strings and the crew duty 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 multiple columns at once, a tolerance parameter $\epsilon>0$ is defined and all columns whose path p prices out less than ϵ are then added. This is sometimes known as path generation through an ϵ-residual network (see Ahuja, Magnanti, and Orlin 1993 for a general description; Shaw 2003 gives an example pertinent to a traditional crew pairing problem). A summary of this method is shown in Algorithm 1.

Algorithm 1 (Path Generation Through ϵ-Residual Network)

Given: Set of resources R, general resource network $G=(V,A)$, dual information $\pi_v\ \forall v\in V$, and tolerance parameter $\epsilon>0$
Initialize: Newly generated variables $X=\varnothing$
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: $A\to A\backslash\{a\}$
    end if
  end for
  Let $X^i=\{\bigcup p:\ p$ is an $s-t$ path s.t. $\sum_{a\in p}\bar{c}_a<\epsilon\}$
  $X\to X^i$
end for
return new columns X

4.3. Simultaneous Row and Column Generation

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

We discuss for two cases how these methods are used together. The first deals with linking variables that are done in a brute force way. The second 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.

4.3.1. Flight Strings. A general Benders cut is valid over all *generated* flight strings $S'\subseteq S$. As new strings are added, the Benders cut may be invalid for some $s\in S\backslash S'$. Although to the best of our knowledge, there does not exist a way to overcome this barrier, we simply remove the Benders cuts any time new strings are added (a related problem introduced by Van Roy 1983 is that of cross decomposition). 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.

4.3.2. Repaired Crew Pairings. A Benders cut is valid over all generated linking variables as well as those local to the subproblem. However, if columns are being added to the subproblem, new columns may violate the previous cuts, rendering them as invalid to all variables. Therefore any cut initially generated becomes a *candidate cut* because it is feasible only over all *generated* variables. In the context of the CRM, we denote $P' \subseteq P$ to be the set of all generated pairings. Below we handle the simultaneity of these two procedures by first obtaining a certificate of infeasibility that proves the CRM is infeasible over all P for a given master solution. If the candidate cut meets this criterion, then the cut is added to the master. Otherwise, it is discarded. In both cases, new columns are being generated.

As discussed in §4.2, columns (repaired crew pairings) are generated through a resource network referred to as the *master crew duty network* $G = (\mathscr{D}, \mathscr{A})$, where $\mathscr{D}$ denotes the set of all duties that have been enumerated a priori, and $\mathscr{A}$ is the set of arcs that can legally connect consecutive 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 $G^k = (D^k, A^k)$ 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 that begins at the station where the crew is at the beginning of the disruption and ends at the station the crew is required to be at by the end of the time window such that all legality requirements are met.

Recall the CRM formulation given in §2.3. Let π_f and ρ_k denote dual variables for constraints (14) and (15), respectively. The reduced cost of a pairing p for crew k is given by $\bar{c}_{k,p} = c^{\text{assign}}_{k,p} - \sum_{f\in p} \pi_f - \rho_k$. Let P_k denote the set of all pairings for some crew $k \in K$ where a subset $P'_k \subseteq P'$ has been generated. The following results give a certificate for which the CRM feasibility cuts remain valid over all P.

Theorem 4.1 (Extending CRM Feasibility Cuts Over New Pairings). *Suppose the CRM is infeasible over a subset of pairings $P' \subset P$. Let $\{\pi_f\}, \{\rho_k\}$ denote the duals corresponding to the Phase I LP-relaxation of the CRM. If*

$$\sum_{f\in F}(1-\kappa_f)\pi_f + \sum_{k\in K}\rho_k > \max_{k\in K}\left\{\max_{p\in P_k\setminus P'_k}\left(\sum_{f\in p}\pi_f + \rho_k\right)\right\},$$

then the CRM is infeasible over all P, and the candidate Benders feasibility cut is valid over all strings and pairings.

Proof. As a corollary to Farkas' Lemma the CRM is infeasible over all P if and only if there exists $(\alpha, \beta, \Delta) \in \mathbb{R}^{|F|} \times \mathbb{R}^{|K|} \times \mathbb{R}$ such that

$$\begin{gathered}\sum_{f\in F}(1-\kappa_f)\alpha_f + \sum_{k\in K}\beta_k > \Delta,\\ \sum_{f\in p_k}\alpha_f + \beta_k \le \Delta \quad \forall k\in K, \forall p_k \in P_k,\\ -\alpha_f \le \Delta \quad \forall f\in F,\\ \beta_k \le \Delta \quad \forall k \in K.\end{gathered} \tag{$\clubsuit$}$$

If the CRM is infeasible, let π_f and ρ_k denote the duals associated with the Phase I LP of the CRM. For all $k \in K$ let P^{new}_k denote the newly generated pairings on G^k that have been generated. Moreover, for all $k \in K$ let $\Delta_k = \max\{\sum_{f\in p}\pi_f + \rho_k \colon p \in P_k\}$ denote the value of the pricing problem (recall that in the Phase I problem $c^{\text{assign}}_{k,p} = 0\ \forall k \in K, p \in P'_k$). By letting $0 < \Delta \equiv \min_{k\in K}\{\Delta_k\}$, the second condition of ($\clubsuit$) holds by construction, and the first condition holds by assumption. The latter two conditions trivially hold by Phase I duality and because $\Delta > 0$. Hence $(\pi, \rho, \Delta) \in (\clubsuit)$, and the CRM is infeasible over all P. □

The analogue for the case of Benders optimality cuts is seen in the following result.

Theorem 4.2 (Extending CRM Optimality Cuts Over New Pairings). *Suppose the CRM is suboptimal over a subset of pairings $P' \subset P$. Let $\{\pi_f\}, \{\rho_k\}$ denote the CRM duals, and let η^* denote the continuous master variable corresponding to the optimal objective of the CRM. If*

$$\sum_{f\in F}(1-\kappa_f)\pi_f + \sum_{k\in K}\rho_k + \min_{k\in K}\left\{\min_{p\in P_k\setminus P'_k}\bar{c}_{k,p}\right\} > \eta^*,$$

then the CRM is suboptimal over all P, and the candidate Benders optimality cut is valid over all strings and pairings.

Proof. Consider two classes of CRM constraints (14) and (15) together with the single equality

$$\sum_{k\in K}\sum_{p\in P_k} c^{\text{assign}}_{k,p} y_{k,p} + \sum_{f\in F} d^{\text{pairing}}_f s_f + \sum_{k\in K} d^{\text{base}}_k \nu_k = \eta^*.$$

The preceding has no solution over all P if and only if there exists $(\alpha, \beta, \gamma, \Delta)$ to the following system:

$$\begin{gathered}\sum_{f\in F}(1-\kappa_f)\alpha_f + \sum_{k\in K}\beta_k + \gamma\eta^* > \Delta,\\ \sum_{f\in p_k}\alpha_f + \beta_k + \gamma c_{k,p} \le \Delta \quad \forall k\in K, \forall p_k\in P_k,\\ -\alpha_f + \gamma d^{\text{pairing}}_f \le \Delta \quad \forall f\in F,\\ \beta_k + \gamma d^{\text{home}}_f \le \Delta \quad \forall k\in K.\end{gathered} \tag{$\clubsuit\clubsuit$}$$

If $\gamma = -1$, the latter two conditions are satisfied by dual feasibility, and the second amounts to $\bar{c}_{k,p} \leq \Delta$. For all k let $\Delta_k = \min\{-\bar{c}_{k,p}\}$ denote the minimum reduced cost generated over G^k. As before, let $\Delta = \min_{k\in K}\{\Delta_k\}$. It is assumed that $\Delta > 0$ because a new column prices out. Therefore the latter two conditions hold by duality, the second holds by construction, and the first by assumption. Thus $(\pi, \rho, \Delta, -1) \in$ (♣♣), and the CRM is suboptimal over all strings and pairings. □

Algorithm 2 summarizes the implementation of the two preceding theorems in the context of our solution strategy for the case of an infeasible CRM. A similar algorithm is implemented for when the CRM is feasible, but we do not include the details herein.

Algorithm 2 (Handling Column Generation and Constraint Generation Together in CRM [Infeasibility])

Solve LP-Relaxation for CRM
Initialize *validCut* = false
if CRM is infeasible over P' **then**
  Extract dual extreme ray (π_f, ρ_k) and
    Phase-I duals (π_f^I, ρ_k^I).
  Let $\sum_{f\in F}(1-\kappa_f)\pi_f + \sum_{k\in K}\rho_k \leq 0$ denote the
    candidate Benders feasibility cut
  for all crew $k \in K$ **do**
    Construct subgraph $\tilde{G}^k(D, A)$ of crew
      duty network G^k from (π_f^I, ρ_k^I)
    Generate new columns P_k^{new} over the
      ϵ-residual network over $\tilde{G}^k(D, A)$
    if a new column exhibits a negative reduced
      cost **then**
      Set $\Delta_k = \max_{p\in P_k} \sum_{f\in p} \pi_f^I + \rho_k^I$
    else
      Set $\Delta_k = 0$
    end if
  end for
  Set $\Delta = \max_{k\in K} \Delta_k$
  if $\sum_{f\in F}(1-\kappa_f)\pi_f^I + \sum_{k\in K}\rho_k^I > \Delta$ **then**
    set *validCut* = true
  end if
  if *validCut* = true **then**
    add candidate Benders cut to master problem
  else
    update columns $P' \to P' \bigcup_{k\in K} P_k^{\text{new}}$, and
      re-solve CRM relaxation
  end if
end if

4.4. Integrality

The iterative Benders scheme 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 a fractional solution is present. If no feasible integer solution is found by branching, the node returned by the algorithm is then rejected and the procedure is to continue until an integer solution is delivered. We discuss how integrality is obtained in each of the three subproblems.

SRM Integrality. The SRM module is solved to integrality using branch-and-cut. One particularly useful strategy is to branch on *follow-ons*. This concept was introduced by Falkner and Ryan (1987). A *follow-on* is a pair of flights that are contained in the same fractional-valued string. The branching dichotomy either forces or forbids the given follow-on. Anbil, Tanga, and Johnson (1992) and Lettovsky, Johnson, and Nemhauser (2000) show follow-on branching to be successful in driving integrality of crew recovery models in particular. We find this branching strategy to also be very effective in the SRM.

ARM Integrality. One of the advantages of the flight string models is it makes the routing problem considerably easier to solve as shown in Theorem 4.3.

Theorem 4.3 (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 assigned strings from the master problem $\{x_{e,s}^*\}$. This class of problems is well known to be integral (Nemhauser and Wolsey 1999). □

CRM Integrality. Solving the LP-relaxation of the CRM induces integer solutions in many scenarios. However, the polytope is itself not integral. Similar to the case of driving SRM integrality, we employ branching on follow-ons with respect to fractional crew pairings.

IRM Integrality. Solving the PRM could be done through a multicommodity 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.

4.5. Overview

Figure 4 summarizes our approach to solving the airline integrated recovery (AIR) model.

5. Computational Results

Our model is tested using 2007 data from a hub-and-spoke regional airline based in the United States with approximately 800 daily flights and two fleet types. The main disruption of interest is a flow rate reduction into and out of 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. As shown in the table, the SRM cost

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^{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^{pairing}	Cost of deadheading on flight f within a pairing	\$1,000
d_k^{base}	Cost of crew k deadheading to crew base	\$2,000
$c_{i,\gamma}^{\text{delay}}$	Cost in passenger goodwill per hour of delay	\$38
c_i^{unassign}	Cost of unassigned itinerary for passenger $i \in OD$	\$2,500
$\omega_{i,\gamma}$	Weight of passenger itinerary cost in IRM	$\frac{n_i^{\text{PAX}}}{\sum_i n_i^{\text{PAX}}}\ \forall \gamma \in \Gamma_i$

objective is only to minimize the cost associated with canceling flights while 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 in the ARM. The cost of \$38 per hour of passenger delay is given by Ball et al. (2010).

Note that we consider a zero objective on individual crew pairing assignments. This is because the crew recovery problem is quite different from the well-known crew pairing problem, where the objective is to minimize the sum of crew pairing assignments known as *pay-and-credit*, a complex objective that factors in the total time the crew is away from base, flying hours, and number of duties in a pairing. Deadhead costs are influential to the cost of the entire pairing, and therefore by minimizing deadhead costs during the broken part of a crew pairing, pay-and-credit can be reduced.

The data represented in Table 2 come from a priori knowledge about the given network and airline under consideration. Of course, different airlines could incorporate their own set of parameters characterizing their own idiosyncratic values. We emphasize that what is important are not the specific values per se, but rather the methodology that determines the set of rescheduling decisions because different sets of parameters could be used to reflect other carriers.

Our goal is to deliver a solution within 30 minutes as agreed upon by our industry partners. Although this number is likely greater than the allowable time posed by an OCC coordinator, we emphasize the challenges posed by this particular regional carrier are among the most complex and difficult-to-solve class of problems. Moreover, our implementation serves only as a prototype versus production software. A number of ways to expedite our implementation exist, including utilizing parallelization and improved computational infrastructure that is likely to be found at an OCC. We emphasize that our model is scalable. For small disruptions that airlines have to deal with every day, much less time is needed, but the model is able to provide an answer for larger scenarios. Even by sacrificing on optimality, our model is likely able to provide an improvement over incumbent methods that often rely on the manual construction of rescheduling decisions.

Our model has been implemented in C++ using Concert/CPLEX 12.2 on a quad-core computing cluster whose head node is a 2.66 GHz Xeon X5355 processor.

Problem Size and Length of Disruption. Section 3 discussed how the scope of the recovery operation was limited. Figure 5 shows how the number of disruptable flights grows with respect to the duration of closure at the hub beginning at 8:00 A.M. local time. Although a one-hour disruption affects nearly half the flights, every flight is disruptable when the length of the disruption reaches 105 minutes. This is partly because 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 because every tail number has some activity at the hub between 8:00 and 9:15 A.M. local time.

Build vs. 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. 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 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

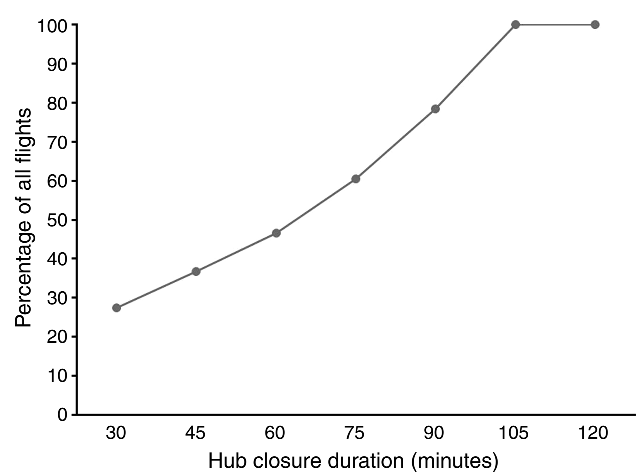

Figure 5 Disruptable Flights and Length of Hub Closure

resources spent constructing the crew duty network and information about the true network. If the time window includes more than one day, the number of connecting duties increases substantially thereby making the CRM even more complex, and the former approach is more plausible. As a first attempt to study the AIR problem, we begin by restricting our analysis to a one-day time window 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. The multiday problem would require a different set of algorithms.

5.1. Disruption Scenarios

We model three classes of disruption scenarios:

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

Each scenario will examine four different disruption events characterized by a disruption time, disruption location, and time window shown in Table 3. Scenario 4 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 5), we consider a maximum hub disruption to be 75 minutes, which our heuristic search procedure includes for every flight after the disruption. The final column represents the maximum delay considered, which has a profound effect on the number of strings being generated. For a two-hour hub disruption, the total number of flight strings (that contain no more than seven flights) increases from under 200,000 using a one-hour maximum delay period to more than 2.6 million using a three-hour maximum delay period. If a set of passenger itineraries is suboptimal after the 30-minute threshold, the best incumbent solution is given and passed to the ARM and CRM subproblems. The algorithm has timed out only for the largest scenarios in our study.

5.2. Integrated vs. Sequential Recovery

We report costs for all subproblems and important metrics that determine quality of solution. We do not report costs for the ARM in the integrated model because it amounts to a feasibility problem and is always feasible in the sequential module. All times are reported in MM:SS format.

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	08:00–09:15	Hub	08:00–23:59	150
4	08:00–09:00	Hub	08:00–23:59	120
	09:00–14:00	Spoke		

Table 4 Sequential Recovery Summary (50% Flow Rate Reduction)

	Event			
	1	2	3	4
Subproblem costs ($)				
SRM	0	0	0	150,000
ARM	0	0	0	INFEAS
CRM	0	0	0	INFEAS
PRM	11,653	28,257	55,665	116,471
Solution metrics				
Mean flt delay	20:05	23:34	42:21	41:36
Cancelled flts (%)	0	0	0	4.6
Delayed flts (%)	12.8	59.4	56.2	52.0
Total deadheads	0	0	0	INFEAS
Mean PAX delay	23:09	24:28	45:39	39:15
Unassigned PAX	0	4	5	31
CPU time	0:58	07:28	17:20	12:02

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.

Disruption Scenario 2: 75% Flow Rate Capacity Reduction. Tables 6 and 7 show the results from reducing capacity by 75%.

Disruption Scenario 3: Hub Closure. Finally we consider a full closure into and out of a set of stations prohibiting all arrivals and departures within the disruption time, which is shown in Tables 8 and 9.

In both environments, a warm start is provided to the initial SRM that preserves all scheduled routings incorporating the minimum possible delay with each flight (thereby initially not considering flight cancellations). As a result the integrated and sequential

Table 5 Integrated Recovery Summary (50% Flow Rate Reduction)

	Event			
	1	2	3	4
Subproblem costs ($)				
SRM	0	0	0	50,000
CRM	0	0	0	4,000
PRM	11,653	22,942	46,057	54,820
Solution metrics				
Mean flt delay	20:05	20:34	39:50	33:41
Cancelled flts (%)	0	0	0	1.6
Delayed flts (%)	12.8	35.1	38.0	49.3
Total deadheads	0	0	0	2
Mean PAX delay	23:09	21:47	39:22	33:37
Unassigned PAX	0	3	3	5
CPU time	1:02	24:41	32:28	36:34

Table 6 **Sequential Recovery Summary (75% Flow Rate Reduction)**

	Event			
	1	2	3	4
Subproblem costs ($)				
SRM	0	0	0	150,000
ARM	0	0	INFEAS	INFEAS
CRM	0	0	INFEAS	INFEAS
PRM	15,316	29,440	62,316	85,039
Solution metrics				
Mean flt delay	17:57	28:24	46:58	44:01
Cancelled flts (%)	0	0	0	4.6
Delayed flts (%)	28.7	37.7	52.3	54.4
Total deadheads	0	0	INFEAS	INFEAS
Mean PAX delay	22:19	28:52	50:23	44:41
Unassigned PAX	2	4	8	24
CPU time	1:01	10:02	14:11	14:29

Table 8 **Sequential Recovery Summary (Hub Closure)**

	Event			
	1	2	3	4
Subproblem costs ($)				
SRM	0	0	0	175,000
ARM	0	0	0	INFEAS
CRM	0	0	0	INFEAS
PRM	17,979	32,057	56,730	133,573
Solution metrics				
Mean flt delay	28:10	25:28	27:51	49:47
Cancelled flts (%)	0	0	0	4.0
Delayed flts (%)		66.9	64.2	59.3
Total deadheads	0	0	0	INFEAS
Mean PAX delay	17:56	31:46	41:58	43:41
Unassigned PAX	4	4	3	36
CPU time	0:35	17:50	31:01	20:41

solutions may coincide if the warm start is optimal. This occurs in two of the scenarios, which explains why the integrated recovery framework provides no improvement. Of course, relaxing the warm start will induce the integrated solution to dominate its sequential counterpart. No scenarios were encountered from the integrated model where no integer feasible solution was found to a subproblem after the Benders framework has terminated.

We note that the 75-minute disruption seems to prohibit obtaining a solution in our 30-minute runtime goal. Although about 60% of the flights are initially disruptable from the scheduled routings, all flights are disruptable through the process by which we limit the scope (§3). Moreover, the number of strings is vastly higher because of a longer maximum flight delay period. The multiple disruption scenario performs better, but it does not always meet the runtime goal in the integrated setting (Tables 5 and 7).

Moreover we note the improvement in solution quality the integrated approach delivers over the sequential one. First, note that 25% of the scenarios show the sequential approach is infeasible where the integrated approach always delivers a solution. Secondly, we note a reduction in the key performance metrics that include flight delay, passenger delay, and cost of recovery. Table 10 shows how the integrated module reduces mean passenger delay, mean flight delay, and passenger reaccommodation costs by averaging across the 50%, 75%, and 100% capacity reduction scenarios. Of particular interest in the behavior of mean passenger delay, which is reduced by as much as 14.5% in the 75-minute disruption. The integrated model also reduces passenger reaccommodation costs considerably.

Another question of interest is how the solution quality changes with respect to input parameters. Figure 6 shows two experiments of interest using the 60-minute hub closure disruption scenario. Panel 6(a) shows how the cancellation rate changes with respect to the cost of flight cancellations c_f^{cancel}. As mentioned previously, the airline under consideration is highly adverse to flight cancellations because of their own idiosyncratic requirements. The figure shows that as

Table 7 **Integrated Recovery Summary (75% Flow Rate Reduction)**

	Event			
	1	2	3	4
Subproblem costs ($)				
SRM	0	0	0	100,000
CRM	0	0	0	5,000
PRM	15,316	22,198	51,336	40,489
Solution metrics				
Mean flt delay	17:57	28:31	44:01	33:05
Cancelled flts (%)	0	0	0	2.3
Delayed flts (%)	28.7	38.1	42.1	56.4
Total deadheads	0	0	0	3
Mean PAX delay	22:19	20:36	41:44	36:19
Unassigned PAX	2	3	6	7
CPU time	1:04	23:20	30:56	32:27

Table 9 **Integrated Recovery Summary (Hub Closure)**

	Event			
	1	2	3	4
Subproblem costs ($)				
SRM	0	0	0	100,000
CRM	0	0	0	5,000
PRM	12,186	24,566	41,993	58,300
Solution metrics				
Mean flt delay	19:25	23:24	29:39	34:40
Cancelled flts (%)	0	0	0	2.3
Delayed flts (%)	26.4	40.6	59.1	58.5
Total deadheads	0	0	0	3
Mean PAX delay	16:04	21:02	36:41	41:54
Unassigned PAX	2	4	3	7
CPU time	1:46	24:09	31:00	24:22

Table 10 **Summary of Improvement from Integrated Model**

	Performance metric improvement (%)		
Event	Mean passenger delay	Mean flight delay	PRM cost
30-minute disruption	2.9	13.2	12.9
60-minute disruption	13.7	6.2	22.3
75-minute disruption	14.5	3.3	20.3
Multiple disruptions	12.4	25.1	54.2

long as the cost associated with a cancellation exceeds \$15,000 per flight, the same recovery tactic that considers only delays remains optimal. Cancellations only become desirable when the cancellation penalty is between \$10,000 and \$15,000 per flight. Panel 6(b) illustrates the tradeoff between the severity of passenger delay and cancellations by changing the cost of unassigned passengers c_i^{unassign}. The solution summarized in Table 9 (setting c_i^{unassign} to \$2,500 for all $i \in \text{OD}$) remains the optimal solution for all values c_i^{unassign} that exceed \$1,000. The tradeoffs between passenger delay and flight cancellations change the solution only when the penalty parameter is between \$500 and \$1,000 per passenger. Therefore, the optimal solution attained in the integrated model for the one-hour hub closure is robust with respect to these two input parameters under consideration. For brevity we do not report the full set of parametric studies in this paper, but they are available in Petersen (2011).

6. Conclusion and Future Work

This paper seeks 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 allow for efficient decomposition 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 reasonably sized disruptions.

This paper is one of the first attempts to computationally 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. Our approach may well handle longer time horizons on other networks. An efficient procedure was 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 connecting duties is relatively small. Extending this to a multiday 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 on the original flight schedule and locally repair the network within each iteration, as opposed to building it anew every iteration. Our model could also be implemented by parallelizing components of the problem to reduce computational effort. Another relevant question is to seek whether the Benders cuts can be strengthened to obtain a tighter LP-relaxation of the master problem.

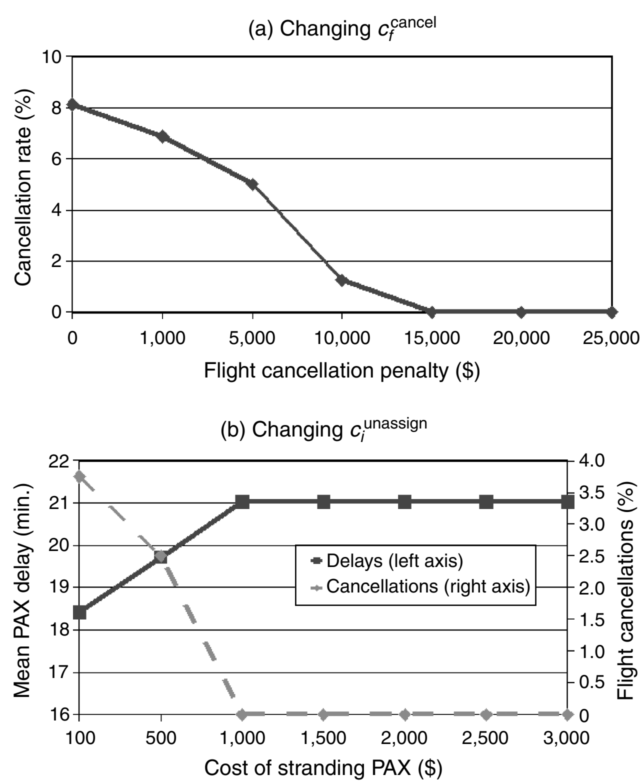

Figure 6 **Sensitivity Analyses for a One-Hour Hub Closure**

One possible approach would be to lift in additional coefficients whose dual was zero. Another would be to shift in the Benders cut further to the interior of the feasible region for the SRM.

Acknowledgments

The authors are grateful that this work has been supported in large part by Sabre Holdings. The authors have also benefited from fruitful discussions with Michael Clarke, Barry Smith, George Nemhauser, Shabbir Ahmed, and Tim Niznik. The authors also thank two anonymous referees for exceptional comments and suggestions that greatly enhanced the quality of the work.

References

Ahuja RK, Magnanti TL, Orlin JB (1993) *Network Flows: Theory, Algorithms, and Applications* (Prentice Hall, Upper Saddle River, NJ).

Anbil, R., Tanga R, Johnson EL (1992) A global approach to crew pairing optimization. *IBM Systems J.* 31(1):71–78.

Ball M, Barnhart C, Nemhauser GL, Odoni A (2007) Air transportation: Irregular operations and control. Barnhart C, Laporte G, eds. *Transportation Volume* 14: *Handbook of Operations Research and Management Science* (Elsevier, Amsterdam), 1–73.

Ball M, Barnhart C, Dresner M, Hansen M, Neels K, Odoni A, Peterson E, Sherry L, Trani A, Zou B (2010) Total delay impact study: A comprehensive assessment of the costs and impacts of flight delay in the United States. Technical report, Federal Aviation Administration, Washington, DC.

Barnhart C, Boland NL, Clarke LW, Johnson EL, Nemhauser GL (1998) Flight string models for aircraft fleeting and routing. *Transportation Sci.* 32(3):208–220.

Bisaillon S, Cordeau J-F, Laporte G, Pasin F (2010) A large neighbourhood search heuristic for the aircraft and passenger recovery problem. Accessed July 2010, http://neumann.hec.ca/chairelogistique/jfc.shtml.

Bonnefoy PA (2008) Development of multiairport systems: A worldwide approach. Unpublished Ph.D. thesis, Massachusetts Institute of Technology, Cambridge.

Bratu S, Barnhart C (2006) Flight operations recovery: New approaches considering passenger recovery. *J. Sched.* 9(3): 279–298.

Clarke MDD (1998a) Development of heuristic procedures for flight rescheduling in the aftermath of irregular airline operations. Unpublished Ph.D. thesis, Massachusetts Institute of Technology, Cambridge.

Clarke MDD (1998b) Irregular airline operations: A review of the state-of-the-art practice in airline operations control center. *J. Air Transportation Management* 4(2):67–76.

Clausen J, Larsen A, Larsen J, Rezanova NJ (2010) Disruption management in the airline industry—Concepts, models, and methods. *Comput. Oper. Res.* 37:809–821.

Falkner JC, Ryan DM (1987) A bus crew scheduling system using a set partitioning model. *Asia-Pac. J. Oper. Res.* 4:39–56.

Filar J, Manyem P, White K (2001) How airlines and airports recover from schedule perturbations: A survey. *Ann. Oper. Res.* 108:315–333.

Gabteni S (2009) ROADEF competition winners methods review. (May) AGIFORS Airline Operations Meeting, Montreal.

Gao C (2007) Airline integrated planning and operations. Ph.D. thesis, Georgia Institute of Technology, Atlanta.

Gao C, Johnson E, Smith B (2009) Integrated airline fleet and crew robust planning. *Transportation Sci.* 43(1):2–16.

Jarrah AIZ, Yu G, Krishnamurthy N, Rakshit A (1993) A decision support framework for airline flight cancellations and delays. *Transportation Sci.* 27(3):266–280.

Klabjan D, Schaefer AJ, Johnson EL, Nemhauser GL (2002) Robust airlines crew scheduling. Technical report, The Logistics Institute, Georgia Institute of Technology, Atlanta.

Kohl N, Larsen A, Larsen J, Ross A, Tiourine S (2007) Airline disruption management—Perspectives, experiences, and outlook. *J. Air Transportation Management* 13(3):149–162.

Lan S, Clarke J-P, Barnhart C (2006) Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Sci.* 40(1):15–28.

Lettovsky L (1997) Airline operations recovery: An optimization approach. Ph.D. thesis, Georgia Institute of Technology, Atlanta.

Lettovsky L, Johnson EL, Nemhauser GL (2000) Airline crew recovery. *Transportation Sci.* 34(4):337–348.

Nemhauser GL, Wolsey LA (1999) *Integer and Combinatorial Optimization* (John Wiley & Sons, New York).

Palpant M, Boudia M, Robelin C-A, Gabteni S, Laburthe F (2009) ROADEF 2009 challenge: Disruption management for commercial aviation. Accessed June 2010, http://challenge.roadef.org/2009.

Petersen JD (2011) Large-scale mixed integer optimization approaches for scheduling airline operations under irregularity. Ph.D. thesis, Georgia Institute of Technology, Atlanta.

Rosenberger JM, Johnson EL, Nemhauser GL (2003) Rerouting aircraft for airline recovery. *Transportation Sci.* 37(4):408–421.

Rosenberger JM, Johnson EL, Nemhauser GL (2004) A robust fleet-assignment model with hub isolation and short cycles. *Transportation Sci.* 38(3):357–368.

Shaw T (2003) Hybrid column generation for large networks routing problems: With implementations in airline crew scheduling. Ph.D. thesis, Georgia Institute of Technology, Atlanta.

Shebalov S, Klabjan D (2006) Robust airline crew pairing: Move-up crews. *Transportation Sci.* 40(3):300–312.

Smith B (2004) Robust airline fleet assignment. Ph.D. thesis, Georgia Institute of Technology, Atlanta.

Smith BC, Johnson EL (2006) Robust airline fleet assignment: Imposting station purity using station decomposition. *Transportation Sci.* 40(4):497–516.

Stojković G, Soumis F, Desrosiers J, Solomon MM (2002) An optimization model for a real-time flight scheduling problem. *Transportation Res. A* 36(9):779–788.

Stojković M, Soumis F (2001) An optimization model for the simultaneous operational flight and pilot scheduling problem. *Management Sci.* 47(9):1290–1305.

Stojković M, Soumis F, Desrosiers J (1998) The operational airline crew scheduling problem. *Transportation Sci.* 32(3): 232–245.

Teodorovic D, Guberinic G (1984) Optimal dispatching strategy on an airline network after a schedule perturbation. *Eur. J. Oper. Res.* 15(2):178–182.

Teodorovic D, Stojković G (1990) Model for operational daily airline scheduling. *Transportation Plan. Tech.* 14(4):273–285.

Teodorovic D, Stojković G (1995) Model to reduce airline schedule disturbances. *J. Transportation Engrg.* 121(4):324–331.

Thengvall BG, Yu G, Bard JF (2001) Multiple fleet aircraft schedule recovery following hub closures. *Transportation Res. A* 35(4):289–308.

Van Roy TJ (1983) Cross decomposition for mixed integer programming. *Math. Programming* 25(1):46–63.

Wei G, Yu G, Song M (1997) Optimization model and algorithm for crew management during airline irregular operations. *J. Comb. Optim.* 1(3):305–321.

Zhang Y, Hansen M (2008) Real-time intermodal substitution: Strategy for airline recovery from schedule perturbation and for mitigation of airport congestion. *Transportation Res. Rec.* 2052:90–99.