# IMPROVING CREW SCHEDULING BY INCORPORATING KEY MAINTENANCE ROUTING DECISIONS


## AMY MAINVILLE COHN

*University of Michigan, Ann Arbor, Michigan 48109, amycohn@umich.edu*

## CYNTHIA BARNHART

*Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, cbarnhar@mit.edu*



Crew costs are the second-largest operating expense faced by the airline industry, after fuel. Thus, even a small improvement in the quality of a crew schedule can have significant financial impact. Decisions made earlier in the airline planning process, however, can reduce the number of options available to the crew scheduler. We address this limitation by delaying some of these earlier planning decisions—specifically, key maintenance routing decisions—and incorporating them within the crew scheduling problem. We present an *extended crew pairing model* that integrates crew scheduling and maintenance routing decisions. We prove theoretical results that allow us to improve the tractability of this model by decreasing the number of variables needed and by relaxing the integrality requirement of many of the remaining variables. We discuss how to solve the model both heuristically and to optimality, providing the user with the flexibility to trade off solution time and quality. We present a computational proof-of-concept to support the tractability and effectiveness of our approach.



Received December 2000; revision received February 2002; accepted March 2002.

*Subject classifications:* Programming, integers, branch and bound: branch and price. Transportation, models, assignment: crews and aircraft.

*Area of review:* Transportation.


## 1. INTRODUCTION

The airline planning process is made up of many large and complex problems. Of these, crew scheduling is of particular importance. Crew costs are the second-largest operating expense faced by the airlines, after fuel; thus, improving the quality of the crew schedule by even a small amount can have significant financial benefits (Anbil et al. 1993, 1991, Gershkoff 1989, Graves et al. 1993).

A key step in solving the crew scheduling problem is to select a minimum cost set of *crew pairings*—sequences of flights that can be flown by a single crew. This step is known as the *crew pairing problem*. The chosen pairings are then combined to form complete schedules for individual crews. The quality of the crew schedule therefore depends on the set of feasible crew pairings. This feasible set can be impacted significantly by decisions made earlier in the planning process.

One restriction on a valid pairing is that two sequential flights cannot be assigned to the same crew unless the time between these flights (known as *connection* or *sit time*) is sufficient for the crew members to travel through the terminal, from the arrival gate of one flight to the departure gate of the next. This minimum connection time can be relaxed if both flights have been assigned to the same aircraft, because the crew remains with the aircraft. We use the term *short connect* to refer to a connection which is feasible for a crew only if the two sequential flights comprising that connection have been assigned to a common aircraft.

The assignment of aircraft to flights occurs in the *maintenance routing problem*, which typically precedes crew scheduling in the airline planning process. In the maintenance routing problem, aircraft are assigned to strings of flights so as to ensure that every aircraft will have adequate opportunity to receive maintenance. This assignment of aircraft to flights determines the set of short connects in the network, thereby impacting the set of feasible pairings permitted in the subsequent crew scheduling problem. Given that the maintenance routing problem does not consider the impact of short connects on the crew scheduling problem, solving the maintenance routing and crew scheduling problems sequentially can result in a suboptimal solution. The goal of our research is to address this limitation.

Prior works by Klabjan et al. (1999) and Cordeau et al. (2000) demonstrate the impact of short connect selection on crew scheduling, and present models for addressing this limitation. Klabjan et al. (1999) reverse the order in which they consider the maintenance routing and crew pairing problems. They solve the crew pairing problem first, assuming all short connects to be valid. They next solve the maintenance routing problem, in which all short connects used in the crew pairing solution are required to be included in the maintenance routing solution. This approach yields significant improvements for many real-world problem instances. It does not guarantee maintenance feasibility, however, and nonpathological instances exist for which the short connects in a crew pairing solution lead to maintenance infeasibility.

Cordeau et al. (2000) present an integrated approach with approximate crew costs, in which they link maintenance routing and crew pairing models by a set of additional constraints. There is one constraint for each short connect, which enforces the rule that we can choose a crew pairing containing that short connect only if we also choose a maintenance routing string containing it. They present a Benders decomposition approach, consisting of a maintenance routing master problem and a crew pairing subproblem, along with a heuristic branching strategy. They present computational results for a number of problem instances and compare the quality of their heuristic to a more basic approach.

In our research, we have developed an *extended crew pairing model (ECP)* which further contributes to this literature. We have focused on three key objectives:

- First, we want to guarantee a maintenance-feasible crew pairing solution;
- Second, we want to provide the user with the flexibility to trade off solution time and quality. When solved completely, our model yields an exact solution to the integrated maintenance routing and crew pairing problem. Alternatively, our approach can be used heuristically to find quality solutions more quickly while still guaranteeing maintenance feasibility;
- Third, we want to leverage the fact that only a fraction of the decisions made in the maintenance routing phase impact the crew pairing problem. By including in the extended crew pairing model only those maintenance decisions which are relevant, we can reduce the size of the model significantly.

The remainder of the paper is structured as follows. In §2 we present the extended crew pairing model. We prove theoretical results that allow us to improve the tractability of this model by eliminating a large number of variables and by relaxing the integrality requirement of many others. In §3 we discuss implementation details associated with *ECP*. In §4 we discuss an alternative approach, the *constrained crew pairing problem*, which can be solved in parallel to *ECP* for improved performance. We provide our conclusions and suggested areas for future research in §5.

## 2. THE EXTENDED CREW PAIRING MODEL

Before introducing the extended crew pairing model, we briefly review the crew pairing and maintenance routing problems.

### 2.1. The Crew Pairing Problem

The crew pairing problem is typically formulated as a *set partitioning problem*, in which we want to find a minimum cost set of feasible crew pairings such that every flight is included in exactly one pairing.

We define the following notation:

- $P$ is the set of feasible pairings;
- $F$ is the set of flights;
- $c_p$ is the cost of pairing $p$;
- $\delta_{fp}$ is an indicator variable that has value 1 if flight $f$ is included in pairing $p$ and 0 otherwise;
- $y_p$ is the binary decision variable associated with pairing $p$. If $y_p = 1$, then pairing $p$ is included in the solution; otherwise, $y_p = 0$.

Given this, we write the crew pairing problem (*CP*) as:

$$\min \sum_{p\in P} c_p y_p \tag{1}$$

st

$$\sum_{p\in P} \delta_{fp} y_p = 1 \qquad \forall\, f \in F \tag{2}$$

$$y_p \in \{0, 1\} \qquad \forall\, p \in P. \tag{3}$$

The objective (1) minimizes the cost of the chosen set of pairings. The cover constraints (2) and integrality constraints (3) state that, for each flight $f$, the number of chosen pairings containing that flight must be exactly one. This formulation eliminates the need to incorporate complicated feasibility rules. It also allows us to linearize the cost function, because the cost associated with each pairing is computed "offline." Note, however, that the number of variables (that is, the number of feasible pairings in the flight network) is exponentially large—often exceeding hundreds of millions. The literature on solving this large-scale integer program includes Anbil et al. (1992), Ball and Roberts (1985), Beasley and Cao (1996), Chu et al. (1997), Crainic and Rousseau (1987), Desaulniers et al. (1997), Hoffman and Padberg (1993), Klabjan et al. (1999), Klabjan and Schwan (1999), Lavoie et al. (1988), Levine (1996), and Vance et al. (1997).

### 2.2. The Maintenance Routing Problem

As is the case with crew pairing, a string-based approach is often used in formulating the maintenance routing problem (for example, see Barnhart et al. 1998).

We define the following additional notation:

- $R$ is the set of feasible route strings;
- $c_r$ is the cost of route string $r$;
- $\alpha_{fr}$ is an indicator variable that has value 1 if route string $r$ contains flight $f$ and 0 otherwise;
- $d_r$ is the binary decision variable associated with route string $r$. If $d_r = 1$, then route string $r$ is included in the solution; otherwise, $d_r = 0$;
- $N$ is the set of nodes which represent points in space and time at which route strings begin or end (and thus, aircraft are needed or become available);
- $g_n^-$ and $g_n^+$ are the ground arc variables representing the number of aircraft on the ground at station $s$ immediately prior to and immediately following time $t$, given a node $n$ that represents time $t$ at station $s$;
- $R^T$ is the set of route strings that span time $T$, an arbitrary time known as the *countline*;
- $N^T$ is the set of nodes with corresponding ground arcs $g_n^+$ spanning the countline;
- $K$ is the available number of aircraft.

Given this, we write the maintenance routing problem (*MR*) as:

$$\min \sum_{r\in R} c_r d_r, \tag{4}$$

st

$$\sum_{r\in R} \alpha_{fr} d_r = 1 \qquad \forall\, f \in F, \tag{5}$$

$$\sum_{\substack{r \text{ ends at}\\ \text{node } n}} d_r + g_n^- - \sum_{\substack{r \text{ starts at}\\ \text{node } n}} d_r - g_n^+ = 0 \qquad \forall\, n \in N, \tag{6}$$

$$\sum_{r\in R^T} d_r + \sum_{n\in Z^T} g_n^+ \leqslant K, \tag{7}$$

$$d_r \in \{0, 1\} \qquad \forall r \in R, \tag{8}$$

$$g_n^+,\ g_n^- \geqslant 0 \qquad \forall n \in N. \tag{9}$$

The objective function (4) minimizes the cost of the chosen route strings—we set the coefficients $c_r$ to zero, given that we are concerned only with finding a feasible solution. The first set of constraints (5) are cover constraints, which state that each flight must be included in exactly one chosen route string. The second set of constraints (6) are *balance constraints*. They ensure that the flow on route strings and ground arcs forms a circulation. The balance constraint for a specific node $n$ states that the number of aircraft on route strings terminating at $n$ plus the flow on the ground arc into $n$ must equal the number of aircraft on route strings originating at $n$ plus the flow on the ground arc out of $n$. Constraint (7) ensures that the total number of aircraft in use at time $T$ (and thus at any point in time, given that the flow forms a circulation) does not exceed the number of aircraft in the fleet. Finally, note that although the route string variables are required to be binary (8), the integrality of the ground arc variables can be relaxed (9), as discussed in Hane et al. (1995).

As in the crew pairing problem, maintenance routing is complicated by an exponentially large number of valid route strings. Barnhart and Talluri (1997) provide a survey of the solution literature.

## 2.3. The Basic Integrated Model

In the crew pairing problem, we want to select a minimum cost set of feasible pairings such that every flight is included in exactly one pairing. Solving this problem after solving the maintenance routing problem can be suboptimal, because the maintenance routing solution can limit the set of feasible pairings from which we can choose. We can therefore improve the quality of the crew scheduling solution by solving the crew pairing and maintenance routing problems simultaneously.

One way to do this is to integrate the existing basic models. To provide as many feasible pairings as possible, we include all short connects in the crew pairing network. We then insure maintenance compatibility by adding one constraint per short connect. This constraint states that we cannot choose a crew pairing containing that short connect unless we also choose a maintenance routing string containing that short connect.

We define the following additional notation:

- $C$ is the set of short connects;
- $\vartheta_{cr}$ is an indicator variable that has value 1 if route string $r$ contains short connect $c$ and 0 otherwise;
- $\eta_{cp}$ is an indicator variable that has value 1 if pairing $p$ contains short connect $c$ and 0 otherwise.

Given this, we write the *basic integrated model (BIM)* as:

$$\min \sum_{p\in P} c_p y_p, \tag{10}$$

st

$$\sum_{p\in P} \delta_{fp} y_p = 1 \quad \forall\ f \in F, \tag{11}$$

$$\sum_{r\in R} \alpha_{fr} d_r = 1 \qquad \forall\ f \in F, \tag{12}$$

$$\sum_{\substack{r \text{ ends at}\\ \text{node } n}} d_r + g_n^- - \sum_{\substack{r \text{ starts at}\\ \text{node } n}} d_r - g_n^+ = 0 \qquad \forall\ n \in N, \tag{13}$$

$$\sum_{r\in R^T} d_r + \sum_{n\in Z^T} g_n^+ \leqslant K, \tag{14}$$

$$\sum_{r\in R} \vartheta_{cr} d_r - \sum_{p\in P} \eta_{cp} y_p \geqslant 0 \qquad \forall\ c \in C, \tag{15}$$

$$d_r \in \{0, 1\} \qquad \forall r \in R, \tag{16}$$

$$g_n^+,\ g_n^- \geqslant 0 \qquad \forall n \in N, \tag{17}$$

$$y_p \in \{0, 1\} \quad \forall p \in P. \tag{18}$$

Objective Function (10) and Constraint Sets (11) and (18) are the same as in the *CP* model. Constraint Sets (12), (13), (14), (16), and (17) are the same as in the *MR* model. These two models are linked together by Constraint Set (15), which states that we cannot choose a pairing containing a short connect unless we also choose a maintenance solution containing that short connect.

## 2.4. The ECP Formulation

The basic integrated model has two important deficiencies—its large size and its weak LP relaxation—leading to intractability for many problem instances. To address this, we present an alternative model called the *extended crew pairing model (ECP)*. In this model, we start with the basic crew pairing model and add a collection of variables, each of which represents a *complete solution* to the maintenance routing problem. Thus, we can eliminate all the constraints associated with the original maintenance routing problem, replacing them with a single *convexity constraint*. This constraint ensures that we select exactly one maintenance routing solution. We ensure compatibility between this maintenance routing solution and the chosen crew pairings by including one constraint for each short connect, which specifies that we cannot choose a pairing containing that short connect unless we also choose a maintenance routing solution containing it.

We define the following additional notation:

- $S$ is the set of feasible maintenance routing solutions;
- $\beta_{cs}$ is an indicator variable that has value 1 if short connect $c$ is included in maintenance solution $s$ and 0 otherwise;
- $x_s$ is the binary decision variable associated with maintenance solution $s$. If $x_s = 1$, then maintenance solution $s$ is chosen; otherwise, $x_s = 0$.

Given this notation, we write the extended crew pairing model (*ECP*) as:

$$\min \sum_{p\in P} c_p y_p, \tag{19}$$

st

$$\sum_{p\in P} \delta_{fp} y_p = 1 \qquad \forall\, f \in F, \tag{20}$$

$$\sum_{s\in S} \beta_{cs} x_s - \sum_{p\in P} \eta_{cp} y_p \geqslant 0 \qquad \forall\, c \in C, \tag{21}$$

$$\sum_{s\in S} x_s = 1, \tag{22}$$

$$x_s \in \{0, 1\} \qquad \forall s \in S, \tag{23}$$

$$y_p \in \{0, 1\} \qquad \forall p \in P. \tag{24}$$

The objective (19) and the cover constraints (20) are the same as in the basic crew pairing model. Constraint (22), in conjunction with (23), ensures the selection of exactly one solution to the maintenance routing problem. Constraint Set (21) eliminates pairings that contain short connects not included in this selected maintenance solution.

This model yields an optimal integrated solution if all maintenance routing solutions are included. Furthermore, if only a subset of the maintenance solutions are considered (including that which would have been provided from the sequential approach currently used in practice), we are still guaranteed a feasible solution to the integrated model, which is at least as good as that found using the sequential approach.

Associated with this model, however, are a number of concerns regarding tractability. How many maintenance solutions will be needed in this new model? How can they be identified? Will the new model have too many binary variables? We address these concerns in the remainder of this section.

## 2.5. Reducing Problem Size Through Variable Elimination

We can solve *ECP* to optimality by including one column for every feasible maintenance routing solution. In practice, however, this is usually not an option, given that the number of feasible maintenance solutions is an exponential function of the number of flights. In this section we leverage the fact that only certain key maintenance routing decisions impact the crew pairing problem, thereby allowing us to significantly decrease the number of maintenance variables required to ensure an optimal solution.

In our initial description of the *ECP* model, we referred to maintenance routing solution variables. An examination of the constraint matrix, however, highlights the fact that most maintenance routing decisions are implicit in the variable definition, with only short connect information stated explicitly. In other words, we can think of a maintenance column not as a full-blown specification of a feasible solution to the maintenance routing problem, but as a *short connect set for which a feasible maintenance routing solution exists*. We refer to this as a *maintenance-feasible short connect set*. This has important ramifications for the size and structure of *ECP*, because there may be many different feasible maintenance routing solutions associated with a given set of short connects. In particular, when solving the crew pairing model for a given short connect set, it is not relevant to crew scheduling *how* the aircraft are routed, but rather that a feasible routing exists.

We claim that in order to guarantee an optimal solution to *ECP*, it is not necessary to include one column for each feasible maintenance routing solution. Instead, it is sufficient to include one column for each *unique and maximal (UM)* maintenance-feasible short connect set. We define this terminology in the following sections.

**2.5.1. Uniqueness.** Consider a basic example with six flights (A, B, C, D, E, and F) and one short connect (A–B). Suppose that there are two feasible maintenance solutions of two route strings each (A–B–C and D–E–F or A–B–F and D–E–C). Note that these two solutions will result in identical columns in *ECP*, with 1s in the row for short connect A–B and in the convexity constraint, and 0s throughout the rest of the column. We need to include only one of these two columns in *ECP*—both solutions define the same set of feasible crew pairings.

We refer to this potential for reducing the required number of columns in *ECP* as *uniqueness*. In other words, we need only one column for each unique maintenance-feasible short connect set, rather than one per distinct maintenance routing solution. To gain some sense of the impact of this reduction, we selected a set of short connects from an actual airline problem instance and began generating distinct feasible maintenance routing solutions, requiring each to contain exactly these selected short connects. We found over 8,700 distinct solutions, all of which could be represented by a single column in *ECP*. Because the number of potential short connects is typically only a small fraction of the number of possible aircraft connections in the network, we believe that similar reductions in the number of required maintenance columns will be found in most real-world instances.

**2.5.2. Maximal Sets.** We take the notion of uniqueness one step further by defining the concept of a *maximal set*. Consider the case where one maintenance solution contains three short connects, denoted by U–V, W–X, and Y–Z. Another solution contains short connects U–V and Y–Z only. Clearly, any crew pairing that is feasible for the second maintenance routing solution is also feasible for the first solution. Therefore, we can discard the second column. In doing so, we do not eliminate from *ECP* any feasible crew pairing solutions, because the chosen maintenance solution tells us which short connects are *permissible* for the crews, not *required*. More generally, we note that it is necessary only to include columns that correspond to *maximal sets*—that is, columns representing maintenance-feasible short connect sets for which adding any additional short connects would result in maintenance infeasibility.

By considering only those maintenance solution columns with *UM* short connect sets, we dramatically reduce the number of columns required in our model. In another example, we looked at a problem instance containing over 25,000 distinct solutions to the maintenance routing problem. Only *four* of these solutions corresponded to unique and maximal short connect sets.

Given the complete set of maintenance routing solutions, we can remove any column that does not correspond to a *UM* short connect set without eliminating any feasible crew pairing solutions. Therefore, it is sufficient to include in *ECP* only those columns that represent unique and maximal maintenance-feasible short connect sets.

## 2.6. Identifying UM Columns

It is not sufficient that the number of required maintenance columns be small. We must also be able to identify these columns in a reasonable fashion—for example, we don't want to have to generate *all* 25,000 feasible maintenance routing solutions from our earlier example in order to isolate those four which contain unique and maximal short connect sets. We can identify *UM* maintenance routing solutions by solving a series of maintenance routing problems with side constraints and a modified objective function.

We begin by solving the maintenance routing problem defined by (5), (6), (7), (8), and (9), with the objective of maximizing the total number of short connects included in the solution. We associate with each route string $r$ a coefficient $c_r$, the number of short connects included in route string $r$. Our objective function is then

$$\min \sum -c_r d_r,$$

where $d_r$ is the binary decision variable associated with route string $r$.

Clearly, a solution to this problem yields a *UM* set of short connects. Let $C^1$ represent the set of short connects included in the solution to this first iteration of the modified maintenance routing problem. We then add a constraint to the modified maintenance routing problem of the form

$$\sum_{c \in C \setminus C^1} \sum_{r \in R} \vartheta_{cr} d_r \geqslant 1.$$

This constraint states that the solution to the next iteration must include at least one route string containing a short connect which is *not* in $C^1$. We then re-solve and generate a new solution with a new corresponding set of short connects $C^2$. This set will be *UM*, because it is the largest cardinality set which is not a subset of another solution. We can continue to iterate, adding cuts for each newly generated solution and re-solving the model until all *UM* short connect sets have been identified.

Thus, it is possible to generate $n$ *UM* columns in at most the amount of time it takes to solve $n$ maintenance routing problems. In fact, it may take significantly less time, because at each iteration we can use the previous iteration's maintenance routing solution as an advanced start.

## 2.7. Structural Properties of ECP

**2.7.1. Relaxing the Integrality of the Maintenance Variables.** Even a relatively small number of additional binary variables can have significant impact on the tractability of a model. This is one of the challenges posed by the *BIM* model—in addition to all the binary crew pairing variables, we add one binary variable for each route string as well. For the *ECP* model, however, Cohn (2002) proves that the binary constraints on the maintenance solution variables can be relaxed. That is, we can replace the integrality constraints with nonnegativity constraints—once we have fixed the crew pairing variables to binary values, the resulting polyhedron will have integer extreme points. This is in contrast to the *BIM* model, where we may have to branch not only on crew pairing variables, but on maintenance routing variables as well. The result is that *the* ECP *model has no more integer variables than the original crew pairing model by itself*. Moreover, this property holds true even if the maintenance routing problem is posed as an optimization problem, contributing to the objective function, rather than as a feasibility problem.

**2.7.2. Quality of the LP Relaxation.** The *ECP* model not only has fewer integer variables than the basic integrated model, but has a provably tighter linear programming relaxation as well. To see this, we first note that any solution to the LP relaxation of *ECP* has a corresponding solution to the LP relaxation of *BIM* with the same cost. To construct this corresponding solution, consider each maintenance routing solution column $x_s$ in the solution to the LP relaxation of *ECP* with strictly positive value. By definition, each of these columns must correspond to at least one feasible maintenance routing solution. Choose one feasible maintenance solution for each such $s$—we denote this by $X_s$—and assign the value $x_s$ to each of the variables in *BIM* which correspond to the route strings in the set $X_s$. Using a one-to-one mapping of crew pairing variables, it is clear that the constructed solution is feasible for *BIM* and has the same cost as the *ECP* solution. Thus, the LP relaxation of *ECP* is at least as tight as the LP relaxation of *BIM*. An example is provided in Cohn (2002) to show that the *ECP* LP relaxation can produce strictly tighter bounds than that of *BIM*.

## 3. IMPLEMENTING ECP

The extended crew pairing model is an extension of the basic crew pairing model, a problem that is itself computationally challenging. The crew pairing problem is often solved using a *branch-and-price* solution approach (Barnhart et al. 1998), in which column generation is used to solve the LP relaxations at nodes of the branch-and-bound tree. The branch-and-price approach to solving crew pairing problems can be extended quite easily to incorporate the added variables and constraints associated with the maintenance routing decisions in *ECP* (see Figure 1). In the sections that follow, we discuss the details of the implementation.

**Figure 1.** *ECP* Branch-and-price algorithm.

(1) Initialize root node
  (a) Set lower bound to $-\infty$.
  (b) Set upper bound to $\infty$.
  (c) Create initial set of crew pairings.
  (d) Create initial set of maintenance solutions.
(2) Choose a pending node and solve.
  (a) Solve current LP and compute dual values.
  (b) For each crew base, look for a negative reduced cost crew pairing.
  (c) Look for a negative reduced cost maintenance solution.
  (d) If any new columns are identified, add to the current LP and return to Step 2a.
(3) Update lower bound.
(4) If solution is fractional:
  (a) If objective value is not strictly less than upper bound, discard the node.
  (b) If objective value is stricly less than upper bound, create two new nodes.
(5) If solution is integer:
  (a) If objective value is strictly less than upper bound, update upper bound.
  (b) Discard the node.
(6) If pending nodes exist, return to Step 2.

## 3.1. Initializing the Root Node

In solving the basic crew pairing problem, we begin by creating an initial set of feasible pairings. In *ECP*, we must also generate an initial set of feasible maintenance solutions. We can generate one such feasible solution by solving the original maintenance routing problem. Note that if we were to include only this column in *ECP*, we would achieve the same result as that found using the sequential solution process currently employed by many airlines.

To identify additional maintenance columns, we can follow the steps described in §2.6. Note that $n$ iterations of this process will yield the $n$ *UM* short connect sets of largest cardinality. As illustrated in the proof-of-concept later in this section, even a small number of such *UM* columns can have a significant impact on the quality of the solution. Users may also choose to provide some additional columns based on their knowledge of the problem domain. For example, they might choose to include columns associated with the maintenance routing solutions implemented over the past several planning periods, or choose to seek maintenance solutions that contain short connects known to be beneficial to the crew schedule.

## 3.2. Generating Crew Pairings

To see how crew pairings can be generated in *ECP*, we first review how pairings are generated in the basic crew pairing model. In *CP*, the reduced cost of a pairing is the cost of the pairing minus the sum of the duals associated with the flights contained in it. If we denote the dual variable associated with the cover constraint for flight $f$ by $\pi_f$, then we can write the *pricing problem* (that is, the optimization problem used to identify the most negative reduced cost variables) as:

$$\min c_p - \sum_{f\in F} \delta_{fp}\pi_f,$$

st

$$p \in P.$$

Barnhart et al. (1999) provide a survey of techniques used in solving such problems. Typically, a crew pairing network is defined, for example, with nodes representing flights and arcs representing feasible flight connections. Specialized algorithms are then used to find the shortest path in this network, where side constraints ensure that a path corresponds to a feasible pairing and the length of a path is equivalent to the reduced cost of the pairing.

In the extended crew pairing model, we must also take into account the dual variables associated with the short connect linking constraints. If we denote by $\gamma_c$ the dual variable associated with short connect $c$, then the reduced cost of a pairing becomes

$$c_p - \sum_{f\in F} \delta_{fp}\pi_f + \sum_{c\in C} \eta_{cp}\gamma_c.$$

By assigning the dual values associated with short connects to the corresponding connection arcs in the crew pairing network, we can solve this pricing problem using the same techniques as those used when solving the basic crew pairing problem. In other words, by simply modifying some of the input parameters, we can use the same pairing generator for both the basic and the extended crew pairing models.

## 3.3. Generating Maintenance Columns

Column generation may also be used to identify additional maintenance solution columns. If we denote by $\sigma$ the dual variable associated with the convexity constraint, then the reduced cost of a maintenance column is

$$-\sum_{c\in C} \beta_{cs}\gamma_c - \sigma.$$

(Recall that the cost of a maintenance column in *ECP* is 0.) The pricing problem can thus be written as

$$\min -\sum_{c\in C} \beta_{cs}\gamma c,$$

st

$$s \in S.$$

If the optimal solution to this pricing problem has objective value less than $\sigma$, then we have identified a new negative reduced cost column. Otherwise, we have established that no new negative reduced cost maintenance columns exist for the current dual values.

Let $R(s)$ denote the set of route strings found in maintenance routing solution $s$. Then,

$$-\sum_{c\in C} \beta_{cs}\gamma_c = \sum_{r\in R(s)} \left(-\sum_{c\in C} \vartheta_{cr}\gamma_c\right).$$

We can therefore formulate the maintenance pricing problem as a basic maintenance routing problem in which the cost coefficient $c_r$ associated with route string $r$ is

$$-\sum_{c\in C} \vartheta_{cr}\gamma_c.$$

We can enhance this pricing problem by adding an appropriately small constant $\Delta$ to each of the duals $\gamma_c$. This ensures that if the short connects associated with one feasible solution are a subset of the short connects in another feasible solution, then the second solution will have an objective value that is lower than that of the first. (This addresses the case where some of the dual values are zero; note that the $\Delta$s must be removed from the duals before comparing the optimal solution to $\sigma$.) Thus, any solution to the maintenance pricing problem will be *UM*.

### 3.4. Branching Strategy

As discussed in §2.7.1, the maintenance variables are guaranteed to take integer values once the crew pairing variables have been fixed to integer values. We can therefore directly apply the branching strategy used in basic crew pairing solvers.

### 3.5. Proof-of-Concept

Given that a significant optimality gap has been shown to exist between the sequential and integrated approaches (Cordeau et al. 2000), we wanted to get some sense of how difficult it is to capture a significant portion of this potential for improvement using *ECP* heuristically. To test the efficacy of our approach, we performed a limited experiment on two small test examples. We considered two actual airline problem instances, both containing approximately 125 flights. We used the following strategy for generating maintenance columns in both instances. First, we generated the 10 *UM* maintenance-feasible short connect sets of largest cardinality. Then, we identified those short connects that were not part of any of these columns. For each of these short connects, we generated the largest cardinality *UM* maintenance-feasible short connect set that contains it, thereby ensuring that each short connect had the potential to be included in a crew pairing. As a result of this approach, problem instance *A* had 16 maintenance columns and problem instance *B* had 20 maintenance columns. We then solved *ECP*, completely enumerating all feasible pairings, but not generating any further maintenance variables.

In order to analyze the quality of our solution, we generated a lower bound by computing the optimal crew pairing solution when all short connects are permitted.

As shown in Table 1, with no more than 20 maintenance solution columns we generate a solution at most 2% away from optimal in both problem instances. We believe that this is largely due to the fact that the number of short connects used in an optimal integrated solution is often much smaller than the total number of short connects. For example, problem instance A had 58 possible short connects.

**Table 1.** Proof-of-Concept.

| Problem Instance | ECP Solution | Lower Bound | Optimality Gap |
|---|---|---|---|
| A | 31,396.10 | 31,396.10 | 0.0% |
| B | 25,498.60 | 25,076.60 | 1.7% |

The 16 short connect columns that we generated contained on average just over 38 short connects each. The optimal solution used only 9 short connects. In problem instance B there were 68 possible short connects. The 20 short connect columns generated contained on average 37 short connects each. The optimal solution used only 10. This indicates that with *UM* columns of maximal cardinality, we may capture the optimal set of short connects even with only a small number of short connect columns.

## 4. The Constrained Crew Pairing Model

One benefit of the *ECP* model is that it can either be solved to optimality or it can provide a feasible solution in less time if we limit the number of maintenance routing solutions from which to choose. This is an advantage over the approach of Klabjan et al. (1999), which may result in an infeasible solution. On the other hand, the Klabjan approach has the benefit of quickly finding an optimal solution in those instances where the optimal crew pairing problem is in fact maintenance feasible. In contrast, the *ECP* model may take much longer to solve such problem instances if an appropriate maintenance solution is not included in the model at an early iteration.

To address this drawback, we present an alternative approach, which we call the *constrained crew pairing model (CCP)*. In this approach, we start off by solving the unconstrained crew pairing problem (that is, the crew pairing problem in which all potential short connects are permitted). If the short connects used in the solution are maintenance feasible, we have an optimal solution and the algorithm terminates. If not, we add a cut to the crew pairing problem ruling out the current infeasible solution, and we reiterate.

This approach can require many iterations, each of which requires us to solve an instance of the crew pairing problem (with side constraints) and an instance of the maintenance routing problem. Another drawback is that this approach does not provide us with a feasible solution until we reach the optimal solution. To remedy these issues we suggest solving this model *in parallel* to *ECP*, as discussed in the sections that follow.

### 4.1. Implementing *CCP*

To solve *CCP*, we begin by solving the basic crew pairing problem in which *all* potential short connects are permitted. Given the resulting set of short connects used, denoted $F^0$, we then solve a maintenance routing problem modified to ensure that all short connects in $F^0$ are included in the solution. For example, if short connect *A*–*B* appears in a crew pairing in the current solution, we do not permit route strings which contain flight $A$ followed by any flight other than $B$. Note that this actually simplifies the maintenance problem by restricting the set of feasible strings.

If we are able to find a feasible solution to this maintenance routing problem, we have an optimal solution to the crew pairing problem and the algorithm terminates. Otherwise, we have a lower bound which we can use in the *ECP* branch-and-bound algorithm. (Note that *ECP* also generates alternative lower bounds based on its LP relaxation.)

If the current crew solution is maintenance infeasible, we proceed by: 1) adding a cut to the crew pairing model to prohibit the current solution, and 2) re-solving the modified crew pairing problem. For example, if the current optimal solution is $Y^0$ and this solution uses $N^0$ pairings, then we add a cut of the form

$$\sum_{p\in P:\, y_p^0=1} y_p \leqslant N^0 - 1. \tag{25}$$

These cuts are not very efficient, however. For example, when we solve the $i+1$th iteration of *CCP*, the new solution might contain a different set of pairings from the $i$th iteration but the same maintenance-infeasible set of short connects. Thus, a better cut would be one that prohibits the short connect set $F^i$ rather than the set of pairings $Y^i$. Such a cut can be written as

$$\sum_{p\in P}\sum_{c\in F^i} \eta_{cp} y_p \leqslant |F^i| - 1. \tag{26}$$

One inefficiency associated with Cut (26) is that we might generate several "nested" solutions in successive iterations that are all infeasible. For example, consider the case where $F^i = \{A, B, C, D\}$. Perhaps this solution is maintenance infeasible because short connects $A$ and $B$ are incompatible. These two short connects might be desirable to the crew pairing problem, however. Our next three iterations might therefore yield solutions containing short connect sets $\{A, B, C\}$, $\{A, B, D\}$, and then $\{A, B\}$, all of which are maintenance infeasible.

We can bypass these intermediate iterations by prohibiting short connect set $\{A, B\}$ rather than the original set $\{A, B, C, D\}$. More generally, we want our cuts to represent *minimally infeasible* subsets $F^{i'}$ of $F^i$; that is, $F^{i'}$ is also maintenance infeasible, but any proper subset of $F^{i'}$ is maintenance feasible. The new cut is then written as:

$$\sum_{p\in P}\sum_{c\in F^{i'}} \eta_{cp} y_p \leqslant |F^{i'}| - 1.$$

Just as we want maintenance-feasible short connect sets corresponding to columns in *ECP* to be as *large* as possible, we want maintenance-infeasible short connect sets corresponding to constraints in *CCP* to be as *small* as possible. In the same way that using maximal short connect sets allows us to minimize the number of maintenance columns needed in *ECP*, using minimally infeasible short connect sets in *CCP* allows us to minimize the number of maintenance cuts needed.

**4.1.1. Generating Minimally Infeasible Short Connect Sets.** Given a short connect set $F^i$ from the $i$th iteration of *CCP*, we first solve a variation of the maintenance routing problem in which our objective is to maximize the number of short connects in $F^i$ included in the solution. If the optimal objective value to this problem contains all the short connects in $F^i$, then the crew pairing solution is maintenance feasible and therefore optimal for *ECP*. Otherwise, the short connect set is infeasible and we seek the smallest maintenance-infeasible subset of $F^i$ to generate a new cut for *CCP*—we refer to this problem as the *minimally infeasible set problem (MIS)*.

Consider a minimally infeasible subset $F^{i'}$. For every feasible solution to the maintenance routing problem, there must be at least one element of $F^{i'}$ *not* included in that solution; otherwise there would be a maintenance solution containing all short connects in $F^{i'}$. To model *MIS*, we let $f_c$ be a binary decision variable indicating whether or not $c \in F^i$ is part of the minimally infeasible set, and use $C(s)$ to denote the set of short connects occurring in maintenance solution $s$. We can then formulate *MIS* as:

$$\min \sum_{c\in F^i} f_c,$$

st

$$\sum_{c\in F^i\setminus C(s)} f_c \geqslant 1 \qquad \forall\, s \in S,$$

$$f_c \in \{0, 1\} \qquad \forall\, c \in F^i.$$

For example, if the initial infeasible short connect set is $F^i = \{A, B, C, D\}$ and the maintenance routing problem has three feasible solutions, containing short connects $\{A, C\}$, $\{B, C, D\}$, and $\{A, C, D\}$, respectively, then the minimally infeasible set problem would be:

$$\min f_A + f_B + f_C + f_D,$$

st

$$f_B + f_D \geqslant 1, \tag{27}$$

$$f_A \geqslant 1, \tag{28}$$

$$f_B \geqslant 1, \tag{29}$$

$$f_A,\ f_B,\ f_C,\ f_D\ \in \{0, 1\}. \tag{30}$$

The optimal solution to this is $\{A, B\}$, and thus we add the cut

$$\sum_{p\in P}\sum_{c\in\{A,B\}} \eta_{cp} y_p \leqslant 1$$

to eliminate any crew pairing solutions containing both short connects $A$ and $B$.

Note that Constraint (27) is redundant—it is dominated by Constraint (29). Note also that Constraints (28) and (29) correspond to short connect sets $\{B, C, D\}$ and $\{A, C, D\}$, which are both maximal. This demonstrates the fact that in the *MIS* model, we need only include one constraint for each maximal short connect set. This fact is important from an implementation standpoint, as it can significantly decrease the number of constraints required in the model.

Another way to improve performance is to use *cut generation* in solving *MIS* itself. In this approach, we begin by solving a restricted version of *MIS* which contains just the set of constraints corresponding to the *UM* columns currently included in the *ECP* model. Given the solution to *MIS*, we solve a maintenance routing problem attempting to identify a *UM* maintenance solution containing this short connect set. If we find such a *UM* maintenance solution, the short connects in the restricted *MIS* solution are maintenance feasible and we identify a violated inequality. We add the cut corresponding to this *UM* short connect set to our restricted *MIS* and repeat. Note that we can also add a new column to *ECP*, as we have found a *UM* maintenance solution not currently contained in the restricted master.

If we *do not* find a *UM* short connect set containing the short connects in the current solution to the restricted *MIS*, this solution to the restricted *MIS* is also optimal for the original *MIS*. Thus, we have found a minimally infeasible short connect set. We add the corresponding cut to *CCP* and solve to find a new lower bound on *ECP*.

We summarize the steps to *CCP* as follows:

*Step* 1. Initialize *CCP* as the basic crew pairing model, in which all short connects are assumed to be feasible crew connections.

*Step* 2. Solve the current version of *CCP*; the optimal objective value provides an updated lower bound on *ECP*. Let $F^i$ denote the short connects used in this solution.

*Step* 3. Solve a maintenance routing problem to find the *UM* short connect set using the maximum number of elements from $F^i$. If all elements from $F^i$ are included, the current crew pairing solution is optimal and the algorithm terminates. Otherwise, we have established that $F^i$ is a maintenance-infeasible short connect set, and we can construct a new cut for *CCP*.

*Step* 4. Initialize an instance of *MIS*, corresponding to $F^i$, to contain one constraint for every *UM* short connect set in *ECP*.

*Step* 5. Solve the current version of *MIS*; denote the solution by $F^{i'}$.

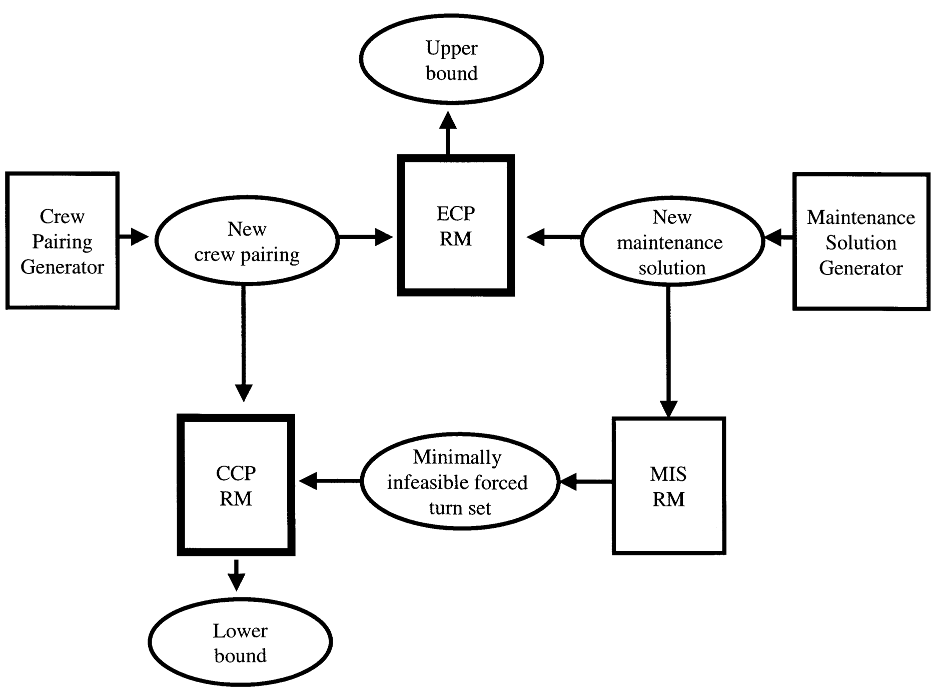
*Step* 6. Solve a maintenance routing problem to find the *UM* short connect set using the maximum number of elements from $F^{i'}$. If one or more elements from $F^{i'}$ *are not* included, then $F^{i'}$ is maintenance infeasible and thus a minimally infeasible short connect set has been identified. Add the cut corresponding to $F^{i'}$ to *CCP* and return to Step 2.

*Step* 7. If all elements from $F^{i'}$ *are* included in the maintenance routing solution, then we have identified a violated cut for *MIS*. We add this cut to the restricted *MIS* and return to Step 5. We also add the maintenance routing solution to *ECP*, given that it is a *UM* short connect set not currently included in *ECP*.

## 4.2. Synergies Between *ECP* and *CCP*

It is interesting to note the synergies between *ECP* and *CCP*. We demonstrate this in Figure 2. This figure presents the output of each of the subproblems that we have described and how this output may be used. *ECP* and *CCP* generate increasingly tighter upper and lower bounds, respectively, allowing us to determine when the optimality gap is sufficiently small to terminate the algorithm. Crew

**Figure 2.** Model synergy.



pairings generated when solving *ECP* can also be added to the restricted master for *CCP* and vice versa. Maintenance solutions generated for *ECP* lead to new cuts for *MIS*, and new cuts for *MIS* lead to new columns for *ECP*. In addition, solutions to *MIS* lead to new cuts for *CCP*. Finally, note that the user can control how to best utilize this collection of models.

## 5. CONCLUSIONS

In this paper we present the extended crew pairing model and solution approach. This new approach solves the important real-world problem of integrating the maintenance routing and crew scheduling problems. We have advanced the existing research in this area by focusing on three main goals: guaranteeing maintenance feasibility, providing user flexibility to trade off solution time and quality, and leveraging the fact that only a fraction of the maintenance routing decisions are relevant to the crew pairing problem. We have satisfied these goals in a model that has no more binary decision variables than the basic crew pairing model alone. Furthermore, this model is flexible in that it can directly incorporate new advances in maintenance solvers and pairing generators and it can be used in a column generation framework. We have also demonstrated that the LP relaxation of *ECP* is tighter than that of the basic integrated approach. We have proposed an alternative approach, *CCP*, that can work in parallel with *ECP*, in some cases more quickly identifying an optimal solution and in others providing useful information in the form of increasingly tight lower bounds and additional maintenance routing solution variables. Finally, we believe that our model has greater potential for extendibility to include additional airline planning decisions.

## ACKNOWLEDGMENTS

This research is supported in part by a National Science Foundation Fellowship, a United Airlines research grant, and the Alfred P. Sloan Foundation research grant establishing the MIT Global Airline Industry Program.

## REFERENCES

Anbil, R., R. Tanga, E. Johnson. 1992. A global approach to crew-pairing optimization. *IBM Systems J.* **31** 71–78.

——, E. Gelman, B. Patty, R. Tanga. 1991. Recent advances in crew-pairing optimization at American Airlines. *Interfaces* **21** 62–74.

——, C. Barnhart, L. Hatay, E. Johnson, V. Ramakrishnan. 1993. Crew-pairing optimization at American Airlines decision technologies. T. Ciriano, R. Leachman, eds. *Optimization in Industry: Mathematical Programming and Modeling Techniques in Practice*. John Wiley and Sons, Chichester, U.K., 31–36.

Ball, M., A. Roberts. 1985. A graph partitioning approach to airline crew scheduling. *Transportation Sci.* **19** 107–126.

Barnhart, C., K. Talluri. 1997. Airline operations research. Arthur McGarity, Charles ReVelle, eds. *Design and Operation of Civil and Environmental Engineering Systems*. John Wiley and Sons, New York, 435–469.

——, E. Johnson, G. Nemhauser, P. Vance. 1999. Crew scheduling. Randolph W. Hall, ed. *Handbook of Transportation Science*. Kluwer Academic Publisher, Norwell, MA, 493–521.

——, ——, ——, M. Savelsbergh, P. Vance. 1998. Branch-and-price: Column generation for solving huge integer programs. *Oper. Res.* **46** 316–329.

——, N. Boland, L. Clarke, E. Johnson, G. Nemhauser, R. Shenoi. 1998. Flight string models for aircraft fleeting and routing. *Transportation Sci.* **32** 208–220.

Beasley, J., B. Cao. 1996. A tree search algorithm for the crew scheduling problem. *Eur. J. Oper. Res.* **94** 517–526.

Chu, H., E. Gelman, E. Johnson. 1997. Solving large scale crew scheduling problems. *Eur. J. Oper. Res.* **97** 260–268.

Cohn, A. M. 2002. Composite-variable modeling for large-scale problems in transportation and logistics. Working paper, Massachusetts Institute of Technology, Cambridge, MA.

Cordeau, J., G. Stojković, F. Soumis, J. Desrosiers. 2000. Benders decomposition for simultaneous aircraft routing and crew scheduling. Technical Report G-2000-37, GERAD, École Polytechnique de Montréal, Montréal, Quebec, Canada.

Crainic, T., K. Rousseau. 1987. The column generation principle and the airline crew scheduling problem. *INFOR* **25** 136–151.

Desaulniers, G., J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M. Solomon, F. Soumis. 1997. Crew pairing at Air France. *Eur. J. Oper. Res.* **97** 245–259.

Gershkoff, I. 1989. Optimizing flight crew schedules. *Interfaces* **19** 29–43.

Graves, G., R. McBride, I. Gershkoff, D. Anderson, D. Mahidhara. 1993. Flight crew scheduling. *Management Sci.* **39** 736–745.

Hane, C., C. Barnhart, E. Johnson, R. Marsten, G. Nemhauser, G. Sigismondi. 1995. The fleet assignment problem: Solving a large-scale integer program. *Math. Programming* **70** 211–232.

Hoffman, K., M. Padberg. 1993. Solving airline crew scheduling problems by branch-and-cut. *Management Sci.* **39** 657–682.

Klabjan, D., K. Schwan. 1999. Airline crew pairing generation in parallel. Technical Report TLI/LEC-99, Georgia Institute of Technology, Atlanta, GA.

——, E. Johnson, G. Nemhauser. 1999. Solving large airline crew scheduling problems: Random pairing generation and strong branching. Technical Report TLI/LEC-99-11, Georgia Institute of Technology, Atlanta, GA.

——, ——, ——, E. Gelman, S. Ramaswamy. 1999. Airline crew scheduling with time windows and plane count constraints. Technical Report TLI/LEC-99-12, Georgia Institute of Technology, Atlanta, GA.

Lavoie, S., M. Minoux, E. Odier. 1988. A new approach for crew pairing problems by column generation with an application to air transportation. *Eur. J. Oper. Res.* **35** 45–58.

Levine, D. 1996. Application of a hybrid genetic algorithm to airline crew scheduling. *Comput. Oper. Res.* **23** 547–558.

Vance, P., C. Barnhart, E. Johnson, G. Nemhauser. 1997. Airline crew scheduling: A new formulation and decomposition algorithm. *Oper. Res.* **45** 188–200.