Air Cargo Allotment Planning

Chan Seng Pun[1] and Diego Klabjan[1]

[1]Industrial Engineering and Management Sciences, Northwestern University

In the mid-term capacity planning process for air cargo, a cargo carrier reserves capacity up to six months in advance for its clients, who provide regular and frequent shipments over multiple flights. We model the underlying capacity allocation problem as a portfolio optimization problem to allocate cargo space on flights while minimizing the demand covariance between allotments and spot market demand. Due to the complexity of the problem, we develop an efficient partitioning algorithm to decompose the problem into subnetworks and cluster demand. The resulting allocation policy is tested using a real world dataset provided by a solution vendor, and it is benchmarked against a risk-neutral allocation policy used in practice. We observed on average revenue improvement by 2%, which approximately accounts for $150,000 per week for major cargo carriers.

Key words: air cargo, revenue management, stochastic optimization

1 Introduction

Air cargo is an indispensable part of airline business. When an aircraft carries passengers to its destination, its belly is utilized to carry cargo shipments. However, with an expected strong growth of cargo demand since 2010, airlines started to purchase dedicated cargo aircraft (freighters) to exclusively handle cargo shipments. Some carriers even set up an independent cargo division to take advantages from the upcoming demand surge. Nowadays, the cargo business accounts for more than 20% of the total revenue for many major carriers (RITA (2012)).

Revenue management (RM) in air cargo consists of short and mid-term allocation processes. The short-term allocation process allocates available flight capacity to volatile spot market demand, and shippers are charged based on the floating market rate. Despite the volatility of the spot market, shippers in the spot market utilize their allocation more effectively as they request capacity only when they are (almost) certain about their shipments. Similarly to the allocation process for passenger RM, the short-term allocation process is run nightly to update the capacity allocation given the remaining capacity and updated demand forecast. The resulting allocation policy is then implemented next day to accept and reject shipment requests. On the other hand, the mid-term allocation process is executed twice per year with a planning horizon of six months. It allocates flight capacity to large and regular shipments that provide a stable revenue stream to the carrier. By promising regular shipments, shippers in return receive a discounted rate and guaranteed space.

The mid-term allocation process is initiated when the carrier releases a new flight schedule. In the process, the carrier sells flight capacity in the form of allotments, which are simply reserved blocks of cargo space, to shippers through various capacity commitments. Those shippers can be freight forwarders (capacity resellers), large clients, and local station managers. Large clients usually receive a deep discount rate, but still account for a significant portion of the total revenue. Local station managers are essentially freight forwarders owned by the carrier. However, they only consider the amount of space that they should acquire on flights that depart from their stations.

Figure 1 provides details about the mid-term allocation process in practice. It starts in the first week when the tentative flight schedule for the next six months is released. After receiving the new flight schedule, each shipper/bidder prepares an allotment bid with a bidding price, shipment schedule, and capacity requirements measured in weight, volume, and the number and type of unit load devices (ULD). A ULD is a cargo container or pallet that can be used to containerize shipments, and can be easily loaded onto compatible aircraft. Once the bids are collected, the carrier prepares input for the optimization system including existing allotments to be honored, capacity forecasts, overbooking rates, available itineraries, operating costs, freight rates, and any business rules that are to be complied. The optimization system then converts the requested capacity of each bid to the smallest ULD type to reduce the complexity of the problem, solves a resource allocation problem, and converts the solution back to the originally requested ULD types. It returns an allocation policy that suggests the carrier which bids should be accepted or rejected, which itineraries are used, and how much space should be reserved on each itinerary (depending on the type of the commitment, the carrier can grant less capacity than requested). However, it neither considers ad-hoc allotment bids, which are bids that come during the planning horizon, nor spot market demand that only appears close to departure. At the end, after fine tuning the allocation for each accepted bid through an iterative negotiation process with its bidder, the carrier constructs allotments by aggregating the allocated weight and volume for each shipper, and uploads the allotments to the booking system for the shippers to book in the future. The entire allocation process takes about seven weeks. Any ad-hoc allotment bids are accepted only if they are profitable, and cargo space is available.

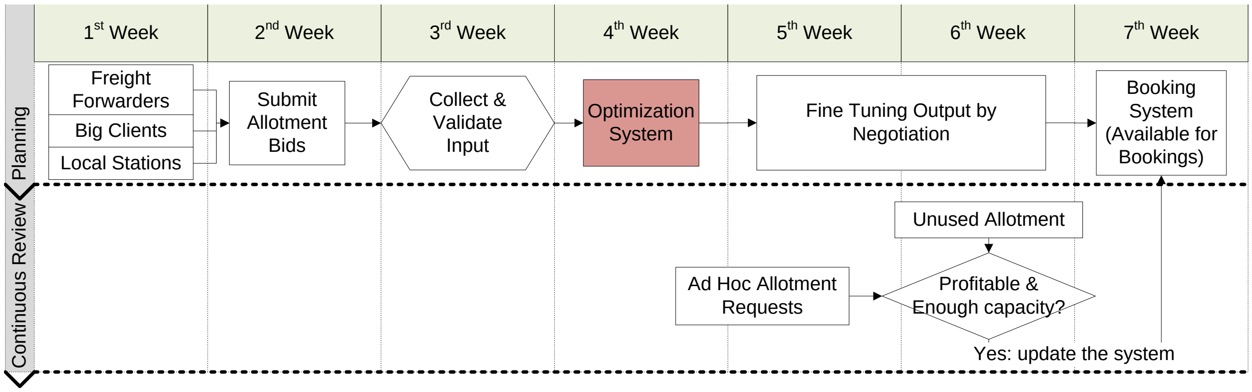

Figure 1: The mid-term allocation process.

The two paragraphs just described outline two allocation processes heavily depending on each other, since flight capacity can be used either as an allotment or reserved for the spot market. When the market rate is high and the allotment utilization is low, the cargo space should be kept for the spot market. Otherwise, allotments can be used to hedge against the volatility of the spot market. Thus, instead of allocating capacity to allotments first and selling the remaining capacity on the spot market later, carriers can utilize their capacity more effectively and economically by considering both allotments and spot market demand simultaneously during the mid-term allocation process.

Although efforts have been made in recent years, RM in cargo has not received comparable attention to RM in passenger, since the cargo business has not been regarded as the main revenue stream for many airlines. In practice, many airlines directly adopt their passenger RM system to manage their cargo business. However, this results in suboptimal capacity controls due to the following major differences between the two RM systems.

- Cargo RM consists of both mid-term and short-term allocation processes while passenger RM has only the latter.
- Passengers are itinerary-dependent, but cargo shipments are mostly defined at the origin-destination level with a delivery deadline.
- While passenger demand is counted in the number of passengers, cargo demand is counted in the units of weight, volume, and ULDs.
- Similarly, in passenger RM, flight capacity is counted in the number of seats. Cargo capacity is measured in the units of weight, volume, and aircraft positions, which are designated aircraft floor areas with a special equipment to fasten compatible ULDs.
- While passengers can only be show-up or no-show, shippers can partially utilize their allocation without paying any penalties.

In fact, cargo RM is more challenging since the mid-term allocation process allocates capacity six months in advance. Let alone the demand stochasticity and the aforementioned differences, the underlying optimization problem is a large multi-dimensional bin packing problem, which is difficult to solve. Thus, our goal is to modify the allocation model so that allocation requirements can be exactly captured, and an allocation policy can be obtained efficiently and of good quality.

There are three major drawbacks in the current optimization system. Firstly, assigning capacity based on the smallest ULD type may not yield a feasible solution when the allocation decision is converted back to the originally requested ULD types, since there are physical limitations on the number of ULDs of a specific type that can be loaded onto an aircraft. If the solution is infeasible, operators either rerun the optimization with additional constraints or manually adjust the solution in the hope that a feasible solution can be obtained. In either case, considerable amount of time and efforts are required to obtain a feasible solution. Secondly, the optimization systems consider allotments first and treat spot market demand secondarily. In fact, as we explained earlier, spot market demand may as well be equally profitable. Thirdly, the process does not capture the variability and correlated nature of allotments and spot market demand. For example, an allotment being frequently underutilized may indicate an upward shift of the spot market demand for the same product. This could happen when there are constant delays in the supply chain of a shipper. In this case, the allotment should be appropriately reduced so that the extracted space could be sold to more profitable shippers or on the spot market.

In this paper, we propose a portfolio optimization problem for the optimization system to resolve the aforementioned drawbacks. Our goal is to capture the necessary allocation requirements while considering demand correlation between allotments and spot market demand. In addition, we capture ULDs exactly on both ends (bids and aircraft positions). Due to the size of the problem, we decompose the problem by a partitioning algorithm. It efficiently partitions flights and groups profitable demand together, and returns a demand cluster for each set of flights in the resulting flight partition. Then, the portfolio optimization problem is applied to each demand cluster to minimize the demand covariance within the demand cluster subject to a revenue lower bound and capacity upper bounds on the corresponding flights. To evaluate the solution, we benchmark the resulting allocation policy against the risk-neutral allocation policy, which is obtained by solving our problem without demand covariance. Both allocation policies are tested using a simulator that we constructed to evaluate their performances and sensitivities on both the revenue lower bound and demand clusters. While varying the revenue lower bound affects the tradeoff between the risk-neutral revenue and variability, demand swapping across clusters perturbs the covariance matrices considered by the portfolio optimization problem and provides a fair performance comparison with the risk-neutral allocation that is cluster-independent.

We make the following contributions.

1. We provide a risk-averse portfolio optimization model that considers demand variability, correlation between allotments and spot market demand, demand requirements at the container level, and capacity requirements at the aircraft position level.
2. We propose an efficient partitioning algorithm to decompose the optimization problem into many smaller prob-

lems by partitioning flights and clustering demand simultaneously. It keeps the portfolio optimization problem tractable by only including highly profitable and connected demand in a cluster whose size is restricted by a threshold.

3. To the best of our knowledge, we are the first to solve a portfolio optimization model over a capacity constrained air cargo flight network, and provide numerical results and sensitivity analyses. Thus, we combine revenue and risk in the same model and algorithm.

The structure of our paper is as follows. Section 2 describes the portfolio optimization problem for the mid-term allocation process, and Section 3 presents the partitioning algorithm. A simulator used to evaluate the performance of the allocations is provided in Section 4. The case study is discussed in Section 5. We conclude our paper in Section 6.

1.1 Literature Review

Only a few works capture allotments, and none of them handles ULD exactly with load positions. In our work, ULD load positions are captured accurately.

Kasilingam (1997) discusses the role of demand forecasting, capacity forecasting, allotment allocation, and overbooking in the overall capacity allocation process. In addition, he provides an overbooking and bucket allocation models. Different from ours, his model allocates capacity to demand buckets, and includes probabilistic chance constraints that can be linearized to model service level requirements. No demand correlation is considered. A similar overview can be found in Slager and Kapteijns (2004), who discuss implementation challenges of a cargo RM system, and several major differences between RM in cargo and passengers.

Hellermann (2004) considers an allotment contract as an option with reservation and exercise fees. His analysis focuses on parameters, structures, and optimality conditions for both the carrier and shipper models. The carrier model maximizes the total return by determining the optimal reservation and exercise fees, which are then fed to the shipper model to obtain an optimal capacity allocation. He also provides an analysis of the integrated model. However, all the models are single-leg and cannot be extended to capture the cargo network.

Amaruchkul et al. (2010) consider a capacity contract, which sets the allocation level, unit price for the capacity used, and unit refund rate for the unused capacity. They express the expected contribution as a utility function for a carrier and shipper separately. The utility functions are then combined to yield a stochastic optimization model that maximizes the total contribution subject to an incentive compatibility constraint and a shipper contribution lower bound. Their problem is also single-leg, and it is inapplicable in a network setting.

Karaesmen (2001) formulates the simplified version of the spot market allocation problem by a continuous linear programming problem. She assumes a single shipment type, and shows that solutions to a sequence of linear problems converge to an optimal solution of the continuous linear programming problem, so do the bid prices, which can be computed by an approximation algorithm. Her result is theoretically interesting but cannot be extended to capture allotment requirements.

Popescu (2006) investigates a spot market allocation problem with backlogs and positive lead time. She divides demand into two groups and designs a different model and algorithm for each group. She also considers cases of her model with one and two time lags, and shows that the optimal allocation, if exists, is a deterministic stationary allocation. Her theories and algorithms are developed directly based on the traditional network RM for passengers (see Talluri and van Ryzin (2004)), and no demand correlation and allotments are considered.

Luo et al. (2009) study the spot market allocation problem in weight and volume while accounting spoilage and offload costs. They analyze the model in terms of booking acceptance regions, which can be circular or rectangular. However, their model is single-leg, and the proposed controls are difficult to implement and store in an information system.

Pak and Dekker (2004) present a dynamic programming (DP) problem to allocate capacity for the spot market. The problem is constrained by flight capacity in both weight and volume. They approximate the value function by a standard knapsack problem, which is further approximated by an ordering algorithm that efficiently generates a set of bid-prices. Amaruchkul et al. (2007) propose several heuristics to decompose and approximate the value function of a DP for the spot market. The DP they model is similar to the traditional passenger RM problem with multiple classes (see Curry (1990), Wollmer (1992), and Brumelle and McGill (1993)). They empirically show that their heuristics perform well. Both Pak and Dekker (2004) and Amaruchkul et al. (2007) address the short-term allocation problem using DP to construct a better allocation policy. However, extending their models to capture allotments significantly increases the complexity of the problem.

Chew et al. (2006) also study a single-leg short-term allocation problem. They focus on updating the allocation as departure approaches. The problem decides how much extra space should be allocated and which shipments should be backlogged in order to minimize the expected cost. Relying on convexity of the cost function, they are able to efficiently construct an optimal solution. Their model is also similar to the traditional passenger RM problem with multiple classes. Hence, it inherits drawbacks similar to the DP in Amaruchkul et al. (2007) and it is a single-leg problem.

Levina et al. (2011) investigate the short-term allocation problem at the network level given the remaining capacity not reserved for allotments. They basically capture all the risk components in the spot market, and propose a simulation-based approach to approximate the optimal allocation. Nonetheless, they do not consider the interaction between the allotments and spot market demand, and adding allotments to their existing model appears to be nontrivial.

Levin et al. (2012) model the capacity allocation problem for a particular origin-destination pair at multiple stages. At the first stage, a mixed integer programming problem is solved to optimally select allotment bids. At the second stage, a DP is run to accept and reject spot market booking requests given the remaining flight capacity. Demand is assumed to be independent. At the final stage, a different model is applied to offload undesirable confirmed bookings. By relaxing the weight and volume capacity constraints of the offloading problem, they obtain an upper bound problem to approximate the original problem that is difficult to solve. Furthermore, the upper bound problem naturally returns a set of bid prices for the spot market. Our work is different in that we allow any pair of demand to be dependent, and we do not rely on solving DPs that are often intractable given a large air cargo network. Instead, we model the problem as a portfolio optimization problem to capture all necessary allocation requirements and demand correlation.

In summary, all past works either are single-leg and not easily extendible to a network setting, or assume independent demand at the network level, or handle the spot market and allotment decisions separately. In addition, they neglect important aspects of ULDs and their positioning requirements on aircraft.

2 Problem Definition

For the mid-term capacity allocation process, we model the underlying cargo capacity allocation problem as a portfolio optimization problem to accept profitable and stable capacity requests and assign their ULDs to aircraft positions. The problem has integer decision variables and a quadratic objective. It minimizes the demand covariance computed based on historical capacity misutilization (over/under-utilizing) of the allotments, forecasting inaccuracy of the spot market demand, as well as the correlation between these two variabilities, and subject to all capacity allocation requirements and a revenue lower bound.

We define demand at the origin-destination level, and each demand unit is either an allotment bid or a spot market capacity request. While an allotment bid is counted in weight, volume, and the number and types of ULDs, and consumes capacity over multiple flights over time, spot market demand is a single shipment and is counted only in the units of weight and volume. We add an auxiliary ULD corresponding to the spot market (details provided later). Each demand unit is assumed to have a fixed rate, i.e. price per weight unit, and a set of available itineraries. For a demand unit, it is allowable to accept only a subset of ULD requests. A single demand unit can be routed through several itineraries. In the case that a demand unit only provides a shipment deadline, any itineraries before the deadline can be selected. Capacity of an aircraft is counted in the number of positions available for a specific position type, and each aircraft has many types of positions available dependent on its configuration. Each ULD may consume multiple positions and vice versa, and there are restrictions on the types and number of ULDs that can be loaded on a given position type. To describe the problem, let us define the following sets:

- $d \in D$ set of demand units, where d can be an allotment bid or a spot market capacity request,
- $i \in I^D(d)$ set of itineraries for demand unit d,
- $i \in I^{DF}(d, f)$ set of itineraries for demand unit d that use flight f,
- $p \in P(f)$ set of position types available on flight f,

- $u \in U^D(d)$ set of ULD types requested by demand unit d,
- $u \in U^P(p)$ set of ULD types that is compatible with position type p,
- $u \in U^F(f) = \cup_{p \in P(f)} U^P(p)$ set of ULD types that can be loaded on flight f,

and decision variables:

- x_{diu} (integer) number of type-u ULDs of demand unit d accepted on itinerary i,
- y_{fpu} (integer) number of type-u ULDs assigned to position type p on flight f.

Additionally, we define the following coefficients:

- d_{du} number of type-u ULDs requested by demand unit d,
- n_{fp} number of type-p positions available on flight f,
- r_{di} unit revenue for carrying each unit of weight of demand unit d on itinerary i,
- w_f available weight on flight f,
- v_f available volume on flight f,
- ρ^w_{du} unit weight for each type-u ULD of demand unit d,
- ρ^v_{du} unit volume for each type-u ULD of demand unit d,
- ρ^P_u number of positions required to accommodate each unit of type-u ULD,
- $\sigma(d, d')$ covariance of demand units d and d' measured in weight,
- τ^w_u tare weight of a type-u ULD,
- τ^v_u tare volume of a type-u ULD.

Furthermore, let φ be a lower bound on the risk neutral revenue (we will explain later how it is derived). Formally, the capacity allocation problem (CAP) is

$$CAP^* = \min \sum_{d \in D} \sum_{d' \in D} \sigma(d, d') \left(\sum_{i \in I^D(d)} \sum_{u \in U^D(d)} r_{di} \rho^w_{du} x_{diu} \right) \left(\sum_{i \in I^D(d')} \sum_{u \in U^D(d')} r_{d'i} \rho^w_{d'u} x_{d'iu} \right) \tag{1}$$

subject to

$$\sum_{d \in D} \sum_{i \in I^D(d)} \sum_{u \in U^D(d)} r_{di} \rho^w_{du} x_{diu} \geq \varphi \tag{2}$$

$$\sum_{d \in D} \sum_{i \in I^{DF}(d,f)} \sum_{u \in U^D(d)} \rho^w_{du} x_{diu} + \sum_{p \in P(f)} \sum_{u \in U^P(p)} \tau^w_u y_{fpu} \leq w_f \qquad f \in F \tag{3}$$

$$\sum_{d \in D} \sum_{i \in I^{DF}(d,f)} \sum_{u \in U^D(d)} \rho^v_{du} x_{diu} + \sum_{p \in P(f)} \sum_{u \in U^P(p)} \tau^v_u y_{fpu} \leq v_f \qquad f \in F \tag{4}$$

$$\sum_{d\in D}\sum_{i\in I^{DF}(d,f)} x_{diu} \le \sum_{p\in P(f)} y_{fpu} \quad u \in U^F(f), f \in F \tag{5}$$

$$\sum_{u\in U^P(p)} \rho_u^p y_{fpu} \le n_{fp} \quad p \in P(f), f \in F \tag{6}$$

$$\sum_{i\in I^D(d)} x_{diu} \le d_{du} \quad u \in U^D(d), d \in D, \tag{7}$$

$$\mathbf{x}, \mathbf{y} \text{ integer.} \tag{8}$$

Objective function (1) minimizes the covariance between each pair of accepted demand units. Since each unit of weight is not valued the same for each demand unit, the associated unit rate is multiplied. The objective can also be modified to capture covariance in volume or volumetric weight (see DHL (2012)), and doing so will not significantly affect the solution as the density of the demand is fixed. Thus, variation in weight corresponds to variation in volume, and vice versa. Constraint (2) imposes the revenue lower bound for a given revenue level φ. Constraints (3) and (4) are the weight and volume upper bounds on each flight. If an ULD of a specific type is used, its tare weight and volume are added. In layman's terms, the left-hand side sums the weight (volume) of the demand accepted and tare or empty weight (volume) of ULD's used. Constraints (5) convert the acceptance decisions x_{diu} to the allocation decisions y_{fpu}. Constraints (6) are the upper bounds on the number of positions for each position type on each flight, and constraints (7) are the demand upper bounds at the ULD level. Other business requirements such as total tonnage upper and lower bounds at the origin-destination and flight levels are implemented but not presented.

To describe how the covariance is estimated, let us refer to the status of a cargo shipment provided by a shipper but not yet shipped as tendered, and the weight of a tendered shipment as tendered weight. We define s^w_{dt} to be the tendered weight of demand unit d on historical date t, μ^w_d to be the targeted weight as anticipated by the bid of demand unit d, and $T(d, d')$ to be the set of shipping dates that both demand units d and d' have shipping records. The targeted weight is the weight granted for the allotment, and is the weight forecast for a spot market demand unit. The covariance between demand units d and d' is estimated based on $\sigma(d, d') = (\sum_{t\in T(d,d')}(s^w_{dt} - \mu^w_d)(s^w_{d't} - \mu^w_{d'}))/|T(d, d')|$.

In our implementation, we additionally introduce two new ULD types. The first type corresponds to bulk cargo that cannot be fitted into any ULDs, and the second type is to hold spot market demand for which only the amount of weight and volume are available at the beginning of the mid-term allocation process. As a consequence of the first ULD type, a new position type is added to represent the bulk compartment on the aircraft. The second ULD type requires the density of the ULD to be demand dependent, and hence, we use ρ^w_{du} instead of ρ^w_u for spot demand d.

In summary, CAP provides an optimal allocation policy that minimizes capacity misutilization of allotments and spot market demand variability while maintaining an acceptable revenue level. However, this problem is difficult to solve, since the covariance matrix is large, and an integer solution is required. Although business experience and intuition help reduce the complexity of the problem, we approach the problem in a general manner by imposing a block diagonal structure to the covariance matrix. This is done by partitioning the set of all flights, and each set of flights in the partition implies a demand cluster. A covariance matrix is then computed for each demand cluster, and the CAP is solved for each corresponding subset of flights. Details are provided next.

3 Solution Methodology

In this section, we discuss an algorithmic framework that decomposes the portfolio optimization problem into many smaller problems, and each of these smaller problems has its own demand cluster and subset of flights. The framework is illustrated in Figure 2. It starts by querying necessary information such as costs, flights, spot market forecast, allotments, and business requirements from a database. A resource allocation problem, the risk neutral problem discussed next, is solved first to yield a preliminary allocation solution. This solution is then used to provide a direction for a partitioning algorithm to generate a flight partition and a set of demand clusters. At the end, for each demand cluster, the covariance matrix is computed using historical bookings and shipping records of the demand units within the cluster, and CAP is applied to each corresponding set of flights to yield the final allocation that we evaluate though a simulator discussed later in Section 4.

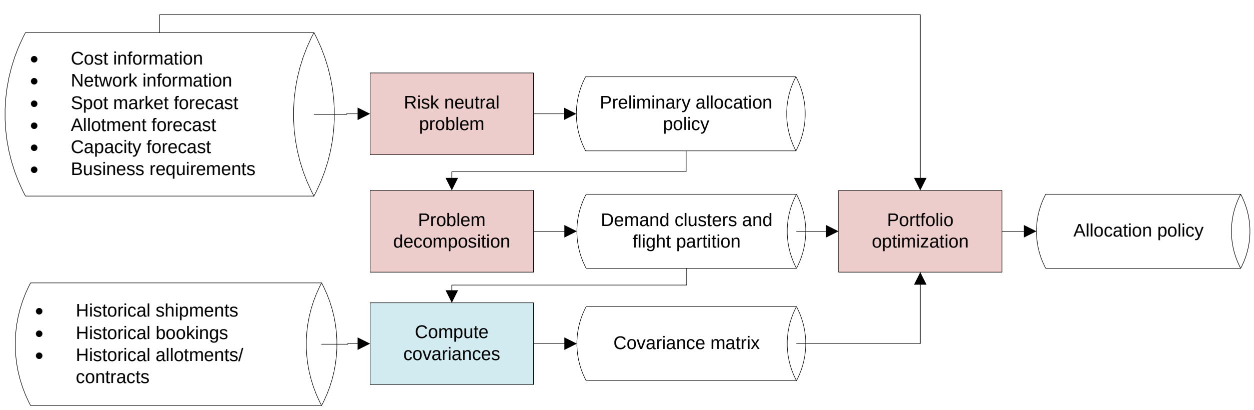

Figure 2: Proposed algorithmic framework for problem decomposition.

3.1 Risk Neutral Problem

In order to provide a direction to partition flights and cluster demand, the risk neutral problem (RNP) is solved. It is the same problem as the portfolio optimization problem, except that its objective is replaced by the left-hand side of constraint (2). The modified problem maximizes the total allocation revenue without accounting for demand covariances. Since RNP is to be solved over a long planning horizon, a rolling horizon approach is implemented. This is accomplished by changing the right-hand side coefficients of constraints (3), (4), and (6) to the remaining weight, volume, and number of positions, respectively. The complete model can be found in Appendix A. Let τ be the length of the time window. For a given starting time $t = 1, \ldots, T$, where T is the length of the time horizon. Only demand units and resources in $[t, t+\tau]$ and across time $t+\tau$ are considered. A demand is classified as across-time $t+\tau$ if it can be assigned to an itinerary that has a flight that departs before time $t+\tau$ and arrives after time $t+\tau$. Any demand units that have previously been accepted or rejected are excluded from the current time window, and are used to update the capacity upper bounds before the RNP for the current time window is solved. Once the incumbent problem is solved, only the solution at time t is kept before moving the starting time from t to $t+1$. When $t+\tau = T$, all allocation decisions from $T-\tau$ to T are kept.

3.2 Problem Decomposition

To decompose CAP into many smaller problems, we partition flights and cluster demand units simultaneously. Our strategy is to group demand units by their contribution evaluated based on the optimal allocation of RNP. The RNP can assign a demand unit to several different itineraries. If a demand unit is assigned to multiple itineraries, and all these itineraries span several flight sets, this demand unit is assigned only to the empty set. Otherwise, there is at least one itinerary of the demand unit that is contained in a flight set. Among all such itineraries, we select the itinerary with the highest revenue based on the RNP solution and assign the demand unit to the demand cluster corresponding to the underlying flight set of the itinerary. In addition, this demand unit is also assigned to the empty set. Note that a demand unit can be simultaneously assigned to a cluster and the empty set, but it cannot belong to two demand clusters at the same time. By the definition of demand clusters, flights for each demand cluster are linked by itineraries. Each flight set implies a demand cluster, and vice versa. In addition, there is an empty set demand cluster that does not correspond to a flight set. At the end, CAP is solved for each flight set with the covariance matrix computed based on the corresponding demand cluster.

Since our strategy groups demand units by their contribution, it does not guarantee that demand units are heavily correlated within a cluster, or that demand units are independent across clusters. Although an optimal clustering strategy would maximize the number of demand units that satisfy these two properties, it requires each demand pair to be first examined before the covariance can be used for one of the clustering criteria. The resulting long solution time and large storage requirement likely render such a strategy impractical. The advantage of our approach is that if two profitable demand units in the same cluster are heavily correlated, we are ensured that CAP has already accounted for their covariance when an allocation is made. Since high revenue demand units are likely assigned to a cluster, their dependencies are captured by CAP.

To describe the partitioning problem mathematically, we define the following:

- $DI = \{(d,i) : i \in I^D(d), d \in D\}$ set of all demand-itinerary pairs,
- F set of all flights,
- $DI(\overline{F}, \overline{D})$ set of demand-itinerary pairs that are formed by $\overline{D} \subseteq D$, and each demand unit in $\overline{D}$ uses flights exclusively in $\overline{F} \subseteq F$,
- K upper bound on the number of demand units in each cluster,
- $R(\overline{DI})$ total revenue from RNP computed based on a set of demand-itinerary pairs $\overline{DI} \subseteq DI$.

The decision variables of the partitioning problem are

- S total number of demand clusters (flight sets),
- F_s flight set corresponding to cluster $s = 1, \ldots, S$,
- $D_s(F_s)$ set of demand units assigned to cluster s given flight set F_s.

For a given solution, we define $DI(\emptyset) = DI \setminus \cup_{s=1}^{S} DI(F_s, D_s(F_s))$. The partitioning problem reads

$$\max_{\substack{S \\ F_s, D_s(F_s), s=1,\ldots,S}} \sum_{s=1}^{S} R(DI(F_s, D_s(F_s))) - R(DI(\emptyset))$$

subject to

$$|D_s(F_s)| \leq K \qquad s = 1, \ldots, S \tag{9}$$

$$D_s(F_s) \cap D_s(F_{s'}) = \emptyset \qquad s \neq s' \text{ and } s, s' = 1, \ldots, S \tag{10}$$

$$F_s \cap F_{s'} = \emptyset \qquad s \neq s' \text{ and } s, s' = 1, \ldots, S \tag{11}$$

$$\cup_{s=1}^{S} F_s = F. \tag{12}$$

The objective is to maximize the total contribution of the demand-itinerary pairs over all clusters minus the contribution of the demand-itinerary pairs in the empty set, which includes both the demand-itinerary pairs that span across multiple flight sets, and the remaining demand-itinerary pairs formed by demand units not in any cluster.

Given solution $\bar{\mathbf{x}}$ to CAP, we define $R(\overline{DI}) = \sum_{(d,i) \in \overline{DI}} r_{di} \sum_{u \in U^D(d)} \rho_{du}^w \bar{x}_{diu}$. Since the total revenue of RNP is fixed, maximizing the total contribution of demand-itinerary pairs is equivalent to minimizing the contribution of the empty set. Thus, without loss of generality, the objective can be rewritten as

$$\max_{S, F_s, D_s(F_s), s=1,\ldots,S} \sum_{s=1}^{S} R(DI(F_s, D_s(F_s))),$$

which simply maximizes the total contribution over all demand-itinerary pairs in the clusters.

This partitioning problem is a large and complex integer programming problem, which is intractable over the entire planning horizon. An efficient flight-based partitioning heuristic is developed to efficiently retrieve a good solution.

The heuristic iteratively includes profitable flights into a flight set in question until the number of demand units in the corresponding cluster exceeds K. It requires as input a feasible solution $\bar{x}_{diu}$ to RNP, set of all flights, and set of all demand-itinerary pairs. It starts a new flight set with the most profitable flight that has not yet been included to any other flight sets, where profitability of a flight is defined by the total revenue collected from the demand units assigned by RNP to the flight. The algorithm then iteratively adds profitable flights so that more profitable demand-itinerary pairs can be assigned to that flight set. At the end, it returns a flight partition and the corresponding demand clusters for each flight set in the partition. Note that the solution returned by the heuristic is feasible to the partitioning problem, and if $K = |D|$, we simply obtain a trivial partition that includes all flights.

To fully describe the heuristic, we define $G(\overline{F}, \overline{D})$ to be the set of demand units that are not in $\overline{D} \subseteq D$, and each demand unit in the set has been assigned by RNP to at least one itinerary that uses flights exclusively in $\overline{F} \subseteq F$. We also define L_i to be the set of flights in itinerary i. The heuristic is presented in Algorithm 1.

Algorithm 1 Flight-based Partitioning Heuristic

Require: $\bar{\mathbf{x}}$, F, and DI

1: Set $s = 1$
2: Set $\bar{D} = D$, $\bar{F} = F$ and $F_j = \emptyset$ for $j = 1, \ldots, |F|$
3: **while** $\bar{F}$ is not empty **do**
4: Let $f^* = \arg\max_{f \in \bar{F}} \sum_{(d,i) \in DI: f \in L_i, d \in \bar{D}} r_{di} \sum_{u \in U^P(d)} \rho^w_{du} \bar{x}_{diu}$
5: $F_s = F_s \cup \{f^*\}$
6: **repeat**
7: Set $\bar{f} = \arg\max_{f \in \bar{F}} \sum_{(d,i) \in DI(F_s \cup \{f\}), d \in \bar{D}} r_{di} \sum_{u \in U^P(d)} \rho^w_{du} \bar{x}_{diu}$
8: $F_s = F_s \cup \{\bar{f}\}$
9: **until** $|G(F_s, \bar{D})| > K$
10: $D_s(F_s) = G(F_s, \bar{D})$
11: $\bar{D} = \bar{D} \backslash G(F_s, \bar{D})$
12: $\bar{F} = \bar{F} \backslash F_s$
13: $s = s + 1$
14: **end while**
15: **return** F_j and $D_j(F_j)$ for $j = 1, \ldots, s$

Step 4 of Algorithm 1 selects the flight that contributes the most and initializes a new flight set with that flight. In steps 6 - 9, we enlarge the flight set by iteratively including flights that bring highly profitable demand-itinerary pairs. Step 10 updates the demand cluster $D_s(F_s)$ accordingly. At the end, the algorithm returns a flight partition and a demand cluster for each flight set in the partition, and its outcome defines the empty set $DI(\emptyset)$.

3.3 Portfolio Optimization

Once the flight partition and demand clusters are found, both the capacity of the flights and demand units in the cluster are adjusted accordingly by subtracting the portion of the demand on the demand-itinerary pairs in the empty set. The corresponding decision variables $\{x_{diu}\}_{u\in U^D(d)}$ are excluded before CAP is applied to each flight set to produce an allocation policy. The allocation policies over all flight sets are then combined to form the final allocation policy to be evaluated through a simulator, which we discuss next.

4 Simulation

We construct a simulator to evaluate the performance of the allocation policy returned by the CAP. The goal is to evaluate the quality of the partition and thus of the fact that only certain covariances are considered. The simulation is divided in two steps (see Figure 3).

Recall that the only uncertainty assumed is the difference between the weight in the bid and the actual tendered amount. The first step is the shipment generation process that generates shipment weight samples. We assume that weight is multinomial dependent. For spot market demand, we further assume that the density is constant. Hence, a weight sample implies a volume sample. For each allotment shipment, given a corresponding weight sample (we discuss how this is generated in the next paragraph), the weight of the sample is proportionally distributed over all originally requested ULD types in the bid, where the proportion is determined based on the required weights of the originally requested ULD types. Given the distributed weight for each ULD type, the weight is then converted to the required number of ULDs.

The shipment weight sample generation process ideally requires means (historical) and the covariances over all demand units. However, due to the fact that the covariance matrix is too large to be stored for generating multinomial shipment weight samples, we, instead of using the covariance matrices computed based on the existing clusters, swap demand units between clusters before shipment weight samples are generated. This demand swapping process provides a fair evaluation of the partitioning heuristic while avoiding the huge storage requirements associated with utilizing the entire covariance matrix. Specifically, for each cluster, the simulator first randomly and uniformly selects some specified percentage of demand units. The extracted demand units are then randomly and uniformly redistributed to other clusters. Note that the number of demand units in each cluster can now go above the upper bound on the number of demand units allowed in the cluster. After demand units are redistributed, a covariance matrix is computed for each new cluster. Shipment weight samples are then generated using a multivariate normal random number generator that takes the new covariance matrices as input. If demand units are not distributed, then sampling would favor our partitioning heuristic and provide a biased assessment. For this reason, we randomly swap demand units to sample "different parts" of the overall covariance matrix.

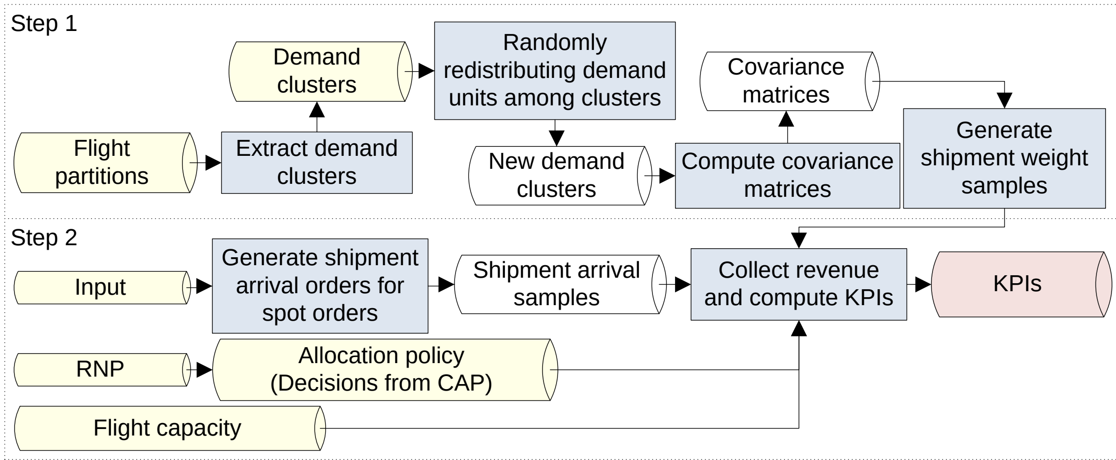

Figure 3: Simulator for Policy Evaluation

The second step generates the arrival orders of the shipments once shipment weight samples are obtained. The arrival of a shipment is determined based on the first date that the shipment is available to be shipped. Such a date can be identified by the earliest itinerary that can carry the shipment to its destination. The shipments are then grouped by the dates that they arrive, and if there are multiple shipments on the same date, the arrival orders of the shipments are randomized to generate the arrival samples.

With the given allocation policy and remaining flight capacity, the simulator collects revenue from each shipment sample by assigning its requested weight, volume, and number of ULDs to all available itineraries that are sorted by their operating costs in ascending order. A spot request might not be accepted by an allocation policy, yet in operations, there might be sufficient buffer capacity to accept it, where the buffer capacity is reserved to account for the volatility of the demand. In CAP, the buffer capacity is the capacity not allocated to any demand unit. Such capacity is possible since the objective of CAP is not to maximize the allocation revenue. Our simulation takes this into account by accepting spot demand without an allocation using the buffer capacity. This essentailly injects the first-come-first-serve (FCFS) policy into the simulator, and it is only appplied to demand units without an allocation.

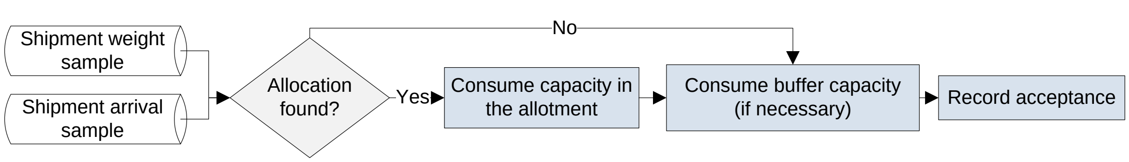

Figure 4: Revenue collecting process

Figure 4 shows the revenue collection process in detail. The process begins with a shipment and checks if an allocation for the shipment can be found from the existing allocation policy. If an allocation is found, its capacity is then reduced until either the demand is fully accepted or no residual allocation remains. In the latter case or when an allocation is not found, buffer capacity on each available itinerary is utilized to carry the remaining spot demand by randomly choosing compatible positions. This is shown by the upper arrow in Figure 4. At the end, the process records the acceptance information to compute the KPIs, and the acceptance information includes the amount of weight and volume, the number of ULDs accepted for each ULD type, and the number of positions consumed on each carrying flight.

The necessary KPIs used to measure the performance of an allocation policy are the total revenue, overtendered demand, and underutilized capacity. The revenue is computed based on the total shipped weight of the demand multiplied by the underlying rate, and it is captured by the left-hand side of constraint (2) instead of the risk-driven objective function of CAP. The overtendered demand is the positive difference of the shipped demand and the weight specified by the bid of an allotment or expected weight of spot demand. It measures how much of the overtendered demand is accepted and shipped not through the allocation policy. The underutilized capacity is the positive difference of the allocation produced by the policy and the utilized capacity. It measures if the allocation is sufficiently utilized. Although the best policy should maximize the total revenue while minimizing both the overtendered demand and underutilized capacity, in practice when underutilized capacity increases, overtendered demand tends to decreases, and vice versa. We discuss this observation further in the next section.

5 Computational Study

We test our proposed allocation policy using real-world data provided by a major cargo RM solution vendor. It has two weeks of cargo data with over 200,000 spot market demand forecasts, 100 allotments, 350,000 itineraries, and 28,000 flights. We benchmark our solution against the allocation policy used in practice, which is obtained by solving the risk-neutral problem (RNP, see Section 3.1) without considering any demand covariance. We want to study if the allocation policy obtained by discounting the risk-neutral revenue to minimize the allocation risk can perform better than a risk-neutral allocation policy that does not consider the risk component.

We solve the RNP in the rolling horizon manner with the rolling horizon window set to be 7 days (the problem is too hard to be solved in one attempt over 14 days). For the CAP, the revenue lower bound φ in constraint (2) is defined to be αz, where α is the revenue discount factor, and z is obtained by solving the risk-neutral version of CAP, which is a maximization problem with the left-hand side of the revenue lower bound (2) as the objective function.

An algorithm by Higham (2002) is implemented to find the nearest semi-positive definite covariance matrix, where the distance is measured by the two-norm. This is to handle covariance matrices with negative eigenvalues, which could happen due to rounding errors. In our experiments, this algorithm is run only for a few clusters, whose corresponding covariance matrices have almost negligible negative eigenvalues.

We present the computational study in three parts. In the first two parts, we benchmarked the CAP allocation policy against the RNP allocation policy. The first part demonstrates the performance of the CAP allocation policy as used in practice, and its performance is evaluated by feeding the simulator presented in Section 4 directly with historical shipments streams. Neither shipment weight and arrival samples are generated, nor clusters are perturbed. To evaluate the effects of changing the revenue lower bound, we vary the revenue discount factor α from 85% to 100% with an increment of 2.5%. Note that each value of α corresponds to a different allocation solution, and when α is 100%, the underlying solution is the RNP allocation policy that provides the baseline for comparisons. This range of α is selected as it is unlikely that a carrier is willing to trade more than 15% off its potentially achievable revenue (if all shipments are tendered as expected) to account for risk.

The second part studies both the performance of the CAP allocation policy with a broader range of α and the sensitivity of the allocation policy to the revenue lower bound and demand clusters. We ran the simulator to generate 100,000 demand-swapping samples. For each of these samples, the simulator generates 100,000 shipment weight and arrival samples. These large numbers of samples ensure that the effect of standard errors is minimized and the results are at least 95% statistically significant. We test α from 55% to 100% with an increment of 5%, and again, $\alpha = 100\%$ corresponds to the RNP allocation policy.

Furthermore, we test the sensitivity of the CAP allocation policy on the demand clusters produced by the partitioning heuristic described in Section 3. It is done by randomly swapping demand units across different clusters before the clusters are used to generate shipment weight samples. Recall from Section 4 that demand swapping is a strategy to avoid the use of the entire covariance matrix to generate shipment weight samples. We vary β, the distribution factor that controls the percentage of demand units to be randomly extracted from each cluster and uniformly distributed to other clusters, from 0% to 90% with an increment of 10%.

The last part reports the behavior of CAP when the revenue lower bound varies. Specially, we show how both the objective value and the risk-neutral revenue computed using (2) change when α decreases from 100%. Furthermore, computational times required for each process in Figure 2 are also reported.

All experiments were conducted on a server with a 64-bit Window 2003 server operating system. Its CPU is Intel Xeon(R) with four 2.67 GHz cores, and has 12 GB of RAM. The data was stored in an Oracle 11g database. The implementation was coded in Java, and ILOG Cplex 12.3 was used for optimization.

5.1 Historical Shipment Results

Table 1 summarizes the results when historical shipment records are directly applied to the simulator using the CAP allocation policies that correspond to various values of α. The revenue, overtendered demand, and underutilized capacity are direct outputs of the simulator, and all percentages are computed in comparison to the output of the RNP allocation policy.

Table 1: Performance summary when historical shipment records are directly applied.

Revenue Discount Factor	97.5%	95%	92.5%	90%	87.5%	85%
Revenue (%)	0.99	1.72	1.84	1.92	1.70	1.50
Overtendered Weight (%)	5.21	8.10	8.75	10.00	10.25	11.82
Overtendered Volume (%)	7.52	10.61	12.94	13.31	14.30	15.19
Underutilized Weight (%)	-1.84	-5.44	-7.45	-13.06	-15.76	-17.37
Underutilized Volume (%)	-0.68	-4.43	-8.94	-13.49	-14.76	-16.10

We observe that revenue can be improved close to 2% when α is set to be 90%. It means that it is worthwhile to reserve some capacity for risk buffering, and the improvement gradually decreases in both extremes of α. Decreasing revenue for larger α is due to the fact that the allocation policy gradually becomes the RNP allocation policy, which leaves no capacity unallocated. Decreasing revenue for smaller α is due to the heavier usage of the FCFS policy, i.e. without any control, as more capacity is set assigned for risk buffering.

Overtendered and underutilized weights go in the opposite direction as α increases. When α increases, allocations get reduced, and consequently, overtendered weight increases while underutilized weight decreases. The trends are similar for both weight and volume.

5.2 Simulation Results

We now present our simulation results and sensitivity analyses. Figure 5 shows the percentage of the revenue change over RNP by using the CAP allocation policy for each selected value of α and β presented in the beginning of this section.

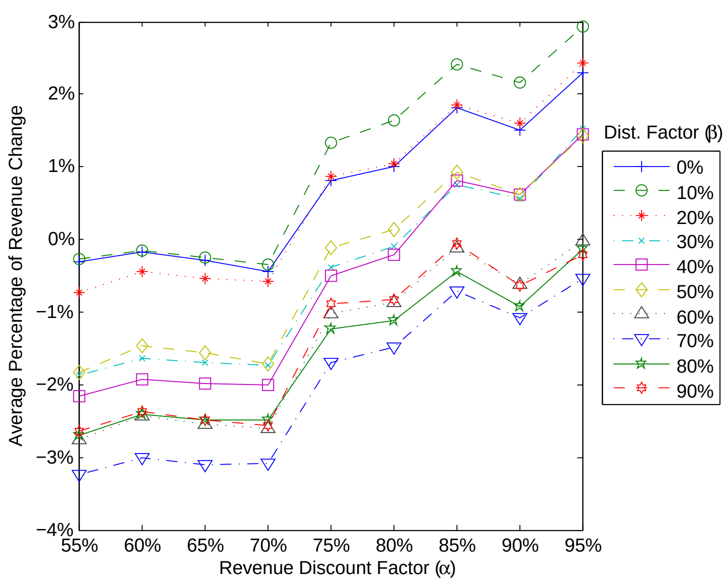

Figure 5: Average percentage of revenue change over RNP

In general, the revenue improvement gradually increases when α increases and β decreases. The trend of improvement is similar over all values of β. Several observations are worth mentioning.

1. The revenue change is not monotonic in α due to the heuristic nature of our approach.
2. Similarly, the revenue improvement is not monotonic in β. In fact, the best revenue improvement is observed when β is set to be 10%, and the worst revenue improvement is obtained when β is set to be 70%. These follow from the fact that the partitioning algorithm (Algorithm 1) is a heuristic, which does not cluster demand units by their correlation but by their risk-neutral revenue contribution. Nonetheless, our results show that our partitioning heuristic works well and is relatively robust to β, since even when the percentage of demand units being swapped is around 50%, and so long as α is at least 85%, the revenue improvement is about 1%.
3. When α is less than 75%, our allocation policy is outperformed by the risk-neutral allocation policy regardless of the value of β. The reason is that too much revenue is traded to reduce the total covariance, i.e., for anticipated uncertainty. This is also due to the fact that CAP naturally discourages allocation, and the unallocated capacity is consumed in the FCFS manner, i.e. without any control, and hence, the risk-neutral allocation policy ultimately prevails.
4. When α is no less than 85% and β is no more than 20%, we observe a revenue improvement about 2% on average by reserving a small amount of capacity and using it as a risk buffer, and as high as 3% of revenue improvement can be achieved, which could account for approximately $225,000 per week for major carriers.

We also measure the overtendered weight and underutilized capacity. Figure 6 shows the percentage of overtended weight change over RNP. In general, the percentage of overtended weight increases when less capacity is reserved. This is due to the fact that more demand units are pushed to be accepted via FCFS. When β increases, the percentage of overtendered weight increases, as the allocated capacity may not be well utilized due to demand swapping. Similarly to the average percentage of revenue change, the percentage of the overtendered weight increases the least when β is 10%. Specifically, about 11% of demand units are accepted on average via FCFS when α is no less than 85% and β is no more than 20%.

On the other hand, Figure 7 shows the percentage of underutilized weight change (a negative number corresponds to a reduction of underutilized capacity). In general, the percentage of undertended weight increases when α increases as more capacity is reserved for risk buffering. When more capacity is reserved, the allocation becomes lower, and hence, is more likely to be underutilized. Similarly, when β increases, the percentage of underutilized weight increases. The reason is that the allocated capacity is not well utilized due to demand swapping. Specifically, when α is no less than 85% and β is no more than 20%, the percentage of underutilized capacity can be reduced by more than 15%.

For both overtendered weight and underutilized capacity, the trends are similar among different values of β, and the same conclusions can be drawn when the overtendered demand and underutilized capacity are measured in volume.

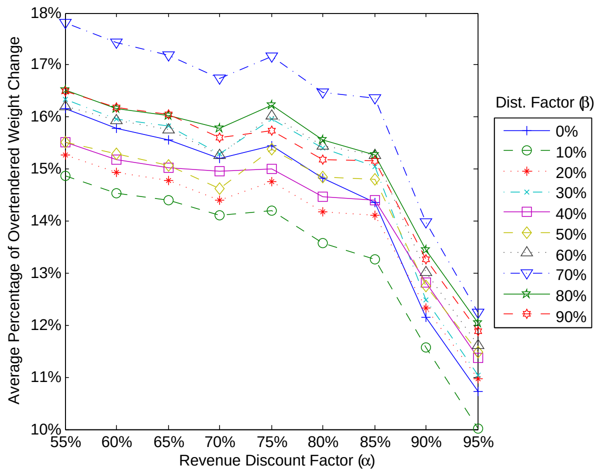

Figure 6: Average percentage change of overtendered weight

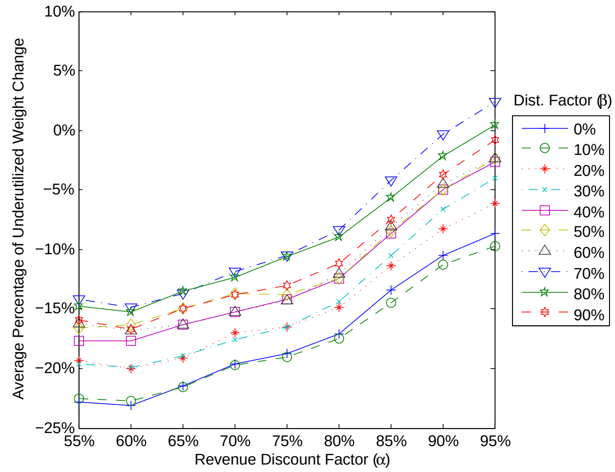

Figure 7: Average percentage change of underutilized weight

5.3 CAP Optimization Results

Lastly, we present results on how the objective of CAP and its total revenue, computed based on the left-hand side of constraint (2), change when different revenue lower bounds are set. Figure 8 illustrates the percentage of revenue reduced for different revenue discount factors, where the percentage is computed using the revenue obtained when α is set to be 100%. It shows that the revenue reduced decreases in a concave manner when α increases. This implies that as the revenue discount factor decreases, the allocation does not have to change as much in exchange for a lower total covariance, and similar revenue levels can be kept albeit decreasing slowly. This is due to the fact that the allocation policy, instead of experiencing substantial changes, is simply reduced to further minimize the objective of CAP when α gradually decreases. On the other hand, Figure 9 shows the percentage of the total covariance reduced when α varies. The covariance reduction curve behaves similarly to the revenue reduction curve that it is concave and decreasing in α. This is again resulting from the fact that reducing revenue lower bound further does not significantly change the underlying allocation policy, and hence, the total covariance is not reduced as much.

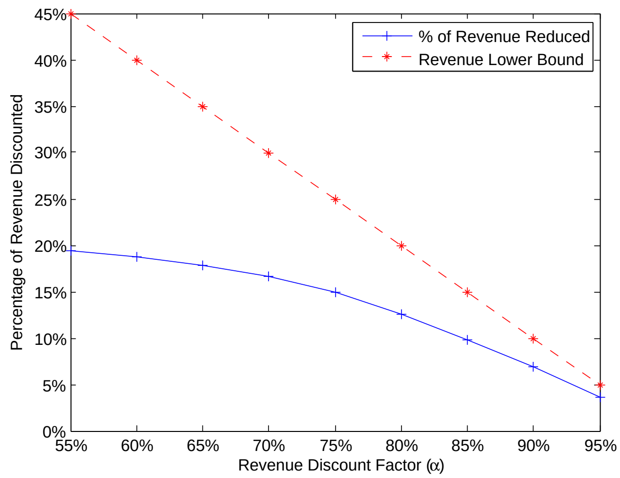

Figure 8: Percentage of revenue reduced

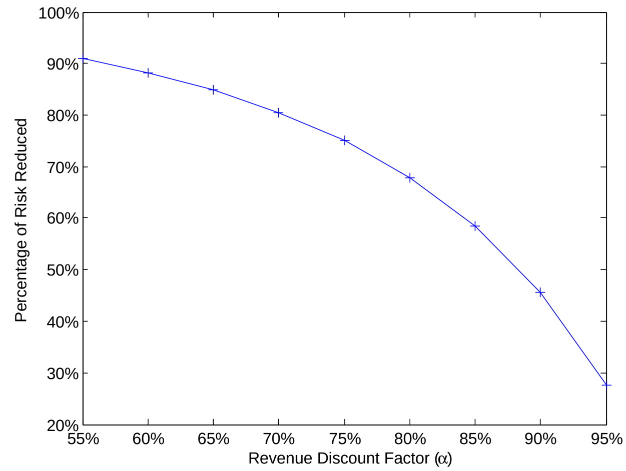

Figure 9: Percentage of covariance reduced

Finally, the running time is about 10 minutes for the risk neutral problem, 20 minutes for the flight partitioning algorithm, and 20 minutes for the CAP over all clusters. Note that the running time of CAP is only twice of the running time for the risk neutral problem. The reason is that multi-thread computing is applied to each demand cluster. However, no parallelization scheme is available to the risk neutral problem due to the solution dependency on the rolling horizon framework.

In summary, the running time for a six month period should be about 13 hours which shows that our proposed algorithm framework is practical.

6 Conclusion

In conclusion, we have developed a framework to solve a difficult optimization problem that considers the interaction between allotments and spot market demand, accepts demand at the ULD level, and allocates flight capacity at the aircraft position level. By conducting a set of comprehensive simulation experiments using a real world dataset provided by a major solution vendor, we demonstrate the practicality of our optimization framework, and show that revenue can be improved by 2% when interaction between demand can be captured, which can be translated to a substantial saving of $150,000 per week for major cargo carriers.

During this study, we have identified several future directions to extend this research. One interesting direction is to derive a way to obtain a bid-price at the aircraft position level. Another direction is to extend the problem to capture the stochastic nature of the demand, and derive an efficient algorithm to solve it. Lastly, improving the partitioning algorithm may provide a more robust solution to various parameters we have tested.

7 Acknowledgments

The authors are grateful that this work has been partially supported in large part by JDA Software Group Inc. The authors also thank Jamison Graff for providing the data and offering his cargo expertise.

REFERENCES

Amaruchkul, K., Copper, W. L., and Gupta, D. (2007). Single-leg air-cargo revenue management. *Transportation Science*, 41(4):457–469.

Amaruchkul, K., Copper, W. L., and Gupta, D. (2010). A note on air-cargo capacity contracts. *Production and Operations Management*. http://dx.doi.org/10.1111/j.1937-5956.2010.01158.x.

Brumelle, S. L. and McGill, J. I. (1993). Airline seat allocation with multiple nested fare classes. *Operations Research*, 41(1):127–137.

Chew, E.-P., Huang, H.-C., Johnson, E. L., Nemhauser, G. L., Sokol, J. S., and Leong, C.-H. (2006). Short-term booking of air cargo space. *European Journal of Operational Research*, 174(3):1979 – 1990.

Curry, R. E. (1990). Optimal airline seat allocation with fare classes nested by origins and destinations. *Transportation Science*, 24(3):193–204.

DHL (2012). Volumetric weight. [Online; accessed 8-October-2012]. http://www.dhl.com/en/tools/volumetric_weight_express.html.

Hellermann, R. (2004). *Capacity Options for Revenue Management: Theory and Applications in The Air Cargo Industry*. Springer, 1st edition.

Higham, N. J. (2002). Computing the nearest correlation matrix - a problem from finance. *IMA Journal of Numerical Analysis*, 22(3):329–343.

Karaesmen, I. (2001). *Three Essays on Revenue Management*. PhD thesis, Columbia University.

Kasilingam, R. G. (1997). Air cargo revenue management: Characteristics and complexities. *European Journal of Operational Research*, 96(1):36 – 44.

Levin, Y., Nediak, M., and Topaloglu, H. (2012). Cargo capacity management with allotments and spot market demand. *Operations Research*, 60(2):351–365.

Levina, T., Levin, Y., McGill, J., and Nediak, M. (2011). Network cargo capacity management. *Operations Research*, 59(4):1008–1023.

Luo, S., Çakanyıldırım, M., and Kasilingam, R. G. (2009). Two-dimensional cargo overbooking models. *European Journal of Operational Research*, 197(3):862 – 883.

Pak, K. and Dekker, R. (2004). Cargo revenue management: Bid-prices for a 0-1 multi knapsack problem. ERIM report series research in management, Rotterdam School of Management, Erasmus Universiteit Rotterdam, The Netherlands.

Popescu, A. (2006). *Air Cargo Revenue and Capacity Management*. PhD thesis, Georgia Institute of Technology.

RITA (2012). U.S. air carrier traffic statistics. [Online; accessed 8-October-2012]. http://www.bts.gov/xml/air_traffic/src/datadisp.xml.

Slager, B. and Kapteijns, L. (2004). Implementation of cargo revenue management at KLM. *Journal of Revenue & Pricing Management*, 3(1):80–90.

Talluri, K. and van Ryzin, G. (2004). *The Theory and Practice of Revenue Management*. International Series in Operations Research & Management Science. Springer, 1st edition.

Wollmer, R. D. (1992). An airline seat management model for a single leg route when lower fare classes book first. *Operations Research*, 40(1):26–37.

Appendix

A Rolling Horizon Implementation of the Risk Neutral Problem

We describe the modifications made to the risk neutral problem so that it can be solved in a rolling horizon manner. The modifications are based on adding a time dimension to the decision variables and coefficients. Let $[\underline{t}, \bar{t}]$, $1 \leq \underline{t} < \bar{t} \leq T$ be the time window in the rolling horizon context. We assume that every itinerary is shorter in duration $\bar{t} - \underline{t}$ days. To describe the problem, in addition to the notation defined for CAP, we add the following:

- $D(\underline{t}, \bar{t})$ set of demand units that can be shipped between time $\underline{t}$ and $\bar{t}$,
- w_{ft} remaining weight on flight f by time t,
- v_{ft} remaining volume on flight f by time t,
- n_{fpt} remaining number of type-p positions consumed on flight f by time t,

The rolling horizon version of RNP is

$$\max \sum_{d \in D(\underline{t}, \bar{t})} \sum_{i \in I^D(d)} \sum_{u \in U^D(d)} r_{di} \rho^w_{du} x_{diu}$$

$$\sum_{d \in D(\underline{t}, \bar{t})} \sum_{i \in I^{DF}(d,f)} \sum_{u \in U^D(d)} \rho^w_{du} x_{diu} + \sum_{p \in P(f)} \sum_{u \in U^P(p)} \tau^w_u y_{fpu} \leq w_{f\underline{t}} \qquad f \in F \tag{13}$$

$$\sum_{d \in D(\underline{t}, \bar{t})} \sum_{i \in I^{DF}(d,f)} \sum_{u \in U^D(d)} \rho^v_{du} x_{diu} + \sum_{p \in P(f)} \sum_{u \in U^P(p)} \tau^v_u y_{fpu} \leq v_{f\underline{t}} \qquad f \in F \tag{14}$$

$$\sum_{d \in D(\underline{t}, \bar{t})} \sum_{i \in I^{DF}(d,f)} x_{diu} \leq \sum_{p \in P(f)} y_{fpu} \qquad u \in U^F(f), f \in F \tag{15}$$

$$\sum_{u \in U^P(p)} \rho^p_u y_{fpu} \leq n_{fp\underline{t}} \qquad p \in P(f), f \in F \tag{16}$$

$$\mathbf{x}, \mathbf{y} \text{ integer.} \tag{17}$$

Constraints (13), (14), and (16) are similar to constraints (3), (4), and (6), which are the upper bounds on the remaining weight, volume, and number of positions respectively.