Integrated Airline Planning
Rivi Sandhu (sandhu@uiuc.edu)
Diego Klabjan (klabjan@uiuc.edu)
Department of Mechanical and Industrial Engineering
University of Illinois at Urbana-Champaign
Urbana, IL

Abstract
The tactical planning process of an airline is typically decomposed into several stages among which fleeting, aircraft routing, and crew pairing form the core. In such a decomposed and sequential approach the output of fleeting forms the input to aircraft routing and crew pairing. In turn, the output to aircraft routing is part of the input to crew pairing. Due to the decomposition, the resulting solution is often suboptimal. We propose a model that integrates these three stages and thus yields a simultaneous solution to all three problems. We design two solution methodologies to solve the model. One is based on a combination of Lagrangian relaxation and column generation while the other one is the Benders decomposition approach. We give computational experiments for a variety of instances obtained by a major carrier.

1 Introduction
Airline business processes related to tactical planning consist of schedule planning, fleeting, aircraft rotation, and crew pairing (see e.g. Klabjan (2003a)). In schedule planning, a set of flights with specific departure and arrival times is constructed. Next is fleeting, which assigns an equipment type (such as Airbus 320, Boeing 737-500, etc.) to each individual flight. The objective of the fleet assignment model (FAM) is to maximize profit subject to the number of available aircraft and other operational constraints. The problems that follow decompose based on the fleeting solution, i.e. there is a separate problem for every equipment type or fleet. The aircraft rotation problem or aircraft routing is to find a set of generic aircraft routes that satisfy maintenance requirements. The crew pairing optimization follows. In crew pairing a set of crew itineraries or pairings is constructed. The goal is to minimize the crew cost and each flight must be covered by exactly one pairing. A few weeks before the day of operations, the actual tail numbers are assigned to each flight and monthly crew rosters or bidlines are assigned to every individual crewmember. Throughout tactical planning, revenue management related processes match demand with supply, which is the sit capacity or the number of sits of a given equipment type.

At present in scholarly publications, only selected two of these problems are modeled and solved as a single integrated problem, e.g., fleeting and aircraft routing, and aircraft routing and crew pairing. In practice, all three models are solved sequentially and the output of one stage is the input to the next stage. Clearly, there is interdependency between the various stages. For example, the fleeting solution decomposes the problems that follow by equipment type. The aircraft rotation and crew pairing problems are then solved over a subset of flights pertaining to a single equipment type. This is due to the fact that each cockpit crew is qualified to fly a particular set of equipment types. But this dependency of crews on equipment types is not captured in fleeting. In the aircraft routing stage decisions are made without considering the impact on the quality of the crew pairing solution. The interaction between these two stages is in the fact that a crew can have a connection shorter than a predefined number only if it stays on the same aircraft. Solutions obtained by using this sequential
methodology can therefore be suboptimal. Ideally all the tactical planning problems should be solved as a single large-scale problem.

We present an integrated approach for these three most important stages. We propose a model that considers fleeting, aircraft rotation, and crew pairing simultaneously. Pairings are modeled explicitly and the rotation problem is captured by the plane count constraints. When aircraft maintenance requirements are easily fulfilled due to the structure of the underlying flight network, our model completely integrates these three stages. We use two different solution methodologies to solve the integrated model. The first methodology entails solving the model by a combination of Lagrangian decomposition, Fisher (1985), and column generation. Column generation is used to price out favorable pairings based on the Lagrangian multipliers. By using Lagrangian relaxation some constraints are penalized and are moved to the objective function. These penalties or multipliers are then used in pricing. The second solution methodology uses Benders decomposition, Benders (1962), on the model that relaxes integrality of pairing variables. The traditional FAM model along with Benders cuts from the crew pairing LP relaxation forms the restricted master problem (RMP). Based on the incumbent fleeting solution in each iteration, for each fleet we solve the crew pairing LP relaxation and add a Benders cut to the RMP. The main contributions of our work are as follows.

- The model, which integrates fleeting, aircraft routing, and crew pairing. Fleeting and crew pairing aspects are accurately captured including fleet families (sets of crew compatible fleets), the aircraft count, etc. The aircraft routing is captured via the concept of plane count constraints, while the maintenance routing constraints are not explicitly modeled.
- The application of solution methodologies to the problem specific nature of the model. We apply Benders decomposition, and a combination of Lagrangian relaxation and delayed column generation.
- The use the constrained shortest path algorithm in delayed column generation, where additional labels are required to capture different equipment types.
- Substantial profit improvements obtained by our integrated approach.

In Section 2 we present a brief review of fleeting, aircraft routing, and crew pairing. The integrated model is presented in Section 3. Section 4 outlines both solution methodologies. Finally, in Section 5 we present the computational experiments. We conclude the introduction with a brief review of the optimization techniques that are used later.

**Delayed Column Generation, Lagrangian Relaxation, and Benders Decomposition**

Large-scale linear programs are often solved by *delayed column generation*. In this algorithm, at every iteration, only a subset of columns is considered. In this context as well the problem with only a subset of columns is called the restricted master problem. In every iteration of the algorithm, first the RMP is solved and let \( \pi \) be the optimal dual vector, which for ease of discuss we assume it exists. Next the so called subproblem is solved. In subproblem solving we identify a set \( S \) of columns with the lowest reduced cost with respect to \( \pi \). If we cannot find a column with negative reduced cost, then we stop since \( \pi \) is an optimal dual solution to the original problem and together with the optimal primal solution to the RMP we have an optimal primal/dual pair. Otherwise, we append columns in \( S \) to the RMP and the entire procedure is iterated. When the RMP includes too many columns after several iterations, columns with large reduced cost are removed from the RMP.

Frequently the most computationally intensive step in delayed column generation is subproblem solving since it needs to scan many columns and typically it is a complex task to generate a single one. When columns correspond to constrained paths in a network, an efficient algorithm known as
constrained shortest path is often employed, see e.g. Desaulniers et al. (1998). In this case, the task is to find the cheapest cost \( s-t \) path (reduced cost in delayed column generation framework) among all paths with certain properties or constraints. We explain the algorithm by an example. Assume we want to find a shortest path with respect to the reduced cost in a network subject to the duration and the number of arcs in paths being below a given number. By duration we mean that every arc has an associated transit time and the duration of a path is the sum of transit times along the path. We introduce label vectors, which in this case have 3 coordinates. The first one corresponds to the reduced cost, the second one to the duration, and the last one to the number of arcs. With every node we associate a set of label vectors. For example, a label vector \((-45,134,4)\) at node \( i \) corresponds to an \( s-i \) path with reduced cost -45, duration 134, and 4 arcs. The constrained shortest path algorithm uses the same framework as standard shortest path algorithms. Suppose the algorithm selects a node \( i \) for scanning. The constrained shortest path algorithm next scans all neighbors \( j \) of \( i \) and all label vectors \( k = (k_1,k_2,k_3) \) at \( i \). Each label vector \( k \) is updated by traversing the arc \((i,j)\) and the updated label vector is appended to node \( j \). In our example, the label update means that the new label vector has \( k_3+1 \) as the third component, \( k_2 \) plus the transit time of arc \((i,j)\) as the second component, and \( k_1 \) plus the “reduced cost” of arc \((i,j)\) as the first component. The key observation is that under some realistic assumptions label vectors that are dominated can be discarded. If we have two label vectors \( \vec{k}, \vec{k} \) at node \( j \) and \( \vec{k} \leq \vec{k} \) component-wise, then the \( s-j \) path corresponding to \( \vec{k} \) is not going to be part of the shortest path. The efficiency of the algorithm depends heavily on the frequency at which dominance occurs. Note that if there is no dominance, the algorithm simply enumerates all paths. It turns out that dominance occurs often in practice and therefore the algorithm is computationally efficient.

Lagrangian relaxation, see e.g. Fisher (1985), is a widely used technique for solving large-scale integer programs. Suppose we can partition constraints into “easy” and “difficult” constraints. The concept behind this classification is that if the difficult constraints are removed, the resulting problem is easily solvable. In Lagrangian relaxation, every difficult constraint gets a linear penalty and it is moved to the objective function. The resulting problem is called the Lagrangian relaxation and it is a function of the penalties. Let us assume that we have a maximization problem. For any given values of penalties, the Lagrangian relaxation is computationally easy. It is easy to see that it always provides an upper bound on the optimal solution. The goal now is to find the best upper bound, i.e., to minimize the Lagrangian relaxation over all possible penalties. This is the Lagrangian dual problem, which is a nonlinear optimization problem. In practice it is solved by variants of the subgradient algorithm. One drawback of this approach is that there is no guarantee to find feasible solutions. They have to be constructed heuristically during the execution of the subgradient algorithm. The algorithm is very appealing since it is easy to implement and it handles complex (difficult) side constraints.

Benders decomposition, see e.g. Benders (1962), is well suited for mixed integer programs with linking integer variables. It requires that for any fixed value of integer variables, the resulting problem is an LP, where the constraint matrix is often block diagonal. The algorithm at every iteration solves a mixed integer program with a single continuous variable, which is the RMP, that provides a bound on the optimal solution. Next the linear program resulting from the original problem by fixing integer variables to the values from the RMP is solved. The optimal dual vector to this LP provides a Benders cut, which is added to the RMP and the procedure is repeated.
2 Traditional Models

2.1 The Fleet Assignment Model

We assume that a daily flight schedule is given, i.e. every flight repeats every day of the week. The daily approximation is reasonable since for many carriers the majority of the flights have this property. In addition, a vast majority of the research publications and the airline practice focus on the daily problem.

Fleeting is to find an assignment of equipment types or fleets to flights in a given flight schedule while maximizing profit, and subject to assignment constraints (each flight must be assigned to a fleet), flow balance (every aircraft that lands must take off), and plane count constraints (not to use more aircraft than there are available). The input is the flight schedule (obtained by the previous stage of schedule planning), the different equipment types, and the available number of aircrafts for each equipment type. The objective function in FAM has two components – revenue and operating cost. The revenue is typically calculated based on the average fare for each leg. The operating cost consists of the costs associated by using a specific equipment type for a flight (fuel cost, landing fee, aircraft depreciation costs, etc.). We also incorporate the cost of spilled passengers for each fleet and flight in the operating cost. Spilled passengers are those passengers that would like to receive service on a flight but are denied due to sit capacity or revenue restrictions. Since the model uses average fare for each leg to compute the revenue, such a model is also known as the leg-based fleet assignment model or traditional FAM.

We next describe the FAM model, which is an important component of our integrated model. We first explain the underlying network required for model description. Consider a station \( o \). An activity or event presents either a landing or a takeoff event. For the departure of flight \( l \), let \( t_l \) be the departure time. For the arrival of flight \( l \), let \( t_l \) be the arrival time plus the minimum aircraft turn time \( m_t \) (typically around 30 minutes), called also the ready time. The ready-time-space network (RTN) has a node \((o,s)\) for every station \( o \) and every activity \( s \) at this station. There is a flight arc between every departure and arrival event of the same flight. For every station \( o \) we order the activities based on \( t_l \), i.e. \( t_1 \leq t_2 \leq \cdots \leq t_n \), where \( n \) is the number of activities at the station (see the left figure in Figure 1). The network has a ground arc \( g = ((o,s_i), (o,s_{i+1})) \) for \( i=1,2,\ldots,n-1 \). In addition, there is a wraparound arc between the first and the last node of the time horizon. Let \( G \) denote the set of all ground arcs. The ground arc time interval is the time between the two activities that define the ground arc.

The FAM related variables are the binary fleet assignment variables \( x \) and the ground arc variables \( z \). For each flight arc \( l \) and each equipment type \( f \), \( x_{fl} \) is 1 if flight \( l \) is assigned to equipment type \( f \). The nonnegative variable \( z_{gf} \) counts the number of planes on the ground of equipment type \( f \) during the ground arc \( g \) time interval. Let \( MD \) be a fixed time, which corresponds to a time with low activity at any station, e.g. 3 am. The FAM model reads
\[
\max \sum_{f \in F} r_{\beta \ell} x_{\beta \ell} \\
\sum_{f \in F} x_{\beta \ell} = 1 \quad \ell \in L \\
\sum_{i \in O(v) \cap f} x_{i \ell} + z_{a(v)f} - \sum_{i \in I(v) \cap f} x_{i \ell} - z_{i(v)f} = 0 \quad v \in V, \, f \in F \\
\sum_{i \in M} x_{i \ell} + \sum_{g \in W} z_{g \ell} \leq N_f \quad f \in F
\]

where

- \( I(v) \): set of flights to node \( v \) in RTN
- \( O(v) \): set of flights from node \( v \) in RTN
- \( M \): set of flights in the air at \( MD \)
- \( N_f \): number of aircraft in fleet \( f \)
- \( W \): set of ground arcs which contain \( MD \)
- \( r_{\beta \ell} \): profit of assigning fleet \( f \) to leg \( \ell \).
- \( L \): set of all flight arcs
- \( F \): set of all fleets
- \( V \): set of all nodes in RTN
- \( i(v) \): ground arc to node \( v \) in RTN
- \( o(v) \): ground arc from node \( v \) in RTN

Constraints (1) require that exactly one equipment type is assigned to each leg, (2) preserve the flow of aircraft (an aircraft that lends must take off), and (3) count the number of aircraft. The left-hand side of (3) captures all the aircraft that are serving a flight at time \( MD \) and all the aircraft that are on the ground at time \( MD \). This traditional FAM model is described thoroughly in Hane et al. (1995).

Clarke et al. (1994) embed limited crew and maintenance considerations to the FAM. An important extension is to capture multi-leg passenger itineraries in the revenue component, Kniker (1998), Barnhart et al. (2002a), Barnhart et al. (2002b). An extensive literature survey on fleeting models can be found in Klabjan (2003a).

### 2.2 Aircraft Routing

An aircraft route is a sequence of flights flown by the same aircraft or tail number. A routing or rotation is a set of aircraft routes, which partition all the flights in the schedule. The FAA imposes four types of maintenance checks. Three of them require a significant overhaul and are not captured in the aircraft routing problem. The so-called A checks must be performed every 3 to 4 days and typically have to be satisfied by a rotation. The aircraft routing problem also known as the maintenance routing problem consists of finding a maintenance feasible rotation (A checks plus additional requirements imposed by individual airlines). It must also ensure that the rotation does not use more than the available number of aircrafts, and clearly it must cover every flight exactly once. Sometimes the so-called big cycle constraint is imposed, which enforces equal aircraft wear. Since the fleeting solution decomposes the problem according to equipment types, the aircraft routing problem is solved for each fleet separately.

Since it is difficult to assign a cost to a given aircraft route, the problem is often treated only as a feasibility problem. There is a vast literature on the topic and we refer the reader to the survey work by Klabjan (2003a) for more detailed information. The problem is modeled either as a set partitioning problem, where columns correspond to maintenance feasible sequences of flights linking two maintenance stations, Barnhart et al. (1998a), or as an Eulerian tour problem, which in turn leads
to an asymmetric traveling salesman problem with side constraints, Clarke et al. (1997). A third formulation is based on multicommodity flow approaches, Cordeau et al. (2000). We do not provide these formulations since they are irrelevant to this work.

2.3 Crew Pairing

The crew pairing problem is to find a subset of pairings or crew itineraries that partition all the flights in the network while minimizing the crew cost. The input to this stage is the fleeting and the aircraft routes obtained in the previous stages. Similarly to the aircraft routing problem, there is a separate crew pairing problem for each fleet family. A fleet family is a crew compatible set of fleets, e.g. 737-100, 737-200, etc., 737-900 all belong to the same fleet family. A pairing is a sequence of flights that satisfies several requirements. Clearly the destination station of a flight should be the same as the origin station of the next flight in the sequence. The origin station of the first flight should be the same as the destination station of the last flight in the sequence and it should be a crew base. A crew base is a designated station where crews are stationed. A pairing is composed of several duties, where a duty is a working day of a crew and is made up of a sequence of flights. A duty is subject to a number of FAA and union rules. Some of the duty legality rules are the maximum and minimum sit connection times between two consecutive flights, the maximum flying and elapsed time in a duty and many more. A crew can violate the minimum sit connection time, i.e. the time between two consecutive flights in a duty, only if it stays on the same aircraft. (This is the only reason of aircraft routing being an input to crew pairing.) The cost of a duty is the maximum of three quantities: the flying time, a fraction of the elapsed time and the minimum guaranteed pay.

A pairing must also satisfy a large number of regulatory rules like the minimum and maximum rest time between two consecutive duties, the elapsed time, upper bound on the flying time, the federal aviation rules of the FAA, etc. The cost of a pairing is also the maximum of three terms: the sum of the duty costs, a fraction of the elapsed time and the minimum guaranteed pay. The problem is usually modeled as a set partitioning model. The decision variables $y_p$ are equal to 1 if pairing $p$ is part of the solution, and 0 otherwise. The cost of a given pairing $p$ is denoted by $c_p$. The resulting model reads

$$\min \sum_p c_p y_p$$

$$\sum_{l \in p} y_p = 1 \quad \text{for each flight } l$$

$$y_p \text{ binary}$$

(4)

We denote by $l \in p$, where $l$ is a leg and $p$ a pairing, that leg $l$ is covered by pairing $p$ or included in pairing $p$. The side constraints, which are not given in (4), typically capture manpower requirements.

The crew pairing problem is difficult to solve due to the following two reasons. The number of pairings and thus variables is in the order of billions even for a medium size fleet family of 200 flights. Also, the calculation of the cost of a pairing is very complex within dynamic generation of pairings and a large number of complicated rules need to be taken into account while generating pairings. Due to the large number of variables, delayed column generation is employed. The pricing problem is traditionally solved as a constrained shortest path problem. Approaches based on finding the $k^{th}$ shortest path or depth-first search enumeration of pairings on a network have also been proposed. The underlying network, which is used to generate pairings, is detailed in Section 4.1.1. More details on the crew pairing problem and a survey of literature on crew scheduling is provided by Barnhart et al. (2003).
2.4 Crew Pairing with Plane Count

Klabjan et al. (2002) propose the following way to integrate aircraft routing and crew pairing. Instead of first obtaining aircraft routes and then crew pairings, they propose to reverse these two problems, i.e. first solving the crew pairing problem and then the aircraft routing problem.

In traditional crew pairing, pairings are generated based on the minimum sit connection time \(ms\), unless the crew follows the aircraft turn. In the absence of maintenance constraints, Klabjan et al. (2002) show that to integrate aircraft rotation and crew pairing it suffices to add the so-called plane count constraints to the set partitioning formulation of the crew pairing problem. Suppose we consider pairings with the minimum sit time equal to \(mt\). Pairings from a solution with a connection shorter than \(ms\) (but by definition larger than \(mt\)) imply a plane turn. Such connections are called *forced turns*. A set of forced turns can be extended into a plane count feasible rotation if and only if the number of planes on the ground at any time imposed by the forced turns does not exceed the plane count obtained from the corresponding ground arc value in the FAM solution. Consider the following illustrative example. Assume that the FAM solution specifies that there are 2 planes of a selected fleet on the ground at a time corresponding to a ground arc \(g\). If the crew pairing solution uses 3 pairings with a forced turn that includes \(g\), then the crew pairing solution implies 3 planes on the ground. This contradicts the plane count imposed by the FAM solution and in turn leads to an overall increase in the number of aircraft. Recall that a forced turn implies a plane turn.

The plane count constraints forbid such occurrences. Let \(P\) be the set of all pairings covering any subset of legs in the flight schedule with minimum sit connection time equal to \(mt\). Let \(P_g\) be the set of pairings, which have a forced turn that includes \(g\) and let \(a_{pg}\) be the number of times pairing \(p \in P_g\) includes \(g\). Note that in the daily problem pairings wrap around in time (24 hours) and therefore a pairing can include a ground arc several times. The plane count constraints from Klabjan et al. (2002) read

\[
\sum_{p \in P_g} a_{pg} y_p \leq b_g \quad g' \text{ ground arc},
\]

where \(b_g\) is the number of aircraft on the ground in the time interval defined by ground arc \(g\).

If these constraints are added to the crew pairing problem (4), then the implied forced turns can always be extended into a plane count feasible aircraft rotation. If maintenance feasibility is easily achievable, which is the case for large hub-and-spoke flight networks, then this approach integrates the aircraft routing and the crew pairing problem.

2.5 Literature Review

Some recent attempts have been made to integrate the various stages in airline planning. All of these attempts either only integrate two consecutive stages or they capture the crew pairing problem only at a very high level. Barnhart et al. (1998c) present a model that to some extent integrates FAM and crew pairing. They consider duties and not pairings in long-haul operations. The drawbacks of their model are that due to the large number of constraints, it is hard to solve, and in addition, pairings are approximated by duties. Barnhart et al. (1998a) propose a model for integrating FAM and the aircraft routing problem by using strings of flights. Desaulniers et al. (1997) present a model for the integration of FAM and time windows. Their model is a set partitioning model with side constraints. Rexing (1998) also presents a model that integrates FAM and time windows.

Cordeau et al. (2000) and Mercier et al. (2003) propose a model, which fully integrates the crew pairing and aircraft routing stages. They solve the model by a combination of branch-and-price and
Benders decomposition. Integration of aircraft routing and crew pairing is also discussed in Cohn and Barnhart (2003), where each feasible routing is modeled as a column. The approach by Klabjan et al. (2002) has already been described above.

Similar integration efforts have been undertaken in mass transit scheduling. Haase et al. (2001) present a model, which minimizes both the crew costs and the number of vehicles. They solve the underlying set partitioning model with side constraints using a branch-and-cut-and-price algorithm. Freling et al. (2003), Freling (1997), and Gatti and Nonato (1999) propose models that integrate vehicle and crew scheduling in a single depot environment. The multiple-depot setting is discussed in Huisman et al. (2003). Our first solution methodologies is similar to the one presented in this publication.

3 The Model

3.1 Approach

The primary costs in an airline’s operations are the operating costs from fleeting and the crew cost. The integration of FAM and crew pairing stages could thus potentially yield much lower crew costs and better profitability. But an integration of just the FAM and crew pairing stages implies that the crew pairing problem is solved prior to the aircraft routing problem. The solution obtained from such an integrated model could potentially cause the maintenance routing problem to be infeasible in terms of maintenance and the plane count. In our integrated model, we directly integrate the FAM and crew pairing problems. To prevent infeasible routings, we add additional constraints, similar to the plane count constraints introduced in Section 2.4, which provide the necessary conditions for the aircraft routing problem to be plane count feasible. These necessary conditions ensure that the pairings do not violate the plane count, i.e. number of aircrafts on the ground at any point of time.

Since we simultaneously assign fleets and pairings, it no longer suffices to merely know which pairings are selected. We have to capture to which fleet family a pairing is assigned to since all legs in a selected pairing must be assigned to the same fleet family.

3.2 Details

We first discuss a complete integration of the crew pairing and FAM problems. We integrate these models by enforcing that a pairing is assigned to a fleet family if and only if all the flights, which constitute the pairing, are assigned to the same fleet family. This implies assigning pairings to fleet families, which requires expanding the crew pairing variables. We then link the new crew pairing variables with the fleet assignment variables. For fleeting, we use the aforementioned variables and constraints from the traditional FAM model described by (1)-(3). In order to capture pairings within fleeting, we modify the pairing variables to \( y_{qp} \), where \( p \) is a pairing covering any subset of flights among all the flights in the schedule and \( q \) is a fleet family index. \( y_{qp} \) is 1 if pairing \( p \) is assigned to fleet family \( q \). The fleet-pairing linking constraints must model that a pairing \( p \) assigned to fleet family \( q \) covers flight \( l \) only if \( l \) is assigned to a fleet \( f \) (determined by the corresponding \( x \) variable) from fleet family \( q \). It is important here to note that pairing assignment is done at the fleet family level (a fleet family is an input to the crew pairing problem), while leg and aircraft assignments are at the fleet level (fleets are part of the fleet assignment problem and the aircraft routing problem). For a given fleet family \( q \) let \( S_q \) be the set of all fleets in fleet family \( q \). Thus we have \( F = \bigcup_{q \in Q} S_q \) and these sets are disjoint. Here \( Q \) is the set of all fleet families. For a fleet \( f \)
let \( h(f) \in Q \) be the corresponding fleet family. By definition we have \( f \in S_{h(f)} \). We now detail the incorporation of the aircraft routing constraints into the integrated model.

Recall the plane count constraints

\[
\sum_{p \in P_g} a_{pg} y_p \leq b_g \quad g \text{ ground arc.}
\]

In order to embed this into our integrated model, we have to observe that the ground arc value now corresponds to a decision variable and is not a fixed value. A technical difficulty is the fact that the RTN uses ready times whereas pairings are based on the actual arrival times. Another technical hurdle is the mapping between fleets and fleet families.

Let the actual-time-space network (ATN) be defined in the same way as RTN except that for each arrival of flight \( l \), \( t_a \) is the actual arrival time \((l_a)\) of leg \( l \). The set of all ground arcs in ATN is denoted by \( G' \) and the ground arcs are denoted by \( g' \). Essential ground arcs are those ground arcs, which are defined by a departure followed by an arrival. The plane count constraints associated with non-essential ground arcs are redundant (for proof see Klabjan et al. (2002)). Let \( E(G') \subset G' \) be the set of essential ground arcs. Figure 1 details the difference between the RTN and ATN networks. When a flight arrives at a station, the arrowhead shows the actual arrival or ready time. For an outbound flight, the tail of the arc shows the actual departure time.

![Figure 1: Comparison between RTN and ATN](image)

It remains to be seen how to convert these constraints into a fleet based setting, i.e. we have to link \( b_g \) with the ground arc variables \( z \). For each \( g' \in E(G') \), there exists a corresponding ground arc \( m_g \) in RTN. Note that \( g' \) is defined by the arrival of a leg \( l \). This ground arc \( m_g \) is defined by the ready time of leg \( l \) and an earlier activity. Let \( dep(g') \) be the set of all flights that depart in the time interval \([l_a, l_a + mt]\). We replace the right hand side of the plane count constraints by

\[
\sum_{f \in S_q} (z_{m_g, f} + \sum_{l \in dep(g')} x_{l}),
\]

where \( q \) is a fleet family. This expression states that the value of the ground arc corresponding to an arrival event of leg \( l \) in ATN is equal to the ground arc value of the same flight in RTN plus the number of departures in the set \( dep(g') \). This correspondence between ground arc values in ATN and RTN is shown in Figure 2. This statement can be easily checked.
Figure 2: Ground arc conversion

The integrated model reads

\[
\begin{align*}
\text{max} & \quad \sum_{f,l} r_{fl} x_{fl} - \sum_{f,p} c_{p} y_{gp} \\
\sum_{l \in L} x_{fl} & = 1 \quad l \in L \tag{5} \\
\sum_{l \in \mathcal{D}(v)} x_{fl} + & \sum_{l \in \mathcal{I}(v)} x_{fl} - \sum_{g \in \mathcal{W}} z_{gf} = 0 \quad v \in V, f \in F \tag{6} \\
\sum_{l \in \mathcal{M}} x_{fl} + \sum_{g \in \mathcal{W}} z_{gf} & \leq N_f \quad f \in F \tag{7} \\
\sum_{p \in \mathcal{L}(f)} y_{hk(f)p} & = x_{fl} \quad l \in L, f \in F \tag{8} \\
\sum_{p \in \mathcal{P}(g')} y_{qp} & \leq \sum_{f \in \mathcal{S}_q} (z_{mf} + \sum_{l \in \text{dep}(g')} x_{fl}) \quad g' \in E(G'), q \in Q \tag{9}
\end{align*}
\]

Constraints (5)-(7) are the standard FAM constraints, (8) ensures that a pairing is assigned to a fleet family if and only if all the legs in the pairing are assigned to the same fleet family, and (9) are the plane count constraints.

Several extensions and enhancements can be easily incorporated. We mention a few of them. If plane turn times depend on the fleet (as is usually the case in practice), then all we have to do is to add dependency on \( f \) to \( g' \) and \( \text{dep}(g') \) in (9). The corresponding RTN network needs to be adjusted accordingly. In practice, each fleet family uses only a subset of crew bases. This is very easy to accommodate by changing the summation range in (8). Suppose that in each fleet family \( q \) the number of crews must be in the range \([l_q, u_q]\), thus limiting the manpower. This is easily captured by adding

\[
l_q \leq \sum_{p} y_{qp} \leq u_q \quad q \in Q
\]

to the model. Other variations of the manpower constraints can be treated in a similar way.
4 Solution Methodologies

The integrated model is too large to be solved by standard optimization software packages even for small instances. We propose two different methodologies. Our approach is to either decompose the problem into smaller problems, which can be easily solved, or to consider only a subset of pairings at a time.

We first sketch the latter, which is in spirit very similar to the one used by Huisman et al. (2003). We described in the introduction the concept of delayed column generation for linear programs. This framework cannot be directly applied to our integrated model since we have integer variables and therefore the dual values are not available. The main idea is to obtain approximate dual values by means of Lagrangian relaxation, where the Lagrangian multipliers take the role of dual values, and as a consequence pairings are generated dynamically as needed.

In this approach we use Lagrangian relaxation over a small subset of pairings to obtain Lagrangian multipliers. These multipliers are then used to price out new favorable pairings. In the Lagrangian relaxation step, we relax constraints (8) and (9) and then solve the resulting problem over a subset of pairings by using Lagrangian relaxation.

In the second approach the key observation is that the reduced problem with only constraints (5)-(7) is the traditional FAM model, which is relatively easy to solve. We first relax the integrality requirements on the pairing variables. The resulting problem can now be solved by Benders decomposition, wherein the crew pairing problem is solved as an LP. The information from the dual of the LP is used to form Benders cuts, which are then added to the FAM model.

Another widely used solution methodology for solving large-scale integer programs is branch-and-price, see e.g. Barnhart et al. (1998b). We believe branch-and-price is not a suitable algorithm for solving our model since it requires too many pricing operations. As shown in Section 4.1.1, the pricing step of our model is computationally harder than usual pricing for crew pairing and therefore its invocations must be limited. In our computational experiments, pricing is called at most 10 times in the Lagrangian relaxation with column generation algorithm. We believe that a branch-and-price type algorithm would require many more pricing operations and therefore it would produce excessive computational times. The appeal of Benders decomposition is that the cut generation procedure is decomposed with respect to the current fleeting and therefore it results in solving several standard crew pairing LP relaxations. The drawback is that the integrality of the pairing assignment variables is relaxed.

In Section 4.1 we describe the solution methodology using Lagrangian relaxation with column generation. A detailed discussion about column generation is postponed to Section 4.1.1. Section 4.2 details the solution methodology using Benders decomposition.

4.1 Lagrangian relaxation with column generation

Traditionally, column generation is used along with the LP relaxations in the branch-and-price framework. We propose a column generation type approach combined with Lagrangian relaxation. First we approximate the model by changing the partitioning requirement (8) into covering constraints. We then relax constraints (8) and (9) and associate nonnegative Lagrangian multipliers λ with constraints (8) and μ with constraints (9). The Lagrangian relaxation of the restricted master problem reads
\[
\phi(\lambda, \mu) = \max \sum_f \sum_l (r_{fl} + \lambda_{fl} - \sum_g \sum_{l \in \text{dep}(g')} \mu_{h(f)g}) x_{fl} - \sum_q \sum_{p \in R} \sum_{l \in \text{dep}(g')} \lambda_{fl} + \sum_g \sum_{p \in \text{dep}(g')} a_{pg} \mu_{ag}) y_{ap} \\
+ \sum_f \sum_g \mu_{h(f)g} z_{fg'}, \\
\text{subject to FAM constraints (5), (6), and (7)},
\]
where \( R \) is a small subset of pairings.

The Lagrangian dual problem \( \min \phi(\lambda, \mu) \) is solved by the subgradient algorithm. The obtained Lagrangian multipliers \( \lambda \) and \( \mu \) are then used to find new pairings, which are added to the RMP, i.e. \( R \). The Lagrangian relaxation of the RMP is the traditional FAM model with a different objective function, which contains the Lagrangian multipliers. The solution methodology is next detailed step-wise.

1. **Initialization**: In this step we find initial pairings in \( R \) by applying the traditional sequential approach. We first solve FAM and next we solve the crew pairing problem with the plane count constraints by using the fleeting obtained from FAM. The pairings obtained from crew pairing are used as the initial set of columns in the RMP. The initial set of columns and this initial fleeting are feasible to integrated model since the pairings generated cover all the flights in the schedule and they do not violate the plane count constraints.

2. **Setting the lower bound**: We add the initial set of columns obtained in step 1 to the RMP. The objective value (profit in the integrated model) of the initial solution, denoted by \( LB \), is equal to the objective value of the Lagrangian dual over this initial set of columns. This follows directly from the fact that the LP relaxation of the RMP in this case yields an integer solution. \( LB \) is clearly a lower bound on the optimal value of the integrated model as well as of the initial Lagrangian dual problem.

3. **Computing Lagrangian multipliers** (major iteration): We use subgradient optimization to solve the Lagrangian dual problem with the current set of columns. We use the step updating heuristic proposed by Caprara et al. (1999). Next we briefly describe the subgradients and the step size formula.

After changing the set partitioning constraints to set covering constraints, constraints (8) read
\[
\sum_{l \in \text{dep}(g')} y_{h(f)g} \geq x_{fl}, \quad l \in L, f \in F.
\]

Let \( x, y \) be an optimal solution to the RMP. The subgradient vector \( s(\lambda) \) associated with a given \( \lambda \) is defined as
\[
s_{l,f}(\lambda) = x_{fl} - \sum_{l \in \text{dep}(g')} y_{h(f)g} \quad l \in L, f \in F.
\]

Similarly, the subgradient vector \( s(\mu) \) associated with a given \( \mu \) is
\[
s_{q,g}(\mu) = \sum_{p \in \text{dep}(g')} y_{ap} - \sum_{f \in \text{dep}(g')} z_{mgf} + \sum_{f \in \text{dep}(g')} \sum_{l \in \text{dep}(g')} x_{fl} \quad g' \in E(G'), \quad q \in Q.
\]

Let \( k \) be the iteration count within the subgradient algorithm. The step size along \( s(\lambda^k) \), denoted by \( \sigma^k \), is calculated as \( \sigma^k = \theta (z(\lambda^k, \mu^k) - LB) / \| s(\lambda^k) \|^2 \), where \( \theta > 0 \) is a parameter, which controls the step size along the subgradient direction. We change this parameter as required to increase or decrease the step size. For example, if the objective value does not change for \( k \) consecutive iterations, we reduce \( \theta \) by half. We replace \( s(\lambda^k) \) by \( s(\mu^k) \) in the above formula and

\[\text{...}\]
compute the step size $\tau^k$ along $s(\mu^k)$. Note that we deliberately do not use the same step size for $\lambda$ and $\mu$. The computational experiments have shown that such a choice yields better convergence.

Next we update the Lagrangian multipliers using the following formula

$$
\lambda^{k+1} = \max \{ \lambda^k + \sigma^k s(\lambda^k), 0 \}
$$

$$
\mu^{k+1} = \max \{ \mu^k + \tau^k s(\mu^k), 0 \},
$$

where the maximum is considered component-wise. We stop the subgradient algorithm either after a given number of iterations or if the norm of the subgradient becomes small.

4. **Pricing**: In this step we use the Lagrangian multipliers to price out favorable pairings using the constrained shortest path algorithm. Formally, we have to solve

$$
\min \min_{q, \phi} \left( c_p - \sum_{f \in S_q} \sum_{(p, q) \in P_f} \lambda_{f, \beta} + \sum_{d \in d(p), \phi \in P_p} a_{pg} \mu_{qg} \right).
$$

This is detailed in Section 4.1.1. Let $S$ be a small subset of pairings that either attain this minimum or are very close to it.

5. **Loop**: We first check the following termination criterion.

- The above minimum (10) is nonnegative, i.e. no columns with negative reduced cost are obtained.
- If $\phi(\lambda, \mu)$ does not change significantly for a given number of iterations.
- If a predefined maximum number of major iterations have been completed.

If any of these termination criteria is satisfied, we go to step 6; otherwise, we add the obtained pairings $S$ from step 4 to the RMP, i.e. we set $R \equiv R \cup S$, and we go to step 3.

6. **Obtaining the final solution**: The final solution is obtained by using the fleeting produced by solving the Lagrangian dual in the last iteration. We use this fleeting to obtain aircraft routes and crew pairings using the traditional approach.

As described earlier, after fleeting is solved, the problem decomposes by fleet family type. In the proposed integrated approach, the crew pairing problem for a given equipment type may be infeasible. This might happen either because we relax the partitioning constraints (8) to covering constraints or it might occur due to the nature of Lagrangian relaxation, i.e. a Lagrangian solution might not be feasible to the original problem. This was not observed in practice. Note that the traditional sequential approach suffers the same drawback. We believe that by using our integrated approach, which explicitly considers pairings and captures the dependency of pairings on fleet families, the likelihood of producing a crew infeasible fleet family is substantially lower.

4.1.1 **Pricing**

Pricing is the problem of generating pairings with the lowest reduced cost. There are two approaches to finding pairings with the least reduced cost. The first one is by enumerating all pairings and the second, typically more efficient one, is by using a variant of a shortest path algorithm. We use the latter approach to price out pairings based on (10). In this section, we first review the traditional constrained shortest path algorithm as it is applied to the crew pairing problem. Additional details were given at the end of the introduction. We then show how to tailor this algorithm to solve (10).

There are two types of networks that can be built to solve the pairing generation problem: flight network or the duty period network. In the flight network each departure and arrival has an associated node in the network. A flight arc connects the departure node of a flight with the
corresponding arrival node of the same flight. We add connection arcs between an arrival and a
departure node at the same station subject to the constraints on the minimum connection time. We
augment the network by adding a source $s$ and a sink $t$. We connect source $s$ to every departure node
originating at a crew base. Similarly, we connect all the arrival nodes from a crew base to sink $t$. The
duty period network is similar except that we replace the flight arcs by duty periods and connection
arcs correspond to legal rest connections. Although we can capture more pairing feasibility rules in
the duty period network (all duty rules are embedded by definition), the duty period network is much
larger than the flight network. Since we deal with the daily problem and to avoid cyclic networks,
we replicate each flight several times until the maximum elapsed time of pairings is reached. For
example, if pairings cannot exceed 5 days, then the network has 5 copies of each flight, each one
offset in time by a day. The resulting network captures all pairings and it is acyclic. The expanded
network also naturally embeds ground arcs. Each ground arc in the original network has now several
copies.

It is clear that each pairing corresponds to an $s$-$t$ path but an $s$-$t$ path might violate pairing
feasibility rules. In order to circumvent this, to find a favorable pairing, the constrained shortest path
algorithm must be employed (see the introduction for details on constrained shortest path). In such
an algorithm, a label is maintained for each feasibility rule and cost resource. The latter are required
if the cost of a pairing is non linear. Examples of labels are those corresponding to the maximum
number of duties, the maximum elapsed time, the sum of the duty costs, etc. In addition, to capture
the dual prices (in our case the Lagrangian multipliers), an additional label is required. Each $s$-$i$ path
is represented as a vector consisting of the values of all resources. Thus every node $i$ contains a set of
label vectors.

For the integrated approach pricing (10), we can reduce the problem to the constrained shortest
path problem. Assume that to solve the traditional crew pairing pricing problem we require $k$ labels
and let the corresponding label vector at node $i$ be denoted by $u$. Furthermore, assume that it encodes
a path $p$ (partial pairing) from $s$ to $i$. Since the Lagrangian multipliers contribute to the calculation of
the reduced cost (10) we need to augment the underlying network and incorporate these values. We
have $|Q|$ Lagrangian multipliers for each flight and each ground arc. Therefore we need to add $|Q|
new labels to the existing labels. Thus the number of required labels becomes $k + |Q|$. This clearly
results into higher computing times; however, since the pricing step is not performed often (at most
10 times) this is acceptable.

The new label vector becomes

$$v = (u, - \sum_{f \in S_1} \sum_{l \in p} \lambda_{fl} + \sum_{g' : p \in P_{g'}} \mu_{l_{g'}}, \cdots, \sum_{f \in S_1} \sum_{l \in p} \lambda_{fl} + \sum_{g' : p \in P_{g'}} \mu_{l_{g'}}, \cdots, \sum_{f \in S_1} \sum_{l \in p} \lambda_{fl} + \sum_{g' : p \in P_{g'}} \mu_{l_{g'}}).$$

It is easy to see that if $v \leq \bar{v}$, then the path corresponding to $\bar{v}$ can be discarded and therefore this
label vector can be removed.

If the duty period network is used, then for each duty $d$ we precompute

$$(- \sum_{f \in S_1} \sum_{l \in d} \lambda_{fl} + \sum_{g' : d \in P_{g'}} \mu_{l_{g'}}, \cdots, \sum_{f \in S_1} \sum_{l \in d} \lambda_{fl} + \sum_{g' : d \in P_{g'}} \mu_{l_{g'}}, \cdots, \sum_{f \in S_1} \sum_{l \in d} \lambda_{fl} + \sum_{g' : d \in P_{g'}} \mu_{l_{g'}}).$$

These are then used as node weights in the network and the labels are updated upon scanning of duty
d by adding this vector to the last $|Q|$ coordinates of the label vector.

If the flight network is used, then the treatment is slightly different. For each flight arc $l$ in the
network we first form the vector $\alpha(l) = (- \sum_{f \in S_1} \lambda_{fl}, \sum_{f \in S_2} \lambda_{fl}, \cdots, - \sum_{f \in S_{|Q|}} \lambda_{fl}).$ To handle ground arcs,
consider a connection arc \( ca \) corresponding to a sit connection between an arrival of a flight and a departure of a different flight. Let the notation \( g' \in ca \) mean that ground arc \( g' \) is included in connection \( ca \). Next we form the vector

\[
\beta(ca) = \left( \sum_{g': g' \in ca} \mu_{1g'}, \sum_{g': g' \in ca} \mu_{2g'}, \cdots, \sum_{g': g' \in ca} \mu_{|g'|}\right).
\]

An example is given in Figure 3, where the dotted lines show the actual flights, which constitute the essential ground arc. The solid lines are the flight arcs.

If we treat label vector \( v \) by scanning flight arc \( l \), then the new label vector \( \overline{v} \) is given by

\[
\overline{v} = \left( \overline{u}, v_{k+1} + \alpha(l)_1, v_{k+2} + \alpha(l)_2, \cdots, v_{k+|Q|} + \alpha(l)_{|Q|} \right),
\]

where \( \overline{u} \) is the treatment of the standard crew pairing resources. If we are scanning a connection arc \( ca \) that corresponds to a sit connection, then the updated label vector is given by

\[
\overline{v} = \left( \overline{u}, v_{k+1} + \beta(l)_1, v_{k+2} + \beta(l)_2, \cdots, v_{k+|Q|} + \beta(l)_{|Q|} \right).
\]

If the connection arc corresponds to an overnight connection, then we treat only \( u \) as in standard crew pairing pricing.

---

**Figure 3: Connection arc weights for \( |Q| = 2, |S_1| = 1, |S_2| = 1 \)**

We remark that ground arcs are captured correctly. The contribution of ground arcs is

\[
\sum_{g' \in P} a_{pg'} \mu_{qg'},
\]

where \( g' \) are ground arcs in the network without replications. In the network with replications this quantity becomes

\[
\sum_{g \in P} \mu_{qg},
\]

where \( g \)'s are ground arcs in the expanded network.

This shows that all coefficients in front of \( \mu \) are 1.

In our computational experiments we solve (10) based on the duty period network.
4.2 Benders Decomposition

In this approach we relax the integrality of the pairing variables while maintaining the integrality on the fleet assignment variables. Clearly this yields a relaxation of the integrated model. By preserving the integrality of the fleet assignment variables, we do not relax the important revenue based decisions. In essence, we comply with the hierarchy of the current sequential decision making process.

The resulting model is now suited for the Benders decomposition approach. The RMP consists of the FAM constraints (5)-(7) and the Benders cuts. Given a solution to the RMP, i.e. a feasible fleeting, the subproblem then decomposes into the LP relaxations of the crew pairing problems with plane count constraints. The subproblem clearly is decomposed by fleet family and this substantially reduces the computational burden. More precisely, in each iteration we solve the RMP and then use the underlying fleeting to solve the LP relaxations of the crew pairing problems with embedded plane count constraints. By using the dual values of these LP relaxations, we add Benders cuts. If an LP relaxation is infeasible, then a Benders feasibility cut is added.

Next we elaborate on the most significant steps. For a given fleet family \( q \), we define \( L_q \) as the subset of flights that have been assigned to fleet family \( q \) after solving the RMP, i.e. given fleeting decision variables \( x \). The LP subproblem for fleet family \( q \) then reads

\[
\begin{align*}
\min \, & \sum_p c_p y_p \\
\sum_{l \in P} & y_p = 1 \quad l \in L_q \quad \text{(11)} \\
\sum_{p \in P(g')} & a_{pg'} y_p \leq b_{g'} \quad g' \in E(G'_q) \quad \text{(12)} \\
y_p & \geq 0.
\end{align*}
\]

Here \( E(G'_q) \) denotes the set of the essential ground arcs with respect to the flights in \( L_q \) and \( b_{g'} = \sum_{f \in S_q} (z_{mf} + \sum_{l \in \text{dep}(g')} x_{l}) \) is the ground arc value.

Let the duals for (11) be represented by \( \theta \) and the duals for (12) by \( \Pi \), whenever the LP is feasible. If the LP is infeasible, let \((\alpha, \beta)\) be an extreme ray with

\[
\sum_{l \in L_f} \alpha_{gl} + \sum_{g' \in E(G'_f)} b_{g'} \beta_{g'} > 0,
\]

where \( \alpha \) corresponds to (11) and \( \beta \) to (12). Let \( \eta \) represent the upper bound on the crew cost in the relaxed integrated model. We index the Benders cut by \( k \), where \( k \) is the iteration count. The RMP at the beginning of a new iteration reads
\[
\max \sum_{f,i} r_{fi} x_{fi} - \eta
\]

subject to FAM constraints (5), (6) and (7),

\[
\sum_{f} \sum_{l} \left[ \theta_{h(f)l}^k + \sum_{g : l \in \text{dep}(g')} \prod_{g : l \in \text{dep}(g')}^{k} \right] x_{fi} + \sum_{f} \sum_{g} \prod_{g : l \in \text{dep}(g')}^{k} z_{m_{gf}^i} \leq \eta \quad k \in K
\] (14)

\[
\sum_{l} \left[ \alpha_{q_i}^k + \sum_{g : l \in \text{dep}(g')} \beta_{gq_i}^k \right] x_{h(f)l} + \sum_{g} \beta_{gq_i}^k z_{m_{gf}^i} \leq 0 \quad k \in J
\] (15)

\[
\eta \text{ unrestricted, } x \text{ binary, } z \geq 0.
\]

Constraints (14) are the Benders cuts and (15) are the Benders feasibility cuts. Here \( q_k \) denotes an infeasible subproblem LP at iteration \( k \).

The entire solution methodology is described stepwise as follows. Unlike the first solution methodology, we do not need to generate an initial feasible solution in this case.

1. **Solving the Restricted Master Problem:** Solve the restricted master problem and obtain a fleeting.

2. **Decompose based on fleeting:** Based on the solution of the RMP, for each fleet family we generate legs and plane count information.

3. **Solve the crew pairing LP relaxations:** For a given fleet family, we use the corresponding leg and plane count information obtained in step 2 to solve the LPs (11)-(12). Once the LPs have been solved, one of the following two cases may arise: the solution is optimal for each fleet family or there is a fleet family that yields an infeasible subproblem.

4. **Generate Benders cut:** If all the subproblems are feasible, we obtain the duals \( \theta^k \) and \( \Pi^k \) for constraints (11) and (12), respectively, for the current iteration \( k \). Use these duals to add a new Benders cut (14) and go to step 6.

5. **Generate a feasibility cut:** For each infeasible subproblem, we obtain an extreme ray satisfying (13). For each such infeasible fleet family, we add the corresponding feasibility cut (15).

6. **Iterate:** If we reach the maximum number of iterations, we go to step 7. Otherwise we go to step 1.

7. **Obtaining the final solution:** We use the fleeting from the last iteration to generate aircraft routes and crew pairings using the traditional approach.

## 5 Computational Experiments

We tested the integrated model on four data sets. Real world data from a major US carrier were used. The carrier has a heavy hub-and-spoke network structure with five crew bases and 8 hubs. Crew feasibility rules and cost function comply with the airline rules. For discretionary purposes, the real profit numbers are fudged but the presented numbers show correct proportions and magnitudes. Due to the lack of data, we do not use manpower constraints. However, the minimum turn time is fleet dependent and each fleet family uses only a specific subset of crew bases. The computing environment consists of a cluster of 27 dual 900 MHz Itanium 2 processors running Red Hat 7.3 operating system and the gcc 3.2 development environment. For solving small LPs and the fleeting integer programming models we used CPLEX from ILOG Inc., version 8.1.
In order to make a more fair comparison between the proposed solution methodologies and the traditional sequential approach, in our implementation we have also modeled constraints to prevent crew double overnights. Crew double overnights occur when only very late night flights of a fleet arrive at a station and only early morning flights of the same fleet departure with no other activity during the day. If this is the case, then the crew must have an extra day of rest due to the insufficient rest time between the arrival and the next departure. We add constraints to FAM using legal rest arcs and mid-day breakouts such that the obtained fleeting is more “crew friendly”, see Clarke et al. (1994) for details.

Each crew pairing problem is solved by branch-and-price. The pricing problem (10) and the pricing in solving the crew pairing LP relaxations in the second approach is carried out by using the parallel constrained shortest path algorithm, Klabjan (2003b).

We control tractability in the following way. First FAM is solved over all fleets and flights. Next we pick a subset of fleet families and the corresponding flights. The integrated model is then solved by considering only this subset of fleet families and flights. In other words, instead of considering all fleet families at once, we can consider subsets of fleet families. The number of flights in each instance and the number of considered fleet families are given in Table 1. Cases 3 and 4 are not full fleet families, i.e. after obtaining a fleeting, we choose a subset of fleet families and then from each set of flights for a given fleet family we pick a subset of flights. They were generated as proof of concept instances. Cases 1 and 2 on the other hand consist of the entire set of flights for a subset of fleet families. The first case consists of a majority of flights since the carrier uses 6 fleet families.

In this section we present the computational results for the four test cases. We first discuss all the benefits and then give a more detailed analysis of some of the test cases.

The increase in profits obtained by using the integrated approach as opposed to the sequential one is shown in Table 1. The monetary unit is US dollar. Both approaches were stopped after approximately the same elapsed time, which is discussed later. Profit here is defined as the combined profit resulting from FAM and the crew pairing costs, i.e. for the integrated approach it corresponds to the objective value. In the last two columns, we show the increase in profit obtained by solving the integrated model by Lagrangian and Benders respectively. Except for case 4, Lagrangian relaxation approach outperforms the Benders decomposition. We stress that these are daily profit increases that produce on an average 50 million dollars of additional profit per year for this particular airline. In addition, we selected a subset of fleet families just once. By repeating the process several times additional profit can be squeezed.

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Increase in Profit ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Fleet Families</td>
</tr>
<tr>
<td>Case 1</td>
<td>4</td>
</tr>
<tr>
<td>Case 2</td>
<td>2</td>
</tr>
<tr>
<td>Case 3</td>
<td>3</td>
</tr>
<tr>
<td>Case 4</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1: Profits
Table 2 details the breakup of the profit by revenue, operating cost for operating the flights, and the crew cost. The breakup has been presented for both solution methodologies. (The abbreviation for Lagrangian is Lgr and for Benders it is Bns.) All values are shown as percentage increase or decrease values with respect to the corresponding value obtained by using the traditional methodology. To ensure data confidentiality, we present only the range instead of the actual percentage increase or decrease. For example, for case 1 solved by the Lagrangian approach the revenue decrease is lower than 2% with respect to the traditional approach. Most of the increased profit does not come from diminishing revenue in FAM but it comes from the reduced crew cost. The change in the operating cost is negligible. This is a desired property since for historical and cultural reasons the carriers are not willing to sacrifice too much on the revenue side (even though it would increase the total profit). The last column shows the increased profit with respect to the traditional approach.

<table>
<thead>
<tr>
<th>Revenue %</th>
<th>Operating Cost %</th>
<th>Crew Costs %</th>
<th>Savings %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Lgr</td>
<td>Bns</td>
<td>Lgr</td>
</tr>
<tr>
<td>Case1</td>
<td>[-0.1,0]</td>
<td>[-0.5,0]</td>
<td>[0,0.5]</td>
</tr>
<tr>
<td>Case2</td>
<td>[-1,0]</td>
<td>[-1,0]</td>
<td>[0,1]</td>
</tr>
<tr>
<td>Case3</td>
<td>[-1,0]</td>
<td>[-1,0]</td>
<td>[0,1]</td>
</tr>
<tr>
<td>Case4</td>
<td>[-2,0]</td>
<td>[-1,0]</td>
<td>[0,0.5]</td>
</tr>
</tbody>
</table>

Table 2: Breakup of profit

We next study in more details cases 2 and 4 when solved using the Lagrangian approach. Figure 4 shows the improvement in the objective value after every major iteration (steps 3-5 in the algorithm) for the two test cases. Figure 4a shows the improvement for case 2 while Figure 4b shows the improvement for case 4. Obviously in initial iterations the improvements are minimal and then the objective value increases substantially. It is obvious that based on this trend it would be beneficial to perform additional iterations.

Figure 4: Objective value improvements
Within a given major iteration the objective values $\phi(\lambda, \mu)$ of the Lagrangian relaxation tend to decrease and eventually they converge towards the value of the Lagrangian dual. Figure 5 and Figure 6 show the decreasing trend of the Lagrangian relaxation objective values within a given major iteration for cases 4 and 2, respectively. Figure 5a and Figure 6a show the trend in a relatively early major iteration while Figure 5b and Figure 6b show the trend in a later major iteration. In few iterations this objective value significantly increases. We do not have an explanation for such a behavior. Fortunately it decreases back to the previous value relatively soon.

It also seems that the objective value is more stabilized in early major iterations than in the late iterations. Also with an increased data set size, the oscillation tends to increase. It is intuitive that convergence is slower when the data set size increases, which is apparent by comparing Figure 6 with Figure 5. Case 2 is larger than case 4.

![Figure 5: Trend of the Lagrangian relaxation objective values for case 4](image1)

![Figure 6: Trend of the Lagrangian relaxation objective values for case 2](image2)

Now we focus on the Benders decomposition. In Benders decomposition the objective value is non-increasing across iterations. The trends for cases 2 and case 4 are shown in Figure 7. As we can see, due to degeneracy in subproblems, the objective value can be constant for some iterations. We did not experiment with the core approach from Magnanti and Wong [22] since it is a non trivial task to find a core point.
In order to get a sense on the quality of the Lagrangian approach, we created a small instance where it is possible to obtain an optimal solution. This instance has 74 legs and 2 fleets, which results in 73,514 total pairings. The resulting full integrated model (5)-(9) has 402 rows and 147,286 variables. CPLEX is able to find an optimal solution in 35 seconds. We ran our Lagrangian approach and the traditional methodology. The results are summarized in Table 3. The top row shows the relative gap with respect to the optimal solution. It is encouraging to see that even for such a small instance with not many opportunities for improvement the Lagrangian approach is less than 4% from the optimal solution. We believe this number to be even smaller for real size problems. The remaining two rows of the table show the fleet decomposition in terms of the number of legs.

<table>
<thead>
<tr>
<th></th>
<th>Optimal</th>
<th>Lagrangian</th>
<th>Traditional</th>
</tr>
</thead>
<tbody>
<tr>
<td>gap</td>
<td>0%</td>
<td>3.8%</td>
<td>14.3%</td>
</tr>
<tr>
<td>No. of legs in fleet 1</td>
<td>35</td>
<td>28</td>
<td>40</td>
</tr>
<tr>
<td>No. of legs in fleet 2</td>
<td>39</td>
<td>46</td>
<td>34</td>
</tr>
</tbody>
</table>

Table 3: Small test instance with 74 legs

The breakup of computation times obtained by solving the integrated model using the Lagrangian approach is shown in Table 4. The units of measurement are listed in each column. The Lgr column shows the time taken to solve one subgradient iteration. The “average” column for each of the FAM, Lgr, and Pricing steps shows the average cumulative running time over all iterations. The percentile column for each of these steps shows the percentage of the total time that is spent in solving the step over all iterations. (We sum up the time taken for a given step in all minor and major iterations and calculate the % of the total time, which can be attributed to the given step.) The last column shows the total time taken to solve a given data set using the integrated model.
Table 4: Computation times for the Lagrangian approach

The running time for each major iteration increases significantly with an increasing number of legs. There is not a considerable increase in computational time with an increase in the number of fleet families as shown by cases 3 and 4. The subgradient optimization accounts for more than three-fourths of the time. The most intensive step is solving the RMP in each iteration and all other operations per iteration are negligible. The reason is in the slow convergence of the subgradient algorithm. It is surprising that the pricing step is not more computationally intensive. This is primarily due to the employment of the parallel algorithm.

The computation times for Benders are shown in Table 5. The “average” column for the RMP and LP steps shows the absolute value of the average over all iterations. The “%” column shows the percentile of the total time that is spent in solving a step over all iterations. The last column gives the overall running time. The increase in the running time to solve the RMP is striking. The traditional FAM used in the Lagrangian approach takes less than 1 minute. On the other hand, after adding Benders cuts (we never add more than 30 of them), this running times increases to almost an hour.

Table 5: Computation times of Benders decomposition

With an increase in the number of legs and fleet families, the computational time using the Benders approach increases. The distribution of time across RMP and LP shows a significant trend. For smaller test cases, the RMP takes about three-fourth of the total time to solve the problem. But with an increase in the size of the data set, the time taken to solve the LP subproblem increases significantly. For larger data sets, half of the total time is spent in solving the crew pairing LP relaxation.
6 Conclusions and Future Directions

The Lagrangian relaxation methodology on average outperforms the Benders decomposition algorithm. Only in test case 4 the latter algorithm produces a slightly better result. Their behaviors in terms of the objective value improvements are complementary. The Benders algorithm makes substantial improvements in initial iterations and then it stalls due to the known phenomena of degeneracy. On the other hand, the Lagrangian methodology makes only minor improvements in early iterations and substantial improvements in later iterations. In early iterations Lagrangian multipliers are unstable. Once they stabilize, they reveal more accurate information, which results into significant gains. As a conclusion, if an improvement is desirable in a short amount of time, Benders decomposition should be used. If several hours of computing time are available, then Lagrangian relaxation should be used. Since this is a tactical planning problem, usually the latter condition holds. We find the Lagrangian relaxation approach more robust and practical.

Next we discuss some future directions. The convergence rate of the Lagrangian algorithm using the subgradient optimization is very slow. The bundle method is known to have a better convergence rate as compared to subgradient variants to solve Lagrangian relaxation. We see a potential improvement in the running time by using the bundle algorithm instead of the vanilla subgradient optimization.

On the algorithmic front, we plan to test an alternative algorithm, which is aligned with the existing one. We propose to directly solve the Lagrangian dual. In the current solution methodology, we consider only a subset of pairings in each iteration. We then generate columns and add them to the current subset. As opposed to this, in every subgradient or bundle iteration, we can price out all pairings to adjust the current iterate.

The revenue obtained from the current integrated model takes only the average fare per leg into account, i.e. fleeting is leg based. The average fare for each leg is not representative of the actual revenue due to the multi leg passenger itineraries. Leg-based FAM solutions tend to be biased toward larger aircraft. This leads to the requirement of better revenue modeling practices. The origin-destination fleet assignment model (O-D FAM) models passenger revenues for each origin-destination itinerary rather than for each leg. The O-D FAM model is a combination of the leg-based FAM model and the passenger mix model. Given a fixed sit capacity, the passenger mix model decides the number of passengers that should be booked on a given O-D itinerary.

To overcome this problem and to be consistent with yield management practices, we can integrate O-D FAM, crew pairing, and aircraft routing. O-D FAM can be solved using Benders decomposition. The newly formed integrated model can be solved along the same lines as the current integrated model. Unfortunately we believe this still might be out of reach since the solution times for O-D fleeting are too high. An alternative might be to use clever high performance algorithms.

In our integrated model, although we present a complete integration of FAM and crew pairing problems, the integration with the aircraft routing model is only partial. Recall that out solution yields plane count feasible aircraft routes but not necessarily maintenance feasible routes. In addition, we do not capture the potential aircraft routing objective function. An integrated model, which fully integrates the aircraft routing model with the FAM and crew pairing models instead of just ensuring feasibility, is an interesting topic for future research.

7 Acknowledgments

We would like to thank Michael Clarke from Sabre Inc. for providing the data and valuable discussions. We are indebted to Professor Karsten Shwan from Georgia Institute of Technology for
allowing us to use their computing resources. Special thanks to Professor Eva Lee from the Georgia Institute of Technology, Atlanta, GA for assistance with CPLEX licenses. In addition, we are thankful to ILOG Inc. for providing the licenses.

8 References


