A Stochastic Programming Approach to the Airline Crew Scheduling Problem

Joyce W. Yen[1]
Industrial and Operations Engineering Department
University of Michigan
Ann Arbor, MI USA
jyen@engin.umich.edu

John R. Birge
Industrial Engineering and Management Sciences Department
Northwestern University
Evanston, IL USA
jrbirge@northwestern.edu

Abstract

Traditional methods model the billion-dollar airline crew scheduling problem as deterministic and do not explicitly include information on potential disruptions. Unfortunately, the effort used to optimize crew schedules and reduce crew costs is often wasted as flight schedules are often disrupted. Consequently, airlines spend a great deal of time and money optimizing crew schedules in the planning phase only to change and readjust the crew schedules in the operational phase in response to disruptions. Disruptions are expensive and lead to loss of time, money, and customer goodwill.

Instead of modeling the crew scheduling problem as deterministic, we consider a stochastic crew scheduling model and devise a solution methodology for integrating disruptions in the evaluation of crew schedules. The goal is to use that information to find robust solutions that better withstand disruptions. Such an approach is important because we can proactively consider the effects of certain scheduling decisions. By identifying more robust schedules, cascading delay effects will be minimized.

In this paper we describe our stochastic integer programming model for the airline crew scheduling problem and offer a branching algorithm to identify expensive flight connections and find alternative solutions. The branching algorithm uses the structure of the problem to branch simultaneously on multiple variables without invalidating the optimality of the algorithm. We present computational results demonstrating the effectiveness of our branching algorithm.

1 Introduction

To increase profits, airlines continually look for ways to better use their resources and to improve scheduling decisions. In 1991 American Airlines spent \$1.3 billion on crew costs [1] while in 1993 United Airlines spent \$0.6 billion just on pilot costs [8]. Combined crew costs involve billions of dollars of investment, second only to fuel costs among all airline costs, giving airlines incentives to efficiently use their crew resources

[1] This work was supported in part by the National Science Foundation under Grant DMI-9523275.

[8]. Airlines need to efficiently schedule the time crews work because schedule inefficiencies mean wasted time and money. When schedules become disrupted, the potential for even more scheduling inefficiencies increases.

How one chooses to model this problem greatly affects what solution techniques will be used. Traditional approaches to this problem model the long-term planning problem separate from models used to respond to disruptions in the short-term. (See [2], [5], and [7] for an overview of the long-range crew planning problem. See [3], [14], and [15] for examples of short-range planning techniques.) Such models may not be completely appropriate if one wishes to proactively capture the effects of disruptions on schedules. In reality long-range planning and short-range planning intersect and interact because short-range decisions inevitably alter long-range planning decisions. Hence, crew schedule assignments are actually based on short-range and long-range decisions. New models and new techniques must be developed to address this relationship between short-range and long-range crew scheduling decisions.

We propose a new model that addresses long-range scheduling issues while providing for short-range interaction. We call this problem the stochastic crew scheduling problem because stochastic disruptions (short range effects) are considered in the long range crew scheduling problem.

We integrate short-range and long-range planning by introducing randomness in the form of a short-term variable in the long-range problem. In doing so, we move away from the deterministic integer program to a stochastic integer program. We include the cost of disruptions in the scheduling formulation to gain a more realistic appreciation for the actual cost of a schedule. Furthermore, we propose a model that captures the interaction and interdependence between crew assignments. Such interactions are important because they can magnify disruption costs by facilitating cascading delay effects.

While Schaefer et al. [13] propose a stochastic extension to the deterministic crew scheduling problem, their methodology does not capture interaction effects of planes, crews, and schedule recovery. They modify the objective function's coefficient vector to reflect the expected cost of each decision variable rather than deterministic cost. They then solve the regular crew scheduling problem with this new coefficient vector. Their method does not account for disruption interactions between potential crew schedules.

In this paper we first define the deterministic crew scheduling problem, and then describe our stochastic integer programming model for the airline crew scheduling problem. This stochastic model considers interaction between crew schedules and the cascading effect delays can have on the system. Next we describe a branching algorithm to identify expensive flight connections and find alternative solutions. The branching algorithm uses the structure of the problem to branch simultaneously on multiple variables without invalidating the optimality of the algorithm. Finally we present computational results demonstrating the effectiveness of our branching algorithm.

2 Problem Definition

The traditional deterministic crew scheduling problem is modeled as a set partitioning problem. Let a *pairing* be defined as a round trip itinerary a crew member might fly. Let a *flight segment* be defined as a single flight from origin to destination. Then the deterministic crew scheduling problem can be formulated as follows:

$$
\begin{aligned}
&\textbf{minimize} && \boldsymbol{c^T x} \\
&\textbf{subject to} && \textstyle\sum_{j=1}^{n} \mathbf{e}_{\mathbf{ij}} \boldsymbol{x_j} = \mathbf{1} \qquad \textbf{for all } \boldsymbol{i} = \mathbf{1,...,}\boldsymbol{m}\textbf{;} \\
& && \boldsymbol{x_j} = \mathbf{0,1} \qquad \textbf{for all } \boldsymbol{i} = \mathbf{1,...,}\boldsymbol{m}\textbf{;}
\end{aligned}
$$

where

$\boldsymbol{x_j} = \mathbf{1}$ **if pairing j is selected in the solution, 0 otherwise;**

$\mathbf{e}_{\mathbf{ij}} = \mathbf{1}$ **if flight segment i is covered in pairing j , 0 otherwise;**

$\mathbf{c}_{\mathbf{j}} =$ **cost of pairing j.**

1. General crew scheduling formulation

This formulation may be relaxed to allow a flight to be serviced (covered) by more than one crew.

Since plane usage and crew assignment turn-around times are often short, long delays can have a cascading effect on future plane and crew assignments. As a result crew schedules which are cost effective in the deterministic case may no longer be so when random disruptions are considered. For example, in 1997 almost 55% of the delays on Air New Zealand's domestic flights, totally over 90,000 minutes of delay, were directly attributed to a flight crew delay or an upstream delay (a delay occurring earlier in the day) affecting downstream flights (flights occurring later in the day). Since the deterministic crew scheduling model fails to address this problem, we turn to stochastic modeling methods.

2.1 Stochastic Programming Formulation

We propose a method to adjust crew assignments during the planning phase so as to continue to minimize crew costs while creating more robust solutions. We strive to identify pairings that are less sensitive to schedule disturbances. In our formulation, we add delay costs to the deterministic objective function and consider how delays affect flight segments constraints.

We suppose that we will observe the disruptions and obtain a recovery cost. We model the disruption as increases in flight operation times, such as ground holds and actual flight times. These times form a random vector, $\xi(\omega)$, where ω is a random element in some space Ω. We call each ω a disruption scenario. It represents all disruptions that might happen to the schedule. For each disruption scenario ω we have a different recourse.

Therefore, we reformulate the problem as:

$$
\begin{aligned}
&\textbf{minimize} && \mathbf{c}^{\mathbf{T}} \boldsymbol{x} + \boldsymbol{Q}(\boldsymbol{x}) \\
&\textbf{subject to} && \boldsymbol{Ax} = \boldsymbol{b} \\
& && \boldsymbol{x} = \mathbf{0,1}
\end{aligned}
$$

2. General stochastic crew scheduling formulation

where the constraints **Ax = b** represent flight coverage equations from the deterministic formulation (see Figure 1) and where $Q(\mathbf{x}) = \int_{\omega} Q(\mathbf{x},\omega)P(d\omega)$ is the expected value of future actions due to disruptions in the original schedule. This formulation is a standard two-stage stochastic program with recourse [4].

2.2 Recourse Problem Formulation

To solve the recourse problem, we propose a nonlinear recourse model whose objective is to minimize the cost of delays. The recourse problem captures the dependent, interactive relationship among pairings. Any solution to the crew scheduling problem consists of a collection of pairings. When delays occur, one pairing's delays may affect flight departure and arrival times in another pairing. This effect is captured in our recourse model.

The recourse problem constraints represent time consistency requirements such as requiring a flight's departure time proceed its arrival time. The constraints also model the plane assignment precedence requirements and the crew assignment precedence requirements. Note that crew assignment precedence depends on the collection of pairings selected as a solution.

The recourse problem formulation is as follows:

$$\boldsymbol{Q(x,\omega) = \text{minimize} \quad \sum_j penalty \cdot delay(j,\omega)}$$

Subject to:

Consistency constraints for flight arrival and departure times
Consistency constraints for plane predecessor requirements
Consistency constraints for crew predecessor requirements

3. General recourse problem formulation

The crew predecessor constraints are nonlinear and make the entire formulation nonconvex. More generally, the recourse problem formulation can be given as:

$$\begin{aligned} \boldsymbol{Q(x,\omega) =} \quad & \textbf{minimize } \boldsymbol{q^T y} \\ \textbf{subject to} \quad & \boldsymbol{(W + Gx^T)y = h(\omega),} \\ & \boldsymbol{y \geq 0,} \end{aligned}$$

4. Two-stage nonlinear stochastic program

which demonstrates the nonconvex relationship between the first stage variable, x, and the second stage variable, y. (Details on the recourse problem can be found in [16].)

3 Delay Branching Algorithm

As we are examining how delays propagate through the network, we focus on the key idea that scheduling decisions only create additional delays when crews are assigned to switch planes. When crew assignments follow plane assignments, no additional delays are created in the system due to crew schedules. Our algorithm takes advantage of this structure.

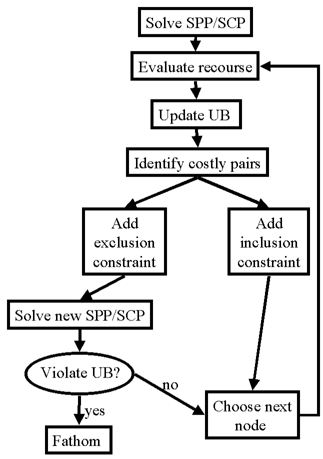
3.1 Background information for Algorithm

Our methodology finds an optimal solution to the no-delay problem and then examines key flight pairs where crews switch planes. This strategy allows us to consider how pairings interact. Because we want to create more robust solutions where crew plane changes are minimized, such plane switches are undesirable. Our algorithm forcibly allows or disallows such connections through a variation of constraint branching [12].

In order to create such branching constraints, we focus on crews switching planes. We define a specific type of delay, called the *switch delay*. A *switch delay* is a delay due to a plane change. Suppose a crew is assigned to service two flights, flight i and flight j. Suppose there is a delay to flight i. If flight i and flight j are assigned to different planes, then flight j would incur a *switch delay* because of this crew assignment and the delay to flight i.

3.1 Description of Algorithm

A visual representation of the algorithm is presented in Figure 5. First we solve the set partitioning problem (as outlined in Figure 1) using state-of-the-art deterministic crew scheduling knowledge. Note that our algorithm essentially augments existing crew scheduling methods by including the evaluation of a recourse problem.



5. Outline of Delay Branching Algorithm

We evaluate the expected cost of disruptions via the recourse problem. In order to determine which flight pair to consider next in the algorithm, we devise a hierarchy for the flight pairs based on the delay costs, placing those pairs with larger delays higher in the hierarchy.

Along one branch, we forcibly include the identified flight pair in a pairing selected in the optimal solution at that node. On the other branch, we forcibly exclude said flight pair from any pairing selected in the solution for that node.

At any point, upper and lower bounds can be used to select the next node from which to branch until the algorithm terminates. The user may choose, however, to prematurely terminate the algorithm if the current solution's disruption cost is satisfactorily limited. Furthermore, at any iteration of the algorithm we have a feasible solution which is more robust than the initial solution.

We have the following theorem:

Theorem: *The Delay Branching Algorithm terminates in a finite number of iterations with an optimal solution to the stochastic crew scheduling problem given by Figure 2 and recourse problem in Figure 3.*

(For details on the proof of convergence, see [16].)

With the exception of the SPP/SCP steps, all other steps illustrated in Figure 5 are relatively easy to evaluate. Using the solution to the set partitioning problem, the recourse problem reduces to a simple, albeit large, linear program. We can solve such linear programs efficiently. Identifying costly flight pairs is also simple. We need to evaluate each flight's *switch delay* value which is a simple addition/subtraction operation.

4 Computational Results

We implement our algorithm on small sample problems as well as larger problems based on airline data from Air New Zealand's 1997 domestic operations. The algorithm performs well and we can obtain significant cost savings. In this paper we report the results from the Air New Zealand test problems.

4.1 Implementation Assumptions and Preproccessing

We make the following simplifying assumptions when implementing the algorithm:

- Access to a state-of-the-art set partitioning solver that effectively and efficiently solves the deterministic problem. We use the solver described in [10].
- All crews fly their schedules as planned regardless of the delay circumstances. Legality considerations and other constraints make this assumption unrealistic. However, the solution obtained is a lowerbound for the actual expected cost of delays.
- Plane assignments are static. Flights are assigned to the same plane every day.
- All planes have same ground time requirements.
- All crew have same ground time requirements.
- Maximum delay times are bounded.

Now that we have established our assumptions, we preprocess the data. First we generate our set of possible crew pairings. Next we generate the random disruption scenarios ω. For each test problem we sample 100 scenarios using a truncated gamma or lognormal distribution for the length of delays. We assume that each scenario is equally likely.

4.2 Computational Results

We implement our algorithm on three test problems representing a simplification of the Air New Zealand domestic schedule: 9 planes covering 61 daily flights, 10 planes covering 66 flights, and 11 planes covering 79 flights. Each of these three problems represents a possible domestic schedule for Air New Zealand's 737 fleet. The schedule services seven cities (Auckland, Christchurch, Dunedin, Hamilton, Rotorua, Queenstown, and Wellington) with the same schedule being flown each day of the week.

Using the set parititioning solver described in [10], we enumerate 1911 feasible pairings for the 9-plane problem, 2737 pairings for the 10-plane problem, and 3296 for the 11-plane problem. Our implementation of the Delay Branching Algorithm uses AMPL [6] to formulate the model and CPLEX [9] to solve the resulting linear program. A master C program, which uses a general purpose priority queue to maintain the branching decisions, is used to manage the branching process [11].

The branching tree becomes very large and, due to computational limitations, we do not explore the entire tree. Nevertheless, our results indicate that we quickly find good solutions and that subsequent improvements may be quite minimal.

We test four varations of each test problem examining how different penalty parameter values affect the solution. We also calculate the value of the stochastic solution (VSS) (see [4] for a definition). We obtain large savings in the recourse objective without significant sacrifice in the cost of the first stage objective.

4.2.1 General Results

When the penalty parameter equals one, delay costs are valued at face-value. The recourse cost is exactly the sum of the delay minutes over all flights averaged over 100 scenarios. Note that the first stage objective is order 10^3 while the recourse objective is order 10^2. These orders of magnitude are consistent with the 1997 data. The simulated total delay time minutes are the same order of magnitude as the average delay times from the 1997 data. In 1997 the average number of delay minutes per day was 392.65 minutes.

The four variations per test problem set the penalty parameter at 1, 10, 100, and 1000. See Figure 6 for results from the 9-plane problem. Similar results hold for the 10-plane and 11-plane problems. The first column lists the penalty parameter value. The second and third columns list the initial solution's value for the first and second stage, respectively. Note this initial solution, $\mathbf{x^0}$, is the solution to the deterministic problem from Figure 1. The fourth and fifth column have the best solution's first stage and second stage values, respectively. The sixth column gives the value of the stochastic solution, while the seventh column gives the best solution's objective value relative to the initial solution objective. For example, in Figure 6 row 3, the current optimal solution costs 2332.81 units less than the initial solution and is equal to 90.582% of the value of the initial solution.

Penalty	$\mathbf{c^T x^0}$	$\boldsymbol{Q}(\mathbf{x^0})$	$\mathbf{c^T x^*}$	$\boldsymbol{Q}(\mathbf{x^*})$	VSS	% VSS
1	3889	208.803	3889	208.803	0	100
10	3889	2088.035	3937	1957.513	82.5218	98.619
100	3889	20880.35	4297	18139.53	2332.81	90.582
1000	3889	208803.5	4819	179628	28245.5	86.72

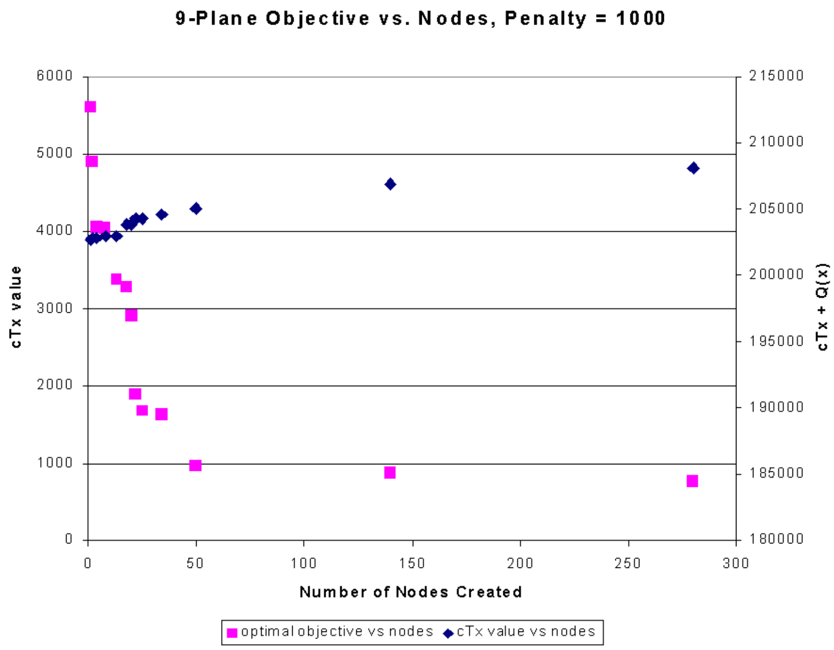
6. 9-plane problem results

While we do not explore the entire branching tree, we do explore enough of the tree such that we believe our solutions are near optimal or optimal since the curve of the solution values flattens (see Figure 8). However, we do not have proof of optimality. Figure 7 gives a summary of the number of iterations executed until finding the best solutions listed in Figure 6 as compared to the total number of iterations calculated for each problem.

Penalty	No. of iterations at optimal	Total number of iterations
1	1	1287
10	7	1127
100	25	995
1000	140	386

7. 9-plane Problem: Number of iterations before finding x*

Even though we may execute many iterations before finding the best solution, we find that we are very close to the best solution after just a few iterations. For example, consider the 9-plane problem with penalty of 1000. We are within five percent of the best solution after only eleven iterations and within one percent after 25 iterations. We get quick savings early in the implementation of the algorithm (see Figure 8). In this case, we see small changes to $\mathbf{c}^T\mathbf{x}$ while simultaneously seeing relatively large changes to the overall objective. Similar results hold for the 10-plane and 11-plane problems.

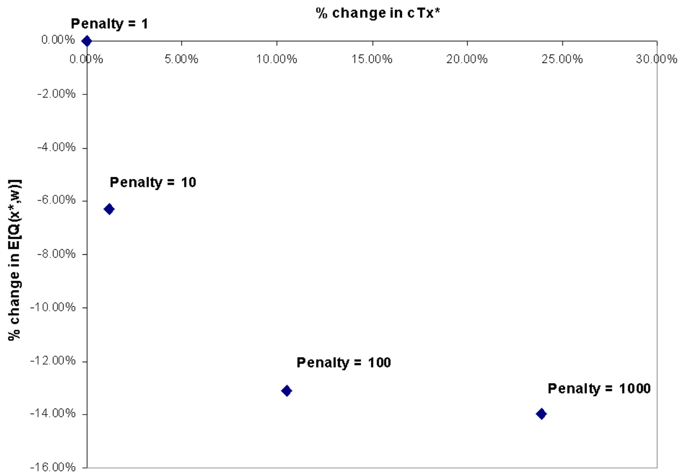


8. 9-plane problem optimal objective function values

4.2.2 Penalty Parameter Analysis

Finally we comment on the effects of the penalty parameter value. The penalty parameter is a reflection of how much a disruption is worth. Increasing the penalty parameter increases the cost of disruptions. The total objective function can become dominated by the recourse objective giving priority to disruption costs. The penalty parameter represents a trade-off between disruption costs and crew costs. As we increase the penalty parameter, we can track this trade-off (see Figure 9.)

Management or other crew scheduling decision makers can use this trade-off curve to assess the value of disruptions with respect to crew costs. Furthermore, the model allows flexibility through the penalty parameter. Decision makers can use the penalty parameter to evaluate different scenarios and find alternative solutions. They might also use it as a way to include “scheduling expertise” or “company policies” in the scheduling process.



9. Penalty parameter trade-off curve

5 Conclusion

We propose a new model for the crew scheduling problem. We introduce randomness into the model taking into account the expected cost of delays. We demonstrate the value of a stochastic formulation which addresses the interaction between pairings. Our Delay Branching Algorithm branches on a collection of variables allowing or disallowing certain flight connections and quickly finds a good solution. Furthermore, it allows flexibility in the anlysis of the trade-off between crew costs and disruption costs.

Significant saving can be gained if delay effects on crew schedules, which consequently effects the entire system, are considered during the planning phase. In this paper we demonstrate the effectiveness of our method. In particular, we show we obtain significant gains early on, quickly reducing the expected cost of disruption while minimizing additional crew costs.

Further saving might be obtained by applying these ideas to the pairing generation process. By directing pairing generation to exclude costly flight connections, we hypothesize we would create a better set of pairings from which to identify a solution.

In summary crew scheduling can benefit from an analysis which includes random elements and which consider pairings collectively rather than just individually.

References

[1] R. Anbil, E. Gelman, B. Patty, and R. Tanga. *Recent Advances in Crew-pairing Optimization at American Airlines*. Interfaces, 21(1) (January – February 1991), pp 62-74.

[2] J. Arabeyre, J. Fearnley, F. C. Steiger, and W. Teather. *The Airline Crew Scheduling Problem: A survey*. Transportation Science, 3(2) (May 1969), pp 140-163.

[3] M. F. Argüello, J.F. Bard, and G. Yu. *Models and Methods for Managing Airline Irregular Operations* in Operations Research in the Airline Industry. Kluwer Academic Publishers, (1998), pp 1 - 45.

[4] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, (1997).

[5] M. M. Etschmaier and D. F. X. Mathaisel. *Airline Scheduling: An Overview*. Transportation Science, 19(2) (May 1985), pp 127-138.

[6] R. Fourer, D. M. Gay, and B. W. Dernighan. *AMPL: A Modeling Language for Mathematical Programming*. The Scientific Press (1993).

[7] I. Gershkoff. *Optimizing Flight Crew Schedules*. Interfaces, 19(4)(July – August 1989), pp 29-43.

[8] G. W. Graves, R. D. McBride, I. Gershkoff, D. Anderson, and D. Mahidhara. *Flight Crew Scheduling*. Management Science, 39(9) (June 1993), pp 736-745.

[9] ILOG, Inc. *CPLEX v. 6.0*, (1998).

[10] E. Johnson, T. Shaw, and R. Ho. *Modeling Tools for Airline Crew-scheduling and Fleet-assignment Problems* in Operational Research in Industry, MacMillan Press, (1999).

[11] T. P. Kelly. *Heap-based Priority Queue Implementation in ANSI C*.URL: http://www-personal.engin.umich.edu/tpkelly/heap/heap.tar.gz (18 Febuary 2000).

[12] D. Ryan and B. Foster. *An Integer Programming Approach to Scheduling* in Computer Scheduling of Public Transport, North-Holland Publishing Company (1981), pp 269-280.

[13] A. Schaefer, E. Johnson, A. Klegwegt, and G. Nemhauser. *Robust Airline Crew Scheduling*. Institute for Operations Research and Management Science Conference, Salt Lake City, Utah, (Spring 2000).

[14] M. Stojkovic, F. Soumis, and J. Desrosiers. *The Operational Airline Crew Scheduling Problem*. Transportation Science, 32(3) (August 1998), pp 232-245.

[15] G. Wei, G. Yu, and M. Song. *Optimization Model and Algorithm for Crew Management During Airline Irregular Operations*. Journal of Combinatorial Optimization, 1(1997), pp 305-321.

[16] J. W. Yen, *A Stochastic Programming Formulation of the Stochastic Crew Scheduling Problem*. PhD thesis, University of Michigan, Ann Arbor, MI 48109, (December 2000).