**A MATHEURISTIC APPROACH TO THE AIR-CARGO RECOVERY PROBLEM UNDER DEMAND DISRUPTION**

**ABSTRACT**

Air cargo transport is subject to unpredictable changes in expected demand, necessitating adjustments to itinerary planning to recover from such disruptions. This study proposes a matheuristic based on column generation in which each subproblem is solved using an ad-hoc heuristic. The approach is tested on demand disruption instances containing up to 75 air cargo orders with different penalty levels. The results show that within a two-hour time limit, the proposed method improves profit by 33% over the solution generated by a commercial MIP solver and by 9.53% over the solution with the routes fixed as in the original flight planning.

*Keywords:  Air Cargo Schedule Recovery; Airline Schedule Recovery; Disruption Management; Air cargo Rescheduling; Pickup and delivery, Column Generation, matheuristic*

**1. INTRODUCTION**

Air cargo can be carried either in dedicated freighters, that is, aircraft designed exclusively for cargo transport, or in the spare capacity in passenger aircraft bellies after baggage has been loaded. Although in recent years the annual added payload capacity in bellies has surpassed that of freighters by ever increasing margins (in 2015 it was almost 3 times greater) (IATA, 2015a), freighters have certain advantages that continue to make them indispensable to any air cargo operation. They can, for example, carry types of loads that passenger aircraft cannot, such as oversize items or dangerous cargo. Furthermore, the supply characteristics of bellies on passenger flights, either because of their schedules, markets served or spare capacity, do not necessarily coordinate well with the characteristics of air cargo demand (Boeing, 2014). These observations lie behind the analysis offered in the present study.

One of the principal factors differentiating air cargo transport from air passenger transport is the high level of uncertainty in demand for the former. The sources of this uncertainty are the very short time windows in which cargo bookings are made and the frequent occurrence of cargo no-shows or only partial show-ups (Wada et al., 2017). The problem typically arises just hours before a departure, causing an imbalance between planned supply and real demand. This results in losses for carriers that are reflected in low air freight load factors, which in 2015 averaged 45% compared to an average passenger load factor for the same year of 80.3% (IATA, 2015b).

To deal with these demand disruptions, which are difficult to anticipate, cargo airlines make focussed, last-minute adjustments to their schedule planning based on the experience of decision-makers charged with modifying schedules manually. A range of measures are employed, alone or in combination, such as flight cancellations, aircraft rerouting, flying aircraft empty to reallocate cargo (where doing makes economic sense), and adding intermediary stops or roundtrips. These actions must be adopted within short time frames and are local and corrective in nature, without attempting to design a new itinerary from scratch. In practice, however, they will affect other stages in the original scheduling such as aircraft rotation, crew assignments and cargo routing. The idea behind the use of minor corrections is to upset previously assigned crew shifts as little as possible, but these solutions are generally sub-optimal and directly impact carriers’ costs.

Schedule design in the air transport industry has been extensively studied for the passenger business. Etschmaier and Mathaisel (1985) were the first to review the state of the art in airline scheduling. Since that time, numerous works have been published on airline schedule planning, covering topics such as incremental design of passenger schedules (Barnhart et al., 2003; Berge and Hopperstad, 1993), the simultaneous schedule design and fleet assignment problem (Levin, 1971; Rexing et al., 2000; Lohatepanont and Barnhart, 2004) and disturbances or unexpected events. The latter have been studied from two different perspectives, the first one based on operational schedule recovery models (Bratu and Barnhart, 2006; Rosenberger et al., 2003) and the second one on the development of robust schedule planning models (Gao et al., 2009; Lan et al., 2006; Rosenberger et al., 2004; Pita et al., 2013; Froyland and Maher, 2014).

Although schedule recovery models have been analyzed in depth for passenger air transport, little work has yet been done on the recovery problem in the air cargo business. Disturbances in the passenger industry are of a different nature from those arising in the cargo market and are due mainly to weather or the propagation of delays caused by late or cancelled flights rather than demand fluctuations. Another major difference lies in the type of corrective actions taken when disturbances occur. For passenger transport, the legs of a route cannot be changed in the short run due to passenger preferences for certain itineraries, but with air freight, there is greater flexibility in the legs actually flown since what counts is that the cargo arrive at its destination by the stipulated deadline regardless of the route actually taken.

One of the first published studies in a cargo context is that of Marsten and Muller (1980), who propose a mixed integer programming model to solve two strategic problems, route design and fleet planning, for a hub-and-spoke network. Lin & Chen (2003) develop an embedded multi-commodity minimum cost flow problem (MCFP) that chooses transit (stopover) airports for the design of a cargo network linking China and Taiwan using belly space on passenger flights.

Yan et al. (2006) present a model that integrates the strategic processes of cargo airline route design and fleet assignment. By using a weekly planning horizon, the solutions can be applied in short-term operations. The authors formulate the problem as an MCFP assuming a single fleet. The idea is to integrate a flight network with a cargo network (O-D pairs) and design a route starting from zero independent of any base plan that considers all possible flight legs and incurs no flight cancellation penalties. The solution employs various heuristics based on the number of stops that must be made to pick up and deliver cargo to its destination. Yan and Chen (2008) extend this model to scenarios where there are alliances between airlines that require coordinated scheduling. Tang et al. (2008) devise a model that integrates route design for passenger, cargo and combi flights. The problem is formulated as an integer MCFP and a series of heuristics are proposed for solving it.

Derigs et al. (2009) address freighter route planning using a formulation that simultaneously optimizes flight choice, aircraft rotations and cargo routing. The emphasis in the authors’ approach is on incrementally improving schedules through flight choice changes. Continuing in this vein, Derigs and Frederichs (2013), after surveying the general problem of cargo flight planning and its subproblems, then turn their attention to the air cargo scheduling problem. They present a model that integrates several stages including itinerary design, fleet assignment, and aircraft and cargo routing. However, the planning generated by the model is confined to an externally predefined list of mandatory and optional flights, the latter incorporated only to the extent they generate profits.

What emerges from the above review is that the problem of cargo route adjustment in response to last-minute demand disruptions has not been treated in the literature at the operational level. This problem has a number of special characteristics: time is a limited resource, there is no predefined flight list and minor adjustments to the base planning scheme must be doable. The principal contributions of the present article that distinguish it from prior studies may be summed up as follows:

1. A model is developed that can react to cargo demand disruptions in the short run. This capability has not previously been addressed and the proposed formulation is not comparable to reactive models of passenger air transport where demand disruptions are of a different nature, caused typically by weather and flight delays or cancellations.
2. The proposed model simultaneously handles aircraft routing, cargo routing and changes in the base plan, while attempting to impact crew scheduling as little as possible through the use of penalties for base plan modifications. This approach imposes constraints not generally found in planning problems, which typically resort to cyclic itineraries that require, for example, that flights begin and end at specific airports within given time periods.

The remainder of this paper is organized into four sections. Section 2 models the problem to be solved, setting out the notation and the main assumptions. Section 3 describes the proposed solution method, including the column generation technique and the heuristic developed to solve each subproblem. Section 4 reports the computational results for a set of air cargo instances with demand disruption used to test the model. Finally, Section 5 presents our conclusions.

**2. THE MODEL**

The proposed problem is a variant of the pickup and delivery problem with time windows (PDPTW) (Dumas et al., 1991; Cordeau et al., 2002; Parragh et al., 2008). Adjustments and constraints have been added to the classic formulation to adapt it to the case of air cargo. In this new version, customer cargo bookings (hereafter “orders”) are continuously divisible and can therefore be delivered by multiple planes, each one picking up a part of the order. Each node in the cargo network is an event representing a takeoff, landing or stopover at a given airport with its corresponding loading and unloading of an order or parts thereof. An order is defined as a pickup node where the cargo specified in the order is loaded and a delivery node where it is unloaded, both nodes located at an airport and with its respective time window and loading/unloading time.

The arcs connecting the nodes are ground arcs (representing airport stopovers) if both are located at the same airport, and flight legs if they are located at different airports. Each aircraft has a subnetwork it can operate over and which is delimited by various operating restrictions or landing rights that prevent it from operating freely over the entire airport network. The fleet is heterogeneous, with each plane having a given load capacity, initial and final locations at either end of the planning horizon and a base plan to which adjustments must be as few as possible.

In what follows we set forth the main assumptions of the model, its notation and mathematical formulation, and then introduce the decomposition method for solving it.

**2.1. Assumptions**

The proposed model makes the following assumptions:

1. Loading and unloading times are deterministic and specific to each type of order to be carried.
2. Cargo cannot be transferred between aircraft (i.e., no transhipments).
3. Orders are continuously divisible, thus allowing the cargo for a given order to be transported by various aircraft, each one picking up a partial order.
4. Load capacity restrictions are expressed in terms of weight, in keeping with industry practice.
5. No penalties are applied for orders in the base plan that are partially or fully undelivered within the planning horizon.

**2.2. Notation**

**Sets**

|  |  |  |
| --- | --- | --- |
| *K* | : | Set of aircraft, |
| *P* | : | Set of orders . Let . The pickup node for order *i* is then and its delivery node is *.* |
|  | : | Set of orders that can be carried by aircraft *k*, . |
|  | : | Set of pickup and delivery nodes that can be visited by aircraft *k*, including the initial and final location nodes {} at the start and end of the planning horizon, respectively. |
|  | : | Set of arcs that can be flown by aircraft *k*, , where . Note that the arcs connecting pickup or delivery nodes with the aircraft’s initial and final locations, denoted () and (), are fictitious arcs that do not represent the transport of any cargo but are included to ensure the routes end at the right airport as required by each aircraft. |
|  | : | Set of predecessor nodes to *i* for aircraft *k*, . |
|  | : | Set of successor nodes to *i* for aircraft *k*, . |
|  | : | Set of pickup and/or delivery nodes constituting a macronode . A macronode consolidates all the pickup and/or delivery nodes for the orders to be transported by a given aircraft to or from a single airport and thus defines an origin or destination node for the corresponding flight leg in the base plan. |
|  | : | Set of macronodes. |
|  | : | Set of flight legs in the base plan for aircraft *k*, indexed as where . |

**Parameters**

|  |  |  |
| --- | --- | --- |
|  | : | Cost of flying arc/flight by aircraft [US$/arc]. |
|  | : | Penalty cost imposed for a modification to flight of aircraft . Modifications may include putting on additional crew or bringing in crew from other airports, among other possibilities [US$/flight]. |
|  | : | Freight rate for transporting order [US$/tonnes]. |
|  | : | Travel time between node *i* and  *j* for aircraft *k* [hrs]. |
|  | : | Loading or unloading time at node [hrs]. |
|  | : | Earliest moment at which node can be visited [hrs]. |
|  | : | Latest moment at which node can be visited [hrs]. |
|  | : | Total cargo demand for order [tonnes]. |
|  | : | Aircraft load capacity [tonnes]. |

**Decision variables**

|  |  |  |
| --- | --- | --- |
|  | : | Binary variable, equal to 1 if aircraft *k* flies arc , otherwise 0. |
|  | : | Time at which aircraft *k* begins its visit at node *i*[hrs]. |
|  | : | Amount of cargo assigned to node *i.* Represents the cargo picked up () if node *i* is a pickup node, and the cargo delivered () if node *i* is a delivery node [tonnes]. |
|  | : | Amount of cargo carried by aircraft *k* when departing node [tonnes]. |
|  | : | Auxiliary variable, equal to 1 if flight leg is cancelled, otherwise 0. |

**2.3. Mathematical model**

The mathematical formulation of the proposed model is set out below.

|  |  |  |  |
| --- | --- | --- | --- |
| subject to: | | | (1) |
|  | |  | (2) |
|  | |  | (3) |
|  | |  | (4) |
|  | |  | (5) |
|  | |  | (6) |
|  |  |  | (7) |
|  |  |  | (8) |
|  | |  | (9) |
|  | |  | (10) |
|  | |  | (11) |
|  | |  | (12) |
|  | |  | (13) |
|  | |  | (14) |
|  | |  | (15) |
|  | |  | (16) |
|  | |  | (17) |
|  | |  | (18) |

The objective (1) of the model is to maximize profit, that is, the revenue received on the set of orders less the operating costs and any penalty costs imposed for base plan flight modifications. Constraints (2) and (3) together trigger a penalty each time a base plan flight is cancelled or operated by a different aircraft. Constraints (4) and (5) impose that each aircraft k starts (ends) at the initial (final) node of the planning horizon. Constraint (6) ensures continuity between consecutive nodes of a feasible route. Constraint (7) sets the arrival time for each node. Constraints (8), (9), (10) and (11) relate to the load capacity restrictions for aircraft *k*. Constraint (12) sets the maximum amount of cargo that can be picked up at a given pickup node. Constraint (13) ensures that the amount of cargo picked up at a pickup node is the same as the amount delivered at the corresponding delivery node. Constraint (14) obligates an aircraft visiting a pickup node to visit the corresponding delivery node. Constraint (15) imposes that a pickup node is visited before the corresponding delivery node is visited. Constraint (16) restricts arrival times for each node to their time windows. Finally, constraints (17)-(18) define the nature of the variables.

The model is not linear in constraints (7), (8) and (10). For (7) and (8), the non-linearity is corrected by introducing big-M parameters as proposed by Cordeau (2006). Constraint (10), on the other hand, can easily be replaced by a set of linear restrictions. Lastly, although the flight crew is not directly modelled as a constraint, it is implicitly present in the objective of making the least possible number of modifications to the base plan by minimizing total penalty costs (the third component in the objective function).

**3. SOLUTION METHOD**

Since the problem as set out above in (1)-(18) is NP-hard (Savelsbergh and Sol, 1995), solving it exactly would be very costly in computation time. This could well mean the exact solutions the model eventually finds in response to last-minute changes in cargo demand would not be generated quickly enough to be useful in the real-world applications the model is intended for. Some other approach is therefore required.

Various solution methods have been developed for solving PDPTW problems, most of which have been either exact (Sigurd et al., 2004, Ropke et al., 2007; Ropke and Cordeau, 2009) or metaheuristics (Ropke and Pisinger, 2006; Qu and Bard, 2012; Cherkesly et al., 2015). For our PDPTW-variant problem we propose to combine the two methods in the form of a matheuristic algorithm, a technique that combines the use of heuristics with mathematical programming models (Maniezzo et al., 2010). Since the problem has a block structure, we can solve it using the Dantzig-Wolfe decomposition (Dantzig and Wolfe, 1960). In the present case, this means the formulation is first decomposed into a restricted master problem(RMP) and a set of *k* subproblems (KSPs), each of which generates feasible routes for a given aircraft. The linear relaxation of the RMP is then solved using the column generation approach (Wolsey, 1998; Barnhart et al., 1998; Desrosiers and Lübbecke, 2005; Feillet, 2010) while the KSPs are tackled under a heuristic approach. The definitive solution employs the approach suggested by Xu et al. (2003), which consists in solving a restricted version of the RMP that includes only those columns generated by solving the linear relaxation.

**3.1 General Framework**

The formulations of the restricted master problem and the kth subproblem are set out below.

**3.1.1 Restricted Master Problem (RMP)**

The sets, parameters and additional variables in the RMP are as follows:

**Sets**

|  |  |  |
| --- | --- | --- |
|  | : | Set of feasible routes for aircraft *k.* |
|  | : | Set of nodes visited on route *r.* |
|  | : | Set of orders corresponding to the nodes visited on route *r*. |

**Parameters**

|  |  |  |
| --- | --- | --- |
|  | : | Represents the route generated by the *k*th subproblem; a vector defined as |
|  |  |  |
|  | : | The profit associated with route *r* of aircraft *k*; a vector whose elements are the coefficients of each family of variables , defined as |
|  |  |  |
|  |  | The dot product is therefore the value of the term in the original objective function for aircraft *k* evaluated in the solution or route generated by the *k*th subproblem. |
|  | : | The cargo at pickup node assigned to aircraft *k* flying route *r*; part of the family in vector . |

**Variables**

|  |  |  |
| --- | --- | --- |
|  | : | A binary variable, equal to 1 if route *r* is assigned to aircraft *k*, otherwise 0. |

The RMP can now be formulated as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | sujeto a: | | (19) |
|  | |  | (20) |
|  | |  | (21) |
|  | |  | (22) |

The objective (19) of the RMP can be described as a convex linear combination of the values of the solutions or routes generated by the subproblems. The subproblem solutions are used to generate a feasible solution of the original problem. Constraint (20) is a linking condition imposing the interdependence of the different subproblems, restricting cargo demand at each pickup node for order in analogous fashion to constraint (12) in the original problem. Constraint (21) is known as the convexity constraint and may be interpreted as a restriction that ensures each aircraft *k* is assigned only one route *r* from the set of routes it can feasibly fly. Constraint (22) expresses the binary nature of the decision variable, which must be relaxed to facilitate the solution of the problem and thus obtain the dual values that are then entered into the *k* subproblemas.

Each time the relaxed RMP is solved, we obtain the dual values for each pickup node and for each subproblem. may be interpreted as the opportunity cost of transporting an additional unit of order *i* before a unit of any other order while may be interpreted as the profit on the optimal route generated for aircraft *k* up to that point.

**3.1.2** **The *k*th subproblem (*k*-SP)**

Each satellite problem, hereafter denoted k-SP, attempts to find the optimal route for a given aircraft *k* and consists of all of the constraints in the original problem except linking condition (12).

|  |  |  |
| --- | --- | --- |
|  |  | |
|  | subject to: | (23) |
|  | (2) - (11) y (13) -(18) | |

Objective function (23) represents the reduced costs ( in the original problem. The solution obtained is a feasible route for aircraft *k* without considering the demand constraints. If the OF value or reduced cost given by the solution to (23) is greater than 0, the route is beneficial for the problem and is incorporated into the RMP.

Since the search for an exact solution of the RMP would involve an exponential number of columns (every feasible route for each aircraft), we propose to solve it using a column generation approach in which the RMP has a limited number of initial columns and new columns are added as the *k-*SP subproblems are solved heuristically. Then for any *k*-SP, if the heuristic solution for the final route generated is a positive reduced cost, the corresponding column is added to the RMP and the contrary case is discarded. The column generation algorithm is iterated until no more subproblems are found that have generated columns with positive reduced cost. Once this procedure is completed, the RMP is solved (imposing the binary variable constraint) for a restricted version of the problem that includes only those columns generated up to that point. This solution methodology is summarized in Figure 1.



Figure 1: Solution algorithm flow diagram.

Since at the first iteration no columns have yet been added to the RMP, a set of initial columns must be generated. Three such sets are in fact generated for each aircraft as follows:

1. *Base Plan*. This set consists of three columns for each aircraft. One of them is the base plan itself while the other two are obtained by applying to it respectively two local search operators, one denoted drop and add, the other swap, as will be described below in Section 3.2.2. If the base plan is no longer feasible after disruptions have occurred, it is made feasible by adjusting the time windows and loads.
2. *Deadheading*. This set of columns is for the case in which an aircraft flies directly from the initial to the final airport ([]) without taking any orders.
3. *Additional Columns*. This set consists of three columns for each aircraft. One of them is a route specified by a construction heuristic while the other two are obtained by applying to the route respectively the same two operators applied for the base plan column set.

The construction heuristic generates routes for all of the aircraft in parallel. This is done to get routes that are more balanced (Labadie et al., 2016). The heuristic proceeds by first adding an order to each of the routes, then adding a second one, and continuing iteratively until all orders have been routed. Each new order *i* so added is inserted at the end of the route, its pickup node *i+* first and its delivery node *i-* immediately thereafter. In each case, the new order is the one with the lowest insertion cost. In formal terms, this procedure is as follows: Let *j* be the last order inserted into route *k* so far, and *Nk* be the number of pending orders that can be carried by aircraft *k*. Then the next order to be inserted will be the one that satisfies and the time window and load capacity constraints. When the procedure terminates, each aircraft *k* will have a route of type . There may, however, be some orders that have not been routed, in which case the procedure attempts to insert both their pickup and delivery nodes into the positions that generates the greatest profit.

Seven initial columns are thus generated for each aircraft. In other words, the RMP begins with 7*k* columns.

The heuristic for solving the subproblems is described in detail in what follows.

**3.2. Description of *k*-SP heuristic**

The *k*-SP heuristic proceeds in two major sequential stages: initialization and local search. In the initialization stage, an initial feasible route is created for each subproblem. This route is then subjected to various modifications in the local search stage by different operators whose probability of being chosen for the task is updated over time. The local search stage terminates when either the maximum number of iterations Ω is reached or all of the operators have been executed with no further improvements in the objective function.

A simplified flow diagram of the heuristic algorithm is shown in Figure 2. Below we explain in detail the two stages, the operators used in the local search stage and the updating of each operator’s probability of being chosen.

*#Construction of the initial feasible solution*

BestSolution = Initialization()

Best OF = OF(BestSolution) #*Best objective function*

[All operators initialized with equal probability]

#Local search stage

**while** (stopping criterion not satisfied) **do**

[Operator chosen according to probabilities]

CurrentSolution = Operator(BestSolution)

[Probabilities of all heuristics updated]

**if** (OF(BestSolution) > BestOF) **then**

BestSolution = CurrentSolution

BestOF = OF(CurrentSolution)

**end**

**end**

Figure 2:Simplified flow diagram of proposed heuristic algorithm.

**3.2.1 Initialization**

In the initialization stage, the heuristic randomly chooses an initial route for each *k-*SP from among the feasible routes for the corresponding aircraft *k* inserted up to that point into the RMP (i.e., the RMP columns) and recalculates the chosen route’s profit as given by the objective function for *k-*SP (Eq. 23). Thus, as the heuristic advances through the iterations, the set of candidate routes for the initial solution increases in number.

**3.2.2 Local search**

In the local search stage, certain routes are modified without taking account of the existence of routes created for other aircraft up to that point. Interaction with other subproblems takes place in the process of solving RMP, which supplies the dual values for the subproblem’s objective function.

Seven different operators are utilized to modify the initial route (column) added to the RMP. To be accepted, a modification must improve the level of profit as given by the OF. Four of the operators change the positions of orders on a single route, the fifth exchanges an order with an unassigned one and the remaining two change the number of orders either by eliminating one already on the route or inserting a new one. This is explained in greater detail below in Section 3.2.2.1.

The choice of operators is made randomly on the basis of their weighted efficiency , defined as follows:



where is the weighted efficiency of operator *h* in the previous iteration and is its efficiency in the current iteration, the latter given by



where is the percentage improvement in the objective function value achieved by using operator *h*, is that operator’s execution time and is a very small value included so that an operator that generates no improvement in current iteration has a non-zero probability of being chosen in the next one. The parameter is defined by the modeller to give more or less weight to the operator’s current or historical efficiency. Finally, the probability of choosing an operator *h* is defined as



This formulation increases the probability of choosing operators that have achieved high levels of weighted efficiency in improving the OF value.

**3.2.2.1 Operators**

The operators are described in detail below.

Intra-route operators:

These four order operators are the intra-route operators used in Cherkesly et al. (2015). The intra-route request exchange operator modifies the positions of two orders *i* and *j* on a given route by exchanging pickup position *i+* with *j+* and delivery position *i-* with *j-*. Intra-route request relocate attempts to relocate order *i* to its best position on the route. The remaining two operators, intra-route multiple request exchange and intra-route multiple request relocate, exchange and reposition truncated routes, i.e., subroutes of an order. The truncated route for an order *i* is the subset of a route whose initial and final nodes are that order’s pickup and delivery nodes. It also satisfies the conditions that the aircraft departs the initial node with no other cargo than that for order *i* and departs the final node with no cargo at all.

Swap:

The swap operator, based on the swap routed and covered customers operator developed by Talerian and Salari (2015), exchanges an order belonging to route *k* with another order belonging to the list *Lp* of pending orders. These latter are orders that have not yet been incorporated into a route, and are ordered from higher to lower unit profit. For an order *i*, unit profit is given by , where is the freight rate for *i* and is the dual of the cost derived from Eq. (20) in the RMP problem. For a given route *k*, this operator proceeds through the entire list *Lp* searching, for each order *i Lp* , its best exchange option with order *j* on route *k*, where “best” means the option that most improves total route profit. Note that in any swap of orders between *k* and *Lp*, neither of the two change in size.

Drop\_and\_Add:

This operator, also proposed by Talerian and Salari (2015), increases the length of the route by inserting orders into route *k* from the pending order list *Lp*. The operator inserts each order *i Lp* in its best position, where “best” means the position that most improves total route profit.

Delete:

This operator proceeds through each route *k*,eliminating orders one by one to test whether doing so improves the route’s total profit.

Note that although the operators described above modify the sequence of nodes that are visited, they do not optimize the nodes’ cargo assignments. Since the proposed model allows aircraft to pick up part of an order from a node and the nodes can be visited by more than one aircraft, a cargo assignment (CA) method is needed. The method must assign an amount of cargo () to each node *i* on each route *r* generated by the operators while satisfying the load capacity restrictions (8)-(11) in the original problem. It does this by greedily assigning each of the nodes its full demand sequentially, in the order in which the aircraft visits them. If this results in the aircraft’s load capacity being exceeded, the cargo reassignment (CR) linear optimization problem shown below is executed to redistribute the cargo among the nodes. The method thus has a hybrid function. This sequential approach, first applying the greedy algorithm and then the CR program only if capacity constraints are violated, is much faster than simply applying the LP to every case.

|  |  |  |  |
| --- | --- | --- | --- |
| (CR) |  | | (27) |
|  | subject to: | |  |
|  |  |  | (28) |
|  |  |  | (29) |
|  |  |  | (30) |
|  |  |  | (31) |
|  |  |  | (32) |
|  |  |  | (33) |
|  |  |  | (34) |
|  |  |  | (35) |
|  |  |  | (36) |

where is the set of orders that correspond to the nodes visited on route *r*. The objective of this problem is to maximize profit on the transport of the set of orders associated with the route. Constraint (28) prevents any cargo from being assigned to the initial and final nodes on the route associated with the start and end of the planning horizon. Constraint (29) ensures that the cargo at the pickup and delivery nodes for each order is the same. Constraints (30) and (31) impose that the cargo associated with a pickup be non-negative and less than the order’s total demand. Constraints (32) and (33) restrict the continuity of the cargo. Constraint (34) is the aircraft load capacity restriction. Finally, (35) and (36) define the nature of the variables.

**4. COMPUTATIONAL RESULTS**

The proposed matheuristic algorithm was coded in Python 2.7, and Gurobi 6.5 was used to solve the optimization models (RMP) at each iteration. The different instances were run on a computer with an Intel Core i5 2.6 GHz processor and 8 GB of RAM. For every instance tested, the number of iterations was Ω=35, the values of the operator weighted efficiency parameters were and , and the size of the fleet of aircraft was set at 7.

**4.1 Benchmarks**

To test the performance of the proposed matheuristic, we compared its results with two benchmarks: i) MIP, the solution of the optimization model (1)-(18) using Gurobi’s MIP solver; and ii) FIX Route, the solution of the model with the routes fixed as in the base plan. The latter benchmark in effect measures the impact of disruptions on the base plan when only the amount of each order to be carried can be optimized. Since the model is intended to be used in a real operating context in which decisions must be made within short time periods, a solution time limit of 2 hours was imposed. Both benchmarks were solved using the Gurobi solver.

Given the random nature of the choice of initial columns and of operators in the local search stage of the heuristic, it is highly likely that successive runs of the same instance will produce different results. Also, in a certain period of time – say, one hour – the heuristic may be executed more than once. A stopping criterion was therefore imposed, consisting of whichever of the following three conditions is satisfied first: i) , the maximum number of global runs, reaches 10; ii) Δ, the number of global iterations with no improvement in the objective function value, reaches 3; iii) τ, the maximum run time, reaches 3,600 seconds.

**4.2 Instances**

The testing of our model was carried out using a modification of the AA instances in Ropke and Cordeau (2009) for the PDPTW. A total of 10 demand disruption instances were tested, the number of orders varying between 30 and 75. Each instance has a planning horizon of T=600, a time window width of W=60 and an aircraft load capacity of CAP=15. For each instance a square space is assumed measuring [0,50]x[0,50] and the initial and final locations are chosen randomly. The base plan is then drawn up reflecting the initial condition of the orders and the aircraft locations at the start and end of the planning horizon. The instances with disruptions are constructed from the changes to the initial condition, which may be either of two types: i) a modification in the demand (weight) of the original orders, and ii) the appearance of new orders. The modifications in the original orders are determined by a factor . If , the order is unchanged; if , the order changes and if , the order disappears, that is, falls to zero (in the wake of the disruption). In the case of a change, the new demand is given by where .

For each instance, new orders are generated where and <1. For each new order, both its pickup and delivery locations are also generated according to a uniform distribution over the square space and demand is . To define the time windows for order *i*, the earliest time () is first set for the pickup node by a uniform distribution on the interval [0,T-], where is the direct travel time between the pickup and delivery nodes, and the latest time () is then calculated as . The time windows for the delivery node are then given by and .

Since in the case of a disruption, the cargo airline has the option of not taking an order or taking it only partially if it is not profitable, a separate freight rate is generated for each one. For this purpose it is assumed that the cost of carrying a complete order *i* from its pickup node to its delivery node () is a percentage θ of the total customer freight charge *Fi* (, where . Thus, the freight rate for *i* is given by .

The values for the parameters in our tests were set at (0.3,0.85,0.7,1.2,0.2,0.05,0.2).

Note that even if an order disappears (demand falls to zero), it must still be taken into account given that it is part of the base plan. This may occur in situations where the flight crew of the aircraft go off duty at a specific airport or the aircraft must undergo maintenance.

A summary of the 10 instances is given in Table 1, showing for each one the numbers of original, new, disappeared and modified orders.

Table 1: Characteristics of instances in model tests.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Original** | **New** | **Disappeared** | **Modified (weight)** |
| AA30 | 30 | 5 | 4 | 12 |
| AA35 | 35 | 6 | 2 | 18 |
| AA40 | 40 | 5 | 8 | 13 |
| AA45 | 45 | 2 | 9 | 20 |
| AA50 | 50 | 2 | 7 | 24 |
| AA55 | 55 | 3 | 7 | 26 |
| AA60 | 60 | 9 | 8 | 25 |
| AA65 | 65 | 11 | 10 | 26 |
| AA70 | 70 | 3 | 9 | 36 |
| AA75 | 75 | 3 | 17 | 30 |

Finally, each of the 10 instances was tested with 3 alternative penalty values (=20,15,10) representing greater or lesser degrees of flexibility in changes to the base plan. Thus, for the 10 instances there were 30 test cases.

**4.3 Analysis of results**

The results for the test cases are set out in Table 2. For each one, the first column lists the type of instance (AA), the number of orders and the penalty value (S=20,15,10). Columns 2 through 4 give the results for the MIP benchmark while columns 5 through 8 and 9 through 11 do the same respectively for the proposed matheuristic and the FIX Route benchmark. In each case, the values reported are the objective function value (i.e., profit), the solution time, and the gap between the OF value and an upper bound defined as the value found by the MIP in two hours of run time. The figure shown for the gap is the difference between these two values as a percentage of the upper bound. Note regarding the latter that it is not necessarily the best integer solution of the problem. In the case of the matheuristic, the number of global iterations is also given in the table. The best solution for each instance among the three approaches is indicated in bold type.

Table 2: Comparison of model test results for 10 instances, each with 3 penalty values.

****

As can be seen, in 22 of the 30 cases the matheuristic clearly produced the best OF value while in two additional cases (AA30\_S20 and AA45\_S20) it equalled that of the MIP. In four of the remaining six cases, all of which had few orders (30, 35, 40), MIP performed best but the matheuristic averaged just 0.63% below it and in no case was more than 1% lower. The two remaining cases (AA70\_S15 and AA70\_S10) were the only ones in which FIX Route outperformed the others. Over all 30 cases, the matheuristic OF values reveal that the profits it obtained averaged 33% higher than MIP and 9.53% higher than FIX Route. The average gap for the matheuristic was just 5.89%, much smaller than for MIP (31.15%) and FIX route (14.96%).

Regarding the effect of the different penalty values, the MIP results generally deteriorated as the penalty was reduced. Even FIX Route produced higher OF levels than MIP for cases of more than 35 orders when the penalty S=15, and more than 30 orders with S=10. The reason for these results might be that the smaller is the penalty, the less restrictive is the problem in the sense that there are more alternative routing possibilities, making the solution more difficult to find. This in turn may explain why a reasonably good solution within the run time limit might not be found. Indeed, there were cases such as AA45\_S15 and AA65\_15 where MIP simply did not identify a feasible integer solution at all within the time limit.

As for solution times, the matheuristic was able to execute multiple global iterations for smaller cases of up to 45 orders. As the number of orders increased, the number of global iterations declined, falling to just one for the largest cases. As can be seen in the table, there were many cases where the solution time exceeded the 3,600-second limit. This was due to the functioning of the stopping criterion. The procedure checks the criterion before embarking on a new global iteration, and if the time is below 3,600 seconds the iteration will start and continue till completed before the criterion is checked again. This meant that there were instances (e.g., AA45\_S15) that executed more than one global iteration and ended up with solution times that were over the limit.

As regards the benchmarks, MIP managed to solve to optimality only in four cases within the time limit while FIX Route achieved the best times simply because it solved a restricted version of the original problem in which cargo was routed across a network with fixed flights.

These test outcomes taken together demonstrate that the proposed matheuristic delivered better results than either of the two benchmarks within the imposed time limit.

**5. CONCLUSIONS**

A solution approach was proposed for the problem of last-minute adjustments to cargo flight schedules made to recover from disruptions in short-run demand. It is built around a reactive model based on the pickup and delivery problem with time windows (PDPTW) that is designed to be used with real-world air cargo operations. The formulation simultaneously solves the flight scheduling, aircraft routing and cargo routing stages while mitigating crew scheduling impacts through the imposition of penalties for modifications to flight leg planning.

To solve the model, a matheuristic algorithm was developed that combines heuristics with mathematical programming techniques. It employs the column generation method in which subproblems are solved heuristically in order to determine the optimal route for each cargo aircraft.

The performance of the matheuristic was measured on 10 instances with disruption of demand that vary from 30 to 75 cargo orders. Each instance was tested at three different penalty levels for departures from the base flight planning, giving a total of 30 test cases. The results of the matheuristic were compared to those generated by two benchmarks, one being the direct solution of the model using a commercial MIP solver and the other simply being the solution with the routes fixed as in the original flight planning. To reflect the real context of air cargo decision-making, maximum run time was set at 2 hours. It was found that the matheuristic performed best in 24 of the 30 cases, with an average profit over all 30 that was 33% greater than for MIP and 9.53% greater than for the fixed (original planning) routes. In the 6 cases where the matheuristic result was not the best, the gap between it and the benchmarks was never more than 1%. The foregoing suggests that the proposed approach combining mathematical programming and heuristics holds significant promise for use as an operational tool in making cargo flight planning decisions under demand uncertainty and the need for rapid response.

Future research will aim at developing a robust planning approach for cargo flights that can readily incorporate changes in demand in the construction of efficient itineraries.

**REFERENCES**

Barnhart, C., Johnson, E., Nemhauser, G., Salvelsbergh, M. and Vance, P. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3), 316-329.

Barnhart, C., Belobaba, P., and Odoni, A. (2003). Applications of operations research in the air transport industry. *Transportation Science*, 37(4), 368–391.

Berge, M. and Hopperstad, C. (1993). Demand driven dispatch: A method for dynamic aircraft capacity assignment, models and algorithms. *Operations Research,* 41(1). 153-168.

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

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

Cherkesly, M., Desaulniers, G. and Laporte, G. (2015). A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading. *Computers and Operation Research*, 62, 23–35.

Cordeau, J.-F., Desaulniers, G., Desrosiers J., Solomon M.M. and Soumis F. (2002). VRP with time windows. In: Toth P, Vigo D (eds.), *The Vehicle Routing Problem, SIAM monographs on discrete mathematics and applications* (pp. 157-193). Philadelphia: SIAM.

Cordeau, J.-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3), 573–586.

Dantzig, G. and Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research,* 8(1), 101-111.

Derigs, U., Friederichs, S. and Schäfer, S. (2009). A new approach for air cargo network planning. *Transportation Science*, 43(3), 370–380.

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

Desrosiers, J. and Lübbecke, M.E. (2005). *Column Generation* (1ra. ed.). New York: Springer.

Dumas, Y., Desrosiers, J. and Soumis, F. (1991). The pickup and delivery problem with time windows, *European Journal of Operational Research*, 54, 7–22.

Etschmaier, M. and Mathaisel, D. (1985). Airline scheduling: an overview. *Transportation Science*, *19*(2), 127–138.

Feillet, D. (2010). A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR***,** 8(4), 407–424.

Froyland, G., Maher, S. and Wu, C.L. (2014). The recoverable robust tail assignment problem. *Transportation Science,* 48(3), 351–372.

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

IATA (2015a). Cargo eChartbook Q3.

IATA (2015b). Air Freight Market Analysis, December 2015.

Labadie, N., Prins, C., & Prodhon, C. (2016). *Metaheuristics for Vehicle Routing Problems*. John Wiley & Sons.

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

Levin, A. (1971). Scheduling and fleet routing models for transportation systems. *Transportation Science*, 5(3), 232–255.

Lin, C. and Chen, Y. (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.

Lohatepanont, M. and Barnhart, C. (2004). Airline schedule planning: integrated models and algorithms for schedule design and fleet assignment. *Transportation Science*, 38(1), 19-32.

Maniezzo, V., Stützle, T., Voß, S. eds. (2010) Matheuristics – Hybridizing Metaheuristics and Mathematical Programming. Annals of Information Systems, vol. 10. Springer

Marsten, R. and Muller, M. (1980). A mixed-integer programming approach to air cargo fleet planning. *Management Science*, 26(11), 1096–1107.

Parragh, S. N., Doerner, K. F. and Hartl, R. F. (2008). A survey on pickup and delivery problems. *Journal Für Betriebswirtschaft*, 58(1), 21–51

Pita, J., Barnhart, C. and Antunes, A. (2013). Integrated flight scheduling and fleet assignment under airport congestion. *Transportation Science*, 47(4), 477–492.

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

Rexing, B., Barnhart, C., Kniker, T., Jarrah, A. and Krishnamurthy, N. (2000). Airline fleet assignment with time windows. *Transportation Science*, 34(1), 1–20.

Ropke, S. and Pisinger, D. (2006). An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, *40*(4), 455–472.

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

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

Rosenberger, J., Johnson, E. and Nemhauser, G. (2003). Rerouting aircraft for airline recovery. *Transportation Science*, 37(4), 408–421.

Rosenberger, J., Johnson, E. and Nemhauser, G. (2004). A robust fleet-assignment model with hub isolation and short cycles. *Transportation Science*, 38(3), 357–368.

Savelsbergh, M. and Sol, M. (1995). General pickup and delivery problem. *Transportation Science*, 29(1), 17.

Sigurd, M., Pisinger, D. and Sig M. (2004). Scheduling transportation of live animals to avoid the spread of diseases. *Transportation Science*, 38, 197–209.

Tang, C.-H., Yan, S. and 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.

Wada, M., Delgado, F. and Pagnoncelli, B. (2017). A risk averse approach to the capacity allocation problem in the airline cargo industry. *Journal of the Operational Research Society*, 68(6), 643-651

Wolsey, L. A. (1998). *Integer Programming* (1ra. ed.). New York: Wiley.

Xu, H., Chen, Z.-L., Rajagopal, S. and Arunapuram, S. (2003). Solving a practical pickup and delivery problem. *Transportation Science*, 37(3), 347–364.

Yan, S., Chen, S. and 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.

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