

# Robust Airline Crew Pairing: Move-up Crews

Sergey Shebalov  
Department of Mathematics  
University of Illinois at Urbana-Champaign  
Urbana, IL  
shebalov@uiuc.edu

Diego Klabjan  
Department of Mechanical and Industrial Engineering  
University of Illinois at Urbana-Champaign  
Urbana, IL  
klabjan@uiuc.edu

February 20, 2004

## Abstract

Due to irregular operations, the crew cost at the end of a month is typically substantially higher than the projected crew cost in planning. We present a model and a solution methodology that produces robust crew schedules in planning. Besides the objective of minimizing the crew cost, we introduce the objective of maximizing the number of move-up crews, i.e. the crews that can potentially be swapped in operations. We use delayed column generation, and Lagrangian decomposition for solving the restricted master problem. We report computational experiments, which demonstrate benefits of using the robust crew schedule instead of the traditional one.

## 1 Introduction

In the last decade of the 20th century the international and the US domestic air traffic was rapidly growing and was expected to double in the next 10 to 15 years. As a result of this trend, in the US the airports and the national air space became extremely congested. In addition, the airlines, to be competitive, were notorious to keep the market presence by scheduling many flights. The result was a large number of flight delays that kept growing every year. In the US, according to the Department of Transportation, from 1995 to 1999 the total number of delayed minutes, not counting delays that are shorter than 15 minutes, had increased by 11%, [Bond \(2000\)](#). The piling delays brought a nation-wide attention in Summer of 2000, when the airline with the best on-time performance had only 75% on-time flights. The events of September 11th 2001 led to a crisis the consequences of which still have effects. However, the airline industry is strongly recovering and in the next two years and a half revenue passenger mile (RPM) in the US is expected to reach the level of 2000, [Donoghue \(2004\)](#), [FAA \(2003\)](#). Therefore, the problems caused by over extensive air traffic demand and other uncontrollable events will soon raise again.

To fight congestion, the infrastructure needs to be expanded, e.g. by building new runways. On the other hand, the airlines can improve their own operations. Inclement weather, congestions, employees' sickness and other uncontrollable factors lead to delayed arrivals, which in turn snowball to delaying other flights due to missing or late resources such as aircraft and crews. In particular, a disrupted crew schedule can result into delayed or cancelled flights. Since the crew cost is second only to the fuel cost, the airlines incur substantial additional crew cost as a result of disruptions or irregular operations.

The crew pairing problem is to find crew itineraries or *pairings*, that cover all of the flights and minimize the crew cost. The crew pairing problem is solved several months before the actual flight schedule goes in operation. It is assumed that all of the flights will be flown as planned, i.e. on-time. Due to irregular operations, the crew cost at the end of a month, called the *operational crew cost*, can be up to 5% larger than the planned crew cost obtained by the optimal crew schedule in planning. To decrease the operational crew cost the airlines can use better recovery procedures in disruption management, see e.g. [Stojković et al. \(1998\)](#), or an alternative is to develop robust crew pairing solutions in planning. Robust crew schedules take into account possible disruptions in operations and therefore they should perform better in operations. This work focuses on the robust airline crew pairing problem.

When a disruption occurs, a low cost solution to an airline is to swap two crews. In a crew swapping scenario, a crew whose arrival is delayed, next flies a flight with a later departure time than its originally assigned flight. A different crew, called the *move-up crew*, then covers the flight of the disrupted crew, [Figure 1](#). In addition to low cost of crew swapping, they are appealing to crew controllers since typically it is easy to find a crew swap. If the crew schedule does not allow many such opportunities, then often a controller has to opt for alternative solutions like a more complicated crew recovery, delaying flights or even cancelling flights. We present a tactical planning crew pairing model that considers move-up crews. In addition to capturing the crew cost, our model obtains crew schedules that have many opportunities for crew swapping, thus the model is a bicriteria optimization model. The resulting large-scale model is solved by a combination of Lagrangian relaxation and delayed column generation. We evaluate solutions of our model by means of simulation. The experiments show that our robust solutions yield lower operational crew cost and fewer flight cancellations. One important conclusion is that the trade-off between the planned crew cost and robustness has to be considered judiciously since sacrificing too much crew cost is often not beneficial.

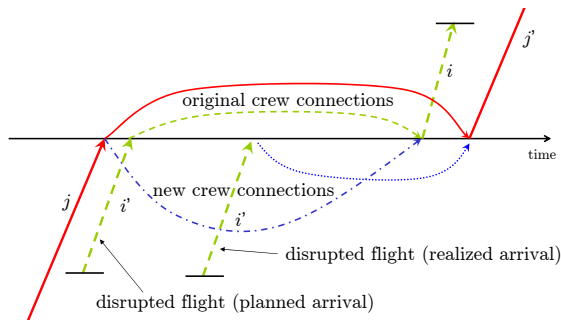


Figure 1: Crew swapping

In [Section 2](#) we give a brief description of the airline crew pairing problem and we review previous work on robust airline crew pairing. We present in [Section 3](#) the model. [Section 4](#) presents the solution methodology, and the computational experiments are given in [Section 5](#).

## 2 Crew Pairing and Robustness

In this section we overview the crew pairing problem and we discuss previous literature on robustness in the airline industry.

### 2.1 The Crew Pairing Problem

In tactical planning, first an equipment type is assigned to each flight. Since cockpit crews are not cross qualified, the crew scheduling problem that follows decomposes based on different equipment types. We assume that a flight schedule of a given equipment type is selected.

A *flight leg* or *segment* is a nonstop flight. We denote by  $L$  the set of all flight legs and let  $dt_i/at_i$  be the departure/arrival time of leg  $i$ . A *hub* is a station with high activity and every other station is called a *spoke*.

A *duty* is a working day of a crew. It consists of a sequence of flights, where the arrival station of a flight equals to the departure station of the next flight. A duty is subject to regulatory (such as FAA) and company rules. Among other rules, there is a minimum and maximum connection time between two consecutive flights in a duty, denoted by  $minSit$  and  $maxSit$ , respectively. A connection within a duty is called a *sit connection*. The minimum sit connection time requirement can be violated only if the crew follows the plane turn, i.e. it does not change planes. The cost of a duty is usually the maximum of three quantities: the flying time, a fraction of the elapsed time, and the duty minimum guaranteed pay. All three quantities are measured in minutes.

*Crew bases* are designated stations where crews are stationed. A crew base is often a hub. A *pairing* is a sequence of duties, starting and ending at the same crew base. Notation  $p = (d_1, \dots, d_k)$  encodes pairing  $p$  in terms of its duties, i.e.  $d_1$  is the first duty in  $p$ ,  $d_2$  is the second duty, and so forth until the last duty  $d_k$ . A connection between two duties is called an *overnight connection* or a *layover*. We refer to the time of a layover as the *rest*. Like sit connection times, there is a lower and an upper bound on the rest, denoted by  $minRest$  and  $maxRest$ , respectively. The cost of a pairing is also the maximum of three quantities: the sum of the duty costs in the pairing, a fraction of the time away from base and a minimum guaranteed pay times the number of duties. The *excess cost* or *pay-and-credit* of a pairing is defined as the cost minus the flying time of the pairing. Note that the excess cost is always nonnegative. The *flight time credit* (FTC) of a pairing is the excess cost times 100 divided by the flying time. Occasionally a crew needs to be repositioned and in this case the crew members fly as passengers, i.e. the crew is *deadheaded*.

The crew pairing problem has drawn a lot of attention in the past. In recent years, due to novel algorithmic methodologies and advances in computer hardware and software, the excess cost for large fleets has been pushed close to zero. A survey on crew scheduling is given by Barnhart *et al.* (2003). This survey also gives a detailed discussion of legality rules and the cost structure.

The *airline crew pairing problem* is to find the minimum cost pairings that partition all the legs. The daily airline crew pairing problem is the crew pairing problem with the assumption that each leg is flown every day of the week. In practice, some legs are not operated during weekends. Since the number of such irregular legs is small, the daily problem forms a good approximation (see Barnhart *et al.* (2003) how to obtain a workable crew schedule from a daily one). This work deals exclusively with the daily problem.

In a typical problem, the number of pairings varies from 200,000 for small fleets, to about a billion for medium size fleets and to billions for large fleets. Furthermore, since the cost function of a pairing is nonlinear and the legality rules are complex, it is hard to perform delayed column generation, i.e. generating columns only as needed. Traditionally the crew pairing problem is modeled as the set partitioning problem

$$\min\{cx : Ax = \mathbf{1}, x \text{ binary}\}, \quad (1)$$

where each variable corresponds to a pairing,  $a_{ij} = 1$  if leg  $i$  is in pairing  $j$  and 0 otherwise, and  $c_j$  is the cost of pairing  $j$ .

At the day of the operations, if an irregular event occurs, most airlines first recover the flight schedule, then the new crew assignments are created, and at the end passengers are reaccomodated. In crew recovery, crew itineraries can be changed and if need be, additional crew members are required. If new flying duties are created, first regular crew members are asked for voluntary overtime flying. Since frequently their response is not sufficient, the next step is to consider standby reserve crews. Standby reserve crew members have predefined duty times and they are located at the corresponding crew base. Since these crew members are located at the airport, their response time is quick. If there is still shortage of crew members, then regular reserve crews are contacted. These are typically not on duty and their response time can be as high as 20 hours. An extremely high cost solution for the airline is to ask for involuntary flying of regular crews at a premium pay. As the last resort, the airline might cancel a flight due to crew unavailability. Models for solving the crew recovery problem are surveyed in Barnhart *et al.* (2003).

Although heavily studied and used in practice, the set partitioning model (1) has a serious drawback. Namely, a solution to this model is an optimal or near optimal crew schedule under the assumption that all

the flights will be operated on-time. The airline then incurs a much larger crew cost in operations. Typically for large fleets the FTC in planning is below 1% and it increases to 3% to 4% in operations whereas for smaller fleets the planning FTC is approximately 3% or higher and it jumps to around 8% in operations. These numbers translate into millions of dollars of the increased crew cost. Solutions of large fleets are much more sensitive to disruptions since they have many tight connections. A disrupted short connection can have significant impact on the entire flight schedule due to snowball effects. This fact shows severe limitations in existing approaches.

## 2.2 Robust Optimization in the Airline Industry

Only recently practical approaches to robust crew pairing were introduced. Schaefer *et al.* (2000) investigate two approaches. In one approach they consider the effect of less restrictive feasibility parameters, e.g. what is the impact on robustness and the crew cost if the minimum sit connection time is increased. In the second approach they solve the crew pairing model (1) with a modified crew cost. Instead of considering the pairing cost, they estimate the expected pairing cost. The estimation is done by means of simulation, where they assume a specific crew recovery procedure called push-back, which, however, is not often used in practice. In the push-back recovery procedure a flight is delayed until all the resources are available. Yen and Birge (2000) and Ehrgott and Ryan (2001) consider the effect of maximizing the sit connection time whenever the crew does not change the plane, but they do not take into account the recovery procedure at all. The former work presents a stochastic model and the latter a deterministic variant. The stochastic programming approach is computationally difficult and Yen and Birge (2000) report computational results only for small fleets. In addition, both of these two approaches use weights to encourage long sit connections and it is not clear how to set these weights. Our model is completely different from all these approaches. We assume an underlying recovery procedure that considers crew swapping, which is widely used in practice.

Robustness paved its way also in other areas of airline tactical planing. Work in robust fleetling, Rosenberger *et al.* (2003), Kang and Clarke (2003), Listes and Dekker (2003), robust aircraft routing, Ageeva (2000), Lan *et al.* (2003) and the robust approach to passengers rerouting in disruption management by Karow (2003) show this emerging trend. Another growing area is the development of simulation systems of airline operations, e.g. SimAir by Rosenberger *et al.* (2002) and MEANS by Bly *et al.* (2003). These systems play a crucial role in evaluating and comparing the performance of different schedules.

## 3 The Crew Pairing Model with Move-Up Crew Count

In this section we present a model, called the crew pairing model with move-up crew count, whose solution yields robust crew schedules. We also give extensions to the model.

### 3.1 Move-up Crews

We first introduce the notion of a move-up crew. Let crew  $crew_1$  cover flight  $j$  that is followed by flight  $j'$  in a pairing. Let a different crew  $crew_2$  fly flights  $i'$  and  $i$  based on the crew schedule produced in the planning stage, see Figure 2. We assume that the two crews are based in the same crew base. Suppose that flight  $i'$  is delayed in operations and as a consequence crew  $crew_2$  cannot cover flight  $i$  since it would violate the minimum sit or rest connection time (whichever occurs in the planed crew schedule). If the departure time of flight  $j'$  is after the departure time of flight  $i$  and crew  $crew_1$  is ready to fly before the departure time of  $i$ , i.e. it is on the ground longer than either  $minSit$  or  $minRest$ , than crews  $crew_1$  and  $crew_2$  can potentially be swapped. Precisely, crew  $crew_1$  can fly flight  $i$  and crew  $crew_2$  can potentially cover flight  $j'$ . Such a swap is completely feasible if the two swapped pairings are feasible. In addition to pairing feasibility rules, we would like that the two involved crews finish their respective pairings on the same day. If, for example, crew  $crew_1$  is scheduled to finish the pairing on Wednesday and crew  $crew_2$  on Thursday, then if we swap the pairings on Monday, we severely disrupt their monthly schedules since it is likely that they will not be able to fly the next pairing in the schedule. In addition, the union rules may prohibit extending the length of a pairing. Therefore in addition to pairing feasibility, we impose that the two pairings must have

an equal number of days till the end. If these requirements are met, then we say that crew  $crew_1$  has a *move-up* opportunity since it moves up in the crew schedule.

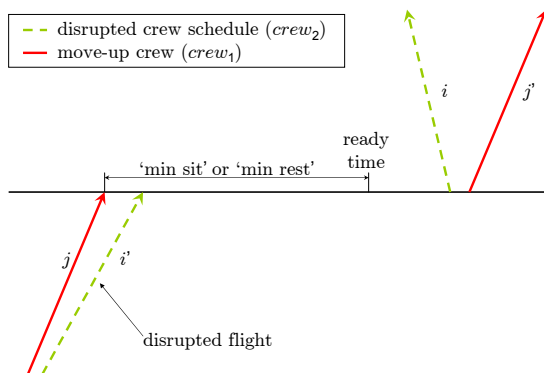


Figure 2: A move-up crew

Move-up crews have an additional benefit besides the crew swapping opportunities. In a presence of a move-up crew, the move-up crew can cover either flight  $i$  or  $j'$ . If the airline has to cancel a flight, it gives the opportunity to cancel either of the two flights and therefore the airline may consider cancelling a flight that minimizes passenger disruptions. If the move-up crew is not present, then the airline does not have a crew for leg  $i$  and therefore it has to cancel this flight.

We introduce a new objective function that captures the number of move-up crews. For each flight  $i$  we count the number of move-up crews for this flight, i.e. the number of those crews that are ready to fly before the departure time of this flight and have the same number of days till the end of the pairing as the pairing covering flight  $i$ . In addition we assume that each pairing of a move-up crew has to start at the same crew base as the pairing covering flight  $i$ . We do not take into account other pairing feasibility rules. If we denote by  $z_i$  the number of move-up crews for flight  $i$ , then the new objective function is  $\sum_{i \in L} z_i$ . A crew schedule that has a large number of move-up crews is likely to be more robust since it has many potential swappable crews. Note that besides making the approximation of neglecting some pairing feasibility rules of the swapped pairings, in operations it can happen that the crew flying the disrupted flight  $i'$  cannot connect to flight  $j'$ , but, if there are many move-up crews for this flight, then it is likely that the crew covering  $i'$  will be able to cover at least one flight  $j'$  from a move-up crew.

Clearly there is a trade-off between minimizing the crew cost and maximizing the number of move-up crews in planning. A schedule that maximizes the number of move-up crews may as well have high crew cost. To circumvent this problem, we assume that first the traditional crew pairing problem (1) is solved, thus the planned crew cost is given. Our model maximizes the number of move-up crews and we add a constraint controlling that the crew cost does not increase too much above the planned one.

### 3.2 Model

In order to count the number of move-up crews we need to know, given a flight  $i$ , the crew base of the pairing covering  $i$  and the number of days until the end of the pairing. Therefore the already introduced  $z$  variables need to be augmented.

Let  $HL$  be the set of all legs originating at a hub, and let  $CB$  be the set of all crew bases. We assume that each crew base is a hub. Since at spokes there are not many opportunities to swap crews, we count move-up crews only at hubs. Let  $P_{i,cb,d}$  be the set of all pairings covering leg  $i$  in  $L$ , starting at crew base  $cb$  in  $CB$  and having  $d$  days from flight  $i$  to the end of the pairing. Let  $\bar{P}_{i,cb,d}$  be the set of all pairings that yield a move-up crew for the pairing covering leg  $i$ . Formally, if  $p = (d_1, \dots, d_k) \in \bar{P}_{i,cb,d}$ , then there must exist a leg  $j \in d_s, 1 \leq s \leq k$  with the following properties:

1. the flight following leg  $j$  in  $p$  is not leg  $i$  (a pairing itself does not yield a move-up crew),
2. the arrival station of leg  $j$  is the same as the departure station of leg  $i$ ,
3.  $p$  starts at crew base  $cb$ ,
4.  $d = k - s + 1$  (there must be the same number of days till the end of the pairing),
5.  $at_j + t \leq dt_i$ , where  $t$  is either  $minSit$  or  $minRest$  depending if  $p$  has a sit or an overnight connection at the arrival station of leg  $j$  (the crew must be ready at the departure time of leg  $i$ ),
6.  $dt_{j'} > dt_i$ , where leg  $j'$  is the flight following leg  $j$  in  $p$  (the crew covering leg  $i$  can potentially fly leg  $j'$ ).

In addition, let  $P_i$  be the set of pairings covering leg  $i$  and let  $D = \{1, 2, \dots, K\}$ , where  $K$  is the maximum number of days in a pairing. Since most of the times (double overnight rests are exceptions) the number of flying days of a crew equals to the number of duties in a pairing, we have that if  $p = (d_1, \dots, d_k) \in P_{i,cb,d}$  and  $i \in d_j$ , then  $d = k - j + 1$ . Finally, let  $P$  be the set of all the pairings.

The model has the following variables:

- a binary variable  $y_p$  for each pairing  $p$ , which is 1 if and only if pairing  $p$  is selected, and
- let  $z_{i,cb,d}$  count the number of move-up crews to flight  $i$  if  $i \in p$  for a  $p \in P_{i,cb,d}$ .

The crew pairing model with move-up crew count (CPMC) reads

$$\max \sum_{\substack{i \in HL \\ cb \in CB \\ d \in D}} z_{i,cb,d} \tag{2}$$

$$\sum_{p \in P_i} y_p = 1 \quad i \in L \tag{3}$$

$$z_{i,cb,d} \leq \sum_{p \in \bar{P}_{i,cb,d}} y_p \quad i \in HL, cb \in CB, d \in D \tag{4}$$

$$z_{i,cb,d} \leq M \sum_{p \in P_{i,cb,d}} y_p \quad i \in HL, cb \in CB, d \in D \tag{5}$$

$$\sum_{p \in P} c_p y_p \leq (1 + r) \cdot C_{OPT} \tag{6}$$

$y$  binary .

Constraints (3) are the standard set partitioning constraints. (4) and (5) together count the number of move-up crews. If pairing  $p$  from  $P_{i,cb,d}$  covers leg  $i$ , then (4) by definition of  $\bar{P}_{i,cb,d}$  counts the number of move-up crews for  $i$ . (5) imply that if  $i$  is not covered by a pairing from  $P_{i,cb,d}$ , then  $z_{i,cb,d}$  is 0. Constant  $M$  regulates the maximum number of move-up crews we want to capture per leg. If we want to exclusively maximize the number of move-up crews, we select a large enough  $M$ . However, it makes more sense to require, for example, that we take into account only at most two move-up crews for each flight ( $M = 2$ ). A solution with  $k$  move-up crews, where each move-up crew covers a different leg, is preferable over a solution with  $k$  move-up crews for a single leg. To control the crew cost, we include (6). Here  $C_{OPT}$  is the optimal value of (1), and  $r, r \geq 0$  is the robustness factor, which measures how much extra crew cost we are willing to absorb.

### 3.3 Extensions

The CPMC model allows that a crew is a move-up crew to several other crews, but in reality it can be swapped only with a single crew. This can lead to cases with large objective value, but the solution does

not provide good flexibility during irregular operations. We call this property *double counting*. An example is given in [Figure 3](#). In case *I* crew flying leg 1 serves as a move-up crew for leg 3, and crew flying leg 2 for leg 4. In case *II* crew flying leg 2 serves as a move-up crew for both legs 3 and 4. Although both *I* and *II* have an objective value 2, crew schedule *I* is clearly preferable, since if both legs 3 and 4 are disrupted, only one of them can use the move-up crew covering leg 2.

To account for double counting we augment CPMC as follows. Let  $\overline{HL}$  be the set of all flights arriving at hubs. For every  $i \in \overline{HL}, j \in HL$  such that the arrival station of leg  $i$  is the same as the departure station of leg  $j$  and the connection time is within  $[minSit, maxSit] \cup [minRest, maxRest]$  we define a new binary variable  $v_{ij}$ , which is equal to 1 if leg  $i$  yields a move-up crew to leg  $j$  and 0 otherwise. To prevent double counting the following constraints are added to the model

$$\sum_{j \in HL} v_{ij} \leq 1 \quad i \in \overline{HL} \quad (7)$$

$$\sum_{i \in \overline{HL}} v_{ij} = \sum_{cb \in CB, d \in D} z_{j,cb,d} \quad j \in HL. \quad (8)$$

Constraints (7) guarantee that each leg arriving at a hub can be a move-up crew for at most one leg. Constraints (8) make sure that the number of assigned move-up crews to a leg is equal to the number of available move-up crews. A drawback of this extension is that, for example, a solution with a single flight having one move-up crew has the same objective value as a solution with a single flight having two move-up crews. The latter is preferable since we are neglecting some pairing feasibility rules in crew swaps.

To take this into account we can proceed as follows. We introduce for every  $i \in \overline{HL}, cb \in CB, d \in D$  two binary variables  $v_{i,cb,d}/w_{i,cb,d}$ , which are equal to 1 if leg  $i$  is covered by pairing  $p \in P_{i,cb,d}$ , it can serve as a move-up crew, and the next connection is a sit/rest. In this case we add

$$v_{i,cb,d} \leq \sum_{p \in (\cup_{j \in S(i)} P_{j,cb,d}) \setminus P_i} y_p \quad i \in \overline{HL}, cb \in CB, d \in D \quad (9)$$

$$v_{i,cb,d} \leq \sum_{p \in P_{i,cb,d}^S} y_p \quad i \in \overline{HL}, cb \in CB, d \in D, \quad (10)$$

$$w_{i,cb,d} \leq \sum_{p \in (\cup_{j \in S(i)} P_{j,cb,d}) \setminus P_i} y_p + \sum_{p \in (\cup_{j \in R(i)} P_{j,cb,d-1}) \setminus P_i} y_p \quad i \in \overline{HL}, cb \in CB, d \in D \quad (11)$$

$$w_{i,cb,d} \leq \sum_{p \in P_{i,cb,d}^R} y_p \quad i \in \overline{HL}, cb \in CB, d \in D. \quad (12)$$

Here  $S(i), R(i)$  is the set of all legs departing from the same station as the arrival station of leg  $i$  and with the connection time within  $[minSit, maxSit], [minRest, maxRest]$ , respectively.  $P_{i,cb,d}^S/P_{i,cb,d}^R$  is the set of pairings  $p \in P_{i,cb,d}$  such that the next connection is a sit/rest. Constraints (10) and (12) make sure that the leg is actually covered by a pairing  $p$  from  $P_{i,cb,d}$  and it has the desired connection length. Constraints (9) and (11) count the number of legs which can use the crew flying leg  $i$  as a move-up crew. Note that we have to exclude pairings that cover leg  $i$ . We also add the term  $\alpha \sum_{\substack{i \in \overline{HL} \\ cb \in CB \\ d \in D}} (v_{i,cb,d} + w_{i,cb,d})$  ( $\alpha > 0$  is a parameter)

to the objective function to encourage a larger number of potential move-up crews. In [Figure 3](#) this term is  $2\alpha$  for case *I*, and  $\alpha$  for case *II*. Thus the first case is preferable.

Another property, which the CPMC model does not captures, is the dispersion of move-up crews in the crew schedule. In [Figure 4](#) both *I* and *II* have 4 move-up crews, but *II* is preferable, since it is more likely that both legs 1 and 2 will be able to use a move-up crew in case of a disruption. To capture this feature we introduce integer variables  $u_{i,j,cb,d}$  for each  $i \in HL, j \in HL$ , which depart from the same hub, and  $cb \in CB, d \in D$ . We add constraints

$$\begin{aligned} u_{i,j,cb,d} &\geq z_{i,cb,d} & i \in HL, j \in HL, cb \in CB, d \in D \\ u_{i,j,cb,d} &\geq z_{j,cb,d} & i \in HL, j \in HL, cb \in CB, d \in D. \end{aligned}$$

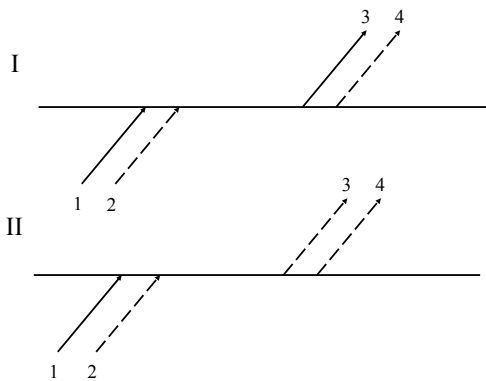


Figure 3: Double counting

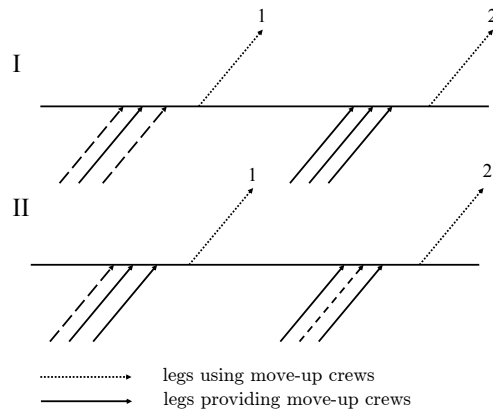


Figure 4: The dispersion effect

Thus  $u_{i,j,cb,d} = \max\{z_{i,cb,d}, z_{j,cb,d}\}$ . To encourage even distribution of move-up crews we add the term  $\beta \sum_{\substack{i \in HL, j \in HL \\ cb \in CB, d \in D}} u_{i,j,cb,d}$  ( $\beta < 0$  is a parameter) to the objective function. In our example, in case *I* this term equals to  $3\beta$ , and in the second case to  $2\beta$ .

We can also enhance the objective function (2). The objective function can be generalized by assigning weights  $q_{i,cb,d}$  to each  $z_{i,cb,d}$ , thus maximizing  $\sum_{\substack{i \in HL \\ cb \in CB \\ d \in D}} q_{i,cb,d} z_{i,cb,d}$ . In this way we can control the number of move-up crews corresponding to different legs. Airlines have strategic flights, which yield substantial revenue. Higher weight could be assign to such flights. An airline might also give weight based on the on-time performance of a leg. We can also assign higher weights for lower values of  $d$ . Due to snowball effects, legs which are towards the end of a pairing are more likely to be disrupted due to crew problems and therefore it is desirable that they have move-up crews.

## 4 Solution methodology

In this section we describe the underlying methodology for solving CPMC. Because of the enormous number of pairings, columns need to be generated dynamically. Since there is also a substantial number of constraints, we chose not to apply the standard branch-and-price algorithm. Instead we employ a combination of Lagrangian relaxation, see e.g. Fisher (1981), and column generation. The restricted master problem consists of only a subset of pairings and it is solved by Lagrangian relaxation. Based on Lagrangian multipliers, we generate new pairings, which are added to the restricted master problem. Our model is suited for Lagrangian relaxation, since relaxing (4) and (5) yields the standard set partitioning formulation (1) with a single side constraint (6).

### 4.1 Algorithm

Instead of relaxing both (4) and (5), we can potentially relax just a single family of these constraints even though the corresponding restricted master problem would no longer be well structured. The next proposition shows that relaxing both families yields the same Lagrangian dual value as relaxing only on a single family. For generality, we give a result that covers slightly more general models.



**Proposition 1.** Consider the following optimization problem

$$\begin{aligned} \max \quad & \sum_{i \in N} p_i z_i \\ & x \in S \\ & z \leq Cx \end{aligned} \tag{13}$$

$$z \leq Dx \tag{14}$$

where  $S$  is an arbitrary non-empty bounded set. Let  $z^1$  be the optimal value of the Lagrangian dual if only (13) are relaxed and let  $z^2$  be the optimal value of the Lagrangian dual if both (13) and (14) are relaxed. Then  $z^1 = z^2$ .

*Proof.* Without loss of generality assume that  $p_i > 0$  for any  $i$ . Let  $\lambda$  be Lagrangian multipliers for (13) and  $\mu$  Lagrangian multipliers for (14). Then

$$z^1 = \min_{\lambda \geq 0} \max_{\substack{x \in S \\ z \leq Dx}} \left( \sum_{i \in N} p_i z_i + \sum_{i \in N} \lambda_i (c^i x - z_i) \right) = \min_{\lambda \geq 0} \max_{\substack{x \in S \\ z \leq Dx}} \left( \sum_{i \in N} (p_i - \lambda_i) z_i + \sum_{i \in N} \lambda_i c^i x \right),$$

where  $c^i$  is the  $i$ th row of  $C$  and  $d^i$  is the  $i$ th row of  $D$ . For  $\lambda$  such that  $\lambda_i > p_i$  for some  $i$ , we take  $z_i = -\infty$ , and therefore the inner maximum equals to  $+\infty$ . The outer minimum is obtained at  $\lambda$  such that  $\lambda_i \leq p_i$  for every  $i$ , since in this case it is finite. To obtain the inner maximum we consider as large  $z_i$  as possible, and therefore  $z_i = d^i x$ . Thus

$$z^1 = \min_{0 \leq \lambda \leq p_i} \max_{x \in S} \left( \sum_{i \in N} (p_i - \lambda_i) d^i x + \sum_{i \in N} \lambda_i c^i x \right). \tag{15}$$

On the other hand

$$z^2 = \min_{\substack{0 \leq \lambda \\ 0 \leq \mu}} \max_{x \in S} \left( \sum_{i \in N} p_i z_i + \sum_{i \in N} (\lambda_i c^i x - z_i) + \sum_{i \in N} (\mu_i d^i x - z_i) \right).$$

Let  $N_+ = \{i \in N : \lambda_i + \mu_i < p_i\}$ ,  $N_0 = \{i \in N : \lambda_i + \mu_i = p_i\}$  and  $N_- = \{i \in N : \lambda_i + \mu_i > p_i\}$ . Then

$$z^2 = \min_{\substack{0 \leq \lambda \\ 0 \leq \mu}} \max_{x \in S} \left( \sum_{i \in N_+} (p_i - \lambda_i - \mu_i) z_i - \sum_{i \in N_-} (\lambda_i + \mu_i - p_i) z_i + \sum_{i \in N} (\lambda_i c^i + \mu_i d^i) x \right).$$

For  $\lambda, \mu$  such that  $N_+ \neq \emptyset$ , we consider  $x_i = 0$ ,  $z_i = +\infty$  for  $i \in N_+$ , and  $z_i = 0$  for  $i \notin N_+$ . For this choice the inner maximum is equal to  $+\infty$ . For  $\lambda, \mu$  such that  $N_+ = \emptyset$  and  $N_- \neq \emptyset$ , we consider  $x_i = 0$ ,  $z_i = 0$  for  $i \notin N_-$ , and  $z_i = -\infty$  for  $i \in N_-$ . Then the inner maximum is again equal to  $+\infty$ . For  $\lambda, \mu$  such that  $N_+ = N_- = \emptyset$  the inner maximum is finite, since  $S$  is bounded. Thus

$$z^2 = \min_{\substack{0 \leq \lambda, 0 \leq \mu \\ \lambda + \mu = p_i}} \max_{x \in S} \sum_{i \in N_0} (\lambda_i c^i + \mu_i d^i) x = \min_{0 \leq \lambda \leq p_i} \max_{x \in S} \left( \sum_{i \in N} (1 - \lambda_i) d^i x + \sum_{i \in N} \lambda_i c^i x \right). \tag{16}$$

Comparing (15) and (16) we conclude  $z^1 = z^2$ .  $\square$

Based on this proposition we relax both constraints (4) and (5). After relaxing (4) and (5), and accord-

ingly fixing the values of  $z$  variables, the restricted master problem reads

$$\max \sum_{\substack{i \in CBL \\ cb \in CB \\ d \in D}} \left( \lambda_{i,cb,d} \sum_{p \in \bar{P}_{i,cb,d} \cap S} y_p + M \mu_{i,cb,d} \sum_{p \in P_{i,cb,d} \cap S} y_p \right) \quad (17a)$$

$$\sum_{p \in P_i \cap S} y_p = 1 \quad i \in L \quad (17b)$$

$$\sum_{p \in P \cap S} c_p y_p \leq (1+r) \cdot C_{OPT} \quad (17c)$$

$$y \text{ binary}, \quad (17d)$$

where  $S$  is a given subset of pairings. Note that this problem is the traditional set partitioning problem with a single side constraint. Since  $z$  variables are not present in constraints, they can be fixed and therefore eliminated from the restricted master problem (this also follows from (16)). Given an optimal solution  $y^*$  to the restricted master problem, we can obtain a feasible solution to CPMC as  $z_{i,cb,d}^* = \min(\sum_{p \in \bar{P}_{i,cb,d} \cap S} y_p^*, M(1 - \sum_{p \in P_{i,cb,d} \cap S} y_p^*))$ . It is clear that  $(y^*, z^*)$  is a feasible solution to CPMC.

The overall solution methodology is given in [Algorithm 1](#). In step 5 we use Lagrangian multipliers to dynamically generate new pairings. A difficulty arises in the fact that the partitioning constraints (3) and the cost bounding constraint (6) are not relaxed and therefore it is not clear how to assign ‘‘dual values’’ to these constraints. At the iteration of the subgradient algorithm that yields the best solution, we solve the LP relaxation of the restricted master problem and we consider the corresponding dual values. We experiment in [Section 5](#) by considering ( $\delta = 1$ ) the dual values of the partitioning constraints or not ( $\delta = 0$ ). The dual value of the cost bounding constraint (6) is always used.

- 1: Let  $S$  be a subset of pairings with low reduced cost based on (1).
- 2: **loop**
- 3: Solve the restricted master problem over pairings in  $S$  by relaxing (4) and (5) and by using a subgradient algorithm.
- 4: Let  $\lambda, \mu$  be the resulting Lagrangian multipliers corresponding to (4) and (5), respectively. In addition, let  $\pi, \gamma$  be the optimal dual solution to the LP relaxation of the restricted master problem at the iteration where the best solution is found corresponding to (17b) and (17c), respectively. Let  $\hat{S}$  be the set of pairings in the incumbent best solution.
- 5: Let  $\tilde{S}$  be a subset of pairings  $p$  with low modified reduced cost defined by

$$\hat{r}c_p = \gamma c_p - \sum_{\substack{i,cb,d \\ p \in \bar{P}_{i,cb,d}}} \lambda_{i,cb,d} - M \sum_{\substack{i,cb,d \\ p \in P_{i,cb,d}}} \mu_{i,cb,d} - \delta \sum_{i \in p} \pi_i, \quad (18)$$

where  $\delta$  is a parameter, which is either 0 or 1.

- 6: Set  $S = \tilde{S} \cup \hat{S}$ .
- 7: **end loop**

**Algorithm 1:** The algorithm

The Lagrangian dual problem of minimizing (17) subject to  $\lambda \geq 0, \mu \geq 0$  is solve with the standard subgradient algorithm, see e.g. [Lemaréchal \(1989\)](#). The Lagrangian multipliers are updated according to the formula  $(\lambda, \mu) = \max\{(0, 0), (\lambda, \mu) - t(u^\lambda, u^\mu)\}$ , where the maximum is taken componentwise. Here  $t$  is the step size defined by

$$t = \frac{t_0}{\alpha^{\lfloor k/\beta \rfloor} \|(u^\lambda, u^\mu)\|^2},$$

where  $k$  is the iteration number,  $\alpha, \beta$  and  $t_0$  are parameters,  $f_k$  is the value of the dual objective function at  $k$ th iteration,  $f_{opt}$  is the best primal objective value found so far and  $(u^\lambda, u^\mu)$  is the subgradient defined as  $u_{i,cb,d}^\lambda = \sum_{p \in \bar{P}_{i,cb,d}} y_p^* - z_{i,cb,d}^*$ ,  $u_{i,cb,d}^\mu = M \sum_{p \in P_{i,cb,d}} y_p^* - z_{i,cb,d}^*$ .

## 4.2 Subproblem Solving

Next we give more details on step 5 of the algorithm. The problem to solve is

$$\min_{p \in P} \{ \gamma c_p - \sum_{\substack{i,cb,d \\ p \in P_{i,cb,d}}} \lambda_{i,cb,d} - M \sum_{\substack{i,cb,d \\ p \in P_{i,cb,d}}} \mu_{i,cb,d} - \delta \sum_{i \in p} \pi_i \}. \quad (19)$$

We use the *constrained shortest path algorithm* with backward node scanning. Details on constrained shortest path can be found in [Desrosiers et al. \(1991\)](#), [Desrosiers et al. \(1995\)](#), and [Desaulniers et al. \(1998\)](#). We first briefly describe the *segment timeline network*, see e.g. [Barnhart et al. \(2003\)](#). The segment timeline network has two distinct nodes for each flight  $i$ , one for the arrival denoted by  $b_i$  and the other for the departure, which is denoted by  $e_i$ . For each flight there is an arc connecting the two nodes. Additionally, the network has an arc between the arrival node of a flight and the departure node of a flight if the connection time between the two flights is within  $[minSit, maxSit] \cup [minRest, maxRest]$  and the arrival station of the first flight is the same as the departure station of the second flight. For ease of exposition we assume that all plane turns are longer than  $minSit$ . The network has two additional nodes  $s, t$ . Every flight with the arrival station being a crew base is connected to  $t$  and  $s$  is connected to every flight originating from a crew base. For every flight  $i$  let  $NS(i), NO(i)$  be the set of all flights  $j$  such that  $(e_i, b_j)$  is an arc in the network with the connection time  $[minSit, maxSit], [minRest, maxRest]$ , respectively. 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. 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.

It is clear that each pairing corresponds to an  $s-t$  path but an  $s-i$  path might violate pairing feasibility rules. In order to circumvent this, to find a favorable pairing the constrained shortest path algorithm must be employed. In such an algorithm, a label is maintained for each feasibility 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, the number of legs in a duty, etc. In addition, to capture the dual prices, an additional label is required. Each  $s-i$  path is represented as a vector consisting of the values of all resources. Thus every node contains a set of label vectors. In the constrained shortest path algorithm a node is selected and then all of its label vectors and adjacent nodes are scanned. For each label vector  $v$  and each neighbor node  $j$  a new label vector is formed by updating  $v$  using the underlying arc and the new label vector is inserted at  $j$ . If a label vector is dominated by another one, it is discarded.

The constrained shortest path algorithm for (19) is based on the segment timeline network and it uses the same resources or labels as the constrained shortest path algorithm for solving the traditional crew pairing problem. For ease of discussion, we assume that the first label corresponds to the dual prices, i.e.  $\widehat{r}c_p - \gamma c_p$ . We show next how to update this label and the necessary changes to the constrained shortest path algorithm.

For each  $i \in L, k \in NS(i) \cup NO(i), cb \in CB, d \in D$  let

$$\alpha_{i,k,cb,d} = \sum_{\substack{j \in NS(i) \\ dt_j < dt_k}} \lambda_{j,cb,d} + \sum_{\substack{j \in NO(i) \\ dt_j < dt_k}} \lambda_{j,cb,d-1}.$$

For each pairing  $p$  and leg  $i \in p$ , let  $cb(p) \in CB$  be the crew base of  $p$ , let  $d(i, p)$  be the number of days in  $p$  from  $i$  till the end of the pairing, and let  $next(i, p)$  be the leg following  $i$  in pairing  $p$ . Now it is easy to see that

$$\widehat{r}c_p - \gamma c_p = -M \sum_{i \in p} \mu_{i,cb(p),d(i,p)} - \delta \sum_{i \in p} \pi_i - \sum_{i \in p} \alpha_{i,next(i,p),cb(p),d(i,p)} = \sum_{i \in p} \beta_{i,next(i,p),cb(p),d(i,p)},$$

where for every  $i \in L, k \in NS(i) \cup NO(i), cb \in CB, d \in D$  we have

$$\beta_{i,k,cb,d} = -M \mu_{i,cb,d} - \delta \pi_i - \alpha_{i,k,cb,d}.$$

We have transformed the dual vector contribution into one that is linear in terms of  $\beta$ 's.

The constrained shortest path starts at  $t$  and it performs a backward scan of nodes. Therefore a label vector encodes an  $i - t$  path. We first precompute all  $\beta$ 's. Given a flight (node)  $i$ , let  $v^i$  be a label vector at  $i$ . In addition, let  $cb(v^i)$  be the crew base corresponding to the underlying  $i - t$  path encoded by  $v^i$  and let  $d(v^i)$  be the number of days on the  $i - t$  path encoded by  $v^i$ . Let  $j$  be a flight such that  $(j, i)$  is an arc in the segment timeline network and we denote by  $v^j$  the newly formed label vector at  $j$ . Then the update of the first label is given by

$$v_1^j = \begin{cases} v_1^i + \beta_{j,i,cb(v^i),d(v^i)} & (j, i) \in NS(j) \\ v_1^i + \beta_{j,i,cb(v^i),d(v^i)+1} & (j, i) \in NO(j) \end{cases}$$

and

$$\begin{aligned} cb(v^j) &= cb(v^i) \\ d(v^j) &= \begin{cases} d(v^i) & (j, i) \in NS(j) \\ d(v^i) + 1 & (j, i) \in NO(j) \end{cases}. \end{aligned}$$

This shows that (19) can be solved by the backward constrained shortest path algorithm.

## 5 Computational Experiments

The computational experiments were carried out on a PC with a 333 MHz Pentium III processor, 520MB of RAM, and Windows 2000 operational system. Microsoft Visual Studio 6.0 was the development environment. We used **ILOG CPLEX 7.5** as the mixed integer programming solver. We have considered 3 instances, which are given in **Table 1**. We have obtained these instances from 2 airlines. The pairing cost structure and the feasibility rules comply with those of the corresponding airline. Both airlines use the hub-and-spoke network structure and they have to comply to the computationally hard 8-in-24 pairing feasibility rule. Even though the first and the third instance do not have many legs, they are computationally hard due to the heavy hub-and-spoke structure. It takes two hours to solve the crew pairing problem (1) for these two instances. An acceptable solution to the crew pairing problem for the second instance is computed in 4 hours. The last column shows the number of move-up crews in the traditional solution.

	No. of legs	No. of crew bases	No. of hubs	No. move-up crews in traditional	
				$M = 1$	$M = 2$
Instance 1	123	3	7	2	3
Instance 2	228	5	11	5	5
Instance 3	80	1	7	4	5

Table 1: Instances

A *robust crew schedule* or robust solution is a solution to CPMC. On the other hand, a traditional crew schedule or traditional solution is a solution to (1). In the first part we study the effects of parameters such as  $M$  and  $r$  on the crew cost and the number of move-up crews. We also exploit the two options in computing the modified reduced cost in the solution methodology. In the second part we provide a computational analysis of the trade-off between the crew cost and robustness. We generate several robust solutions with respect to  $M$  and  $r$  and we evaluate them by generating random disruption. For each disruption we then run a crew recovery decision support system to get the operational crew cost after the disruption. We perform the same analysis for the traditional solution. At the end we compare the resulting operational crew cost, the number of deadheads, the number of used reserved crews, and the number of canceled flights.

### 5.1 Parameter Sensitivity and Solution Methodology Analysis

All presented experiments but the last one in this section were carried out on the first instance. To justify the choice of the Lagrangian decomposition approach for solving CPMC we first give a comparison between

the performance of CPLEX and our Lagrangian decomposition approach. Since CPLEX cannot handle all pairings at once due to their prohibitive number, we first solve the root node LP relaxation of (1). Next we select 10,000 pairings with the lowest reduced cost and for both approaches CPMC is solved only by considering these selected pairings. The results are given in Table 2 and we used  $r = 0.01, M = 2$ . All presented times are CPU times in minutes. The stopping criteria for CPLEX is 900 minutes and the stopping criteria for our approach is always 30 iterations of loop 2-7. Our methodology finds a solution with the same objective value, i.e. the number of move-up crews, in substantially lower computational time. The resulting crew cost shown as “FTC offset” in the second column and measured as the deviation from the FTC of the traditional solution is higher for the solution obtained by Lagrangian decomposition but this is a pure coincidence since the identical constraint (6) is used. The last column “Gap” gives the gap at the end of the computation, which is defined as  $(\bar{z} - z^*)/\bar{z}$ , where  $z^*$  is the move-up value of the best obtained solution and  $\bar{z}$  is the upper bound on the optimal solution produced by the underlying algorithm. The gap of both algorithms is unusually high, but nevertheless the gap of our algorithm is substantially lower than the gap produced by CPLEX. It is clear from the presented results that the underlying mixed integer program is very challenging for branch-and-bound solvers and our Lagrangian decomposition approach is a clear winner.

	FTC offset	time	Move-up count	Gap
CPLEX	3.5%	900	12	62%
Lagrangian Decomposition	4.8%	15	12	43%

Table 2: Comparison of CPLEX and our methodology

Next we show in Table 3 how is the running time effected by different choices of  $M$  and  $r$ . The two cases labeled by † use the following “softer” constraint

$$\sum_{p \in P} c_p y_p \leq (1 + r) \cdot c_{\text{OPT}} + t \quad (20)$$

instead of (6). Here  $t$  is a new nonnegative real variable with a large negative objective coefficient. Note that by using (20) the cost upper bound can be violated but the violation is penalized. The choice of  $M$  does not effect substantially the running time, however increasing  $r$  increases the running time. The reason is that for small values of  $r$  the restricted master problem has only few feasible solutions, which results into a small branch-and-bound tree (branch-and-bound is warm started with the solution of the restricted master problem from the previous iteration). On the other hand, the branch-and-bound algorithm struggles in finding feasible solutions. To elevate this, we have developed softer constraints (20). Indeed, by using this constraint the running time decreases, especially for the case  $M = 2, r = 0.015$ . As shown later, unfortunately these solutions do not perform well in practice.

	$M = 1$ $r = 0.005$	† $M = 1$ $r = 0.005$	$M = 2$ $r = 0.005$	$M = 2$ $r = 0.01$	$M = 2$ $r = 0.015$	† $M = 2$ $r = 0.015$
Time	38	34	59	167	202	70

Table 3: Execution times

We have performed additional experiments, which are geared toward sensitivity analysis and algorithmic choices of the our methodology.

1. Constraint (6) imposes an upper bound on the crew cost. Clearly, larger values of  $r$  produce more move-up crews. For  $r \in \{0.005, 0.001, 0.015, 0.025\}$  and  $M = 2$  the corresponding crew cost and the number of move-up crews are given in Table 4. Recall that  $r$  is not the maximum crew cost, but it is the allowed margin over the crew cost of the traditional solution. As expected, the number of move-up crews grows with increasing  $r$ . As shown later in Section 5.2, the best schedules from the operational

point of view were obtained at  $r = 0.005$ . This shows that although the number of move-up crews is increasing with an increase of  $r$ , it does not provide large enough recovery flexibility to compensate cost loss during regular operations.

$r$	0	0.005	0.01	0.015	0.025
Schedule cost	26015	26132	26255	26375	26627
Move-up count	3	14	17	18	21

Table 4: Effects of  $r$

- Increasing  $M$  in (5) results in a larger number of move-up crews, as shown in Table 5. Here we use  $r = 0.005$ . The last column shows that number of legs that actually have at least one move-up crew. We note that this number increases up until  $M = 3$  and then the solution for  $M = 5$  is clearly inferior. The solutions for  $M = 2$  and  $M = 3$  are very similar. In the experiments that follow we use either  $M = 1$  or  $M = 2$ .

$M$	1	2	3	5
Move-up count	14	21	26	26
No. legs with a move-up crew	14	16	17	13

Table 5: Effects of  $M$

- The size of the restricted master problem and the choice of  $\delta$  in calculation of the modified reduced cost given by (18) effect the solution quality and the running time. Clearly the larger the size of the restricted master problem, the better the solution. However, with the increasing size of the restricted master problem the running times grow. Table 6 confirms these trade-offs. These results were obtained for the second instance with  $M = 2$  and  $r = 0.005$ . Even though the running time for  $\delta = 0$  and the size of 5,000 is the highest one, it gives by far the best quality solution. The running time is acceptable since this is a tactical planning problem. This is the default strategy in the experiments that follow.

	$\delta = 0$		$\delta = 1$	
Size of the restricted master problem	3,000	5,000	3,000	5,000
Number of move up crews	7	15	5	5
Time for 30 iterations	97	287	21	29

Table 6: Algorithmic choices

## 5.2 Robust vs. Traditional

In order to evaluate practical performance of robust schedules, we generate several random disruptions. The number of disruptions is proportional to the number of legs. We use 80 minute shut-downs at airports throughout the day with the number of disruptions proportional to the number of flights. For every disruption, the new crew assignments are obtained by an automated crew recovery module. The recovery module uses sophisticated optimization techniques such as integer programming and heuristics. Every disruption has to be recovered in a 24-hours time window with the objective of minimizing the crew cost of the resulting schedule and minimizing the number of uncovered legs. Crew swapping, reserve crews, deadheading and flight cancellation are all considered and therefore the recovery module accurately mimics recent state-of-the-art recovery procedures.

When comparing the traditional crew schedule with a robust one, we identically disrupt both schedules and recover them. We repeat this for all different scenarios and we report an accumulated difference between performance attributes of the traditional schedule and the robust one. Positive values correspond to robust schedule outperforming the traditional crew schedule. The performance of a schedule is evaluated by the following attributes.

- DH (*number of deadheads*). Often the cheapest recovery solution includes deadheading to reposition crews. The deadhead flight does not necessarily have to be from the considered fleet or even from the same airline.
- RC (*number of stand-by reserve crews*). Stand-by reserve crews are on duty at crew bases and therefore they are being paid even if they are not used. An advantage here is not only in the reduction of the number of used reserve crews, but also in the potential reduction of the number of on-duty reserve crews.
- UL (*number of uncovered legs*). Sometimes, as a result of a disruption, it is impossible to cover all legs with crews. For example, a flight from a spoke, (no reserve crews) whose inbound crew is late and there is not sufficient time to deadhead another crew from a near crew base, requires a cancellation.
- FTC pl (*planned flight-time-credit*). This is the FTC based on planned departure and arrival times and it is known before the day of operation. We report the difference in FTC between the traditional solution and the incumbent solution. If the traditional solution is optimal to (1), then this number is always nonpositive.
- FTC opt (*operational flight-time-credit*). This is the FTC at the end of the day and thus it is based on the operational crew cost. At the first glance, since robust solutions provide more recovery opportunities, their operational FTC should be smaller. However, recall that FTC is based on the block time and therefore, since traditional schedules tend to have more uncovered legs, their operation cost can decrease, which effects FTC.

Since the first three attributes are intangibles, we list them separately and we do not attempt to assign any monetary value.

For the first instance a total of 200 disruptions in 7 stations were generated. The results are given in Table 7. All numbers are cumulative numbers across all disruptions. The last two cases denoted by † correspond to employing the soft cost bounding constraint (20). It is obvious that the first crew schedule with  $M = 2, r = 0.005$  gives by far the best results. It is important to note that theoretically more robust solutions do not necessarily produce lower operational FTC solutions (recall that the flying time is influenced by the number of uncovered legs). Another argument is in the fact that some disruptions do not disrupt the crew schedule and therefore in such cases solutions with lower  $r$  perform better. Due to such predominant behavior of the first solution, for the remaining two instances we always consider  $r = 0.005$ .

	FTC pl %	FTC opt %	DH	RC	UL	MC
$M = 2, r = 0.005$	-0.5	0.32	61	41	18	14
$M = 2, r = 0.015$	-1.5	-0.13	59	69	-7	18
$M = 1, r = 0.005$	-0.5	0.18	25	31	-17	10
$M = 2, r = 0.01$	-1.0	0.17	30	68	-17	17
† $M = 1, r = 0.005$	-0.8	0.19	36	38	-2	14
† $M = 2, r = 0.015$	-1.6	-0.71	17	53	-20	20

Table 7: Comparison of traditional and robust solutions for instance 1

For the second instance a total of 240 disruptions in 11 stations were generated. The results are reported in Table 8. The resulting solution again outperforms the traditional one in all aspects.

	FTC pl %	FTC opt %	DH	RC	UL	MC
$M = 2, r = 0.005$	-0.4	2.5	15	7	35	15

Table 8: Comparison of traditional and robust solutions for instance 2

Instance 3 corresponds to a different airline and therefore many pairing feasibility rules and the pairing cost structure are different. For this fleet a total of 140 disruptions in 7 stations were generated. The comparison of robust and traditional solutions is given in [Table 9](#). Due to a different airline, we performed the comparison also for  $M = 1, r = 0.005$  setting, which seems promising from [Table 7](#). Both of these crew schedules outperform the traditional one. The winner between these two solutions is not clear since the first one outperforms the second one in the number of used deadheads and reserved crews.

	FTC pl %	FTC opt %	DH	RC	UL	MC
$M = 1, r = 0.005$	-0.5	0.27	3	16	1	9
$M = 2, r = 0.005$	-0.4	0.51	1	14	10	11

Table 9: Comparison of traditional and robust solutions for instance 3

Note that the above cost gains are based only on irregular operations. Whenever irregular operations do not occur, the robust crew schedule bears higher crew cost. If there is a single disruption per day, then the second column in [Table 10](#) shows the yearly gains in thousands of dollars. The third column gives the same for a single disruption every two days. Note that in this case the robust solution for the first instance is not profitable solely based on the crew cost. All robust solution in this table have  $M = 2, r = 0.005$ . These annual savings do not take into account the number of deadheads, reserve crews, and the number of uncovered legs. Based on statistical analysis reported by [Bratu and Barnhart \(2000\)](#), the delays tend to be longer and longer in time and therefore our 80 minute disruptions are realistic. In addition, based on the same report a single disruption per day is conservative. When these savings are added over all fleets, an airline can achieve significant savings by using our robust crew solutions.

	1 disruption per day	1 disruption every 2 days
Instance 1	320	-80
Instance 2	2,400	800
Instance 3	200	15

Table 10: Annual savings in thousands of US dollars

## 6 Conclusions

The main contribution of this work is to show that robust crew pairing solutions produce lower operational crew cost. We achieve this by introducing a new model that is based on a realistic and commonly used crew recovery practice of crew swapping. We performed detailed experimental analysis. The main conclusion is that one needs to be cautious in trading off the crew cost for robustness. Sacrificing too much crew cost quickly leads to solutions that do not have operational gains. Nevertheless there is a fine line where the trade-off is beneficial and robust solutions produce significant annual savings.



## Acknowledgement

This research was supported by NSF grant DMII-0084826. We are also obliged to M. Clarke, D. Güenther, and B. Smith from Sabre Holdings, Southlake, TX for their valuable feedback and assistance with data. M. Sohoni from Delta Technology Inc., Atlanta, GA provided us with the state-of-the-art crew recovery module and we are indebted for his contribution.

## References

- Ageeva, Y. (2000). *Approaches to incorporating robustness into airline scheduling*. Master’s thesis, Massachusetts Institute of Technology.
- Barnhart, C., Cohn, A., Johnson, E., Klabjan, D., Nemhauser, G., and Vance, P. (2003). Airline crew scheduling. In R. W. Hall, editor, *Handbook of Transportation Science*. Kluwer Scientific Publishers.
- Bly, E., Clarke, J., Loy, J., Melconian, T., and Rabbani, F. (2003). MIT extensible air network simulation (MEANS). Technical report, Massachusetts Institute of Technology.
- Bond, D. (2000). Commercial aviation on the ropes. *Aviation Week & Space Technology*. September issue.
- Bratu, S. and Barnhart, C. (2000). Air traffic congestions: A macroscopic view. Presentation for the Global Airline Industry Program, Massachusetts Institute of Technology.
- Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, M., and Soumis, F. (1998). A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In T. Crainic and G. Laporte, editors, *Fleet Management and Logistics*, pages 57–93. Kluwer Publishing Company.
- Desrosiers, J., Dumas, Y., Desrochers, M., Soumis, F., Sanso, B., and Trudeau, P. (1991). A breakthrough in airline crew scheduling. Technical Report G-91-11, GERAD.
- Desrosiers, J., Dumas, Y., Solomon, M., and Soumis, F. (1995). Time constrained routing and scheduling. In M. Ball, T. Magnanti, C. Monma, and G. Nemhauser, editors, *Handbook in Operations Research and Management Science, Network Routing*, pages 35–139. Elsevier Science Publishers.
- Donoghue, J. (2004). Gaining strenght. *Air Transport World*, pages 28–34. January issue.
- Ehrgott, M. and Ryan, D. (2001). Bicriteria robustness versus cost optimization in tour of duty planning at Air New Zealand. Technical report, Univeristy of Auckland.
- FAA. <http://api.hq.faa.gov>.
- Fisher, M. (1981). The Lagrangian relaxation method for solving integer programming problems. *Management Science*, **27**, 1–18.
- Kang, L. and Clarke, J. (2003). Degradable airline scheduling. Technical report, Massachusetts Institute of Technology.
- Karow, M. (2003). *Virtual Hubs: An Airline Schedule Recovery Concept and Model*. Master’s thesis, Massachusetts Institute of Technology.
- Lan, S., Clarke, J., and Barnhart, C. (2003). Robust aircraft routing. Technical report, Massachusetts Institute of Technology.
- Lemaréchal, C. (1989). Nondifferentiable optimization. In G. Nemhauser, A. R. Kan, and M. Todd, editors, *Optimization*, pages 529–572. Elsevier Science.
- Listes, O. and Dekker, R. (2003). A scenario aggregation based approach for determining a robust airline fleet composition. Technical Report EI 2002-17, Rotterdam School of Economics.

- Rosenberger, J., Schaefer, A., Goldsman, D., Johnson, E., Kleywegt, A., and Nemhauser, G. (2002). A stochastic model of airline operations. *Transportation Science*, **36**, 357–377.
- Rosenberger, J., Johnson, E., and Nemhauser, G. (2003). A robust fleet assignment model with hub isolation and short cycles. *Transportation Science*. To appear.
- Schaefer, A., Johnson, E., Kleywegt, A., and Nemhauser, G. (2000). Airline crew scheduling under uncertainty. Technical Report TLI-01-01, Georgia Institute of Technology.
- Stojković, G., Soumis, M., and Desrosiers, J. (1998). The operational airline crew scheduling problem. *Transportation Science*, **32**, 232–245.
- Yen, J. and Birge, J. (2000). A stochastic programming approach to the airline crew scheduling problem. Technical report, University of Washington.