Air-cargo Schedule Recovery with Transshipment Considering Demand Disruptions

Abstract

Demand volatility of air cargo transportation forces airlines to quickly adapt their schedules for disruptions. In this article, we introduce the Air Cargo Schedule Recovery Problem (ACSRP) with transshipment. The objective is to redesign a schedule to minimize costs, considering a penalty for schedule modifications. We propose three ways to penalize schedule changes as a function of the additional crews and flights. Our model was tested using real-life data, for three different demand disruption levels. Our experiments show the applicability of our methodology, which yields cost savings from 12.1% to 32.4% measured against an optimized base schedule.

Keywords: Air Cargo Schedule Recovery; Airline Schedule Recovery; Disruption Management; Air cargo Rescheduling; Pickup and delivery with transshipment.

1 Introduction

Air cargo is essential for the global economy because it enables the seamless integration of markets worldwide. It is a fast and reliable alternative, which makes it particularly attractive in markets where capital costs are high, and where it is consequently crucial to try to minimize shipping times. Although aircraft shipments currently represent 1% of the total tonnage moved by the cargo market, this figure increases to 35% when measured by value instead of weight (Boeing, 2015).

For the past 30 years, the number of air cargo shipments has grown consistently, but growth rates are decreasing year on year. There are two main ways that air cargo can be transported: using dedicated freighters, or using the remaining space in the luggage compartments of passenger airplanes (or, as it is usually called, in the "belly" of the plane). The utilization of belly space is very convenient for airlines, and therefore, even though there is more air freight than ever before, the yields from freighters have been declining on a yearly basis (IATA, 2016). For instance, between 2011 and 2015, although the total freight ton-kilometers (FTK) of the industry increased by 7.5%, freighter profits dropped by almost 12%. Furthermore, load factors for freighters achieved a historic low average of 44% in the year 2015. (IATA, 2015). Nevertheless, in spite of this decline in the utilization of freighters, there are two key elements that make them irreplaceable for the airlines. First, they provide a reliable capacity level that can be known in advance, as opposed to belly space which depends on how much luggage passengers check in for a specific flight. Second, there are many kinds of shipments that can only be transported in freighters, such as products that are too large for the belly of passenger aircraft, or products that require special handling such as live animals, or certain chemical substances.

The planning required for freighter operations is not an easy task. Airlines need to plan their schedules in advance to enable them to reserve the necessary crews and aircraft. However, it is very hard to make good demand forecasts in the air cargo industry, and this is owing to its high unpredictability. This variability comes from many sources. First, requests tend to be large in size (smaller ones are easier to fulfill using belly space), and they are placed by a reduced number of customers. Second, requests are usually placed on relatively short notice. Third, the routes of the passenger aircraft (i.e., belly space capacity) are planned in accordance to the passengers' demand for trips, which does not usually match the demand for air cargo. Finally, requests placed in advance often arrive just partially, or sometimes they do not even show up. This happens because there are usually no penalties in place for order no-shows, as explained in Wada, Delgado, and Pagnoncelli (2016).

Given the unreliability of demand forecasting, airlines are forced to redesign their schedules on a daily basis. In many airlines, this process is done manually by an experienced planner, who can change the routing of an aircraft, reroute cargo, or even cancel or add flights. Unfortunately, this manual process does not guarantee an efficient solution. The impact of uncertainty can be significant to the airline, in that the airplane may end up flying with very low load factors, or in some cases, it may even be unable to serve some requests, which in a tight margin industry could result in losses.

The aim of this paper is to formally introduce and solve this re-scheduling problem that arises in the short-term planning problem for freighters. We call this problem the air cargo schedule recovery problem (ACSRP). We borrow the term "recovery" from the passenger transportation version of the problem. In this last case, the disruptions occur on the supply side (e.g., an unscheduled maintenance of an aircraft) which forces the airline to find a way to return to (i.e., recover) the original schedule within a given time frame, minimizing the negative effects on operative costs and passengers. In our version of the problem, the objective is to reschedule the freighters in light of the demand disruptions, taking into account the operative costs and the costs that follow from adjusting the schedule. In this work, we will propose three alternative approaches to measuring this particular cost component.

When dealing with the ACSRP, it is of special interest to allow cargo to be transshipped. Unlike when dealing with passengers, the total travel time between the origin and destination, and even the number of connections that are made by a request is not relevant as long as it is delivered to its final destination within the desired time window. This consideration, which has not been taken into account by previous works dealing with this recovery problem, can lead to better solutions; however, it needs to be considered during the formulation of the problem, and this could make it harder to solve.

The rest of the paper is organized as follows. In section 2, we present a brief literature review regarding the problem. In Section 3, we provide a formal description of the ACSRP, and we introduce three alternative approaches for modeling the penalization for changes in the schedule. In Section 3.3, we formulate three different ways of comparing new solutions to the initial routing, and we discuss the implications of each one. Section 4 contains numerical experiments that are based on real data, as well as a discussion on the results obtained. Finally, in Section 5, we conclude the paper and present some possible future lines of research.

2 Literature Review

Adapting a schedule to account for different perturbations is a problem that has been addressed by various authors in recent years. In this section, we discuss the existing literature for the recovery problem, both for air cargo and passenger airlines. In the case of air cargo, schedule recovery is a rather unexplored research topic, so we will also discuss the schedule planning problem for this case to provide more context. In addition, given that the nature of the problem at hand involves routing aircraft and requests from given initial and final locations, we also provide a brief literature review on the pickup and delivery problem.

2.1 Schedule Planning and Recovery

As mentioned above, the concept of recovery was first introduced for passenger airline operations. This problem has been investigated by numerous authors, including Teodorović and Guberinić (1984), Rosenberger, Johnson, and Nemhauser (2003), and Bratu and Barnhart (2006). These authors study the problem of schedule recovery for disruptions affecting capacity, such as weather conditions, crew absences, congestion in airports, and last-minute maintenances. These works propose models that recover the original schedule while aiming to minimize a combination of operating costs (crew and aircraft utilization) and passenger costs (delays and cancellations).

The literature regarding the recovery problem for freighters is scarce in comparison to that for the case of passengers, and it is focused on schedule planning rather than recovery. Marsten and Muller (1980) solved the planning problem by addressing two strategic issues: deciding which Origin-Destination (OD) pairs to serve, and routing the freighter fleet accordingly. Lin and Chen (2003) studied the integration of air cargo networks from Taiwan and China. They designed a model based on a multi-commodity flow formulation, and decided which airports to use and where to locate transshipment points, if convenient. Yan, Chen, and Chen (2006) introduced an integrated model that solves simultaneously the airport selection, aircraft routing, and departure time setting for each flight. The model was formulated as a MIP problem that maximizes profits subject to operational constraints. Yan and Chen (2008) extended this formulation to consider the coordination between the schedules of various air cargo airlines. Tang, Yan, and Chen (2008) formulated a model based on an integer multi-commodity flow, combining freighter capacity with the use of bellies, applying a similar approach to Yan et al. (2006). Derigs and Friederichs (2013) solved the planning problem by exploring how to optimally modify an original schedule. Note that they use an initial schedule as a way of speeding up the solution of the problem, but they do not deal with recovery. Feng, Li, and Shen (2015) presented a review paper for the literature regarding air cargo operations, comparing theoretical studies with practical problems faced by different industry players.

To the best of our knowledge, there is no literature that addresses the AC-SRP, especially with transshipment. This problem has the following characteristics that set it apart from the ones presented previously.

(i) Schedule design is usually done for a planning horizon of one week, while a reactive approach has a more immediate scope (i.e.,: three days), where specific locations for airplanes are set at the beginning and end of the horizon.

(ii) Even though it compares the new solution with an existing original schedule, it solves simultaneously issues related to schedule design, aircraft routing and cargo routing. This means that it creates a new solution from scratch.

2.2 Pickup and Delivery Problem

Designing aircraft routes implies picking up and delivering requests in different locations, which calls for a pickup and delivery (PDP-) type formulation of the problem. Various authors provide literature reviews both in problem formulation and solution methods for the PDP, such as Parragh, Doerner, and Hartl (2008) and Toth and Vigo (2014).

Ropke, Cordeau, and Laporte (2007) formulated the PDP with time windows (PDPTW) in two different ways, followed by various methods to strengthen the formulation, using diverse valid inequalities. Ropke and Cordeau (2009) extended their previous work by solving the problem through branch-and-cut-and-price.

Mitrović-Minić and Laporte (2006) made the first contributions to the PDPTW with transshipment. In this problem, a request may be fulfilled by two vehicles, one performing the pick-up and delivering the request to a transshipment point, and another vehicle picking up the request at this point and taking it to its final destination. The results show that transshipment can reduce costs in certain circumstances, especially when requests are uniformly distributed over the plane.

Inspired by a dial-a-ride problem where passengers are allowed to transfer, Cortés, Matamala, and Contardo (2010) present a general formulation of the PDPTW with transshipment. They propose an exact algorithm based on Bender's decomposition method, and report optimal solutions for instances involving up to six nodes, two vehicles, and one transfer location. Qu and Bard

(2012) propose a greedy randomized adaptive search procedure (GRASP) using an adaptive large neighborhood search heuristic for a PDPTW with transshipment for air cargo. They validate their heuristic using the benchmark instances of Li and Lim (2003) for the VRP with time windows, providing solutions for instances of up to 50 requests. Rais, Alvelos, and Carvalho (2014) generalize the PDPTW for the heterogeneous vehicle and flexible fleet size case. They validate their models on 10 and 14 node instances obtained from the Li and Lim (2003) data set. Using a commercial solver, they show that allowing for transshipment could improve the quality of the solution by an average of 1.17% for the 14 node instances.

3 Model

Next, we show that the ACSRP can be described as a problem of determining, for a given planning horizon, a set of aircraft routes and the routing that the cargo has to follow. We refer to the set of aircraft and cargo routes the schedule for the period. This rescheduling problem aims to minimize the operational costs, while trying to prevent the new schedule from differing too much from the original one. This difference is measured using some particular metric (in this article, we propose three alternatives) and introduced directly in the objective function as a penalty. Below, we provide a formal definition and formulation of the ACSRP, and present and discuss three comparison schemes between the original and new schedules.

3.1 Definitions and Notation

An airline operates with a fleet of $k \in \mathcal{K}$ aircraft with capacity κ_k, serving a set of airports $l \in \mathcal{L}$. The planning horizon for the recovery problem is divided into a set of relatively short periods of time $t \in \mathcal{T}$. There is a set of requests $r \in \mathcal{R}$ to be fulfilled, each of which is defined by its weight w_r, the origin and destination airports, l_r^+ and l_r^-, respectively, and the time window during which it becomes available at the origin and is required to be at is destination, t_r^+ and t_r^-, respectively. The time that it takes to fly from airport i to j, which is assumed to be fixed and independent of the aircraft, is denoted as d_{ij}.

To formulate the existing problem, we construct a time–space network where every node consists of an airport paired with a period, i.e., $\mathcal{N} := \{i = (l_i, t_i) : l_i \in \mathcal{L} \wedge t_i \in \mathcal{T}\}$. Over this set of nodes, we defined three different sets of arcs. The first set $\mathcal{A}^F$ contains flight arcs, and is defined as every pair of nodes that connect an airport to a different one in the future, i.e., $\mathcal{A}^F := \{(i,j) : l_i \neq l_j \wedge t_i < t_j\}$. These arcs can be used for routing both aircraft and cargo. The second set $\mathcal{A}^G$ contains ground arcs, which denote when an aircraft (or request) remains in an airport until the next period of time. This set is defined as $\mathcal{A}^G := \{(i,j) : l_i = l_j \wedge t_i + 1 = t_j\}$. The third set of arcs $\mathcal{A}^N$ corresponds to the no-service arcs, which are arcs connecting the pickup and delivery nodes of a request. If we define i_r^+ and i_r^- as these nodes (e.g., $i_r^+ := (l_i^+, t_i^+)$), this set can be defined as $\mathcal{A}^N := \{(i,j) : \exists r \in \mathcal{R} : i = i_r^+ \wedge j = i_r^-\}$. Thus, the set of arcs of the problem is $\mathcal{A} := \mathcal{A}^F \cup \mathcal{A}^G \cup \mathcal{A}^N$.

The original routing of the aircraft are known beforehand, starting at node i_k^+ and finishing at node i_k^-. These nodes cannot be modified in the new solution. Each aircraft is allowed to fly in specific zones. Binary parameter α_{ij}^k indicates whether aircraft k can fly from node i to node j, and considers the validity of the flight leg in terms of permits and travel times. Note that parameter α_{ij}^k can be used to reduce the size of set $\mathcal{A}^F$ to make the solution of the problem more efficient. Because of labor laws, crews are limited to shifts of h hours a day. If an aircraft has to fly more than h hours a day, then an additional crew has to be assigned to that aircraft on that day.

The objective of the problem is to find a set of new itineraries that minimizes operation costs plus the costs that are associated with rescheduling flights. Parameter F_{ij}^k denotes the fixed cost incurred by aircraft k traversing arc (i,j), while parameter V_{ij} represents the cost incurred by a unit of weight of a request traversing arc (i,j). Note that in the case where $(i,j) \in \mathcal{A}^N$, this parameter represents the cost of not serving a request.

3.2 Mathematical Formulation

To formulate the ACSRP as an MIP problem, let us define binary variables $X := \{x_{ij}^k\}$ to denote whether aircraft $k \in \mathcal{K}$ uses arc $(i,j) \in \mathcal{A}$ as part of its schedule. In an analogous way, binary variables $Q := \{q_{ij}^r\}$ will denote whether a request $r \in \mathcal{R}$ traverses a particular arc.

In order to compute the costs for a specific solution, it is necessary to define a way of quantifying the extent to which the new itineraries deviate from the original ones. Let us define $\mathcal{C}(X,Q)$ as the comparison function that determines the cost associated with the deviation of the new solution with reference to the original one. Three different formulations for this function will be proposed and compared in Section 3.3.

In the formulation of the model, we define sets $\delta^+(i)$ and $\delta^-(i)$ as all the arcs belonging to $\mathcal{A}$ that emanate from (or are incident to) node i. In other words, $\delta^+(i) := \{(i,j) : (i,j) \in \mathcal{N}\}$, and $\delta^-(i) := \{(j,i) : (j,i) \in \mathcal{N}\}$.

The MIP formulation for the ACSRP is then defined as minimizing the total objective function Z, and is defined as

$$Z = \sum_{k\in\mathcal{K}} \sum_{(i,j)\in\mathcal{A}} F_{ij}^k x_{ij}^k + \sum_{r\in\mathcal{R}} \sum_{(i,j)\in\mathcal{A}} V_{ij} w_r q_{ij}^r + \mathcal{C}(X,Q), \tag{1}$$

subject to:

$$\sum_{r\in\mathcal{R}} w_r q_{ij}^r \leq \sum_{k\in\mathcal{K}} \kappa_k x_{ij}^k, \quad (i,j) \in \mathcal{A}^F \tag{2}$$

$$\sum_{(i,j)\in\delta^+(i)} x_{ij}^k - \sum_{(j,i)\in\delta^-(i)} x_{ji}^k = \begin{cases} 1, & i = i_k^+ \\ -1, & i = i_k^- \\ 0, & \text{i.o.c.} \end{cases}, \quad i \in \mathcal{N}, k \in \mathcal{K} \tag{3}$$

$$\sum_{(i,j)\in\delta^+(i)} q_{ij}^r - \sum_{(j,i)\in\delta^-(i)} q_{ji}^r = \begin{cases} 1, & i = i_r^+ \\ -1, & i = i_r^- \\ 0, & \text{i.o.c.} \end{cases}, \quad i \in \mathcal{N}, r \in \mathcal{R} \tag{4}$$

$$\sum_{(j,i)\in\delta^-(i)\cap\mathcal{A}^F} x_{ji}^k \leq \sum_{(i,j)\in\delta^+(i)\cap\mathcal{A}^G} x_{ij}^k, \quad i \in \mathcal{N} : t_i < |\mathcal{T}|, k \in \mathcal{K} \tag{5}$$

$$\sum_{(j,i)\in\delta^-(i)\cap\mathcal{A}^F} q_{ji}^r \leq \sum_{(i,j)\in\delta^+(i)\cap\mathcal{A}^G} q_{ij}^r, \quad i \in \mathcal{N} : t_i < |\mathcal{T}|, r \in \mathcal{R} \tag{6}$$

$$x_{ij}^k \leq \alpha_{ij}^k, \quad (i,j) \in \mathcal{A}^F, k \in \mathcal{K} \tag{7}$$

$$\sum_{r\in\mathcal{R}} \sum_{(i,j)\in\mathcal{A}^N} q_{ij}^r \leq \lfloor (1-\lambda)|\mathcal{R}| \rfloor \tag{8}$$

$$x_{ij}^k \in \{0,1\}, \quad (i,j) \in \mathcal{A}, k \in \mathcal{K} \tag{9}$$

$$q_{ij}^r \in \{0,1\}, \quad (i,j) \in \mathcal{A}, r \in \mathcal{R}. \tag{10}$$

The objective function (1) corresponds to the total operational costs, and consists of the sum of fixed and variable costs, plus the penalty function $\mathcal{C}(X, Q)$. Note that the variable costs include the cost of not serving a particular request. Constraints (2) ensure that requests can traverse a flight arc only if there is enough aircraft capacity assigned to this arc. Constraints (3) and (4) impose conservation of flow for aircraft and requests, respectively.

Constraints (5) and (6) are included to ensure aircraft and cargo coordination. This is done by forcing aircraft or requests that arrive at an airport by a flight arc to spend at least one time period on the ground, unless they arrive on the last period of the planning horizon. Note that the sum on the right-hand side of these inequalities contains a single variable, corresponding to the ground arc connecting nodes (l_i, t_i) and $(l_i, t_i + 1)$.

Constraint (7) indicates whether or not an aircraft can fly a certain flight leg. Constraint (8) ensures that at least a proportion λ of requests in $\mathcal{R}$ are served. Note that this constraint can be trivially adapted to measure the level of service in terms of the total transported weight instead of the number of requests. Finally, constraints (9) and (10) impose the binary nature of the variables on the model.

3.3 Solution Comparison Schemes

In order to compare the original routings with the new ones, there is a need for a method of measuring changes and assigning penalties accordingly. We propose three different ways to do this, and they are explained below. Note that in all three cases, there are penalties when the new schedule exceeds the originally planned resources, but they do not penalize or discount the unused capacity because with such short notice, it becomes a sunken cost.

3.3.1 Additional Daily Crews (C1)

This comparison considers the number of sets of crews used per aircraft on any given day. Labor laws specify the maximum number of hours h that a crew can fly daily. If an airplane flies more than h hours in one day, additional crews are needed. When comparing both schedules, we can identify the number of additional crews needed and penalize the schedule accordingly. We denote this method of comparison as C1.

In order to use this form of comparison, we define $\mathcal{D}$ as the set of days in the planning horizon (note that this is not the same as periods, which are subdivisions of a day). The available number of crew hours for aircraft k on day d on the original schedule is denoted by $\hat{T}_d^k$. This number is obtained as the scheduled hours of flight for aircraft k on day d, rounded up to a multiple of h. Parameter ζ_{ij}^k denotes the number of hours of flight that a particular arc contributes to a given day (note that this is necessary because some arcs may begin and end on different days).

We also define binary variable γ_d^k, which indicates the number of additional crews needed on day $d \in \mathcal{D}$ for aircraft $k \in \mathcal{K}$, and T_d^k as the total hours of flight of aircraft $k \in \mathcal{K}$ in day $d \in \mathcal{D}$ in the new schedule. In a given solution, these variables are implicitly defined by X and Q using the following relations:

$$T_d^k = \sum_{(i,j)\in\mathcal{A}^F} \zeta_{ij}^k x_{ij}^k, \quad d \in \mathcal{D}, k \in \mathcal{K} \tag{11}$$

$$T_d^k \leq \hat{T}_d^k + h\gamma_d^k, \quad d \in \mathcal{D}, k \in \mathcal{K} \tag{12}$$

$$\gamma_d^k \in \{0, 1, ..., \lfloor 24/h \rfloor\}, \quad d \in \mathcal{D}, k \in \mathcal{K} \tag{13}$$

$$0 \leq T_d^k \leq 24, \quad d \in \mathcal{D}, k \in \mathcal{K}. \tag{14}$$

Equation (11) computes the daily hours of flight for an aircraft in the new schedule. Equation (12) ensures that new shifts are created to meet the crew requirements for each aircraft on every day. Equation (13) limits the number of additional crews to an integer bounded by $\lfloor 24/h \rfloor$. Finally, (14) ensures that the rescheduled hours do not exceed the 24 available hours in a day.

Thus, to apply comparison C1, the penalty function $\mathcal{C}(X,Q)$ in the objective function (1) has to be defined as

$$\mathcal{C}(X,Q) = \mathcal{C}_1(X,Q) := \sum_{d\in\mathcal{D}} \sum_{k\in\mathcal{K}} C_1 \gamma_d^k, \tag{15}$$

where C_1 is the cost of an additional crew, and the MIP problem defined by minimizing (1) subject to (2) – (14) has to be solved.

3.3.2 Additional Flights between Airports (C2)

We introduce another scheme, called C2, which compares the total number of flights for airport pairs, regardless of the time period over which they take place, and the aircraft involved. This is an aggregated approach that assumes that the key goal is to not exceed resource utilization on a network-wide scale.

Defining $\hat{\pi}_{pq}$ as the number of flights between airports $p, q \in \mathcal{L}$ in the original solution, and ϕ_{pq} as a variable indicating the number of additional flights between these two airports, the penalty function for this measure is defined by

$$\mathcal{C}_2(X, Q) := \sum_{p \in \mathcal{L}} \sum_{q \in \mathcal{L}} C_2 \phi_{pq}, \tag{16}$$

where C_2 is the cost of an additional flight. In order to link variables ϕ_{pq} to X, the following constraints have to be added:

$$\phi_{pq} \geq \sum_{k \in \mathcal{K}} \sum_{(i,j) \in \mathcal{A}(p,q)} x_{ij}^k - \hat{\pi}_{pq}, \quad p, q \in \mathcal{L}, p \neq q. \tag{17}$$

$$\phi_{pq} \geq 0, \quad p, q \in \mathcal{L}, p \neq q. \tag{18}$$

In (17), the set $\mathcal{A}(p, q)$ is defined as the set of flight arcs connecting airport p to airport q, i.e., $\mathcal{A}(p, q) := \{(i, j) \in \mathcal{A}^F : l_i = p \wedge l_j = q\}$. The MIP problem to be solved using C2 is thus defined as minimizing (1) using $\mathcal{C}(X, Q) = \mathcal{C}_2(X, Q)$ subject to (2) – (10) and (17) – (18).

3.3.3 Additional Flights between Airports for Each Aircraft (C3)

This comparison method has the same idea as C2, but in this case, resources are considered separately according to aircraft, which are no longer interchangeable. This comparison method is denoted as C3. Note that even though the aim is to maintain the same number of flights (as is the case with C2), it does not take into consideration the order of the flights made by each airplane.

In order to use this comparison scheme, $\hat{\pi}_{pq}^k$ is defined as the number of flights between airports $p, q \in \mathcal{L}$ performed by aircraft k in the original solution. Binary variable ϕ_{pq}^k is defined as the number of additional flights between p and q performed by k. Then, the penalty function for C3 is

$$\mathcal{C}_3(X, Q) := \sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{L}} \sum_{q \in \mathcal{L}} C_3 \phi_{pq}^k, \tag{19}$$

where C_3 is the cost of an additional flight associated with $\mathcal{C}_3(X, Q)$. The following constraints need to be added in order to link ϕ_{pq}^k to X:

$$\phi_{pq}^k \geq \sum_{(i,j)\in\mathcal{A}(p,q)} x_{ij}^k - \hat{\pi}_{pq}^k, \quad p,q \in \mathcal{L}, p \neq q, k \in \mathcal{K} \tag{20}$$

$$\phi_{pq}^k \geq 0, \quad p,q \in \mathcal{L}, p \neq q, k \in \mathcal{K}. \tag{21}$$

The set $\mathcal{A}(p,q)$ is defined as in Section 3.3.2. The MIP problem associated with C3 is defined as minimizing (1) using $\mathcal{C}(X,Q) = \mathcal{C}_3(X,Q)$ subject to (2) – (10) and (20) – (21).

3.4 Example of Comparison Schemes

In Figure 1, we present an example to illustrate how the comparison schemes work.

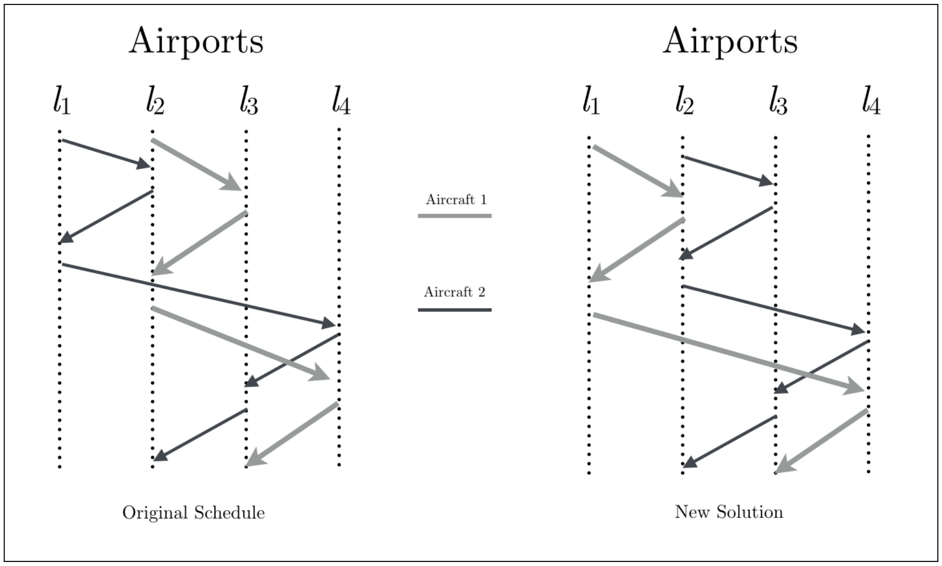

Figure 1: Example of comparison schemes.

Figure 1 depicts two different routes for two aircraft within a one-day period. Several segments are still flown by the same aircraft in the new solution, while others are now flown by other aircraft. In Table 1, we show the data obtained for the example.

For each scenario, we observe differences in the routes of each airplane. Assuming that a crew cannot work for more than six hours ($h = 6$), penalties for the three different comparison schemes can be computed as follows:

Table 1: Example of comparison schemes.

	Number of Flights											
OD Pair	1-2	1-3	1-4	2-1	2-3	2-4	3-1	3-2	3-4	4-1	4-2	4-3
Aircraft 1 - Original Schedule	0	0	0	0	1	1	0	1	0	0	0	1
Aircraft 1 - New Schedule	1	0	1	1	0	0	0	0	0	0	0	1
Aircraft 1 - Additional Flights	1	0	1	1	0	0	0	0	0	0	0	0
Aircraft 2 - Original Schedule	1	0	1	1	0	0	0	1	0	0	0	1
Aircraft 2 - New Schedule	0	0	0	0	1	1	0	2	0	0	0	1
Aircraft 2 - Additional Flights	0	0	0	0	1	1	0	1	0	0	0	0
Total - Original Schedule	1	0	1	1	1	1	0	2	0	0	0	2
Total - New Schedule	1	0	1	1	1	1	0	2	0	0	0	2
Total - Additional Flights	0	0	0	0	0	0	0	0	0	0	0	0

	Flight Hours	
Aircraft	Base Schedule	New Solution
1	5 hours	7 hours
2	7 hours	6 hours

(i) C1: Aircraft 1 now flies seven hours instead of five. This aircraft will require a new crew. Aircraft 2 does not need an additional crew, because it actually flies fewer hours than before, which does not incur a penalty.

(ii) C2: Considering the total number of additional flights in Table 1, we observe that there are no OD pairs where the new schedule exceeds the original number of flights. This means that there are no penalties in this example when using C2.

(iii) C3: From Table 1, we observe that aircraft 1 and 2 each perform three additional flights, incurring a total of six penalties when using this comparison scheme.

The previous example shows that the three schemes can yield radically different results, implying that scheme selection is an important decision that the planner should carefully consider, and which should adequately reflect the policies of the airline.

4 Results

4.1 Data Description

To construct our experiments, we used real data provided by our partner airline. For confidentially issues, we will refer to each airport by a number. The data contains the following:

1. Fixed costs to operate an aircraft.
2. Tariffs for each OD pair in US$/kg. These are the same regardless of the nature of the cargo.
3. The schedule for a full week of operation.
4. Forecasted demand for the schedule mentioned before.

The capacity of each aircraft was obtained from their manufacturers website (Boeing), and crew costs were estimated based the information made available on the website of the Federal Aviation Administration (FAA, 1991). These are shown in Table 2.

Table 2: Cost and capacity parameters for aircraft.

B767 Capacity	70,000 kg
B777 Capacity	100,000 kg
Crew Cost	13,000 USD/shift

To create the original schedule, we first obtained the original routing for each freighter. Then, using this original routing and forecasted demand, we constructed requests as a consolidation of the cargo between each OD pair. The original schedule consists of three B767 aircraft, each flying 9, 12, and 11 legs, respectively, and which serve 14 different airports.

With respect to the planning horizon considered to recover the itineraries, we used $D = 3$ days.

4.2 Scenario Construction

To test our model, we developed three different scenarios which considered different levels of demand disruption: Low, Medium, and High. Each scenario had the following characteristics:

i) Appearance of new requests: we created 20% more new requests. Each new request is generated as follows:
   - the origin and destination are randomly selected from the pool of available airports.
   - the day of service is chosen randomly from a discrete uniform distribution in the interval [0,D]. The day of service is defined as the day

(between 00:00:00 h to 23:59:59 h) in which a request needs to be transported from its pickup to its delivery location.

- the weight is selected randomly from the interval $[0, 0.25 \cdot Q_k]$

ii) Probability of *no-show*: We allow the possibility that any request (i.e., new ones generated after a disruption and originally scheduled ones) may not show up on the day of the flight. The probability of *no-show* is defined by each scenario according to the values shown in Table 3.

iii) Weight distribution for original request. The weights of the original requests are disrupted by a factor that is chosen randomly according to a uniform distribution with the parameters given in Table 3.

Table 3: Scenario characteristics.

Disruption Level	Original Schedule (OS)	Low (L)	Medium (M)	High (H)
New Requests	No	No	Yes	Yes
Probability of No Show	0%	10%	10%	15%
Distribution for weight stochasticity	1	$U(0.84-1.04)$	$U(0.76-1.06)$	$U(0.68-1.08)$
	Original requests			
Requests	38/0	29/0	29/5	28/4
Avg. Weight (kg)	34,633	32,202	29,434	29,146
	Split requests			
Requests	146/0	109/0	110/9	104/7
Avg. Weight (kg)	8,776	8,761	8,610	8,703

Table 3 also shows the characteristics of the original schedule and the number of requests and their average weights for two cases: Original and Split request. In the original case, we assume that a request is a consolidation of different cargo that share the same characteristics, and which are not separable. This leads to requests for large weight, allowing a freighter to carry only a few of them simultaneously, and reducing the flexibility of the network. In the split-request case, we split the original requests into different small ones. To split the requests, we adopt the following heuristic. The number of new requests ρ_r into which an original request r is divided is defined as $\rho_r = \lceil w_r/10 \rceil$, each of them having a weight w_r/ρ_r.

Requests in Table 3 are shown as follows: a/b, where a is the number of original requests and b is the number of new requests generated after disruption. Because new requests are smaller, the average weight decreases in the three disrupted scenarios (L, M, and H). Furthermore, in each of these scenarios, the number of requests is smaller than in the one without disruptions, with the number of requests ranging between 76%-89% with respect to the original schedule.

To compare and understand the impact of the ACSRP model when demand disruptions occur, we use as a benchmark the *optimized base schedule* (OBS). The OBS solves the ACSRP by fixing the flight legs to the original one, thus allowing the optimization of the requests routing.

For the three scenarios and for the original and split request cases, we tested the benchmark OBS and the ACSRP model under the three comparison schemes presented in the previous chapter, imposing at least an 80% service level. Thus, we tested 24 different instances. The results of these experiments are presented in the next section.

4.3 Computational Results

To solve the instances described above, we used a computer equipped with an Intel Core i5 2.4 GHz processor and 8 GB of RAM. The model was coded on Python v.2.7.10 and solved with Gurobi v.6.5.1. We set a time limit of 3,600 s to indicate that rapid solutions are required for operational decisions.

4.3.1 General Results

The results of the 24 instances are shown in Table 4.3.1. For the 12 instances involving original requests, we observe that regardless of the scenario, the objective function values of any of the three comparison schemes decrease when compared with the OBS, with savings that range between 12.1% for the C3-H to 23.9 % in C1-L. For different scenarios, if we compare the objective values for a specific comparison scheme, according to Table 4.3.1, this value increases with the number of requests for each scenario. Hence, for a specific comparison scheme, scenario L shows a smaller objective value compared to H, and H is smaller than M. With respect to the solution times, seven out of the nine instances considering comparison schemes can be solved to optimality during the time limit. The two instances that are not solved to optimality belong to C1 with a gap that does not exceed 4%. Furthermore, C2 shows solution times that are consistently smaller than C3 in all scenarios.

The 12 split-request instances show objective values that are smaller than their corresponding instances in the original request case, even though none of these instances could be solved to optimality during the one-hour time limit. Moreover, the cost savings increase when compared to the OBS with savings that range between 12.9% for the C3-L to 32.4% in C1-M.

Table 4: Computational results for all instances.

	Original requests											
Demand disruption	Low				Medium				High			
Comparison Scheme	OBS	C1	C2	C3	OBS	C1	C2	C3	OBS	C1	C2	C3
Objective Value	$1,058,743	$ 805,555	$869,554	$913,362	$1,071,743	$ 858,340	$935,256	$ 989,035	$ 1,069,371	$ 815,865	$871,004	$939,574
Cost Improvement	-	23.91%	17.87%	13.73%	-	19.91%	12.74%	7.72%	-	23.71%	18.55%	12.14%
Penalization	-	$ 13,000	$ 13,000	$ 65,000	-	$ 13,000	$ 65,000	$ 78,000	-	$ 13,000	$ 52,000	$130,000
# Transshipments	0	2	0	0	1	3	0	3	4	2	0	1
Optimality Gap	-	0.00%	0.00%	0.00%	-	3.82%	0.00%	0.00%	-	3.12%	0.00	0.00%
Solution Time (s)	-	800	182	287	-	3600	536	1366	-	3600	806	2331
	Split requests											
Objective Value	$1,018,673	$ 721,168	$ 787,723	$887,504	$1,167,461	$ 789,287	$844,516	$ 950,867	$1,062,831	$ 731,730	$ 785,954	$899,654
Cost Improvement	-	29.21%	22.67%	12.88%	-	32.39%	27.66%	18.55%	-	31.15%	26.05%	15.35%
Penalization	-	$0	$ 52,000	$ 65,000	-	$ 13,000	$ 78,000	$ 65,000	-	$ 13,000	$ 52,000	$104,000
# Transshipments	0	5	0	0	0	3	0	4	0	2	0	0
Optimality Gap	-	5.01%	4.49%	10.70%	-	8.60%	4.05%	9.73%	-	6.67%	1.43%	8.56%
Solution Time (s)	-	3600	3600	3600	-	3600	3600	3600	-	3600	3600	3600

4.3.2 Analysis of High-Disruption Scenario

Given that the high-disruption scenario generally presents the greatest changes when compared with the original schedule, in this section, we will present a detailed analysis of two key elements: route structure and load factors.

Freighter Route Structure

For each freighter, Figure 2 compares the original routes and the routes obtained for each comparison scheme after optimization.

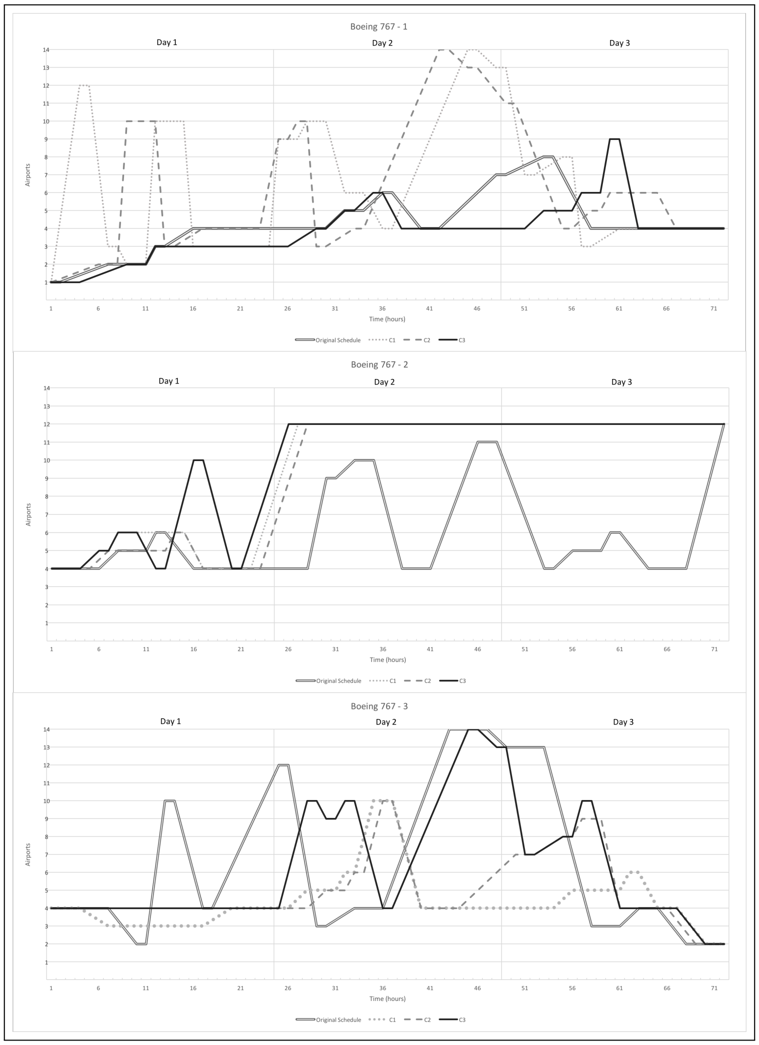

Figure 2: Comparison of route structure after optimization for all comparison schemes.

For the three freighters, the routing of each freighter shows significant differences between the original one and the new routing generated by the ACSRP model. For freighter B767-1, C3 shows a similar routing compared with the original one until hour 40, and the flight legs flown after that differ, while C1 and C2 fly similar legs, but with different departure times. For freighter B767-2, the flight legs flown under C1 and C2 during the first day coincide with the original ones, while under C3, two extra flight legs are flown. However, after hour 28, for the three comparison schemes, the freighter stays on the ground until the end of the horizon, which differs completely from the original schedule. A different picture is shown by freighter B767-3, which stays on the ground for almost all of the first day, starting its activity after that.

Aircraft Load Factors

Table 5 shows the load factor for each freighter under the different comparison schemes. The second column shows the planned load factor, which represents the scenario which will occur under no disruption and the original schedule, while column three shows the resulting load factor under the OBS benchmark. Columns four to six show the load factors that result from the schedule adjustments that occurred after running the ACSRP under the three different comparison schemes.

Table 5: Comparison of Load factors for different comparison schemes.

Aircraft	Original Schedule	OBS	*C1*	*C2*	*C3*
B767-1	0.49	0.39	0.51	0.44	0.50
B767-2	0.69	0.47	0.44	0.51	0.43
B767-3	0.57	0.55	0.63	0.71	0.55
Average	0.58	0.47	0.54	0.56	0.50

From table 5, we first observe that, as expected, the load factor decreases considerably after demand disruption. Without any modifications to the aircraft routing, the airline would suffer a loss of 11% of the average load factor. Optimizing the schedule after disruptions lets the airline reduce this loss to an average of 4.6%, depending on the comparison scheme applied.

5 Conclusions

In this paper, we introduced the ACSRP for demand disruptions, with transshipment. This is a short-term planning problem where decisions regarding the scheduling of aircraft and the routing of cargo should be made simultaneously. To adequately measure the costs of a feasible new schedule, we proposed three different comparison schemes.

We evaluated the ACRSP model under different scenarios of demand disruption, considering the cases of consolidated or split requests. The results show that the ACRSP model outperforms our benchmark optimized base schedule in all the scenarios, with savings that range between 12.1% and 32.4%. Thus, from this perspective, the application of the ACRSP model could significantly benefit air cargo operations.

For the scenarios considered in this paper, we conclude that the use of split requests achieved greater savings than for the case of cargo consolidation, with an increase in cost savings of up to 14.9%, even though none of these last instances were solved to optimality during the imposed time limit.

Potential areas for future research include the study of different time horizons to recover the original schedule because longer periods may yield better results in terms of the cost of delaying the return to the original schedule. In addition, in order to better support the decision-making processes of airlines, it is necessary to keep improving both the run times and the scalability of the solution algorithm. This requires the use of heuristics that are specially designed for the solution of the recovery problem.

Acknowledgments

We acknowledge the support of the Chilean Fund for Scientific and Technological Development (FONDECYT) through Project 11140436, and the CONICYT master thesis grant 22160226 to coauthor Cristóbal Sirhan.

References

Boeing. (2015). World Air Cargo Forecast 2014-2015. *Forecast*.

Bratu, S., & Barnhart, C. (2006). Flight operations recovery: New approaches considering passenger recovery. *Journal of Scheduling*, *9*(3), 279–298.

Cortés, C. E., Matamala, M., & Contardo, C. (2010). The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, *200*(3), 711–724.

Derigs, U., & Friederichs, S. (2013). Air cargo scheduling: Integrated models and solution procedures. *OR Spectrum*, *35*(2), 325–362.

FAA. (1991). *Economic values for faa investment and regulatory decisions, a guide final report (updated september 2016).*

Feng, B., Li, Y., & Shen, Z.-J. M. (2015). Overview Paper: Air cargo operations: Literature review and comparison with practices. *Transportation Research Part C*, 263–280.

IATA. (2015). Air Freight Market Analysis - December 2015. (September), 5–7.

IATA. (2016). August 2016 Air Freight Market Analysis. (August), 4–7.

Li, H., & Lim, A. (2003). A metaheuristic for the pickup and delivery problem with time windows. *International Journal on Artificial Intelligence Tools*, *12*(02), 173–186.

Lin, C. C., & Chen, Y. C. (2003). The integration of Taiwanese and Chinese air networks for direct air cargo services. *Transportation Research Part A: Policy and Practice*, *37*(7), 629–647.

Marsten, R. E., & Muller, M. R. (1980). A Mixed-Integer Programming Approach to Air Cargo Fleet Planning. *Management Science*, *26*(11), 1096–1107.

Mitrović-Minić, S., & Laporte, G. (2006). The pickup and delivery problem with time windows and transshipment. *INFOR: Information Systems and Operational Research*, *44*(3), 217–227.

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pickup and delivery problems. Part I: Transportation between customers and depot. *Journal für Betriebswirtschaft*, *58*(2), 81–117.

Qu, Y., & Bard, J. F. (2012). A GRASP with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers and Operations Research*, *39*(10), 2439–2456.

Rais, A., Alvelos, F., & Carvalho, M. S. (2014). New mixed integer-programming model for the pickup-and-delivery problem with transshipment. *European Journal of Operational Research*, *235*(3), 530–539.

Ropke, S., & Cordeau, J.-F. (2009). Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, *43*(3), 267–286.

Ropke, S., Cordeau, J. F., & Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, *49*(4), 258–272.

Rosenberger, J. M., Johnson, E. L., & Nemhauser, G. L. (2003). Rerouting Aircraft for Airline Recovery. *Transportation Science*, *37*(4), 408–421.

Tang, C. H., Yan, S., & Chen, Y. H. (2008). An integrated model and solution algorithms for passenger, cargo, and combi flight scheduling. *Transportation Research Part E: Logistics and Transportation Review*, *44*(6), 1004–1024.

Teodorović, D., & Guberinić, S. (1984). Optimal dispatching strategy on an airline network after a schedule perturbation. *European Journal of Operational Research*, *15*(2), 178–182.

Toth, P., & Vigo, D. (2014). *Vehicle routing: problems, methods, and applications* (Vol. 18). Siam.

Wada, M., Delgado, F., & Pagnoncelli, B. K. (2016). A risk averse approach to the capacity allocation problem in the airline cargo industry. *Journal of the Operational Research Society*, 1–9.

Yan, S., & Chen, C. H. (2008). Optimal flight scheduling models for cargo airlines under alliances. *Journal of Scheduling*, *11*(3), 175–186.

Yan, S., Chen, S.-C., & Chen, C.-H. (2006). Air cargo fleet routing and timetable setting with multiple on-time demands. *Transportation Research Part E: Logistics and Transportation Review*, *42*(5), 409–430.