



Transportation Science

TRANSPORTATION SCIENCE

Volume 51 • Number 1 • February 2017



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

Flight Network-Based Approach for Integrated Airline Recovery with Cruise Speed Control

Uğur Arıkan, Sinan Gürel, M. Selim Aktürk

To cite this article:

Uğur Arıkan, Sinan Gürel, M. Selim Aktürk (2017) Flight Network-Based Approach for Integrated Airline Recovery with Cruise Speed Control. *Transportation Science* 51(4):1259-1287. <https://doi.org/10.1287/trsc.2016.0716>

Full terms and conditions of use: <http://pubsonline.informs.org/page/terms-and-conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact permissions@informs.org.

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2017, INFORMS

Please scroll down for article—it is on subsequent pages



INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

Flight Network-Based Approach for Integrated Airline Recovery with Cruise Speed Control

Uğur Arıkan,^{a,b} Sinan Gürel,^a M. Selim Aktürk^c

^aDepartment of Industrial Engineering, Middle East Technical University, 06800 Ankara, Turkey; ^bEngineering Systems and Design, Singapore University of Technology and Design, Singapore 487372; ^cDepartment of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey

Contact: uarikan@outlook.com (UA); gsinan@metu.edu.tr (SG); akturk@bilkent.edu.tr (MSA)

Received: February 26, 2014

Revised: August 5, 2015; March 19, 2016

Accepted: July 12, 2016

Published Online in Articles in Advance:
March 29, 2017

<https://doi.org/10.1287/trsc.2016.0716>

Copyright: © 2017 INFORMS

Abstract. Airline schedules are generally tight and fragile to disruptions. Disruptions can have severe effects on existing aircraft routings, crew pairings, and passenger itineraries that lead to high delay and recovery costs. A recovery approach should integrate the recovery decisions for all entities (aircraft, crew, passengers) in the system as recovery decisions about an entity directly affect the others' schedules. Because of the size of airline flight networks and the requirement for quick recovery decisions, the integrated airline recovery problem is highly complex. In the past decade, an increasing effort has been made to integrate passenger and crew related recovery decisions with aircraft recovery decisions both in practice and in the literature. In this paper, we develop a new flight network based representation for the integrated airline recovery problem. Our approach is based on the flow of each aircraft, crew member, and passenger through the flight network of the airline. The proposed network structure allows common recovery decisions such as departure delays, aircraft/crew rerouting, passenger reaccommodation, ticket cancellations, and flight cancellations. Furthermore, we can implement aircraft cruise speed (flight time) decisions on the flight network. For the integrated airline recovery problem defined over this network, we propose a conic quadratic mixed integer programming formulation that can be solved in reasonable CPU times for practical size instances. Moreover, we place a special emphasis on passenger recovery. In addition to aggregation and approximation methods, our model allows explicit modeling of passengers and evaluating a more realistic measure of passenger delay costs. Finally, we propose methods based on the proposed network representation to control the problem size and to deal with large airline networks.

Keywords: airline operations • integrated recovery • disruption management • irregular operations • passenger recovery • cruise speed control • conic quadratic mixed integer programming • flight network

1. Introduction

Poor weather conditions, congestions at hubs, and aircraft mechanical problems are just a few of the causes that prevent airlines from operating their flight schedules as planned. Departure/arrival delays, flight cancellations, and even airport closures can occur. These irregularities in operations are called *disruptions*. When a disruption occurs an airline has to repair aircraft schedules, crew schedules, and passenger itineraries to minimize disruption related costs and maintain customer service quality. In this paper, we propose a recovery approach that integrates aircraft, crew, and passenger recovery decisions. The proposed approach involves a network flow representation of the recovery problem which leads to an efficient mathematical programming formulation.

When we analyze on-time performance data provided by the Bureau of Transportation Statistics (BTS), we observe that disruptions are not rare. About 18.28% of all flights operated in 2015 have experienced more than 15 minutes of arrival delay. Another observation

is that about 1.54% of scheduled flights have been canceled. Airline Operations Control Centers (AOCCs) take actions against disruptions. An AOCC has to seek a quick recovery solution. The objective is to find the optimal set of actions that minimizes the costs of disruptions provided that the original schedule will be resumed at the end of a specified recovery period. In practice, retiming/canceling flights, swapping aircraft among flights, rerouting crew members and passengers, canceling passenger tickets, utilizing spare aircraft, and standby crew members are common recovery actions. Deadheading crew members and ferrying aircraft are not desired actions; however, they may be required as well. Limited solution time in disruption management is challenging. Therefore, airlines typically start by creating recovery plans for aircraft and crew members and then perform passenger recovery. However, this sequential approach results in high itinerary cancellations and passenger delay costs. This necessitates an integrated approach, which has been considered in the recent operations research literature on aviation.

1.1. Literature Review

For a recent review on airline disruption management, we refer to Clausen et al. (2010). Most of the studies in the literature have focused on aircraft recovery; and integration with passenger and crew recovery has not been explicitly considered in most of the recovery models. Jarrah et al. (1993) propose two minimum cost network flow models to recover from aircraft shortages. The first model retimes departure times to minimize delay costs; and the second one determines canceled flights to minimize cancellation costs. Rosenberger, Johnson, and Nemhauser (2003) present a model that reschedules flights and reroutes aircraft with the objective that minimizes rerouting and cancellation costs. The authors also present an extension of the model that they propose by introducing a cost for disrupted itineraries to maintain passenger connections. Abdelghany, Abdelghany, and Ekollu (2008) develop an integrated decision support tool for airlines schedule recovery during irregular operations. The tool is designed for the operators in AOCCs. It is capable of detecting current and future flight delays and aims to generate proactive integrated recovery plans to avoid these delays. Proposed framework integrates a schedule simulation model and a resource assignment optimization model. The models focus on aircraft and crew recovery, however, passenger rebooking costs as a result of flight cancellations are indirectly included in the approach. In a recent study, Maher (2016) focuses on aircraft and crew recovery problems. The author points out the challenge of delivering high quality solutions within short time limits, and proposes a general framework for column-and-row generation as an extension of the existing methods to reduce the problem size. The approach also aims to reduce passenger dissatisfaction through increasing the cancellation cost of a flight.

Passenger recovery decisions have received increased attention in recent studies. Lan, Clarke, and Barnhart (2006) propose two new approaches to minimize passenger disruptions and achieve robust airline schedule plans. The first approach involves aircraft routing and the second one involves retiming the departure times for flights. Aircraft routing problems are considered as a feasibility problem with the aim of achieving robustness with minimal cost implications. In the proposed robust aircraft maintenance routing (RAMR) model, the authors try to minimize the expected total propagated delay. In the second part of their study, the authors consider passengers who miss their flights because of insufficient connection time. The aim of this approach is to minimize the number of passenger misconnections by retiming the departure times of flights within a small time window. For retiming departure times, the authors propose the connection-based flight schedule retiming (CFSR) model. The objective of the model is

to minimize the expected total number of disrupted passengers.

Bratu and Barnhart (2006) aim to find the optimal trade-off between airline operating costs and passenger delay costs, and propose two optimization models. The first one, named Disrupted Passenger Metric (DPM), aims to minimize passenger disruption costs without increasing operating costs. DPM does not consider passenger rerouting decisions and uses an approximate measure for passenger delay costs. The second model, Passenger Delay Metric (PDM), uses a more accurate way of calculating passenger delays.

In a recent paper, Jafari and Zegordi (2010) integrate aircraft and passenger recovery decisions. They present an assignment model for recovering both aircraft and passengers simultaneously with the objective of minimizing the sum of aircraft assignment costs, delay costs, cancellation costs, and disrupted passenger costs.

Marla, Vaaben, and Barnhart (2017) integrate recovery decisions for aircraft and passengers with cruise speed decisions. They utilize the same traditional time-space network representation as Bratu and Barnhart (2006), in which nodes are associated with both time and location. Flights are represented by arcs between two nodes belonging to different locations. To satisfy the ground time between any two consecutive flights, ground arcs starting and ending at the same location are included. Departure time decisions are evaluated by creating flight copies at different departure time alternatives. Marla, Vaaben, and Barnhart (2017) manage to incorporate cruise speed control, or flight planning, within the time-space network by generating a second set of flight copies at each departure time alternative of each flight where each copy corresponds to a different cruise speed alternative. However, this requires discretization of cruise speed options and a large network to be generated. In the proposed mathematical formulation, the authors manage to integrate passenger rerouting decisions in passenger delay cost calculation. In addition, the authors propose an approximation model to deal with large airline networks. In this paper, we show that the recovery decisions like flight timing, aircraft rerouting, and cruise speed control can be formulated over a smaller network without limiting cruise time decisions to the discrete settings.

Petersen et al. (2012) study an integrated airline recovery problem using a single-day horizon, and propose a separate mixed-integer mathematical model for the schedule, aircraft, crew, and passenger recovery problems. Each of the separate formulations uses distinct sets of recovery actions. With the notation used in our paper, the schedule recovery problem deals with flight related decisions such as departure time and cancellation decisions; the aircraft recovery problem

handles aircraft rerouting decisions; the crew recovery problem handles crew rerouting decisions; and the passenger recovery problem decides on the passenger reallocations. The authors utilize a Benders decomposition scheme together with the column generation approach to achieve the coordination among these four mathematical models. They place a 30-minute threshold of computation time for the overall problem; they also propose a sequential recovery algorithm to handle larger problems. In this study, we integrate flight, aircraft, crew, and passenger related recovery decisions using a network flow approach. Different from Petersen et al. (2012), we consider cruise speed control, which is another effective recovery approach in airline disruption management. Moreover, Petersen et al. (2012) utilize flight string representation, while we utilize a flight network representation. A flight string is a sequence of flights with timing decisions to represent the problem. The same sequence of flights might be present in multiple strings, each with a different set of retiming decisions. Flight strings allow to handle complicated airline constraints such as crew rest restrictions depending on the sequence of flights and flying hours by evaluating the feasibility of flight strings beforehand. Moreover, complex and sequence dependent cost functions can be associated by evaluating the costs of flight strings in the preprocessing step. On the other hand, a disadvantage of this representation is that it requires discretization in retiming decisions. It is possible to associate cruise speed control action through generation of copies of flight strings, where each copy corresponds to a different cruise speed. However, this tends to increase the problem size and requires discretization of cruise speeds. Petersen et al. (2012) also discuss the importance of reducing the problem size when dealing with large airline networks, and the authors propose a simple algorithm that limits the scope of recovery to flights in the routing of the disrupted entities. Based on our network representation, we propose a systematic approach using the interdependencies among the recovery actions of entities to accurately control the problem size.

Arıkan, Gürel, and Aktürk (2016) focus on the integrated aircraft and passenger recovery problem. The authors propose a mathematical formulation that is able to evaluate several aircraft and passenger recovery actions such as holding departure times, maintaining or canceling passenger itineraries, and cruise speed control simultaneously. The objective function includes both passenger related costs and fuel costs. The authors manage to reformulate the nonlinear programming model as a conic quadratic mixed integer programming model that can be solved efficiently. The presented results give insights about the impact of cruise speed control action in mitigating delays and reducing

passenger delay costs. However, the proposed mathematical formulation is not flexible for extending the model to other entity types and recovery actions. In this study, we propose a general network structure that allows the integration of aircraft, crew, and passenger recovery, and utilization of a larger set of recovery actions.

Aktürk, Atamtürk, and Gürel (2014) propose an aircraft rescheduling model to deal with aircraft recovery problems. The authors successfully integrate cruise speed control action in the recovery model using a realistic fuel cost function to optimally solve the trade-off between fuel consumption and disturbances of the disruptions. In addition to the additional fuel cost of speeding up flights, the authors manage to integrate environmental costs and constraints. The authors report that cruise speed control can provide significant cost savings. One of the major contributions of Aktürk, Atamtürk, and Gürel (2014) is enabling use of a realistic fuel cost function based on the fuel flow model developed by the Base of Aircraft Data (BADA) project of EUROCONTROL (2009). However, the approach focuses on aircraft schedules and does not deal with the integrated recovery problem. We integrate the fuel cost function and conic quadratic reformulations proposed in Aktürk, Atamtürk, and Gürel (2014) in our network-based approach that deals with the integrated recovery problem and allows a wide range of recovery actions.

We have also benefited from studies that do not directly focus on solving the airline recovery problem. First, Ball et al. (2010) present an extensive analysis on the components of delay costs, such as cost to airlines, cost to passengers, cost of lost demand, etc., as well as flight delays' indirect impact on the U.S. economy. The authors present innovative methodologies to measure the impact of flight delays and estimate cost components. The proposed approach considers a broader consideration of relevant costs than conventional methods.

Second, Barnhart, Fearing, and Vaze (2014) point out the lack of publicly available passenger travel data, which is very important in testing integrated recovery approaches. The authors provide an excellent guide for processing public data to generate possible passenger itineraries. Furthermore, the authors use discrete choice methodology and propose a logit-based choice model to assign the aggregate passenger demand to the possible itineraries.

Finally, Sherali, Bae, and Haouari (2013) focus on schedule design, fleet assignment, and aircraft routing problems, and propose a mixed-integer programming model that integrates certain aspects of these problems. A reformulation-linearization technique is applied to reduce the complexity of the problem. To deal with

the large-scale problem, the authors propose a Benders' decomposition-based solution approach and test their approach using real data from United Airlines. One of the major contributions of the study is that the authors represent the problem with a flight network alternative to the traditional time-space network representation for aircraft routing. Flight networks represent the problem with a much smaller number of nodes and arcs since each scheduled flight is represented by a single node. It is an activity-on-node representation, and hence, both departure and arrival time decisions can be represented by a single continuous variable. Moreover, all aircraft routings can be included in the solution space while avoiding path enumeration. The authors report that the representation is more compact than traditional representations and allows a greater modeling flexibility in routing and timing decisions. In this paper, we extend the flight network representation so that all types of entities (aircraft, crew members, and passengers) are transported through the same network. We manage to integrate a wide range of recovery actions with the proposed representation. Since it is an activity-on-node representation, we manage to model cruise speed control action with continuous decision variables. The unified representation allows to capture the interdependencies in aircraft, crew, and passenger-related recovery decisions, which is crucial in integrated recovery problems. Moreover, it allows to develop fast network-based algorithms to control the problem size. One limitation of flight networks is that attributes cannot be assigned to paths. In the airline recovery context, this leads to limitations in the modeling complicated sequence, the time-dependent crew rest period, and aircraft maintenance restrictions. These limitations can be overcome to some extent. For instance, the proposed formulation in this study can be extended to limit total air time and/or number of flights assigned to aircraft and crew members.

1.2. Contributions

The first contribution of this paper is that we integrate recovery decisions for different entities in an airline system over a simplified network. We propose a flight network representation unlike the time-space network and flight string representations that are used in airline recovery problems. A major advantage of the proposed network representation is that the problem size is kept within reasonable limits so that real-time solutions can be provided. Moreover, unlike traditional representations, it does not require discretization of departure time and cruise speed decisions. Sherali, Bae, and Haouari (2013) have used flight network representation for integrated schedule design, fleet assignment, and aircraft routing problems in which aircraft are transported through the flight network. We extend the flight network representations so that all types of

entities (aircraft, crew members, and passengers) are transported through the same network. Inclusion of all entity types in the unified network representation provides a great opportunity to capture the interdependencies among the recovery actions of different entities, which is crucial in integrated airline recovery. Moreover, it allows to develop network-based algorithms using these relationships to maintain tractability when dealing with large airline networks.

Second, we develop a mathematical formulation that models the recovery decisions of aircraft, crew, and passengers simultaneously to ensure optimality. We have managed to integrate a wide range of recovery actions. These recovery actions include the following:

- departure holding,
- flight cancellation,
- aircraft rerouting,
- aircraft ferrying,
- crew rerouting,
- crew deadheading,
- passenger ticket cancellation,
- passenger reallocation,
- cruise speed control,
- use of spare aircraft,
- calling up reserve crew.

The model indeed formulates a network flow problem to minimize the total recovery costs including fuel costs, that might rise because of speed and swap decisions, and passenger related disruption costs such as delay and ticket cancellation costs.

Third, this paper places a special emphasis on passenger recovery. In addition to itinerary-based modeling (as in most recovery approaches in the literature), we manage to model each passenger explicitly. This representation has several advantages such as assigning various levels of importance and defining different sets of recovery actions for each passenger. Moreover, it allows accurate evaluation of passenger delay costs by simultaneously considering flight delay decisions and passenger rerouting decisions. Despite the increased problem size, we managed to optimally solve recovery problems by explicitly modeling passengers for airline networks including around 288 flights within about 9 minutes. For larger networks including around 473 flights we managed to solve recovery problems by using approximations for passenger delay cost within about 8 minutes.

Finally, the integrated recovery problem is highly complex and a real-time solution requirement is challenging when dealing with large networks. Contributions of this study to this problem are twofold. First, we focus on the compactness of the problem representation to reduce the problem size without changing the optimal cost; second we propose a network-based algorithm to limit the scope of recovery while providing near-optimal solutions.

Although proposed representation keeps the problem size within reasonable limits, a careful investigation of compactness is crucial to provide real-time solutions. For this purpose, we propose two preprocessing approaches that reduce the problem size. The first one, named the partial network approach, aims to identify and eliminate infeasible recovery actions from the solution space. This is carried out by isolating the related portion of the network for each entity. These isolated portions are entity-specific and called partial networks. A partial network of an entity is able to generate all possible recovery actions for the entity, while the common flight nodes construct the interdependencies among the partial networks of different entities. Using partial networks of entities instead of the entire flight network provides a more compact representation. The underlying network representation allows to develop a considerably fast algorithm, named the Partial Network Generation Algorithm (PNGA), to generate partial networks prior to solving the optimization problem. We have observed in our experiments that the reduction in problem size with the partial network approach is significant. In the second approach, we propose a rule to aggregate entities without losing any required information. We provide a procedure that extends the proposed mathematical formulation to handle entity aggregation.

An alternative approach to sequential recovery and approximation models for providing near-optimal solutions is to reduce the problem size. The scope of recovery must carefully be limited to provide real time recovery decisions while maintaining the quality of the solution (Petersen et al. 2012). Literature lacks methodologies that systematically control the problem size to allow real time solutions. Using the interdependencies among the partial networks, we define a measure of likelihood that a rerouting action will be used in the optimal solution. Using the proposed measure and partial network representation, we propose the Problem Size Control Algorithm (PSCA) to limit the problem size. The algorithm iteratively eliminates the rerouting actions that are less likely to be utilized in the solution from the partial networks. The underlying network structure and the proposed measure allow to incorporate fast shortest path algorithms. The proposed algorithm can provide significant reductions in problem size and solution time. Using this approach, we managed to solve integrated recovery problems for airline networks including 1,254 flights within 8 minutes.

1.3. Paper Outline

This paper is organized as follows. In Section 2, the network representation is described in detail and a numerical example is given. Mathematical formulations are constructed in Section 3. In Section 4, a scheme to

reformulate the mixed integer nonlinear programming problem as conic quadratic mixed integer programming is given. In Section 5, preprocessing methods to enhance the performance are described. In Section 6 an algorithm to control the problem size to allow practical solutions is proposed. Results of the computational study are discussed in Section 7. Final remarks are given in Section 8.

2. Problem Representation

In this section, we give the problem definition and present the proposed network structure. An original schedule of an airline is given. A set of disruptions occur on the schedule. We consider a recovery horizon, $[t_0, t_1]$. The aim of the *airline recovery problem* is to find the minimum-cost recovery actions by altering operations of aircraft, crew members, and passengers within the recovery horizon provided that the original schedule will be caught up by t_1 at the latest.

An effective representation of the disruption management problem is crucial because of the size of the flight networks, complexity of the problem, and limited solution times. We have mentioned two important representations used in recovery problems in Section 1.1, namely the flight string representation (Petersen et al. 2012) and the time-space network representation (Bratu and Barnhart 2006, Marla, Vaaben, and Barnhart 2017). Moreover, we have discussed the flight network representation proposed by Sherali, Bae, and Haouari (2013) for integrated schedule design, fleet assignment, and the aircraft routing problem. In this section, we propose an extended flight network representation for the recovery problem that integrates aircraft, crew members, and passengers, as well as a wide variety of recovery actions.

We start our approach by defining state parameters that capture the true state of any entity. These definitions allow modeling of all entity types (aircraft, crew member, or a passenger) in a similar manner. Then, we propose a general flight network representation that allows to integrate any entity type. Therefore, not only aircraft, but all entities are transported through a flight network. By integration on a common flight network, interdependencies among different entity types are easily defined. Moreover, all recovery actions including cruise speed control are included in the model to ensure optimality. Since activity is kept on nodes, departure time, arrival time, and cruise speed decisions can be represented by continuous variables instead of a set of discrete alternatives.

2.1. Network Structure

We start with the notation required to understand a network structure. For ease of reading, we use over-scores and under-scores to denote parameters as upper and lower bounds, respectively. All parameters begin

with an uppercase letter while decision variables start with a lowercase letter throughout the text. Parameters of scheduled flights are defined next.

- Ori_f (Des_f): Origin (destination) airport of flight f ,
 SDT_f (SAT_f): Scheduled departure (arrival) time of flight f in the original schedule,
 \overline{DT}_f (\overline{AT}_f): Latest allowable departure (arrival) time of flight f ,
 ΔT_f^r : Cruise time compression limit of aircraft r for flight f ,
 FT_f^r : Flight time of flight f when operated by aircraft r at max-range cruise speed,
 AT_f : Earliest possible arrival time of flight f ,
 \underline{CT}_{fg} : Minimum of minimum connection times among all entities between flights f and g .

Cruise time compression is the reduction in the cruise time, and hence in the flight time, by speeding up the aircraft during the cruise stage. Cruise speed may be increased only up to a certain extent because of technical limitations and airline policy.

Therefore, cruise time compression is limited. An airline may have different types of aircraft in its fleet. As a recovery action, aircraft swaps may occur between flights. Therefore, we consider aircraft-and-flight-specific cruise time compression limits and flight time parameters, i.e., ΔT_f^r and FT_f^r . Maximum-range cruise speed is the speed of an aircraft that results in minimum fuel consumption which will be discussed in detail in Section 4. To generate all possible rerouting options, we set the value of \underline{CT}_{fg} to the minimum of required connection times among all entities. Practically, a flight can depart whenever the operating aircraft, crew members, and assigned passengers are ready. Without loss of generality, we assume that a flight cannot depart before its scheduled departure time. However, early departures can be associated with the proposed approach by substituting SDT_f with \underline{DT}_f throughout this paper, where \underline{DT}_f is defined as the earliest time that flight f is allowed to depart.

Note that there are two limitations on the earliest arrival time of a flight. The first one is determined by time slot availability. On the other hand, a flight cannot arrive before $SDT_f + \min_r \{FT_f^r - \Delta T_f^r\}$ where the minimum operation is carried out among all aircraft. Therefore, \underline{AT}_f is set to the maximum of these limitations.

2.1.1. Entities. All entities will be transported through the proposed network representation, and hence, aircraft, crew, and passenger related recovery decisions will be integrated. Throughout this paper, we use the term *entity* to refer to an aircraft, a crew member, or a passenger. Let T be the set of entity types relevant to our problem, $r \in R^t$ be an entity of type t , and $R = \bigcup_{t \in T} R^t$ be the set of all entities. We use abbreviations *ac*, *cr*, and *ps* for index t to denote aircraft, crew, and passenger, respectively ($T = \{ac, cr, ps\}$).

2.1.2. Nodes. The proposed network contains four types of nodes: scheduled flight nodes, source nodes, sink nodes, and must-visit-nodes (or must-nodes). For each entity there is a source node which represents the initial state of the entity at t_0 and a sink node which represents the final status of entity at t_1 . For each entity, there might be certain must-nodes. A must-node might represent a maintenance activity of an aircraft at a specific airport at a certain time period, or a scheduled crew rest period. Each node has a demand for each entity.

Let \mathcal{F} be the set of all scheduled flights of the airline. Then, the set of *flight nodes*, F , relevant to the problem are obtained as follows:

$$F = \{f \in \mathcal{F} : SDT_f \geq t_0 \text{ and } \underline{AT}_f \leq t_1\},$$

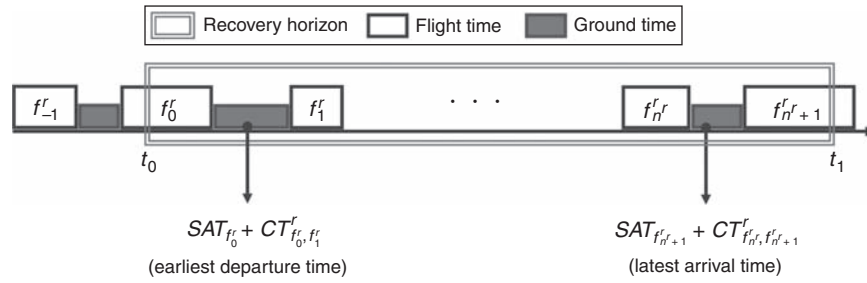
which defines all flights scheduled to depart after t_0 and with the earliest arrival time less than or equal to t_1 as illustrated in Figure 1.

The dynamic state of an entity is obtained and defined by the parameters next. Earliest departure time and latest arrival time parameters that guarantee that operations outside the recovery horizon will be operated as scheduled are illustrated in Figure 1.

- \mathcal{F}^r : Ordered set of scheduled flights originally assigned to entity r where flights with nonpositive subscripts are scheduled to be operated prior to t_0 ; flights with subscripts $n^r + 1, n^r + 2, \dots$ are scheduled to be operated after t_1 ; and the flights with subscripts $1, \dots, n^r$ are included in the recovery horizon
 $\mathcal{F}^r = \{\dots, f_{-1}^r, f_0^r, f_1^r, \dots, f_{n^r}^r, f_{n^r+1}^r, \dots\}$,
 F^r : Ordered set of flights originally assigned to entity r within the recovery horizon $F^r = \{f \in \mathcal{F}^r : SDT_f \geq t_0, \underline{AT}_f \leq t_1\} = \{f_1^r, f_2^r, \dots, f_{n^r}^r\}$,
 CT_{fg}^r : Minimum connection time required for entity r between flights f and g ,
 Ori^r : Location of entity r at the beginning of the recovery horizon (e.g., $Ori^r = Ori_{f_1}^r$),
 \underline{DT}^r : Earliest time that the first flight of entity r can depart (ready time) $\underline{DT}^r = \max\{t_0, SAT_{f_0}^r + CT_{f_0 f_1}^r\}$,
 Des^r : Planned destination of entity r at the end of the recovery horizon (e.g., $Des^r = Des_{f_{n^r}}^r$),
 \overline{AT}^r : Latest time that entity r needs to arrive at Des^r to catch up with its schedule $\overline{AT}^r = \min\{t_1, SDT_{f_{n^r+1}}^r - CT_{f_{n^r} f_{n^r+1}}^r\}$.

Recovery actions such as reserve aircraft and standby crew can be included in the solution space by inserting these entities in set R with corresponding entity parameters. These entities can be generalized as operating resources that can be used within the recovery horizon and have $F^r = \emptyset$.

Figure 1. Earliest Departure Time and Latest Arrival Time of an Entity



The *source node* for entity r is designated by s^r and has the following parameters to represent the initial state of the entity:

$$Des_{s^r} = Ori^r, \quad \overline{AT}_{s^r} = \overline{DT}^r, \quad CT_{s^r, g}^r = 0, \\ \forall g \in F, D_{s^r}^r = -1.$$

Flow of entity r through an arc between its source node and a flight node f means that the first flight assigned to r is f in the recovery. Since such arcs do not correspond to flight connections, connection times of these arcs are set to zero.

The *sink node* for entity r is designated by t^r and has the following parameters to represent the final status of the entity:

$$Ori_{t^r} = Des^r, \quad \overline{DT}_{t^r} = \overline{AT}^r, \quad CT_{f, t^r}^r = 0, \\ \forall f \in F, D_{t^r}^r = +1.$$

Flow of entity r through an arc between a flight node f and its sink node corresponds to the decision that f is the last flight assigned to r in the recovery. Similar to arcs between source and flight nodes, connection times of arcs between flight and sink nodes are set to zero.

Finally, we insert *must-nodes* to model the restrictions of operating entities within the recovery horizon such as scheduled aircraft maintenance, or away-from-home limitations or scheduled rest periods of crew members. On the other hand, we do not use must-nodes for passengers. In the proposed solution approach, we will force entities with such restrictions to visit these nodes. Let M^r be the set of must-nodes of entity r , and $M = \cup_{r \in R} M^r$. For each must-node $m \in M_r$ of entity r , we have

$$Ori_m = Des_m: \text{location of the activity,} \\ \overline{DT}_m(\underline{AT}_m): \text{earliest start (latest completion) time of} \\ \text{the activity,}$$

$$CT_{fm}^r = CT_{mg}^r = 0, \quad f, g \in F, \quad D_m^r = 0.$$

Then, the set of nodes of the network is $\mathcal{N} = F \cup (\cup_{r \in R} \{s^r, t^r\}) \cup M$. Demand of node f for entity r is denoted by D_f^r where

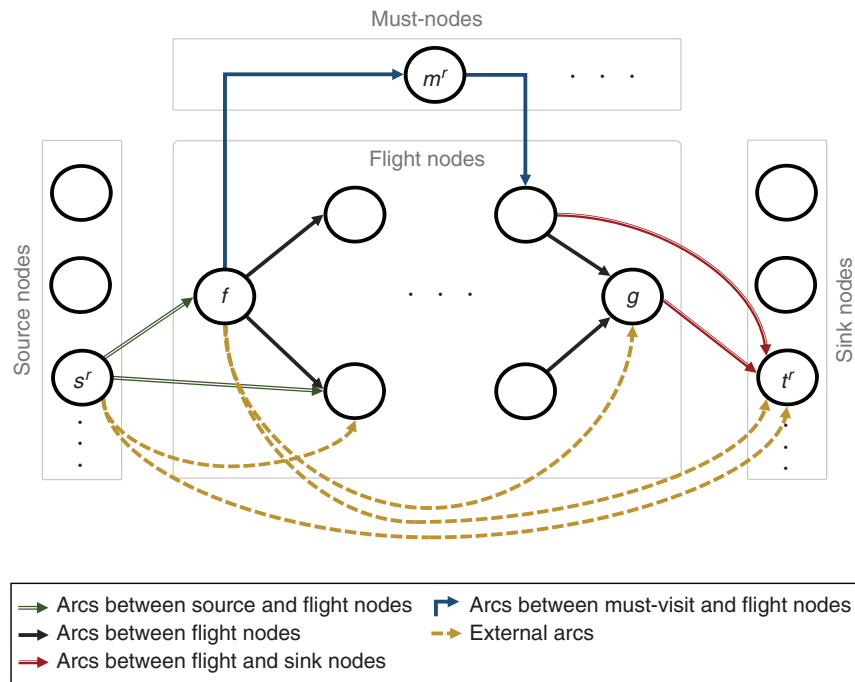
$$D_f^r = \begin{cases} -1 & \text{if } f = s^r, \text{ source node of } r \\ 0 & \text{if } f \text{ is a flight or must-visit node} \\ +1 & \text{if } f = t^r, \text{ sink node of } r. \end{cases}$$

2.1.3. Arcs. An arc (f, g) may correspond to a flight connection (if $f, g \in F$), the beginning of the operations of an entity (if $f = s^r$), the end of the operations of an entity (if $g = t^r$), or connections with must-nodes (if f or $g \in M^r$). The set of arcs is obtained using node parameters as follows:

$$A = \{(f, g): f, g \in \mathcal{N}, Des_f = Ori_g \\ \text{and } \overline{DT}_g \geq \underline{AT}_f + \underline{CT}_{fg}\}. \quad (1)$$

This rule allows to include all possible connections considering the allowed flexibility in departure and arrival times by time slots and by cruise speed options. Therefore, all possible paths can be generated through the proposed network.

To incorporate recovery actions such as ferrying aircraft or deadheading crew members, we insert *external arcs*, i.e., $(f, g) \notin A$, whose arc costs are equal to the costs of the corresponding actions. An external arc from s^r to t^r may represent ferrying the aircraft (deadheading the crew member) from its origin to its destination. An aircraft can also be ferried to its destination after operating some flights, or to the origin of another flight which can be modeled by external arcs from a flight node to the sink, and between two flight nodes, respectively. External arcs between two flight nodes, and between the source and a flight node, may correspond to crew deadheading action which are commonly used in practice. In terms of passengers, ticket cancellations or reallocation to other means of transportation may be modeled by external arcs from source to sink. For one-stop and two-stop passengers, other external arcs may be used. For instance, some of the passengers in a two-flight itinerary may be reallocated to other means of transportation at the connecting airport because of a shortage in seat capacity of the operating aircraft which

Figure 2. (Color online) Network Structure of the Proposed Representation

may be swapped with the originally assigned aircraft by the airline. Let E^r be the set of external arcs available for entity r and $E = \cup_r E^r$. Note that the sets E^r are mutually exclusive, i.e., each external arc corresponds to the recovery action of a particular entity. Then, the set of arcs of the proposed network is $\mathcal{A} = A \cup E$.

External arcs add great flexibility to the flight network representation and increase network connectivity, by relaxing the destination-origin match requirements of the arcs. In theory, any external arc (f, g) is feasible provided that the sum of SAT_f and the flight duration between Des_f and Ori_g is not greater than \overline{AT}_f . Therefore, careful selection of the external arcs to be added to the network is important to not increase the problem size by unrealistic recovery options. An airline's experience is valuable for identifying commonly used external arcs. Moreover, we suggest to add external arcs close to the disrupted nodes (close in terms of location and time). Note that during the time that aircraft or crew members travel through an external arc, they cannot operate flights. For instance, consider an aircraft reaching a flight node at 8:00, and is assigned to an external arc after 30 minutes' connection time to reach another flight node with a scheduled departure time of 20:30. The duration of this external arc is actually 12 hours even if the flight time is less. Considering the tight schedules of airlines, such external arcs would be very costly. Therefore, we suggest to limit the durations of the generated external arcs for disruptions that are unlikely to cause many cancellations. On the other hand, longer external arcs would be more beneficial for more severe disruptions such as hub closures.

The proposed network structure $G = (\mathcal{N}, \mathcal{A})$ is illustrated in Figure 2. Source and sink nodes are displayed on the left and right sides of the network, respectively. The arcs emanating from source nodes (incoming to sink nodes) represent the connection to the first (from the last) flight for the particular entity. For entities with restrictions, we have a set of must-nodes displayed at the top of the network. The nodes within the box in the middle of the network correspond to scheduled flights with incoming and emanating flight connection arcs. All connections are created with respect to the arc generation rule (1). Finally, four external arcs are displayed at the bottom of the network (dashed lines) which may correspond to different recovery actions. We have -1 ($+1$) demand in the source (sink) nodes for the corresponding entities, while all flight and must-visit nodes have zero demand.

2.2. Disruption Types

All disruptions are modeled by updating parameters of entities and specific parts of the network, i.e., no constraints need to be added in the formulation. We have selected and experimented four disruption types which are of major importance with respect to their frequency or severity. After describing how these disruptions are represented, we redefine the problem with the proposed network structure.

2.2.1. Flight Departure Delay. Departure time of a flight may be delayed as a result of various external reasons such as airport congestion or irregularities in ground operations. In cases of disruptions, departure

times of flights may be postponed by the airline as well. The latter type of departure delays are considered recovery actions and they are included in the solution space of the proposed optimization model. Therefore, we use flight departure delay disruption to refer to delays because of external sources. These disruptions are represented by updating $S\overline{D}T_f$ as $S\overline{D}T_f + DD_f$, if flight f experiences a departure delay of DD_f in minutes.

2.2.2. Flight Cancellation. If a flight experiences a severe departure delay, the airline may have no other option but to cancel the flight. Flights may be canceled because of various external sources or by the airline to recover from disruptions. Since flight cancellation is included in the solution space of the optimization model as a recovery action, we use flight cancellation disruption to refer to the cancellations by external sources. Let D^c be the set of canceled flights. Then, all nodes in D^c are removed from the network together with all arcs incoming to and emanating from these nodes.

2.2.3. Delayed Ready Time. Aircraft experiencing an unscheduled maintenance or late arrivals of crew members are examples of this type of disruption. Note that considering these as flight departure delays would eliminate many feasible recovery options and lead to suboptimal solutions. In particular, even if the ready time of an aircraft is delayed, its first flight could still be operated on time by another available aircraft. These disruptions are modeled by updating $\overline{D}T^r$ as $\overline{D}T^r + RD^r$ if entity r experiences a ready time delay of RD^r in minutes.

2.2.4. Airport Closure. Poor weather conditions are one of the major reasons for an airport to cancel all departures and arrivals for a certain time frame. Let $D^{[ac]}$ be the set of closed airports and $a \in D^{[ac]}$ be an airport experiencing a closure during $[ST_a, ET_a]$. The consequences of this closure are handled in two parts. First, as a result of the closure of airports, some flights need to be canceled. These flights are inserted into the set of canceled flights. On the other hand, some flights affected from this closure may still be operated by rescheduling the departure times or increasing their cruise speeds. Time windows of such flights are updated. The procedure to identify whether a flight node f is directly affected from airport closures or not, and to update its parameters accordingly, is presented next

- For each $a \in D^{[ac]}$
 - If $Ori_f = a, S\overline{D}T_f > ST_a$ and $\overline{D}T_f < ET_a$, then $D^{[c]} = D^{[c]} \cup f$;
 - If $Des_f = a, \underline{A}T_f > ST_a$ and $\overline{A}T_f < ET_a$, then $D^{[c]} = D^{[c]} \cup f$;
 - If $Ori_f = a, S\overline{D}T_f < ST_a$ and $\overline{D}T_f > ST_a$, then $\overline{D}T_f = ST_a$;
 - If $Ori_f = a, S\overline{D}T_f < ET_a$ and $\overline{D}T_f > ET_a$, then $S\overline{D}T_f = ET_a$;

—If $Des_f = a, \underline{A}T_f < ST_a$ and $\overline{A}T_f > ST_a$, then $\overline{A}T_f = ST_a$;

—If $Des_f = a, \underline{A}T_f < ET_a$ and $\overline{A}T_f > ET_a$, then $\underline{A}T_f = ET_a$.

In the first two conditions, the flights that need to be canceled are identified. The third condition identifies flights which are scheduled to depart prior to the closure of their origins. The update $\overline{D}T_f = ST_a$ ensures that if departure times of these flights are postponed, they do not depart during closure. In the last condition, we identify the flights for which ending time of closure of the destination airport falls within the arrival time slots. By updating $\underline{A}T_f = ET_a$, it is guaranteed that they do not arrive during closure. Note that flights between two closed airports may be marked to experience both a cancellation and a time window change, in which case the flight is canceled.

Given the network representation, the aim of the *disruption management problem* is to find the minimum-cost flow of entities from their source nodes to their sink nodes provided that must-visit nodes will be visited by corresponding entities. Optimal flows of the proposed network correspond to optimal recovery decisions over a solution space including the following recovery actions: departure delaying, flight cancellation, aircraft and crew swapping, aircraft and crew rerouting, aircraft ferrying, crew deadheading, passenger reaccommodation, ticket cancellation, and cruise speed control.

2.3. Numerical Example

We illustrate the problem representation on a small-sized numerical example. The flight schedule of an airline within the recovery horizon is tabulated in Table 1. The abbreviations Nb, SDT, SAT, Dist, and Nb Pass are used to refer to number, scheduled departure time, scheduled arrival time, distance, and number of passengers, respectively. The abbreviations ORD, DCA, DFW, LAX, and MSP correspond to Chicago O'Hare International Airport, Ronald Reagan Washington National Airport, Dallas/Fort Worth International Airport, Los Angeles International Airport, and Minneapolis–Saint Paul International Airport, respectively. All departure and arrival times presented in the table are converted to the local time at airport ORD. Three aircraft and four crew teams are involved in the problem. In this example, we assume that each flight is operated by a crew team; however, the proposed approach can handle different requirements. All these entities are assumed to be located at the origin of their first scheduled flights at 5:30. Minimum required connection time is set to 30 minutes for all types of entities. Latest departure (arrival) times of flights are set to two hours after their scheduled departure (arrival) times. These limitations may depend on the available

Table 1. Original Flight Schedule of the Example

Tail Nb	Flight Nb	Crew Id	From	To	SDT	SAT	Cruise time	Dist	Nb pass
N322AA	1	C1	ORD	DCA	5:30	7:10	70	610	126
	2	C1	DCA	ORD	7:50	9:30	70	610	149
	3	C1	ORD	DFW	10:00	12:20	110	800	111
	4	C1	DFW	ORD	13:00	15:20	110	800	166
	5	C2	ORD	DCA	16:30	18:10	70	610	153
N345AA	6	C3	LAX	ORD	6:00	9:40	190	1,745	170
	7	C3	ORD	MSP	12:00	13:10	40	335	172
	8	C3	MSP	ORD	14:00	15:10	40	335	135
	9	C1	ORD	LAX	16:00	19:40	190	1,745	139
N5FCAA	10	C4	DCA	ORD	9:00	10:40	70	610	170
	11	C4	ORD	MSP	11:10	12:20	40	335	196
	12	C4	MSP	ORD	13:00	14:10	40	335	200
	13	C3	ORD	DCA	16:00	17:40	70	610	154

time slots and/or the airline policy. Scheduled flights of crew teams C1–C4 are 1-2-3-4-9, 5, 6-7-8-13, and 10-11-12, respectively. The aircraft with tail numbers N322AA and N345AA have a seat capacity of 180, while the seat capacity of N5FCAA is set to 210.

The original routing of N322AA is 1-2-3-4-5. However, it may be rerouted through many alternative paths to reach DCA from ORD. For instance, it may only operate flight 1 in cases of severe disruptions, or follow the path 1-2-5 if flight 3 or 4 is canceled. Moreover, the aircraft may operate the flights scheduled for any other aircraft, i.e., it can follow the path 1-10-11-12-5. On the other hand, only a subset of flight nodes and connections can be used by this entity to construct a feasible path from its origin to its destination. For instance, flight 6 cannot be operated by N322AA since the aircraft is located at ORD at 5:30 and even if it is ferried, it cannot arrive at LAX before the latest departure time of this flight which is 8:00. The part of the proposed network related to N322AA is given in Figure 3. This partial network is able to generate all possible flight paths for the particular entity with an additional external arc (dashed) corresponding to ferrying action. In Section 5.1, we highlight the importance of partial networks

for tractability of the optimization model. To reduce the problem size without sacrificing optimality, we propose to generate partial networks of all entities.

In Figure 4, an example of a partial crew network associated with C3 is illustrated. The original schedule of C3, which is transported from LAX to DCA, is 6-7-8-13. All possible paths such as 6-7-12-13 or 6-13 can be generated through this network with an additional external arc for deadheading. Consider the flight connection arc between flights 7 and 12, which is infeasible in the original schedule. The scheduled arrival time of flight 7 is 13:10 while the scheduled departure time of flight 12 is 13:00. However, there exists a possibility to provide the required connection time between these flights by holding the departure time of flight 12 and speeding up flight 7. Therefore, we include this connection in our solution space as well. Although we have illustrated a single external arc for ferrying and deadheading in Figures 3 and 4, respectively, we note that partial networks include external arcs from source to flight nodes, from flight nodes to flight nodes, and from flight nodes to sink node.

In this example, there exist 13 single-flight itineraries corresponding to each scheduled flight, and

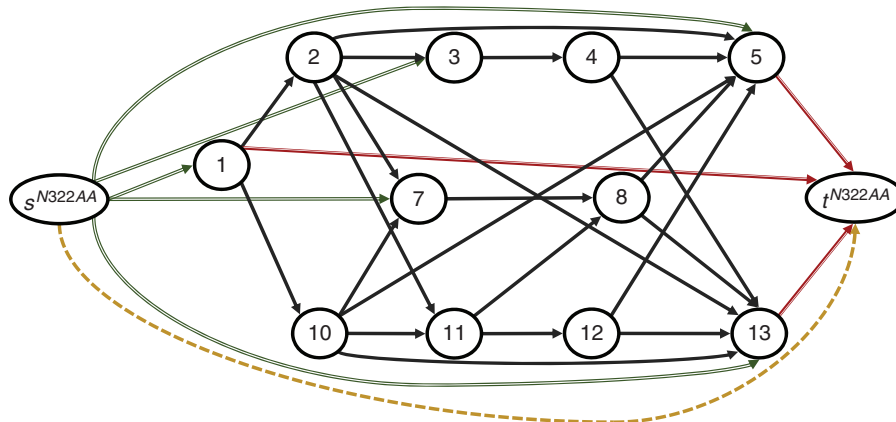
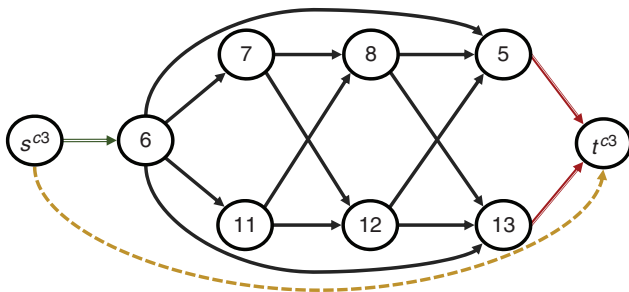
Figure 3. (Color online) Partial Network of Aircraft N322AA

Figure 4. (Color online) Partial Network of Crew Team C3



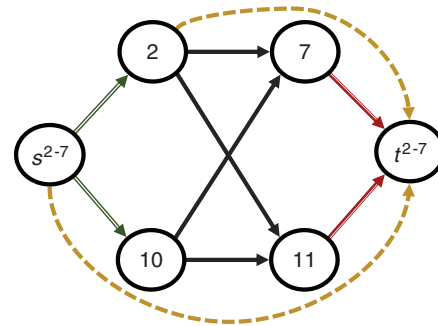
6 two-flight itineraries. All itineraries and number of assigned passengers are tabulated in Table 2. The abbreviation Itin. corresponds to itinerary, and each itinerary is designated by the sequence of flight numbers. An example of a partial passenger network for itinerary 2-7 is illustrated in Figure 5. The external arc from source to sink corresponds to ticket cancellation, while the other one from 2 to the sink corresponds to reallocating this passenger to other means of transportation at ORD.

In the disruption scenario, flight 1 experiences a departure delay of 90 minutes, i.e., it cannot depart before 7:00. This disruption is handled by updating the scheduled departure time of flight 1.

Note that without rerouting options, 90 minutes' delay in flight 1 would propagate through the downstream flights of aircraft N322AA and through those of crew team C1. In the optimal solution of this example, aircraft N322AA follows the path s -1-10-11-12-13- t while the path of aircraft N5FCAA is s -2-3-4-5- t . Since N5FCAA is available at DCA at 7:50, flight 2 does not wait for the arrival of the delayed flight. This swap action mitigates the downstream effect of the delay of flight 1. Since destinations of both flights 13 and 5 are DCA, each aircraft is positioned at their expected locations by the end of the recovery horizon.

Crew rerouting actions are more complicated in this example. Note that the crew team that is originally assigned to flight 2 (C1) also operates flight 1. Since flight 2 does not wait for the arrival of flight 1, flight 2 is assigned to another crew team. In this example, we assume that each crew team can operate each of the flights; however, such technical limitations can be inserted in the proposed approach. In the optimal solution, crew team C1, which is originally located at ORD

Figure 5. (Color online) Partial Network of Passengers in Itinerary 2-7



and needs to arrive at LAX, operates only flight 9. Crew team C2 operates flights 1-10-7-8-13 and reaches its destination (DCA). Flights 6-11-12-5 are operated by C3 with an origin-destination pair LAX-DCA. Finally, flights 2-3-4 are operated by C4. Note that C4 is available in DCA at 7:50, and therefore, flight 2 is not delayed. Also note that a delay can still propagate through the arc 1-10 that is used by crew team C2. Without speeding up, flight 1 would arrive at 8:40. Therefore, C2 would be ready for flight 10 at 9:10 because of minimum connection time requirements, while the scheduled departure time of flight 10 is 9:00.

Allowing interfleet reassignments has two consequences. First, the speed capabilities of different aircraft may vary and this affects the maximum amount of compression of flights; consequently, additional fuel costs are incurred because of the speed increases. In this example, we have assumed that each aircraft has similar speed capabilities. In the optimal solution, flight 1 is compressed by seven minutes for both decreasing the arrival delay of this flight and preventing propagation through the connection 1-10. With the given departure delay and seven minutes of compression, flight 1 departs at 7:00 and arrives at 8:33. Then, as a result of the connection 1-10 used by C2, flight 10 with a scheduled departure time of 9:00 departs at 9:03. In the optimal solution, the speed of this flight is also increased so that it arrives on time at 10:40.

Second, the seat capacities of aircraft may be different and interfleet swaps may result in shortages. In this example, shortages may occur in flights 10, 11, 12, and 13 since the seat capacities of these flights are reduced by 30 seats after the swap action. When we analyze passenger assignments, we observe that flights 11 and 12 will have shortages of 16 and 20 seats, respectively. In Figure 5, it can be seen that 10-7 is an alternative path for passengers in itinerary 10-11. However, there are only eight empty seats available in flight 7. Therefore, eight passengers of itinerary 10-11 are rerouted through path 10-7 and arrive at MSP. However, since flight 7 arrives at 13:10, these passengers experience 50 minutes of arrival delay. The

Table 2. Numbers of Passengers in Passenger Itineraries

Itin.	Nb pass	Itin.	Nb pass	Itin.	Nb pass	Itin.	Nb pass
1	126	4	166	7	67	10-11	91
2	51	5	88	8	70	11	105
2-3	53	6	55	8-5	65	12	200
2-7	45	6-7	60	9	139	13	99
3	58	6-13	55	10	79		

remaining eight passengers are transported through an external arc. Finally, 20 passengers of itinerary 12 are assigned to flight 8 with a 60-minute delay.

This example illustrates the complexity of the problem due to the interrelation among entity types and the necessity of an integrated approach. Moreover, we try to illustrate how passengers in an itinerary may be separated to different paths, and how cruise speed control can be integrated with other recovery actions.

3. Mathematical Formulation

The constraints will be constructed in five groups and calculation of cost terms will be explained after the constraints.

3.1. Flow Balance Constraints

The decision variable x_{fg}^r equals one if entity r flows through arc (f, g) , and zero otherwise. Flow balance is satisfied by Equation (2)

$$\sum_{f:(f,g) \in \mathcal{A}} x_{fg}^r - \sum_{h:(g,h) \in \mathcal{A}} x_{gh}^r = D_g^r \quad r \in R, g \in \mathcal{N}. \quad (2)$$

3.2. Node Closure Constraints

To operate a flight, operating entities should be assigned. For instance, a flight may require an aircraft and a crew team to be operated. We define subset $T^{OP} \subseteq T$ as the set of operating entity types and the parameter Req_f^t as the number of entities of type t needed to operate flight f . The decision variable z_f equals one if flight f is canceled (or node f is closed), and zero otherwise. Recall that $F \subset \mathcal{N}$ is the set of flight nodes. Constraint (3) provides that a flight will be canceled if the required number of operating entities does not flow through the corresponding flight node. Constraint (4) guarantees that other entities cannot flow through a closed flight node as well

$$\sum_{r \in R^t} \left(\sum_{g:(f,g) \in \mathcal{A}} x_{fg}^r \right) = (1 - z_f) Req_f^t \quad t \in T^{OP}, f \in F, \quad (3)$$

$$\sum_{g:(f,g) \in \mathcal{A}} x_{fg}^r \leq (1 - z_f) \quad t \in T \setminus T^{OP}, r \in R^t, f \in F. \quad (4)$$

3.3. Flight Time Constraints

The flight time of a flight node depends on the type of assigned aircraft. Moreover, flight time can be reduced to some extent by increasing the speed of the assigned aircraft. Let nonnegative continuous decision variables dt_f and at_f represent the actual departure and arrival time of flight f , respectively, where $dt_f \in [SDT_f, \overline{DT}_f]$ and $at_f \in [\underline{AT}_f, \overline{AT}_f]$. Note that the value of \underline{AT}_f when cruise speed is utilized is less than or equal to that when cruise speed control is not used (recall the discussion on \underline{AT}_f at the end of Section 2.1), resulting in a larger solution space. Finally, let nonnegative continuous variable δt_f be the amount of cruise time compression of flight f , and R^{ac} be the set of aircraft. Then,

the relation between actual departure and arrival time, and compression is constructed with Equation (5)

$$at_f = dt_f + \sum_{r \in R^{ac}} \left(\sum_{g:(f,g) \in \mathcal{A}} x_{fg}^r \right) FT_f^r - \delta t_f \quad f \in F. \quad (5)$$

Note that $\delta t_f \geq 0$ means that the proposed model allows speed increases but not speed decreases. Although it is not very likely, we would like to note that reducing the cruise speed may be advantageous in certain cases of multiple airport closures.

3.4. Arc Feasibility Constraints

We have four constraints to construct arc feasibility such that each corresponds to a different operational rule.

3.4.1. Arcs Emanating from Source Nodes. These arcs end in flight nodes that may be assigned to an entity as its first flight in the recovered schedule. An entity will use one of these arcs and reach its first flight node, say f_{first} . In this case, f_{first} needs to wait for the ready time of this entity to depart. Therefore, we need a constraint to ensure that the entity is available at the departure time of its first flight. However, only a subset of these arcs are critical for feasibility. They are defined as the set of departure-critical arcs, $DC^r = \{(s^r, g) \in \mathcal{A} : SDT_g < \underline{DT}_r\}$, and the constraint for each entity r is defined over DC^r in (6)

$$dt_g \geq \underline{DT}_r x_{s^r g}^r \quad r \in R, (s^r, g) \in DC^r. \quad (6)$$

3.4.2. Arcs Incoming to Sink Nodes. Similarly, the last flight assigned to entity r cannot arrive later than the latest arrival time of the entity, \overline{AT}^r , to catch up with the original schedule. Constraint (7) is limited to the arrival-critical arcs for entity r , $AC^r = \{(f, t^r) \in \mathcal{A} : \overline{AT}_f > \overline{AT}^r\}$

$$at_f \leq \overline{AT}_f + [\overline{AT}^r - \overline{AT}_f] x_{f t^r}^r \quad r \in R, (f, t^r) \in AC^r. \quad (7)$$

3.4.3. Intermediate Arcs. Intermediate arcs consist of arcs between two flight nodes, and arcs between a flight node and a must-node. If there is a positive flow of entity r between nodes f and g , minimum connection time, CT_{fg}^r , should be provided between these flights. A set of connection-critical arcs for entity r is defined as $CC^r = \{(f, g) \in \mathcal{A} : f, g \in F \cup M, \overline{AT}_f + CT_{fg}^r > SDT_g\}$, and the connection time rule is modeled with Constraint (8)

$$dt_g \geq at_f + CT_{fg}^r x_{fg}^r - \overline{AT}_f (1 - x_{fg}^r) \quad r \in R, (f, g) \in CC^r. \quad (8)$$

Note that when entity r does not use the connection-critical arc (f, g) , this constraint is relaxed, since it reduces to inequality $dt_g \geq at_f - \overline{AT}_f$, the right-hand side of which is always nonpositive ($at_f \in [\underline{AT}_f, \overline{AT}_f]$).

3.4.4. Arcs Emanating from or Incoming to Must-Nodes. Recall that must-nodes represent restrictions of entities. Therefore, entities with such restrictions should visit these nodes as formulated in Constraint (9)

$$\sum_{g:(m,g) \in \mathcal{A}} x_{mg}^r = 1 \quad r \in R, m \in M^r. \quad (9)$$

Constraint (9) is associated with scheduled restrictions, and hence, assumes that must-nodes have rigid locations and periods. However, in cases of disruptions, there may be multiple maintenance stations available. In these situations, this assumption can be relaxed. For instance, consider aircraft r that is required to visit one of the available stations during the recovery horizon. In this case, a must-node can be included in the network corresponding to each maintenance station. Letting \bar{M}^r be the set of these must-nodes, this requirement for aircraft r can be modeled by changing constraint (9) by $\sum_{g:(m,g) \in \mathcal{A}} x_{mg}^r = 1, m \in \bar{M}^r$.

3.5. Aircraft Properties

Some properties of flights depend on the type of assigned aircraft if interfleet aircraft-flight assignments are allowed. Otherwise, these properties would be constant. The first such property is the seat capacity. Let $SCAP^r$ be the seat capacity of aircraft $r \in R^{ac}$. The left-hand side of constraint (10) is the number of passengers assigned to flight f . This number is limited by the seat capacity of the assigned aircraft (right-hand side)

$$\sum_{r \in R^{ps}} \sum_{g:(f,g) \in \mathcal{A}} x_{fg}^r \leq \sum_{r \in R^{ac}} \sum_{g:(f,g) \in \mathcal{A}} x_{fg}^r SCAP^r \quad f \in F. \quad (10)$$

The second property is the limitation on cruise speed. Each aircraft type may speed up to different extents for a particular flight. Maximum cruise speed can be determined by technological constraints or airline policy. This limit can be expressed with an upper bound on cruise speed or equivalently on cruise time compression. We define ΔT_f^r to be the maximum amount of decrease in cruise time of f if it is operated by aircraft r . The cruise time compression variable is bounded by constraint (11)

$$\delta t_f \leq \sum_{r \in R^{ac}} \sum_{g:(f,g) \in \mathcal{A}} x_{fg}^r \Delta T_f^r \quad f \in F. \quad (11)$$

3.6. Aircraft and Crew Compatibility

Crew members cannot operate all types of aircraft in the fleet. We define $R^{ac}(r) \subseteq R^{ac}$ as the set of aircraft that crew member $r \in R^{cr}$ is eligible to operate. Constraint (12) guarantees that the aircraft and crew members assigned to all flights are compatible

$$x_{fg}^r \leq \sum_{s \in R^{ac}(r)} x_{fg}^s \quad (f, g) \in \mathcal{A}, r \in R^{cr}. \quad (12)$$

3.7. External Arc Costs

We define $tc^{[e]}$ to be the total cost of flow on external arcs. Recall that $tc^{[e]}$ represents the sum of costs of actions such as ferrying aircraft, deadheading crew members, ticket cancellations and allocating passengers to other means of transportation, and ticket cancellation. Let $C_e^{[e]}$ be the cost of unit flow on arc e . Then, this cost term is evaluated in (13)

$$tc^{[e]} = \sum_{r \in R} \sum_{e \in E^r} C_e^{[e]} x_e^r. \quad (13)$$

3.8. Flight Cancellation Costs

Let $C_f^{[c]}$ be the flight cancellation cost of flight f . The total flight cancellation cost of the solution, $tc^{[c]}$ is evaluated by (14)

$$tc^{[c]} = \sum_{f \in F} C_f^{[c]} z_f. \quad (14)$$

Flight cancellation results in ticket cancellations or rebooking of passengers. Moreover, it makes the scheduled routings of at least one aircraft and one crew member infeasible. Therefore, the airline may need to cancel other downstream flights or relocate entities. These consequences are already modeled in the proposed formulation as recovery actions. However, the cost of canceling a flight is beyond these direct costs. For instance, it results in passenger inconvenience and a great disturbance in service quality. Moreover, it increases the airline's cancellation rate, which affects passengers' choices. Therefore, $C_f^{[c]}$ should correspond to these indirect costs.

3.9. Additional Fuel Costs

An aircraft is most fuel efficient at its maximum range cruise (MRC) speeds. Fuel consumption is convex and strictly increasing at cruise speeds greater than the MRC speed. However, airlines may still operate their flights with higher speeds because of time and scheduling considerations. We refer the users to the technical reports (Airbus 2004 and Boeing 2007) for detailed analysis on the trade-off between the variable fuel and time related costs depending on cruise speed and time. Considering downstream effects of disruptions and recovery actions on all types of entities, we have already modeled time related costs without isolating the decision to a single flight. Therefore, we require an expression for calculating the fuel consumption of flights to model the trade-off between disruption and recovery costs with the increased fuel cost in the airline recovery context. We integrate the approach proposed by Aktürk, Atamtürk, and Gürel (2014) in our proposed network representation. Based on the fuel flow model developed by the BADA project of EUROCONTROL, the air traffic management organization of Europe (EUROCONTROL 2009), Aktürk, Atamtürk, and Gürel (2014) formulate the total fuel consumption

as a function of speed. Let $fc_{fr}(v_{fr})$ equal the fuel consumption (kg) of flight f if it is operated by aircraft r at cruise speed v_{fr} (km/min) in the recovered schedule, and equal zero otherwise. Nonnegative continuous decision variable v_{fr} equals zero if flight f is not assigned to aircraft r , or takes a value between V_{fr} and \bar{V}_{fr} , which corresponds to the cruise speed of aircraft r for flight f and the maximum cruise speed of aircraft r for flight f , respectively. Finally, let y_{fr} be equal to one if flight f is assigned by aircraft r , and zero otherwise. Then, fc_{fr} can be calculated as follows:

$$fc_{fr}(v_{fr}) = \begin{cases} d_f \left(c_{r1} v_{fr}^2 + c_{r2} v_{fr} + \frac{c_{r3}}{v_{fr}^2} + \frac{c_{r4}}{v_{fr}^3} \right) & \text{if } y_{fr} = 1 \\ 0 & \text{otherwise,} \end{cases} \quad (15)$$

where d_f is the distance flown at the cruise stage of flight f , and parameters $c_{ri}, i = 1, \dots, 4$ depend on several factors such as aircraft specific drag and fuel consumption coefficients, air density at a given altitude, and gravitational acceleration. These parameters can be obtained from the BADA user manual (EUROCONTROL 2012). Then, scheduled fuel consumption of flight f is expressed as $FC_f = fc_{fr_f}(V_{fr_f})$, where r_f is the aircraft that is originally scheduled to operate flight f .

The integration of the proposed network representation and the fuel consumption function proposed by Aktürk, Atamtürk, and Gürel (2014) is through the variable δt_f and constraint (5). Assuming that the distance flown at cruise stage, d_f , is fixed, the cruise time of flight f if assigned to aircraft r can be expressed as $crt_{fr} = d_f/v_{fr}$. Using this relation, the scheduled cruise time of flight f can be expressed as $CRT_f = d_f/V_{fr_f}$. Note that δt_f equals the difference between CRT_f and $crt_{fr'}$, where r' is the aircraft operating flight f in the recovered schedules. These relations and the additional fuel cost of the recovery actions can be formulated with the following constraints:

$$y_{fr} = \sum_{g:(f,g) \in A} x_{fg}^r \quad f \in F, r \in R^{ac}, \quad (16)$$

$$y_{fr} V_{fr} \leq v_{fr} \leq y_{fr} \bar{V}_{fr} \quad f \in F, r \in R^{ac}, \quad (17)$$

$$crt_{fr} \geq 0 \quad f \in F, r \in R^{ac}, \quad (18)$$

$$crt_{fr} v_{fr} = d_f y_{fr} \quad f \in F, r \in R^{ac}, \quad (19)$$

$$\delta t_f = CRT_f - \sum_{r \in R^{ac}} crt_{fr} \quad f \in F, \quad (20)$$

$$fc_{fr} = y_{fr} d_f \left(c_{r1} v_{fr}^2 + c_{r2} v_{fr} + \frac{c_{r3}}{v_{fr}^2} + \frac{c_{r4}}{v_{fr}^3} \right) \quad f \in F, r \in R^{ac}, \quad (21)$$

$$tc^{[f]} = C^{[f]} \sum_f \left(\sum_{r \in R^{ac}} fc_{fr} - FC_f \right) \quad f \in F, r \in R^{ac}, \quad (22)$$

where $C^{[f]}$ is the jet fuel price per kg. The conic quadratic reformulation scheme to handle nonlinearity in constraints (19) and (21) will be discussed in Section 4.

3.10. Passenger Delay Costs

Passenger delay cost includes cost of goodwill loss, and hence, is difficult to calculate in practice. A straightforward calculation method used in many studies is to use a continuous linear delay cost function by utilizing delay cost per passenger per minute. On the other hand, there is also a belief that the relation between goodwill loss and the amount of delay is nonlinear; and hence, a piecewise linear cost function would be more appropriate. As a result of the complexity of the problem, approximate delay costs are utilized in the literature. In this study, we model and experiment approximate and exact delay cost calculation methods for both linear and piecewise linear functions.

3.10.1. Linear Function with Flight Delay Approximation.

Passengers may arrive to their destinations through a set of possible alternative flights as a result of rerouting decisions. Therefore, each possible final flight for a passenger should be investigated to calculate the actual delay, which increases complexity. A common approximation method is to use flight delay instead of using actual delay of individuals. Number of passengers that arrive at their destinations through flight f in the original schedule is designated by N_f^{arr} . Letting Nb^r be the number of passengers in itinerary r , N_f^{arr} is calculated as follows:

$$N_f^{\text{arr}} = \sum_{r \in R^{ps}: Des^r = Des_f} Nb^r.$$

Total passenger delay cost, $tc^{[pd]}$, is approximated with constraints (23) and (24), where the decision variable $delay_f$ is the arrival delay of flight f and $C_f^{[pd]}$ is the per minute delay cost of a passenger whose last scheduled flight is f

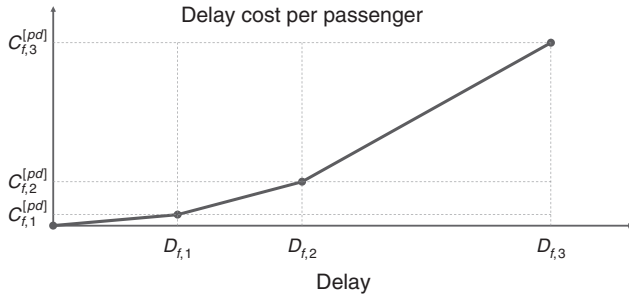
$$delay_f \geq at_f - SAT_f \quad f \in F, \quad (23)$$

$$tc^{[pd]} = \sum_{f \in F} N_f^{\text{arr}} C_f^{[pd]} delay_f. \quad (24)$$

3.10.2. Piecewise Linear Function with Flight Delay Approximation.

In this method, a convex piecewise linear delay cost function is used instead of a linear function. An example of a function is presented in Figure 6. For flight f , the function is defined by delay points $D_{f,i}$ ($D_{f,0} = 0$) and corresponding delay costs $C_{f,i}^{[pd]}$ ($C_{f,0}^{[pd]} = 0$), where I_f is the number of points that the function changes its slope. Let continuous decision variable $delay_f^i$ be defined over $[0,1]$ for each interval i

Figure 6. A Convex Piecewise Linear Passenger Delay Cost Function



of flight f . Total passenger delay cost is approximated with constraints (25)–(27)

$$\sum_{i=1}^{I_f} (D_{f,i} - D_{f,i-1}) \text{delay}_f^i \geq at_f - SAT_f \quad f \in F, \quad (25)$$

$$\text{delay}_f^i \geq \text{delay}_f^{i+1} \quad f \in F, i = 1, \dots, I_f - 1, \quad (26)$$

$$tc^{[pd]} = \sum_{f \in F} N_f^{arr} \sum_{i=1}^{I_f} (C_{f,i}^{[pd]} - C_{f,i-1}^{[pd]}) \text{delay}_f^i. \quad (27)$$

3.10.3. Linear Function with Actual Passenger Delay.

In the last two methods, we propose actual passenger delay cost formulations. These formulations consider both rerouting decisions of passengers and realized arrival times of flights to evaluate exact passenger delays. Let $C_r^{[pd]}$ be the per minute delay cost of passenger r and the decision variable delay_r be the realized delay of this passenger. Then, the total linear delay cost of passengers with actual delays is calculated with constraints (28) and (29), where SAT^r is the scheduled arrival time of passenger r

$$\text{delay}_r \geq at_f - SAT^r - (\overline{AT}_f - SAT^r)(1 - x_{ftr}^r) \quad r \in R^{ps}, f \in F \ni (f, t^r) \in \mathcal{A}, \quad (28)$$

$$tc^{[pd]} = \sum_{r \in R^{ps}} C_r^{[pd]} \text{delay}_r. \quad (29)$$

3.10.4. Piecewise Linear Function with Actual Passenger Delay.

A piecewise convex linear delay cost function can be defined for each passenger in a similar manner. Let $D_{r,i}$ be the delay points that the function changes its slope ($D_{r,0} = 0$) and $C_{r,i}^{[pd]}$ be the corresponding delay costs ($C_{r,0}^{[pd]} = 0$) where there are I_r such points for passenger r . The continuous decision variable $\text{delay}_r^i \in [0, 1]$ is defined for each interval i and the total passenger delay cost is calculated with constraints (30)–(32)

$$\begin{aligned} & \sum_{i=1}^{I_r} (D_{r,i} - D_{r,i-1}) \text{delay}_r^i \\ & \geq at_f - SAT^r - (\overline{AT}_f - SAT^r)(1 - x_{ftr}^r) \\ & \quad r \in R^{ps}, f \in F \ni (f, t^r) \in \mathcal{A}, \quad (30) \end{aligned}$$

$$\text{delay}_r^{i+1} \geq \text{delay}_r^i \quad r \in R^{ps}, i = 1, \dots, I_r - 1, \quad (31)$$

$$tc^{[pd]} = \sum_{r \in R^{ps}} \sum_{i=1}^{I_r} (C_{r,i}^{[pd]} - C_{r,i-1}^{[pd]}) \text{delay}_r^i. \quad (32)$$

3.11. Original Flight Paths

The preceding constraints do not have information about the scheduled flight paths of entities. Since there exists a large number of flight paths that an entity can follow through the flight network, entities may be rerouted without causing flight cancellations or arrival delays even if these rerouting decisions are not related to the disruption. However, it is desirable for entities to use their scheduled routings unless rerouting helps mitigate the disturbances of the disruptions. This desired behavior can be guaranteed by adding a small negative cost to the objective function for each entity flowing through each of its scheduled arcs. Let $SA^r \subseteq A$ be the set of scheduled arcs for entity r , and $\epsilon^r < 0$ be the coefficient corresponding to negative of the benefit of following the original schedule. The absolute value of this coefficient needs to be sufficiently small to not affect the optimal recovery decisions. On the other hand, any strictly negative cost coefficient will guarantee that all entities will follow their original schedules if there is no disruption, since with the described arc cost assignment, this problem has a unique optimal solution. Therefore, ϵ_r can comfortably be assigned to \$1 for aircraft and crew members and to \$0.01 for passengers. The total benefit of following the original schedules, $tc^{[op]}$, is calculated as follows:

$$tc^{[op]} = \sum_{t \in T} \sum_{r \in R^t} \sum_{(f,g) \in SA^r} \epsilon^r x_{fg}^r. \quad (33)$$

3.12. Mathematical Model

The complete mathematical formulation is given next

$$\begin{aligned} & \text{Minimize} \quad tc^{[e]} + tc^{[c]} + tc^{[f]} + tc^{[pd]} + tc^{[op]} \\ & \text{subject to} \quad (2)–(14), (16)–(22), \\ & \quad (23)–(24), \text{ or } (25)–(27), \text{ or } (28)–(29), \\ & \quad \text{or } (30)–(32). \end{aligned}$$

The proposed formulation is a mixed integer nonlinear programming model where the nonlinearity is a result of constraints (19) and (21) that are used to model the trade-off between reduced disruption and recovery costs by reducing cruise times, and additional fuel cost. In Section 4, we will reformulate the problem as a conic quadratic mixed integer programming problem.

4. Conic Quadratic Reformulation

In the conic quadratic reformulation of the proposed mixed integer nonlinear programming model, we follow the propositions of Aktürk, Atamtürk, and Gürel (2014). For the sake of simplicity, we drop aircraft and flight indices from nonlinear constraints (19)

and (21). Constraint (19) evaluates fuel consumption with respect to cruise speed and flight-aircraft assignment variables. This equality is equivalent to the fuel consumption function given in (15) since y_{fr} can only take values zero or one. In Proposition 1 in Aktürk, Atamtürk, and Gürel (2014), the authors show that the convex hull of the epigraph of this function $E_F = \{(v, t) \in R^2: fc(v) \leq t\}$ can be expressed as

$$\begin{aligned} t &\geq d(c_1\tau_1 + c_2v + c_3\tau_3 + c_4\tau_4), \\ v^2 &\leq \tau_1y, \\ y^4 &\leq \tau_3v^2y, \\ y^4 &\leq \tau_4v^3, \end{aligned}$$

in the constraint set, and the last three inequalities can be represented by conic inequalities. With the findings of this proposition, constraint (19) can be substituted by the following constraints:

$$\begin{aligned} fc &\geq d(c_1\tau_1 + c_2v + c_3\tau_3 + c_4\tau_4), & (34) \\ v^2 &\leq \tau_1y, & (35) \\ y^2 &\leq wv, & (36) \\ w^2 &\leq \tau_3y, & (37) \\ w^2 &\leq \tau_4v, & (38) \end{aligned}$$

where τ_1, τ_3, τ_4 , and v are nonnegative continuous variables. Note that when $y = 1$, these constraints are reduced to $\tau_1 \geq v^2$, $\tau_3 \geq w^2 \geq v^{-2}$, and $\tau_4 \geq w^2v^{-1} \geq v^{-3}$. Since fc is increasing in each τ_1, τ_2 , and τ_3 , and the objective function is increasing in fc , the following will hold for every optimal solution to the proposed model: $\tau_1 = v^2$, $\tau_3 = v^{-2}$, $\tau_4 = v^{-4}$, and $fc = d(c_1\tau_1 + c_2v + c_3\tau_3 + c_4\tau_4)$. Therefore, fc corresponds to the fuel consumption defined in (15) for $y = 1$. On the other hand, when $y = 0$, the value of v will be forced to be zero by constraint (17), while the variables τ_1, τ_3 , and τ_4 will be free to take any nonnegative value. As a result of constraint (34) and the objective function, the values of these variables will be set to zero as well at optimality. Therefore, the fuel consumption function (15) holds again for $y = 0$.

Constraint (21) constructs the relation between the cruise time and the cruise speed of a flight. The following inequality presented in Proposition 2 in Aktürk, Atamtürk, and Gürel (2014):

$$dy \leq v \cdot crt,$$

holds for every optimal solution to the proposed mixed integer nonlinear programming model. When $y = 1$, since fuel consumption is increasing in v for $v \geq \bar{V}$, $v = d/crt$ holds. On the other hand, when $y = 0$, the value of v will be forced to be zero by constraint (17), and the equality holds again. Moreover, exploiting the fact that y can only take values zero or one, this inequality

can be represented by the following conic quadratic inequality:

$$dy^2 \leq v \cdot crt. \quad (39)$$

Therefore, the mixed integer nonlinear programming model presented in Section 3 can be reformulated as a conic quadratic mixed integer programming (CQMIP) problem. The modified model with a linear objective function, and linear and conic quadratic constraints can be handled by fast algorithms of commercial CQMIP solvers.

5. Preprocessing

As mentioned previously, flight networks are large, and quick solutions are required in disruption management. Therefore, it is important to eliminate unnecessary variables and constraints without sacrificing optimality. In this section, we describe two preprocessing methods. In the partial network approach, we propose an algorithm to obtain the partial networks of entities. The partial network of an entity is a subset of the complete network that excludes nodes and arcs that will not be visited by the entity. Therefore, we can reduce the number of variables and constraints significantly. In the second method, we propose an entity aggregation rule without losing any information and sacrificing optimality.

5.1. Partial Networks

The partial network approach aims to identify and eliminate infeasible recovery actions from the solution space. In particular, we aim to isolate the related portion of the network, named as the partial network, for each entity. The partial network of an entity is able to generate all possible recovery actions, while it does not include any flight node or connection arc that will not be used by the entity. Therefore, partial networks provide a compact representation of the problem. The interdependencies among the recovery actions of different entity types are maintained through the common flight nodes.

A partial network of entity r , $G^r = (\mathcal{N}^r, \mathcal{A}^r)$, is defined as the subset of the complete network, $G = (\mathcal{N}, \mathcal{A})$, which includes the source and sink nodes (s^r, t^r), and must-visit nodes (M^r) of the entity together with the flight nodes that it can visit in a feasible solution. The idea is to reduce the number of flow variables using the fact that not all arcs can be used to transport a particular entity from its origin to its destination. For instance, consider an entity whose destination is LAX and its latest arrival time is 17:00 (GMT). Then, a flight from ORD to DFW with an earliest arrival time of 17:00 (GMT) is irrelevant to this entity, as well as all arcs incoming to and outgoing from this node. We propose the PNGA for efficiently generating the partial network of an entity which is capable of generating all feasible

paths that can be used by the entity to reach its destination, and does not include any flight nodes that would not be visited.

The steps of the algorithm for entity r are given in Algorithm 1. The algorithm starts with an empty network. The partial network for r is obtained in line 3 by calling GeneratePath subprocedure with $N^{\text{temp}} = \{s^r\}$, where s^r defines the initial state of the entity. We use a temporary path, termed as N^{temp} , that is updated throughout the algorithm. Finally, external arcs related to entity r are included in line 4 and the partial network is returned.

Algorithm 1 (Partial network generation algorithm)

```

1: Procedure PNGA( $r$ )
2:   Initialization:  $\mathcal{N}^r = \emptyset, A^r = \emptyset, N^{\text{temp}} = \{s^r\}$ 
3:    $G^r = (\mathcal{N}^r, A^r) \leftarrow \text{GeneratePath}(N^{\text{temp}})$ 
4:    $A^r \leftarrow A^r \cup E^r$ 
5:   return  $G^r = (\mathcal{N}^r, A^r)$ 
6: end procedure
7: Procedure GENERATEPATH( $N^{\text{temp}}$ )
8:    $f \leftarrow$  last element of  $N^{\text{temp}}$ 
9:   if  $Des_f = Des^r$  and return to destination is
       not allowed then
10:    exit procedure
11:  else
12:     $N^{\text{next}} \leftarrow \{g \in F: Ori_g = Des_f \text{ and } SDT_g \geq AT_f + CT_{fg}^r\}$ 
13:    for each  $g \in N^{\text{next}}$  do
14:       $N^{\text{temp}} \leftarrow N^{\text{temp}} \cup \{g\}$ 
15:      if  $g \in \mathcal{N}^r$  then
16:        Insert( $N^{\text{temp}}$ )
17:      else
18:        if  $Des_f = Des^r$  then
19:          Insert( $N^{\text{temp}} \cup \{t^r\}$ )
20:        end if
21:        GeneratePath( $N^{\text{temp}}$ )
22:      end if
23:    end for
24:  end if
25: end procedure
26: Procedure INSERT( $N^{\text{temp}}$ )
27:    $\mathcal{N}^r \leftarrow \mathcal{N}^r \cup N^{\text{temp}}$ 
28:   Let  $f_i$  be the  $i$ th element of  $N^{\text{temp}}$ 
29:   for  $i = 1$  to  $|N^{\text{temp}}| - 1$  do
30:      $A^r \leftarrow A^r \cup \{f_i, f_{i+1}\}$ 
31:   end for
32: end procedure
    
```

GeneratePath subprocedure starts with a temporary path, N^{temp} , and tries to connect a flight to the final flight of this path. Subprocedure stops at line 10 if the desired destination is reached and return to destination is not allowed. Returning to destination is not allowed for passengers, while it is allowed for aircraft and crew

members. For instance, the path ORD-DCA-DFW-DCA would not be realistic if the entity in consideration is a passenger that will be transported from ORD to DCA. If the destination has not been reached yet (or the entity may leave and return to its destination), N^{next} is created in line 12, which is the set of candidate flights that can be connected to the last flight of the temporary path. The stopping condition in line 15 is crucial for the efficiency of the algorithm. If a flight is already inserted in the partial network of the entity ($g \in \mathcal{N}^r$), we are sure that all subpaths emanating from this node to the sink have already been discovered. Therefore, N^{temp} can be inserted without any further search.

Insert subprocedure inserts the nodes and arcs in the temporary set N^{temp} into the partial network of the entity. Note that this subprocedure is called either in line 16 or in line 19. In the latter one, the temporary path is a complete path from the origin to the destination of the entity. All nodes and arcs in the temporary path are inserted into the partial network. On the other hand, in the prior one, the temporary path is connected to an already inserted node. Since we know that there is a subpath from the already inserted node (g) to the destination, the flights and connections in the temporary path may exist in a feasible path. Therefore, we insert the nodes and arcs of this subpath to the network, as well. Since we do not insert the nodes and arcs of any other path, the generated partial network excludes all nodes and arcs that cannot be visited by the entity through a feasible path. Figures 3–5 are examples of partial networks of a complete network that involves 13 flight nodes.

Let $PA = \{(f, g): Ori_g = Des_f \text{ and } SDT_g \geq AT_f + CT_{fg}^r, \forall f, g \in F\}$ be the set of all potential flight connection arcs. Note that the flight connections in the ordered set N^{temp} are always included in PA because of the definition of N^{next} (line 12). In other words, the algorithm traverses the potential arcs in PA . The condition in line 15 ensures that potential arcs are not visited more than once. Therefore, the worst-case running time of the algorithm is polynomial in the number of potential arcs, i.e., $\mathcal{O}(|PA|)$. Note that $|PA|$ is significantly smaller than $|F|^2$ in airline networks as a result of time and location constraints on the existence of the arcs. For instance, a real airline network having 1,254 flight nodes has 273,372 potential arcs, which is significantly less than $1,254^2$.

The proposed optimization model is extended to the partial network approach by defining x_{fg}^r for $(f, g) \in A^r, r \in R$, and substituting \mathcal{N} and A by \mathcal{N}^r and A^r , respectively, in all constraints. Using partial networks of entities instead of the entire flight network provides a more compact representation, while still being able to generate all feasible solutions (since partial networks are still integrated through the common flight node set). Therefore, the partial network approach helps in

keeping the problem size in reasonable limits. To illustrate the impact of the proposed approach, we use a real airline network mentioned previously. The network contains 1,254 flights operated by 402 aircraft. There exists 55,482 arcs in the entire flight network, while the average number of arcs in the partial aircraft networks is 3,625.58. This means that the number of aircraft flow variables are reduced by a factor of around 15. We observe a similar reduction factor (around 16) for crew flow variables. Partial networks of passengers are much smaller than those of aircraft and crew members. Moreover, the number of passenger itineraries is significantly greater than the number of aircraft and crew members. Therefore, the impact of the partial network approach is much greater. The reduction factor is in tens of thousands for this network. From this analysis, we conclude that the partial network approach significantly reduces the problem size for aircraft-and-crew recovery problems. Furthermore, integration of passenger recovery in real-sized airline networks with the entire flight network representation without using partial networks increases the problem size significantly.

5.2. Entity Aggregation

Each individual (aircraft, crew member, and passenger) is defined as an entity so far. By careful aggregation, the number of entities can be reduced. It is easy to note that individuals of an aggregated entity need to have *exactly the same partial network* to prevent any loss of information. By this observation, we can extend the rule for aggregation of entities as follows:

Aggregation Rule: Individuals with common ready time, latest arrival time, origin, destination, connection time between flights, must-visit nodes, technical properties (such as aircraft speed and seat capacity) and delay cost parameters can be aggregated without sacrificing optimality.

The proposed mathematical formulation is extended to handle entity aggregation by simple modifications. Let r' be defined as the aggregation of Nb^r individual aircraft, crew members, or passengers. In original formulation Nb^r , binary flow variables would be used, while by entity aggregation they may be replaced by a single integer flow variable, i.e., $x_{fg}^{r'} \in \{0, 1, \dots, Nb^{r'}\}$. Let R' be the set of all entity aggregations. Then, the mixed integer nonlinear programming can be modified by the following steps:

- Remove aggregated entities from the model: $R = R \setminus \{r \in R: r \in r', \forall r' \subseteq R'\}$, and set $Nb^r = 1$ for all entities that are not aggregated.
- Add entity aggregations to the model: $R = R \cup R'$.
- Update parameter D_g^r as $-Nb^r$ if $g = s^r$, Nb^r if $g = t^r$, or 0 otherwise, in constraint (2).
- Update constraint (4) by multiplying the right-hand side by Nb^r as follows: $\sum_{g:(f,g) \in \mathcal{A}} x_{fg}^{r'} \leq (1 - z_f) \cdot Nb^r, t \in T \setminus T^{OP}, r \in R^t, f \in F$.

- Substitute one in the right-hand side of constraint (9) by Nb^r .

- For each entity aggregation $r' \in R'$, define binary variables $ff_f^{r'} (g \in DC^{r'})$, $lf_f^{r'} (f \in AC^{r'})$, and $con_{fg}^{r'} ((f, g) \in CC^{r'})$ to equal one if any one of the entities within the aggregated set uses flight g as the first flight, uses flight f as the last flight, and uses flight connection (f, g) , respectively; and equal zero otherwise. Add the following constraints:

$$\begin{aligned} Nb^{r'} ff_f^{r'} &\geq x_{s^r g}^{r'}, & r' \in R', g \in DC^{r'}, \\ Nb^{r'} lf_f^{r'} &\geq x_{f t^r}^{r'}, & r' \in R', g \in AC^{r'}, \\ Nb^{r'} con_{fg}^{r'} &\geq x_{fg}^{r'}, & r' \in R', (f, g) \in CC^{r'}. \end{aligned}$$

Then, substitute flow variable $x_{s^r g}^{r'}$, $x_{f t^r}^{r'}$, and $x_{fg}^{r'}$ in constraints (6)–(8) by $ff_f^{r'}$, $lf_f^{r'}$, and $con_{fg}^{r'}$, respectively.

Note that Req_f^t equals one if t corresponds to aircraft and equals the number of required crew members if it corresponds to crew members. Therefore, constraint (3) will still be valid. In other words, if two or more aircraft are aggregated, they will flow through different flight paths since constraint (3) ensures that at most one aircraft can flow through a flight node. This constraint also maintains the validity of constraints (5), (10), and (11). No further changes are required for the conic quadratic reformulation.

It can be noted that passengers in an itinerary with common delay cost parameters (in the same fare class) can be aggregated without violating the proposed rule. However, for linear and piecewise cost functions with actual passenger delay (Sections 3.10.3 and 3.10.4), passengers should not be aggregated since the flow variables are used as binary assignment variables in constraints (28) and (30), respectively. Finally, we need to note that individuals, which are aggregated, can still be transported through different paths.

6. Controlling Problem Size

Solving the integrated recovery problem in real time is challenging because of the complexity of the problem and the size of the airline networks. We approach this problem in two ways. First, we propose flight delay approximations in Section 3.10 to reduce the problem complexity. Second, we propose to limit the scope of recovery. Note that there exists a huge number of rerouting alternatives of entities in the solution space when dealing with large airline networks. Interdependencies among recovery actions of different entities allow us to define a measure for the likelihood of the connection arcs to be used in the optimal solution. The proposed partial network representation allows for an efficient method to obtain these measures, and hence, to identify the connection arcs that are not likely to be utilized. We propose the PSCA to control the problem

size. The algorithm iteratively reduces the size of partial networks by eliminating connection arcs that are not likely to be used in the optimal solution. Since the number of flow variables in the proposed optimization model is equal to the number of arcs in the partial networks of entities, the algorithm is able to control the problem size accurately.

Recall that $G = (\mathcal{N}, A)$ corresponds to the entire flight network, while $G^r = (\mathcal{N}^r, A^r)$ represents the partial network of entity r . We use the following additional notation in PSCA:

- \mathcal{D} : Set of disrupted flights.
- SF^r : Set of flights that are originally assigned to entity r .
- SA^r : Set of arcs that are used in the original flight path of entity r , $SA^r \subseteq A^r$.
- SA_t : Set of all arcs that are used in the original flight path of an entity of type t , i.e., $SA_t = \bigcup_{r \in R_t} SA^r$.
- $\bar{A} = \{(f, g): (f, g) \in A \text{ or } (g, f) \in A\}$.
- $\bar{SA}_t = \{(f, g): (f, g) \in SA_t \text{ or } (g, f) \in SA_t\}$.
- $G^{r*} = (\mathcal{N}^{r*}, A^{r*})$: Limited partial network of entity r where $\mathcal{N}^{r*} \subseteq \mathcal{N}^r$ and $A^{r*} \subseteq A^r$.
- B : Maximum number of arcs to be included in the limited partial networks.
- w_{at} : Arc length of arc $a \in \bar{A}$ for entity type t ; w_{at} equals 0 if $a \in \bar{SA}_t$; and 1 otherwise.
- $u_{ft}(d)$: Shortest path distance from flight node f to disrupted node $d \in \mathcal{D}$ through network $\bar{G} = (\mathcal{N}, \bar{A})$ with respect to arc lengths w_{at} .
- u_{ft}^* : Shortest path distance from flight node f to the nearest disrupted node in \mathcal{D} through network $\bar{G} = (\mathcal{N}, \bar{A})$ with respect to arc lengths w_{at} .
- v_{fg}^r : A score for $(f, g) \in A^r$ related with the likelihood of being used in the optimal recovery. Arcs with smaller scores correspond to more preferred arcs. Scores are calculated as follows:

$$v_{fg}^r = \begin{cases} (u_{ft}^* + u_{gt}^*)/2, & \text{if either } f \text{ or } g \in F \setminus SF^r \\ 0 & \text{otherwise} \end{cases} \quad (f, g) \in A^r, r \in R_t, t \in T. \quad (40)$$

PSCA follows an arc elimination procedure using arc scores to assign a priority to arcs for elimination (arcs with high scores are likely to be eliminated early). Note that by the definition of v_{fg}^r , the scores of arcs emanating from the source node, entering the sink node, or visiting a must-node, as well as those of the scheduled arcs and unscheduled arcs between two scheduled flight nodes are all zero. The output of PSCA

is the set of limited partial networks G^{r*} for all entities. Since partial passenger networks are smaller when compared with those of aircraft and crew members, and since we place a special emphasis on passenger recovery, we do not limit the scope of passenger recovery. In other words, we apply the algorithm to aircraft and crew networks. Steps of the algorithm are presented in Algorithm 2.

Algorithm 2 (Problem size control algorithm)

- 1: **Procedure** PSCA
- 2: **for** each $t \in \{ac, cr\}$ **do**
- 3: **for** each $d \in \mathcal{D}$ **do**
- 4: Use a shortest path algorithm to calculate $u_{ft}(d)$ for each flight node f
- 5: through network $\bar{G} = (\mathcal{N}, \bar{A})$ with respect to arc lengths w_{at} .
- 6: **end for**
- 7: $u_{ft}^* \leftarrow \min_{d \in \mathcal{D}} \{u_{ft}(d)\}$ for each flight node f .
- 8: **for** each $r \in R_t$ **do**
- 9: Update arc scores v_{fg}^r using expression (40).
- 10: $G^{r*} = (\mathcal{N}^{r*}, A^{r*}) \leftarrow G^r = (\mathcal{N}^r, A^r)$
- 11: **while** $|A^{r*}| > B$ **do**
- 12: $a' \leftarrow \arg \max_{a \in A^{r*}} \{v_{fg}^r\}$
- 13: ArcElimination(G^{r*}, a')
- 14: **end while**
- 15: **end for**
- 16: **end for**
- 17: **end procedure**
- 18: **Procedure** ArcElimination(G^{r*}, a')
- 19: $A^{r*} \leftarrow A^{r*} \setminus \{a'\}$.
- 20: Let $a' = (f', g')$.
- 21: **if** $\nexists g \in \mathcal{N}^{r*}$ such that $(f', g) \in A^{r*}$ **then**
- 22: NodeElimination(G^{r*}, f')
- 23: **end if**
- 24: **if** $\nexists f \in \mathcal{N}^{r*}$ such that $(f, g') \in A^{r*}$ **then**
- 25: NodeElimination(G^{r*}, g')
- 26: **end if**
- 27: **end procedure**
- 28: **Procedure** NodeElimination(G^{r*}, f)
- 29: $\mathcal{N}^{r*} \leftarrow \mathcal{N}^{r*} \setminus \{f\}$.
- 30: **for** each arc $a' \in A^{r*}$ emanating from or incoming to f **do**
- 31: ArcElimination(G^{r*}, a')
- 32: **end for**
- 33: **end procedure**

Once the shortest path distances are calculated and the arc scores are updated, the algorithm starts removing arcs from the partial networks until the number of arcs is reduced to the desired value. The arcs to be eliminated are selected with decreasing order of their scores. Arc elimination is carried out by Arc Elimination subprocedure. Once an arc is removed from the network, the subprocedure checks the start and end nodes. Removal of the arc may leave the start node with

no emanating arcs. This means that the sink node of the entity can no longer be reached from this node. Similarly, if the end node is left with no incoming arcs, this node can no longer be reached from the source node of the entity. In either case, the disconnected node is removed from the network. NodeElimination subprocedure removes the disconnected node and recalls ArcElimination subprocedure to remove all arcs incoming or outgoing from this node. There may be several loops between the two subprocedures before going back to the main procedure. This ensures that the stopping condition in line 11 is never checked before the connectivity of the network is maintained. Therefore, all nodes and arcs in the resulting limited partial network of an entity can be used through a flight path (rerouting alternative) from its source to its sink node.

Nonzero scores of arcs are the average of the shortest distances of its end nodes to the nearest disruption nodes, u_{ft}^* . Therefore, the value of u_{ft}^* is critical in generating limited partial networks that are likely to contain optimal recovery decisions. We have the following property for u_{ft}^* :

Property 1. u_{ft}^* is the minimum number of entities of type t , routings of which need to be altered to assign the entity originally assigned to operate flight f to any of the disrupted flights.

By the definition of the arc lengths w_{at} , u_{ft}^* is also equal to the number of unscheduled arcs used in the shortest path from f to the nearest disrupted node. Note that the start and end nodes of an unscheduled arc are assigned to different entities. Therefore, Property 1 states the equivalence between the minimum number of original routings that will be altered by assigning the entity that is originally assigned to flight f to a disrupted flight, and the shortest path distance from f to the nearest disrupted flight using the proposed arc lengths.

In other words, u_{ft}^* is a measure of disturbance of rerouting the scheduled entity of flight f to recover from disruptions on the original schedules. This information is very insightful in identifying good rerouting

options. For instance, let flight f and a disrupted flight d be scheduled for aircraft r . Then, for $t = ac$, the value of u_{ft}^* would be zero. This means that a disrupted flight can be operated by aircraft r without disturbing the scheduled routings of any other aircraft. An aircraft or crew swap may be defined by two nonscheduled connection arcs between the scheduled routings of two entities. If one of the scheduled routings of these two entities includes a disrupted flight, u_{ft}^* will be one for all f in the scheduled routing of the other entity. On the other hand, to assign the aircraft originally assigned to flight f to a disrupted flight when $u_{ft}^* = 5$, at least five other entities of type t which are not directly disrupted need to be rerouted. Therefore, this aircraft is unlikely to help reduce recovery costs.

For illustration, we use a small network as represented in Figure 7(a). Origins and destinations of flights are attached to flight nodes. The flights in this network are operated by three aircraft with scheduled flight paths of 1-2-3-4, 5-6-7-8, and 9-10-11-12. With a disruption at flight node 1, the values of u_{ft}^* are represented in square brackets above the nodes in Figure 7(b). Two iterations of the main procedure of PSCA for the first aircraft is illustrated in Figure 8, where the arc scores are given in curved braces. Figure 8(a) represents the partial network of this entity. In the first iteration an arc with score 2 is removed from the network, which also results in removal of flight nodes 10, 11, and 12 (Figure 8(b)). Assuming the problem size is still large, we iterate once more. Recall that the number of flow variables is equal to the number of arcs in partial networks. Therefore, the total number of arcs at the end of each iteration provides useful information for deciding if the problem can be solved within the required solution time. The network after elimination of the arcs with score 1.5 is presented in Figure 8(c). Note that eliminated recovery actions alter the original schedules of two aircraft, while the remaining ones alter the schedule of at most one other aircraft. From Figures 8(b) and 8(c), it can be observed that the networks at the end of each iteration are connected.

Figure 7. (Color online) Entire Flight Network

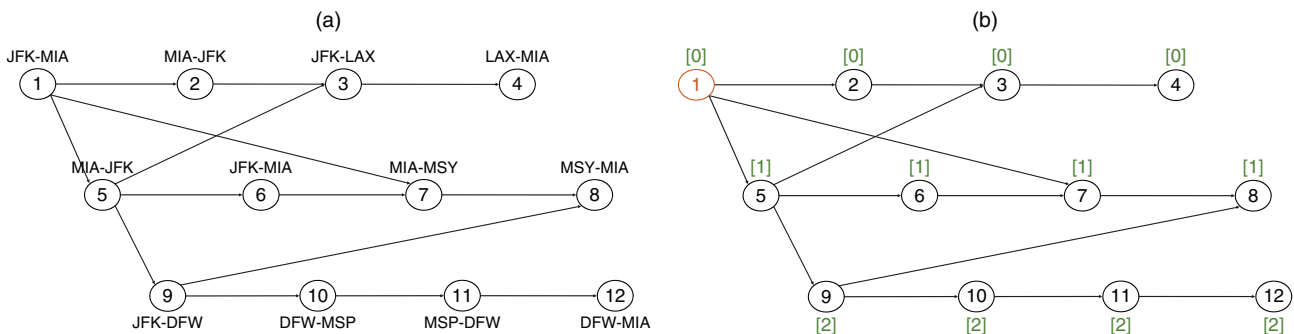


Figure 8. (Color online) PSCA Iterations for the Aircraft with Original Schedule 1-2-3-4

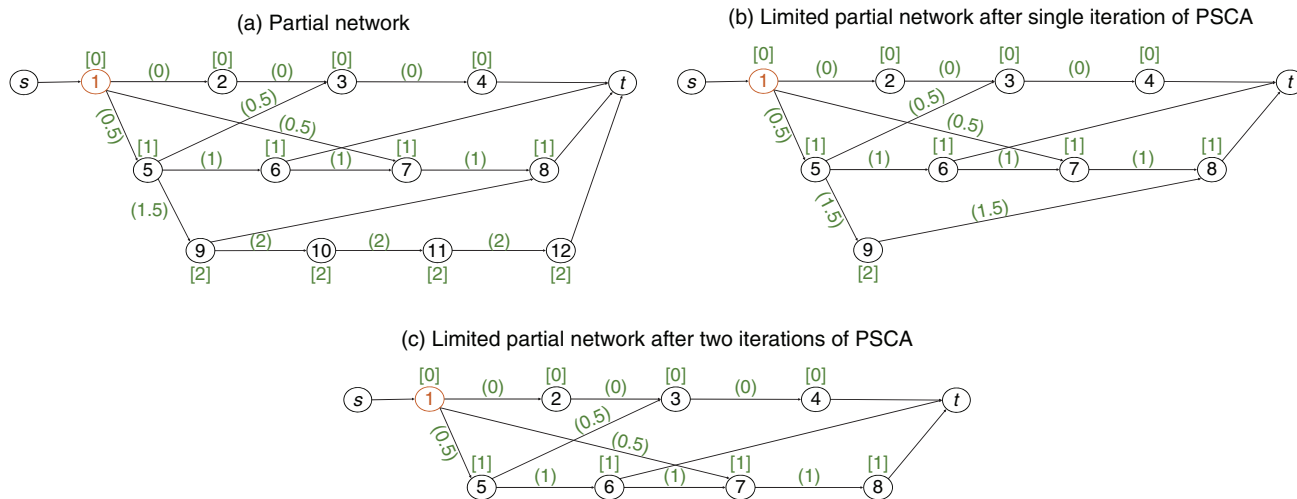
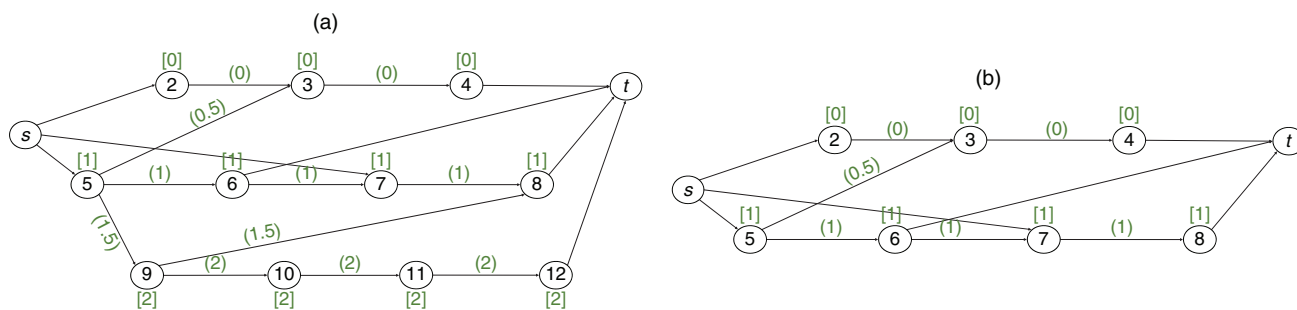


Figure 9. (Color online) PSCA Iterations for the Aircraft with Original Schedule 5-6-7-8



Single iteration of the algorithm for the second aircraft, which is not directly affected from the disruption, is illustrated in Figure 9. Note that this aircraft cannot be assigned to the disrupted flight 1 because of its location. However, it can still be assigned to flights 2, 3, and 4 of the disrupted routing, without altering the routing of any other aircraft. Even though the disrupted flight is not included in the partial network of the aircraft (Figure 9(a)), the algorithm manages to keep these flights while reducing the problem size by eliminating the scheduled flights of the third aircraft (Figure 9(b)). Note that there is no value in swapping the second and third aircraft in this example.

Dijkstra’s algorithm is appropriate for the step described in lines 4 and 5 to obtain the shortest distances from a disruption node to all other nodes. Faster variants are available but with a simple implementation Dijkstra’s algorithm runs in $\mathcal{O}(|\mathcal{N}| \log |\mathcal{N}|)$. In the remaining steps of the algorithm, it is certain that each arc is visited at most once in the worst case. Letting $\bar{a} = \sum_{r \in R^{ac} \cup R^{cr}} |A^r|$, overall complexity of the algorithm is $\mathcal{O}(|\mathcal{D}| |\mathcal{N}| \log |\mathcal{N}| + \bar{a})$.

7. Computational Results

We test the practicality of the proposed network representation, preprocessing methods, and PSCA on a real network of a major U.S. airline in the year 2013. We have also created two subsets of the entire network to test the scalability of the approach. In Section 7.1, we explain parameter settings and methods used to generate problem instances. In the remainder of this section, we discuss the results of our experiments.

7.1. Scenario Generation

We have used several publicly available databases provided by the Bureau of Transportation Statistics (<http://www.transtats.bts.gov/DataIndex.asp>) to generate realistic disruption scenarios. These databases are joined together into a large Oracle database. The first database that we rely on is the Airline On-Time Performance (AOTP) database. AOTP provides dates, scheduled and realized departure, arrival time, flight time, origin, destination, and tail number of flights. In addition, we are able to generate routings of aircraft using tail number information. However, crew routings and passenger itineraries are not publicly available; only

aggregate number of passengers is available. For generating itineraries and estimating the number of passengers using each itinerary, we follow the method proposed by Barnhart, Fearing, and Vaze (2014); while we use an algorithm to randomly generate crew routings from the extracted flight schedules. The second database we use is Schedule B-43 Aircraft Inventory (SB-43). SB-43 provides aircraft inventory information of most airlines. Using tail numbers to join AOTP and SB-43, we can obtain aircraft models and their seat capacities. The airline used in our experimentations uses seven different models in the generated networks. Scheduled maintenances of aircraft is not publicly available as well. However, we have assigned must-nodes for randomly selected aircraft considering their actual flight schedules.

Barnhart, Fearing, and Vaze (2014) provide an excellent guide for estimating itinerary-related data. The approach involves data processing steps for generating itineraries and obtaining aggregate passenger counts. The T-100 Domestic Segment (T-100) database is used to obtain monthly aggregated passenger counts for each carrier segment. A carrier segment (CS) is defined by three fields: carrier, origin, and destination. Since we focus on a single airline, we extract passenger counts of CS's for this carrier. Dropping the carrier field, an example CS can be defined as (ORD-LAX). A carrier route (CR), on the other hand, is defined as a sequence of CS's. A CR is a possible flight path that a passenger can travel from the origin of its first CS to the destination of its last CS. Since a CR does not have information about departure and arrival times, while an itinerary does, a CR may also be considered as sets of itineraries. Note that each itinerary belongs to a single CR. The airline origin and destination survey (DB1B) provides a 10% sample of the number of passengers using each CR. This data is aggregated quarterly. Barnhart, Fearing, and Vaze (2014) propose two data processing steps. In the first step, potential itineraries are generated using the flight route information provided in DB1B. In the second step, the number of passengers belonging to each CR is estimated by combining the information provided by T-100 and DB1B. The approach involves scaling steps because of the difference in aggregation periods of these two databases. Following these steps, we have generated potential itineraries and an aggregate number of passengers for each CR. However, a passenger belonging to a particular CR may have used any of the large number of potential itineraries belonging to that CR. Discrete choice methodology is appropriate for estimating the choice behavior of passengers. Barnhart, Fearing, and Vaze (2014) propose a discrete choice model with a detailed utility function depending on several important factors such as day of week, connection time, seat capacity, etc. The authors use a multinomial logit (MNL) model and

estimate the parameters of the factors used in the utility function. Using this discrete choice model (MNL) and a utility function, we estimate the number of passengers using each itinerary.

A flight delay cost has a complex structure involving a variety of cost components including cost to passengers, cost to airlines, cost of lost demand, etc. Ball et al. (2010) present an analysis on these components, as well as flight delays' indirect impact on the U.S. economy. The authors consider a broader consideration of relevant costs than conventional methods, and use innovative methodologies to assess these costs, one of which is passenger delay cost. It is reported that in 2007, around 487 million passengers experienced an average delay of 31 minutes. The methodology proposed by the authors calculates the total passenger delay cost as \$15,369 million. By disaggregating these values, we obtain a delay cost of \$1.0242 per passenger per minute. We have used this estimate in our linear passenger cost functions. For the piecewise linear passenger delay cost function, we have used four steps: $D_{f,i} = 30, 60, 120,$ and 240. The corresponding delay costs per passenger are set to \$25, \$73, \$192, and \$457.8, respectively. This piecewise linear function coincides with the linear delay cost function at around 40.2, which is the average delay per passenger of the network we have used; and equals passenger ticket cancellation cost at 240.

The problem type, scope of recovery, and cost terms to be minimized, considered by the integrated recovery approach proposed by Marla, Vaaben, and Barnhart (2017), are similar to those in our approach. Therefore, we rely on several parameter estimates of Marla, Vaaben, and Barnhart (2017) which are mostly real estimates of an airline. These estimates are as follows: ticket cancellation cost is \$457.8 per passenger, flight cancellation cost is \$20,000, and jet fuel price is \$0.478/lb. Finally, we used a crew deadheading cost \$1,000 obtained from Petersen et al. (2012), which also deals with the integrated recovery problem.

We adapted parameters related to cruise speed decisions from Aktürk, Atamtürk, and Gürel (2014). It is assumed that in the original schedule each aircraft cruises at 1.02 times its MRC speed. The authors report MRC speeds, fuel consumption function coefficients, and number of seats for six models of aircraft. We assign these coefficients to the models used in our experimentation with respect to similarities in their seat capacities. Cruise speeds can be increased by about 10% of the MRC speed as stated in Delgado and Prats (2009). For the cost of ferrying an aircraft between two airports, we have used the sum of fuel cost at scheduled speeds and crew deadheading cost.

The airline that we use in our experimentations operated an average of 1,442 daily flights in 2013. We extract schedules of a major U.S. airline in July 2013. After elimination of flights with missing data, the network

we use includes 1,254 flights with 402 aircraft. The entire network is denoted N1. To test the scalability of the approach, we generated two subnetworks of N1. For the first subnetwork, N2, we identify aircraft routings that visit ORD or LAX. Then, crew routings and passenger itineraries using these flights are included in the subnetwork. We repeat this procedure for ORD only, to create the third and smallest network, N3.

Four of the disruption types are tested. The AOTP database provides the scheduled and actual departure and arrival times of flights, and hence, we can obtain actual departure delays. We extract the list of flights that experience a departure delay of 30 minutes or more from the AOTP database. These delays may be a result of external reasons or airline’s decisions because of the downstream effect of the delays in the preceding. We assume the latter is the likely reason when multiple sequential flights in an aircraft’s routing experience delays. We mark the first delayed flight in such aircraft routings as disrupted. Flight cancellation scenarios are generated with the same procedure. In addition to scenarios including all disruptions, we create scenarios including a quarter and half of the disrupted flights to account for the dynamic nature of decision making in disruption management. Doing this, we also hope to observe the effect of the number of disrupted flights on the performance of our solution approach. For delayed ready time instances, we have randomly selected an aircraft and delayed its ready time by 60, 120, 180, 240, and 300 minutes. Finally, a hub is closed for 60, 120, 180, 240, and 300 minutes in hub closure scenarios. The solution time is set to 30 minutes for hub closure instances and 15 minutes for the remaining instances.

To observe the effect of cruise speed control action on both solution quality and solution times, we solve all instances with and without this action. We define CS^+ as the proposed approach and CS^- as the proposed approach without using cruise speed control action. Note that CS^- is a mixed integer programming model.

We apply the partial network approach in all instances. Note that the partial networks generated for CS^- are subsets of the corresponding partial networks generated for CS^+ . This is because speeding up flights enables using additional flight connections. This corresponds to adding new rerouting alternatives to the solution space. Throughout this section, we present the characteristics of partial networks for CS^+ , which have approximately 4.1% more connection arcs than those generated for CS^- .

We experiment with four proposed delay cost calculation methods: linear function with flight delay approximation (L^-), piecewise linear function with flight delay approximation (PW^-), linear function with actual delay (L^+), and piecewise linear function with actual delay (PW^+). For experiments with actual delay costs (L^+ and PW^+), passengers are modeled

explicitly while for approximations we have used an aggregation approach for passengers in each itinerary.

Finally, we carry out experiments to understand the effect of limiting the problem size with PSCA on solution time and solution quality. Recall that we do not reduce the partial networks of passengers, but apply PSCA on aircraft and crew networks. The algorithm has a single parameter for each entity used to control the problem size. To observe its effect on the problem size and solution quality, we generate limited partial networks with different upper bounds on the number of arcs and use the optimization approach over the limited solution spaces. We test four different upper bounds: 25, 50, 75, and 100, and denote the corresponding instances with B^{25} , B^{50} , B^{75} , and B^{100} , respectively. We also solve the instances without using PSCA, which are designated by L^∞ . To summarize, for each disruption scenario we analyze 40 different recovery solutions corresponding to the combinations of: CS^+ , CS^- ; L^- , PW^- , L^+ , PW^+ ; B^∞ , B^{25} , B^{50} , B^{75} , B^{100} . In the following sections, we investigate the effects of the Partial Network Approach, cruise speed control action, passenger delay cost evaluation method, PSCA, severity of disruptions, and the length of the recovery horizon on solution times and solution quality.

7.2. Effect of Partial Networks on Scalability

As the size of an airline network increases, the set of possible recovery actions rapidly expands because of the increased number of rerouting options. To observe this behavior, we investigate three networks of different sizes. Characteristics of the extracted networks are presented in Table 3, where DF and CF are the sets of delayed and canceled flights, respectively. We also define $it \in T$ to designate the itinerary that corresponds to passenger aggregation.

Table 4 presents the average number of arcs per entity in the partial networks of aircraft, crew members, and itineraries (*avg* is used to designate average). Recall that the problem size, and hence, the solution times are directly related to the sizes of the partial networks. To understand the effect of the Partial Network Approach, we enumerate all flight paths that can be generated by the partial networks. Partial networks are able to represent 184,535 flight paths in N3, while this number reaches 12,395,883 for N1. Note that path-based representations also require copies of these flight paths with respect to different departure

Table 3. Characteristics of the Networks

Network	Hubs	$ F $	$ R^{ac} $	$ R^{cr} $	$ R^{it} $	$ R^{ps} $	$ DF $	$ CF $
N1	All	1,254	402	634	8,859	150,118	63	11
N2	ORD, LAX	473	146	249	1,551	54,381	21	6
N3	ORD	288	76	143	988	29,918	13	4

Table 4. Effect of the Airline Network on the Size of the Partial Networks

Network	$ F $	$avg A^{ac} $	$avg A^{cr} $	$avg A^{it} $
N1	1,254	3,625.58	3,454.53	6.27
N2	473	818.84	711.78	5.14
N3	288	523.12	459.15	4.49

time and cruise speed decisions. Therefore, we can state the proposed representation with a Partial Network Approach provides a more compact representation. Average running time of the PNGA is around 34.4 seconds for the largest network, while the minimum, median, and maximum of the running times are 24.6, 33.3, and 47.8 seconds, respectively.

7.3. Effect of Cruise Speed Control Action

Recall that the proposed optimization model is a conic quadratic mixed integer programming model, while its variant which does not utilize cruise speed control action is a mixed integer programming model. Therefore, it is expected that enabling cruise speed control increases the complexity of the problem. In this section, we analyze its effect on solution times, as well as on the solution quality.

7.3.1. N3 (ORD). In our experimentations, around 95.8% of all N3 instances are solved to optimality without using PSCA. An average optimality gap of 0.11% and a maximum optimality gap of 4.03% were observed. The instances with 0% optimality gaps are included in average optimality gap statistics. All instances except for hub closure scenarios are solved to optimality. These results suggest that the complexity of the problem increases as the severity of disruptions increases. The effect of disruption severity will be discussed in Section 7.6. Average solution times are displayed in Table 5. The increase in solution times because of cruise speed control in delay scenarios is greater than flight cancellation scenarios. Cruise speed control helps mitigate delays. Therefore, in scenarios with many delayed flights, optimal cruise speeds of

Table 5. Solution Times with B^∞ for N3 (in CPU Seconds)

	Flight delay	Cancellation	Aircraft delay	Hub closure	Average
CS ⁻					
L^-	28.5	53.2	20.0	102.7	51.1
PW^-	29.8	57.3	31.0	117.2	58.8
L^+	124.6	132.8	88.1	444.3	197.4
PW^+	135.2	167.6	234.8	803.0	335.2
CS ⁺					
L^-	195.6	54.5	77.6	224.7	138.1
PW^-	235.5	76.0	139.2	316.8	191.9
L^+	492.9	398.5	535.6	1,248.1	668.8
PW^+	729.4	516.5	725.7	1,475.6	861.8

Table 6. Average Disruption and Recovery Costs for N3

	CS ⁻ (\$)	CS ⁺ (\$)	Improvement (%)
Flight delay	37,434	33,130	11.50
Cancellation	133,440	127,594	4.38
Aircraft delay	344,505	317,825	7.74
Hub closure	1,478,459	1,368,874	7.41

downstream flights of affected aircraft, crew members, and passengers need to be decided. The size of the tree emanating from the disrupted nodes may be very large since the routings of different entities do not overlap.

Table 6 presents the average objective function values of the models using and not using cruise speed control action. Percent improvement values are calculated by dividing the difference in objective function values of CS⁻ and CS⁺ solutions by the objective function value of CS⁻ solutions. Despite the increase in solution times with the integration of cruise speed control, an average improvement of 7.76% suggests that cruise speed control action is beneficial in disruption management. In Figure 10, we illustrate percent improvements in cost terms by cruise speed control option. We observe improvement in passenger delay costs for all disruption scenarios as cruise speed control helps mitigate delays. Moreover, it helps maintain passenger connections so there is an improvement in external arc costs. In hub closure and aircraft delay scenarios, we also observe that there is a reduction in the number of ferried aircraft and deadheaded crew members. Infeasibility by flight cancellations may spread through the schedules of aircraft and crew members, and result in severe disruptions. Hub closure scenarios are the most complex scenarios resulting in many canceled flights as well as departure delays. We observe that network connectivity becomes more valuable than delay mitigation in cancellation and hub closure scenarios. A reduction in the number of canceled flights by cruise speed control option is 0.4 and 1.9 on average for cancellation and hub closure scenarios, respectively.

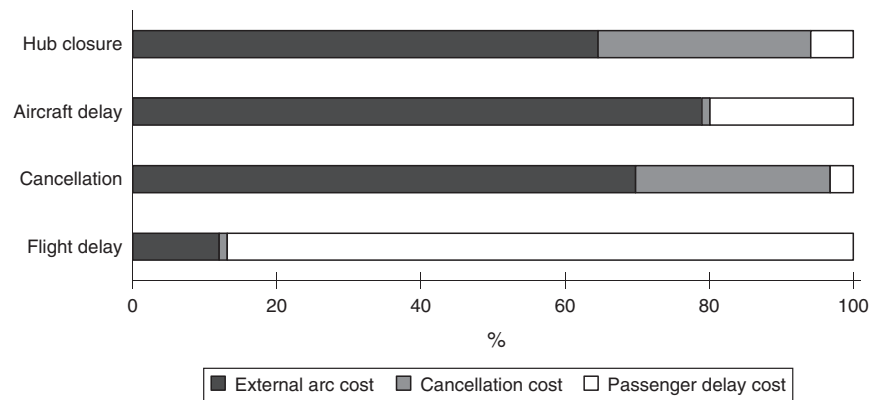
7.3.2. N2 (ORD, LAX). In N2, since the number of passengers increase significantly, we used passenger aggregation to provide real-time solutions. We have been able to solve 93.75% of all N2 instances to optimality, while the average (maximum) optimality gap is 0.15% (2.01%). Average solution times are displayed in Table 7.

On the other hand, we have been able to solve N2 instances using explicit passenger modeling (L^+, PW^+) with PSCA.

7.3.3. N1 (Entire Network). N1 is too large to be handled by using the entire solution space. We have been able to solve all instances with PSCA (B^{100}) and passenger aggregation. The effect of cruise speed control

Downloaded from informs.org by [139.179.72.75] on 29 June 2018, at 10:46. For personal use only, all rights reserved.

Figure 10. Percent Improvement in Cost Terms with Cruise Speed Control



action on solution times and solution quality are presented in Tables 8 and 9, respectively. The reason for the increase in costs in flight delay and cancellation scenarios is the increase in the number of disrupted flights. Especially, in flight delay scenarios, we observe that cruise speed control action becomes more valuable as the number of disrupted flights increases.

Table 7. Solution Times with B^∞ for N2 (in CPU Seconds)

	Flight delay	Cancellation	Aircraft delay	Hub closure	Average
CS ⁻					
L ⁻	135.4	67.8	80.7	583.8	216.9
PW ⁻	142.6	84.1	116.8	656.3	250.0
CS ⁺					
L ⁻	398.9	114.9	250.9	964.4	432.3
PW ⁻	423.4	160.3	398.4	656.3	1,069.9

Table 8. Solution Times with B^{100} for N1 (in CPU Seconds)

	Flight delay	Cancellation	Aircraft delay	Hub closure	Average
CS ⁻					
L ⁻	407.7	104.5	64.3	493.9	267.6
PW ⁻	447.3	152.9	87.3	598.8	321.6
CS ⁺					
L ⁻	811.8	209.7	240.8	1,019.4	570.4
PW ⁻	966.1	325.2	354.6	1,212.4	714.6

Table 9. Average Disruption and Recovery Costs for N1

	CS ⁻ (\$)	CS ⁺ (\$)	Improvement (%)
Flight delay	557,532	470,017	15.70
Cancellation	506,027	480,223	5.10
Aircraft delay	306,790	282,822	7.81
Hub closure	1,771,056	1,635,625	7.65

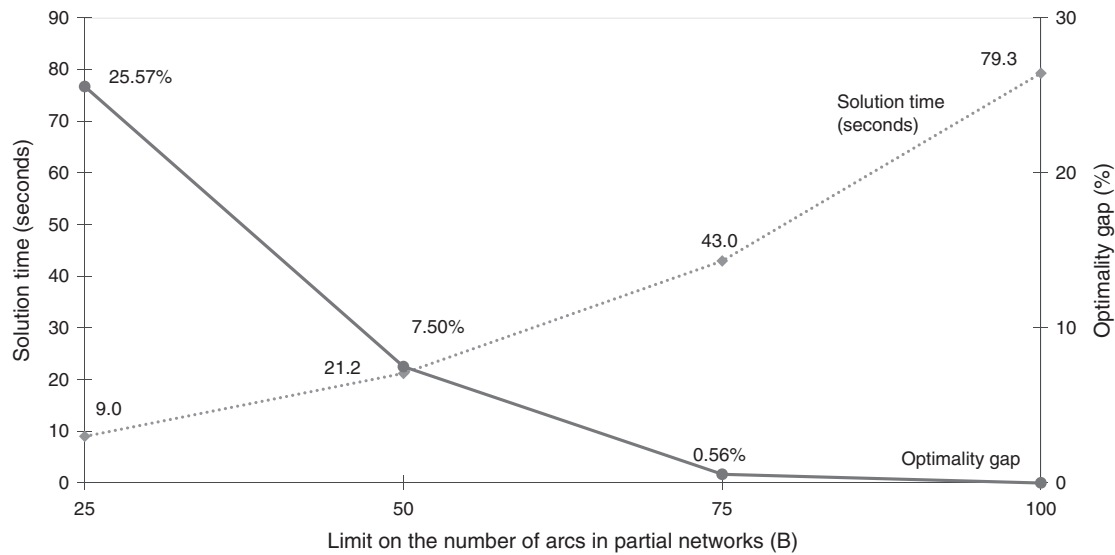
7.4. Effect of Passenger Delay Cost Function

Recall that L^- and PW^- are approximation models, and in these models passengers are aggregated. In L^+ and PW^+ models, on the other hand, passengers are explicitly modeled, and passenger delay cost is evaluated by considering realized delays and realized reallocations simultaneously. It can be observed from Table 5 that solution times are greater with L^+ and PW^+ . As a result of the increase in the number of passengers in N1 and N2, we use approximation models. To understand the effect of the approximation method, we compare the solutions in N3 instances. The approximation method underestimates total costs by \$12,225 on average. This difference may be negligible for severe disruptions, while it probably affects the recovery actions for minor disruptions.

7.5. Effect of PSCA

In large airline networks, there exists a huge number of rerouting opportunities. However, the disturbances of some rescheduling decisions may result in even greater costs than disruption costs. PSCA tries to reduce the problem size by eliminating such rerouting alternatives. In this section, we investigate the effect of the approach on solution time and quality.

We start our analysis with the smallest network, N3, for which optimal solutions to all instances are available. We have tried four different limits on the number of arcs in partial networks: 25, 50, 75, and 100. Solution times and optimality gaps are illustrated in Figure 11. Limiting the number of arcs to 50 or less results in a significant increase in disruption and recovery costs. However, the optimality gaps become negligible at $B = 75$ while the average solution time reduces to 43 seconds (the average solution time for N3 instances is 312.9 seconds over the entire solution space). The rapid decrease in solution gaps as B increases is promising to provide near-optimal solutions in large airline networks.

Figure 11. The Effect of PSCA on Solution Time and Quality (N3)

We observe similar results for N2 instances such that B^{100} instances are solved with a maximum optimality gap of 1.5%. Recall that N1 instances become intractable over the entire solution space. In Figure 12, we compare average solution times and objective function values with different partial network sizes. Objective value gaps are calculated by considering the B^{100} solution as the reference solution. Note that objective value gaps follow a convergent behavior as B increases. The average running time of PSCA in N1 instances is around 21.4 seconds, while the minimum, median, and maximum of the running times are 12.3, 19.8, and 36.0 seconds, respectively.

7.6. Effect of Severity of Disruptions

The severity of the disruption significantly affects disruption and recovery costs. In our experiments, we also observe that its effect on solution times is significant. Figure 13 illustrates the behavior of the objective function value and the solution time as the severity of disruptions increases for N1 instances. First, we observe that cancellations up to 11 or ready time delay of an aircraft does not increase the solution time significantly. However, we observe that departure delay scenarios become as complex as hub closure scenarios when the number of disrupted flights reaches 63. Disruption and recovery costs are not as high as hub closure scenarios. To understand the reasons for this, we investigate the network representation. Extracted departure delays occur at different times of the day at various locations, and hence, the disrupted flight nodes are spread through the network. On the other hand, all canceled and delayed flight nodes in hub closure scenarios are close to each other. This eliminates many rerouting opportunities and results in high recovery costs.

7.7. Effect of the Length of the Recovery Horizon

Recall that original schedules are caught at the end of the recovery horizon, t_1 . In our experimentations, we have used a recovery horizon of 2,000 minutes. It is expected that as $t_1 - t_0$ decreases, the complexity of the problem decreases; however, total recovery and disruption costs may increase. Furthermore, a shorter recovery horizon would be desirable for airlines to not disturb the schedules of many operations. In this section, we investigate this trade-off with N3 instances. We do not change the start time of the recovery horizon, but solve the disruption scenarios for $t_1 = t_0 + 500$, $t_0 + 1,000$, $t_0 + 1,500$, and $t_0 + 2,000$. To make a meaningful comparison, we have delayed and canceled flights that are operated during $[t_0, t_0 + 500]$ (early in the morning). In other words, all disruptions take place within $[t_0, t_0 + 500]$. During this period, the average number of departure delays and cancellations in N3 are three and one, respectively. In addition to actually disrupted flights, we generated nine more scenarios with three random departure delays and one random flight cancellation. The instances are solved with CS^+ , PW^+ , and B^∞ (most complex formulation).

Average solution times and average optimality gaps are presented in Table 10. The box plots of the solution times and optimality gaps are illustrated in Figure 14.

Table 10. Solution Times and Optimality Gaps with Respect to Different Recovery Horizon Lengths for N3

Recovery horizon length (minutes)	Number of flights	Solution time (CPU seconds)	Optimality gap (%)
500	46	83.2	34.24
1,000	174	518.9	11.43
1,500	255	705.3	1.03
2,000	288	739.1	0.00

Figure 12. The Effect of PSCA on Solution Time and Quality (N1)

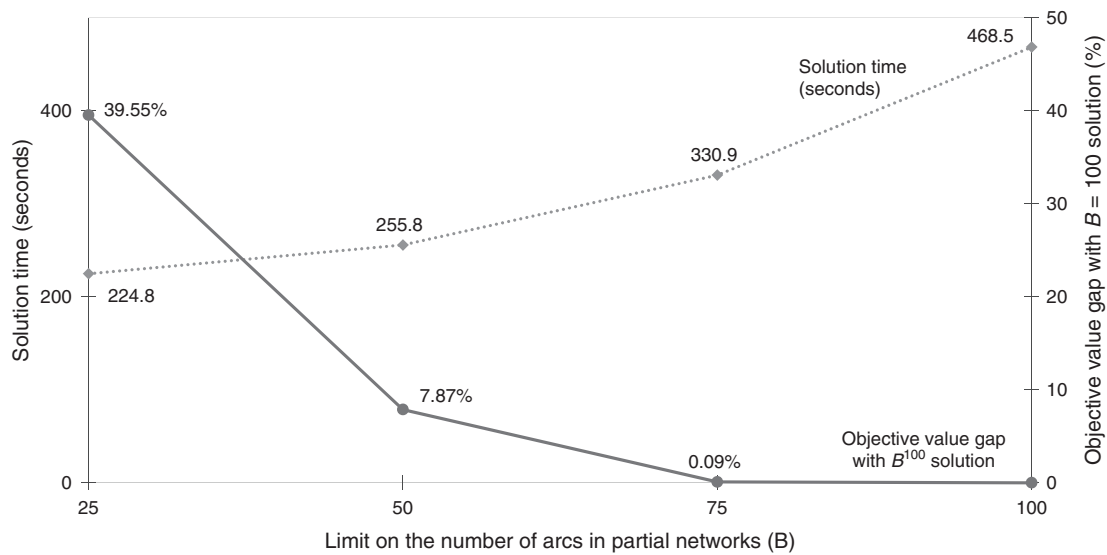
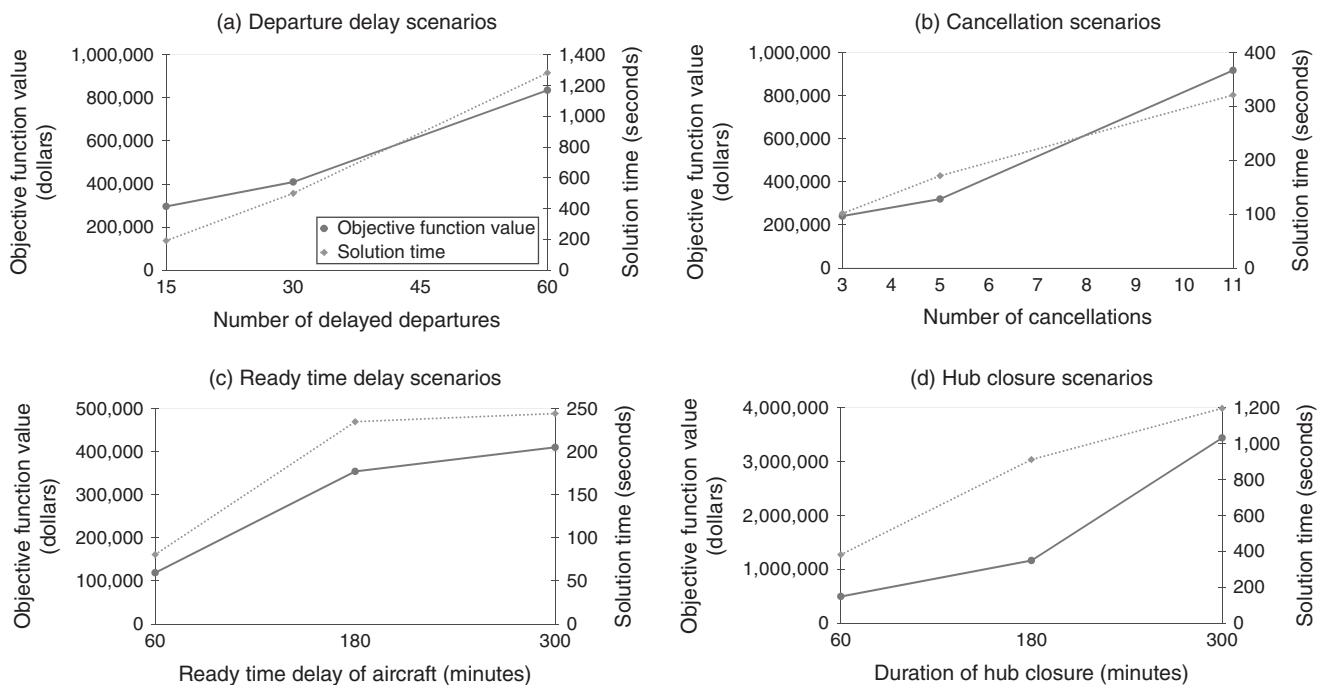
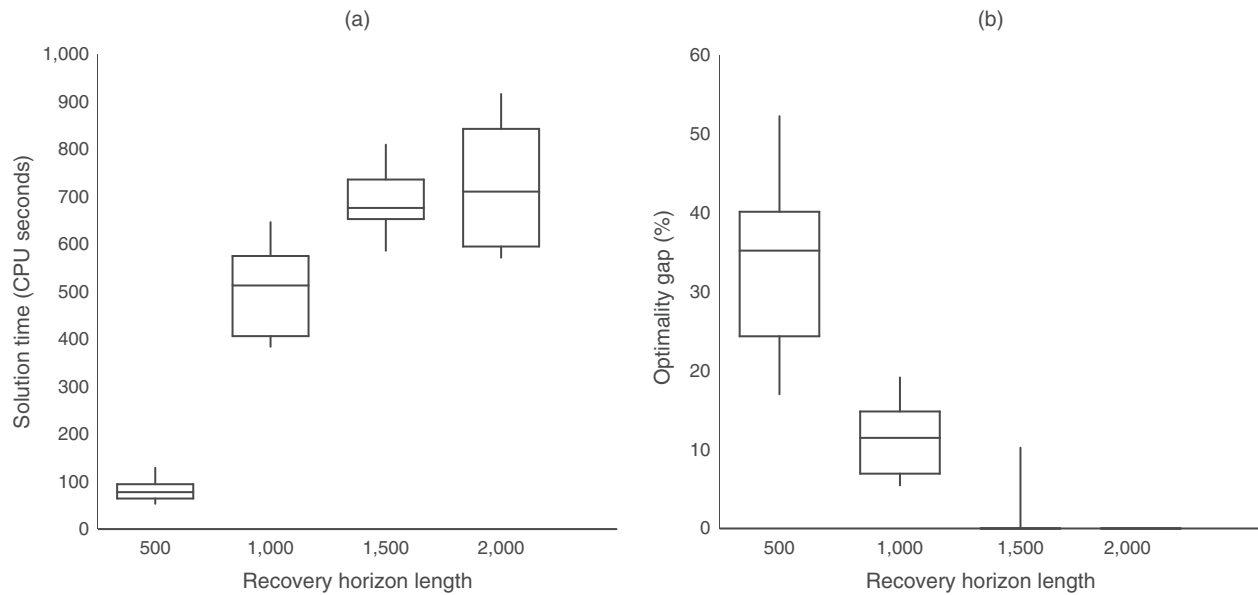


Figure 13. Effect of Severity of Disruptions for N1 Instances



The bottom and top of the boxes are the first and third quartiles, while the bands inside the boxes represent the medians. The ends of the vertical error bars correspond to the minimum and maximum values. The optimality gaps are calculated by considering the solutions to instances with $t_1 = t_0 + 2,000$ as optimal. First, we observe that the flights are not uniformly distributed throughout the day. The effect of the recovery horizon length on the problem size, and hence, on the solution times is significant. However, it is certain that 500 minutes is not sufficient to recover from these disruptions

effectively. The maximum optimality gap reaches 53% in these instances. The optimality gap reduces greatly when the length of the recovery horizon is extended to 1,000 minutes; however, it is still significant. When the recovery length is 1,500 minutes, we observe a suboptimal solution in only 1 instance, however the gap is significant (10.31%). We further investigate suboptimal solutions to understand the reasons of the optimality gaps. In most cases, the increase in costs is a result of cancellation, ferrying, and deadheading decisions that are made to catch up with the original schedules at t_1 .

Figure 14. Box Plots of Solution Times and Optimality Gaps with Respect to Different Recovery Horizon Lengths

8. Conclusion

Recently, there is an increasing effort in integrated recovery approaches for the airline disruption management problem due to high passenger inconvenience and crew recovery costs with sequential approaches. The main challenge in integration is the increased problem size while airlines require real-time solutions. In this study, we propose a general network representation that captures the state of the problem, allows integration of any entity in the same manner, and keeps the problem size within reasonable limits. Another advantage of the proposed representation is that a wide range of recovery actions including all rerouting possibilities for each entity can be integrated.

Service quality is becoming more important because of the high competition in the industry. Therefore, in cases of disruptions, evaluating possible passenger recovery actions is crucial. In this study, we manage to explicitly model each passenger instead of itinerary level modeling. This approach enables to assign passenger-specific cost parameters and generate passenger-specific recovery actions. Moreover, it allows accurate evaluation of passenger delay costs by simultaneously considering passenger rerouting and flight arrival time decisions. We propose a linear and a piecewise linear passenger delay cost function. For larger problems, we also propose approximation approaches similar to the ones proposed in the literature.

In addition to common recovery actions, we also integrate cruise speed control action in our solution space. Our experiments have shown that speeding up flights may be beneficial to help mitigate delays and preserve passenger connections in cases of disruptions. Moreover, we observe an improvement in

the connectivity of the network as new swap and rerouting options are created. However, speeding up a flight increases fuel consumption, and hence, an additional fuel cost is incurred. There is a nonlinear trade-off between fuel consumption and aircraft speed. However, the resulting formulation is second-order cone programming representable. Therefore, we can create conic quadratic constraints for the nonlinear constraints and solve the problem with commercial mixed integer programming solvers such as IBM ILOG CPLEX. With the proposed reformulation, solution times have increased compared to the case ignoring the cruise speed control option, but stayed within reasonable limits. On the other hand, significant improvements in disruption and recovery costs are observed.

We propose two important preprocessing approaches for enhancing the performance of the proposed approach without sacrificing optimality. In the first method, an efficient algorithm to generate partial networks of entities is proposed to eliminate unnecessary variables and constraints. In the second one, we propose a rule to aggregate entities that needs to be satisfied to preserve optimality. In our experimentations, we managed to optimize scenarios with 288 flights by modeling passengers explicitly without aggregation, and scenarios with 473 flights using passenger aggregation.

In addition to the preprocessing methods, we propose an efficient algorithm to control the problem size to allow real-time solutions for larger airline networks. The algorithm uses the proposed network representation to capture relations between entities and identify recovery actions that are likely to be used in the

optimal recovery. Using this approach, we managed to achieve solutions with a maximum optimality gap of 1.5% for networks having 288 and 473 flights, while average solution times are reduced from 6.7 to 1.7 minutes. Moreover, the approach allowed to solve the proposed formulation for airline networks including 1,254 flights within 8 minutes.

Acknowledgments

The authors thank the area editor and two anonymous referees for their constructive comments and suggestions that significantly improved this paper.

References

- Abdelghany KF, Abdelghany AF, Ekollu G (2008) An integrated decision support tool for airlines schedule recovery during irregular operations. *Eur. J. Oper. Res.* 185(2):825–848.
- Airbus (2004) Airbus flight operations support and line assistance, getting to grips with fuel economy. Airbus Customer Services, Blagnac, France.
- Aktürk MS, Atamtürk A, Gürel S (2014) Aircraft rescheduling with cruise speed control. *Oper. Res.* 62(4):829–845.
- Arikan U, Gürel S, Aktürk MS (2016) Integrated aircraft and passenger recovery with cruise time controllability. *Ann. Oper. Res.* 236(2):295–317.
- 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. NEXTOR Report, Federal Aviation Administration, Washington, DC.
- Barnhart C, Fearing D, Vaze V (2014) Modeling passenger travel and delays in the national air transportation system. *Oper. Res.* 62(3):580–601.
- Boeing (2007) Fuel conservation strategies: Cost index explained. AERO Quarterly.
- Bratu S, Barnhart C (2006) Flight operations recovery: New approaches considering passenger recovery. *J. Scheduling* 9(3): 279–298.
- Clausen J, Larsen A, Larsen J, Rezanova NJ (2010) Disruption management in the airline industry—Concepts, models and methods. *Comput. Oper. Res.* 37(5):809–821.
- Delgado L, Prats X (2009) Fuel consumption assessment for speed variation concepts during the cruise phase. *Proc. Conf. Air Traffic Management (ATM) Econom., Belgrade, Serbia.* (German Aviation Research Society and University of Belgrade, Belgrade, Serbia), 1–12.
- EUROCONTROL (2009) Base of Aircraft Data (BADA) aircraft performance modelling report. EEC Technical/Scientific Report 2009-009, Eurocontrol Experimental Centre, Bretigny-sur-Orge, France.
- EUROCONTROL (2012) User manual for the Base of Aircraft Data (BADA) Revision 3.10. EEC Technical/Scientific Report 12/04/10-45, Eurocontrol Experimental Centre, Bretigny-sur-Orge, France.
- Jafari N, Zegordi SH (2010) The airline perturbation problem: Considering disrupted passengers. *Transportation Planning Tech.* 33(2):203–220.
- 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.
- Lan S, Clarke JP, 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.
- Maher SJ (2016) Solving the integrated airline recovery problem using column-and-row generation. *Transportation Sci.* 50(1): 216–239.
- Marla L, Vaaben B, Barnhart C (2017) Integrated disruption management and flight planning to trade off delays and fuel burn. *Transportation Sci.* 51(1):88–111.
- Petersen JD, Sölveling G, Clarke JP, Johnson EL, Shebalov S (2012) An optimization approach to airline integrated recovery. *Transportation Sci.* 46(4):482–500.
- Rosenberger JM, Johnson EL, Nemhauser GL (2003) Rerouting aircraft for airline recovery. *Transportation Sci.* 37(4):408–421.
- Sherali HD, Bae KH, Haouari M (2013) An integrated approach for airline flight selection and timing, fleet assignment, and aircraft routing. *Transportation Sci.* 47(4):455–476.