Airlines’ Crew Pairing Optimization: A Brief Review

Xugang Ye
Applied Mathematics and Statistics, the Johns Hopkins University

Abstract

In most airlines, crew costs are the second largest direct operation cost next to the fuel costs. But unlike the fuel costs, a considerable portion of the crew costs can be saved through optimized utilization of the internal resources of an airline company. And the saving is largely realized through solving the crew pairing problem. This paper aims to give a brief review of the fundamental crew pairing problem: its concepts, its models, and its algorithms.

Keywords: Airlines; Crew pairing; Optimization

1. Introduction

As an important issue of airlines’ operational planning, crew pairing immediately follows fleet assignment phase and right precedes crew rostering phase. A pairing is a sequence of connectable flight legs, within the same fleet, that starts from and ends at the same crew base, where the crew actually lives. A pairing is sometimes called an itinerary for the crew assigned to this journey. It typically spans from one to five days. To create tasks for crew members, the planners of an airline must generate a set of pairings to cover as many flight legs in a planning horizon as possible, with a cost as small as possible. For a large airline, the number of the flight legs in a planning horizon can be very many, hence the planning of pairings can be highly computationally intensive. Although it is possible to model the crew pairing problem as a binary linear programming (BLP) problem if we consider all the subsets of the set of the flight legs in a planning horizon, the number of the decision variables could easily exceed the capability of the current computing technology. Therefore, people rely on more practical methods. The most popular method seems to solve the problem sequentially with two phases. The first phase is to generate as many legal pairings as possible. The second phase is to select a best subset in the sense of least cost.

The reason the sequential method could reduce the computational complexity is that although there can be 2^n subsets of the set of n flight legs in a planning horizon, the number of legal pairings is not such many because the formation of a legal pairing must satisfy many constraints. Those constraints not only follows the pairing’s definition, but also include union rules, government regulations, and contractual agreements. In [1], Andersson *et al*. listed several main characteristics of the crew pairing problem. Those characteristics systematically highlight the structure of crew pairing in both North American airlines and European airlines. Also according to [1], a crew pairing problem can be daily, weekly, or fully dated. The most common version is the daily problem. The problem is formulated upon an assumption that the flight schedule is the same every day. Actually, most published research focuses on this version (see [2]). In [3], Anbil *et al*. provided a pairing example in American Airlines. The pairing is built with the daily assumption.

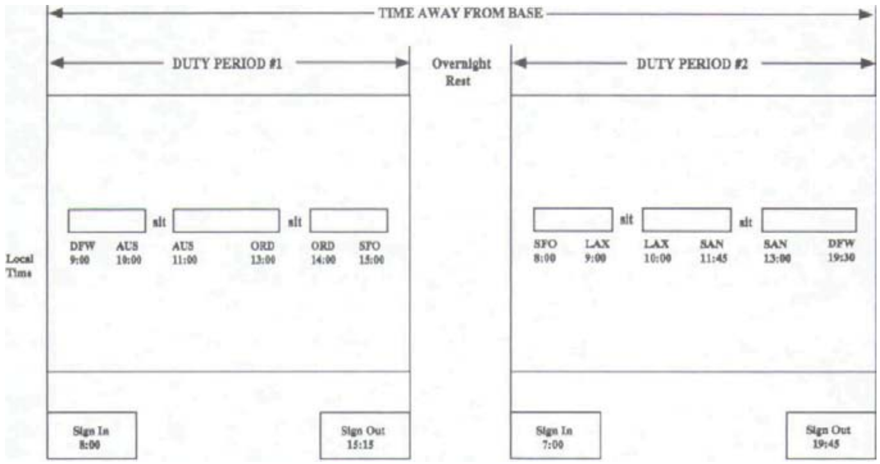


Figure 1.1. A typical pairing with such attributes as duty period, sits within duty periods, overnight rests, and sign-in/sign-out times

As Figure 1.1 shows, a legal pairing is an alternating sequence of events that are mainly divided into duties and overnight rests. Each duty period represents a daily task segment that can be subdivided into flight leg segments separated by sit connections. Hence duties must be formed before pairings. A duty can become a pairing if it starts from and ends at the same city. In United States, the length of a duty is largely determined by Federal Aviation Regulations (FAR). However, the length can vary from airline to airline. The Federal law only requires that a pilot cannot fly more than eight hours within a 24-hour period, and he/she must also be able to rest for eight hours in the same time span. A sit connection during a duty period mainly consists of the waiting time of the crew for changing planes on to the next flight leg. Finally, an overnight rest (or layover) is a rest period between two consecutive duties. If the overnight rest happens away from the home base of the crew, the airline must pay for their hotel expenses and additional compensations.

Crew pairing problem usually has two objectives. One is to cover as many flight legs in a planning horizon as possible. The other is to minimize the total cost associated with the constructed pairings. Note that a more general statement of the first objective is to cover as many important flight legs in a planning horizon as possible. The importance can be quantified with profits, or market values. The rest of this paper is organized as follows. We first introduce several formulations of the crew pairing problem; we then give a brief review of the optimization methods for the crew pairing problem; finally we present a concrete modeling example in American Airlines.

2. Models

Currently, all published methods separate the problem of generating legal pairings from the main problem (see [1, 2, 3, 4, 5, 6]). As we have mentioned earlier, one advantage of doing so is to reduce the number of decision variables (hence the computational complexity of the main problem). Therefore, our review of the models is based the availability of a set of legal pairings and their explicit costs. Also, by the definition of pairing, the crew pairing problem is decomposed by fleet type. Hence we only focus on the flight legs that are carried by one fleet type. No mater what a crew pairing model is, the crew capacity of each fleet at each home base must be respected.

If the crew pairing problem is treated as a multiobjective programming problem, then a bi-objective binary linear programming model can be proposed (by author of this paper) as a revision of the one in [7]. It is stated as follows.

Sets

F = Set of flights

P = Set of legal pairings

K = Set of crew home base cities

Indices

i = Flight index

j = Pairing index

k = Crew home base index

Parameters

v_i = Value of flight i

c_j = Cost of pairing j

a_{ij} = indicator that whether flight i is covered by pairing j ($a_{ij} = 1 \Leftrightarrow$ "yes"; $a_{ij} = 1 \Leftrightarrow$ "no")

h_{kj} = indicator that whether home base city for pairing j is city k ($h_{kj} = 1 \Leftrightarrow$ "yes"; $h_{kj} = 1 \Leftrightarrow$ "no")

b_k = crew capacity at home base city k

Decision variables

$$x_j = \begin{cases} 1 \text{ if pairing } j \text{ is selected} \\ 0 \text{ otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 \text{ if flight } i \text{ is covered} \\ 0 \text{ otherwise} \end{cases}$$

Model

$$\text{Max} \quad \sum_{i \in F} v_i \cdot y_i \tag{2.1}$$

$$\text{Min} \quad \sum_{j \in P} c_j \cdot x_j \tag{2.2}$$

Subject to

$$y_i \leq \sum_{j \in P} a_{ij} \cdot x_j \leq y_i \cdot |P| \text{ for all } i \in F \tag{2.3}$$

$$\sum_{j \in P} h_{kj} \cdot x_j \leq b_k \text{ for all } k \in K \tag{2.4}$$

In model (2.1-2.4), objective function (2.1) is to maximize the total value of the selected pairings. Objective function (2.2) is to minimize the total cost of the selected pairings. Constraint set (2.3) is to ensure that for any $i \in F$, when flight i is covered, $y_i = 1$; otherwise, $y_i = 0$. Constraint set (2.4) is to respect the crew capacity at each home base city.

Multiobjective integer programming is a relatively new research area. Essentially, currently known methods can be reduced to those that are designed to solve single objective integer programming problems. One typical method is to change the multiobjective programming model into its constrained version. For example, we can predetermine a threshold value, say $v > 0$, for objective (2.1) such that the following constraint

$$\sum_{i \in F} v_i \cdot y_i \geq v \tag{2.4}$$

is brought in to replace the objective (2.1). since $\sum_{i \in F} v_i \cdot y_i \leq \sum_{i \in F} v_i$, we have that if $v = \sum_{i \in F} v_i$ and the constrained version (2.2-2.5) still has feasible solution, then it is possible to cover all the flight legs. In fact, under this circumstance, constraint set (2.3) becomes

$$\sum_{j \in P} a_{ij} \cdot x_j \geq 1 \text{ for all } i \in F, \tag{2.5}$$

which are exactly set covering constraints. This constraint set functions as same as another set of set covering constraints in [1]. Note that set covering constraints allow a flight to be overcovered, when that happens, a team of crew is transported as passengers. One advantage of such operation is that the crew resources could be utilized more flexibly, but transporting crew only incurs costs without adding flying credits to the crew, hence this operation is unpopular with the crew itself. In general, most airlines prefer partitioning type of solutions. That is to replace Constraint (2.5) with

$$\sum_{j \in P} a_{ij} \cdot x_j = 1 \text{ for all } i \in F, \tag{2.6}$$

Hence, as long as the set covering constraints can be satisfied, the set partitioning constraints are preferred.

In [1, 8], some consideration is also given to the workload of the crew in each home base city. Both Andersson *et al*. and Medard *et al*. proposed a constraint set that involves the limit on the required workload for each airline base. To mathematically express these constraints, additional parameters are defined as follows:

b_{kj} = Workload of pairing j with home base city k,

b_{upper_k} = Upper limit on the required workload for home base city k,

b_{lower_k} = Lower limit on the required workload for home base city k.

In practice, workload limit is a more popular measure of the crew capacity of a home base city because the workloads of pairings vary from one to another. According to [1, 8], the set of crew capacity constraints can be reformulated, to replace Constraint set (2.4), as

$$\sum_{j \in P} b_{kj} \cdot x_j \leq b_{upper\_k} \text{ for all } k \in K. \tag{2.7}$$

Note that Constraint set (2.7) only deals with upper limits, while the crew pairing model in [7] also deals with lower limits. However, the model in [7] considers the minimum number of crew members to be used at each home base city. More practically, the constraint set that involves the lower limit on the crew capacity of each home base city can be stated as

$$\sum_{j \in P} b_{kj} \cdot x_j \geq b_{lower\_k} \text{ for all } k \in K. \tag{2.8}$$

To introduce lower limits is based on the consideration of balancing the workloads among the home bases. However, to determine proper lower limits is very technical and tedious.

To conclude this section, we remark that the solution to the crew pairing problem is subject to flight schedules, the estimate of pairing costs, and the crew capacities. Most of time, all flight legs can be covered, hence the crew pairing model becomes set partition problem with crew capacity constraints. But what if the requirement for total covering yields no feasible solution? Hence, it usually needs an iterative procedure and feedback system to find a proper pairing solution. That is, the solution is sent to planners of fleet assignment, aircraft routing, and roster building for evaluation. Adjustments in all planning phases might be needed until a mutually agreement is reached. From this perspective, we can see that to be able to find a pairing solution quickly is not only preferred but also required.

3. Algorithms

3.1 Methods for Pairing Generation

Without a set of candidate legal pairings, all the modeling techniques mentioned in section 2 make no sense. Hence, a method for generating legal pairings is crucial if the crew pairing problem is to be well solved. As we mentioned earlier, there can be 2^n subsets of the set of n flight legs in a planning horizon. Hence total generation of all legal pairings is only possible for small problems. For large problems, generation must be restricted to a limited part of the problem, or to a partial selection of all possible legal pairings.

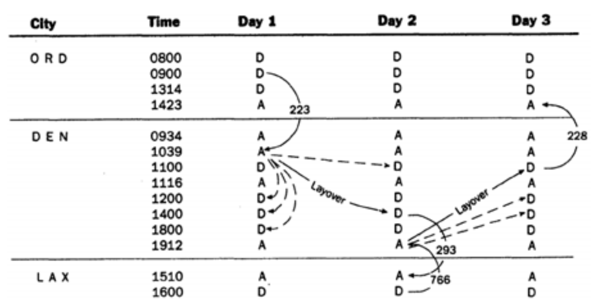
If an old solution to the crew pairing problem is available, then pairing generation can work in the first scenario to improve the solution by generating all possible legal pairings based on all the flight legs that appear in the old pairing solution. It's easy to observe that a new pairing solution based on selecting from the new set of legal pairings is at least no worse than the old one. Also, this new solution, which contains a set of "promising" legal pairings, can be maintained for later use. The logic is as simple as utilizing previous efforts to reduce repeated search. In next round of pairing generation, each old pairing is checked for feasibility under the current setting. Those that are still feasible will join the newly generated legal pairings to form the current pool of candidates. Example of system using this approach includes the TRIP system (see [3]), which had ever been used American Airlines and Continental Airlines, the TPACS system (see [9]) , which had been used by United Airlines, and ALLPS system (see [9]), that had been used by Northwest Airlines and USAir. Although such approach is favored by its efficiency, it is pretty much a local method since it only focuses on a small number of flight legs. Hence this approach is most often used for short planning horizon and small geographical area.

In more broader decision environment, pairing generation in the second scenario is favored. Note that in the first scenario, it is possible to select an optimal subset from the pool of candidate pairings since the size of the pool is within the capability of current integer programming algorithm and computer technology. In the second scenario, however, the number of all possible legal pairings can be astronomical since there can be very many flight legs. A well-established theoretical result is that the selection problem, which is formulated as an integer programming either of set covering type, or of set partition type is NP-complete (see [10]). Hence to reduce the number of legal pairings in the candidate pool is a must. Current state-of-art pairing generators are equipped with some heuristic rules to identify promising pairings. Those rule includes but not restrict to limiting the number of times a flight leg is used in pairings, limiting the connection time, limiting the number of duty periods etc. Currently known systems that are built on pairing generation in the second scenario include CPOS system (see [11]), which is jointly developed by IBM and Sabre group, Carmen Crew Pairing system (see [12]), which is developed by Jeppesen Company, and RALPH system (see [13]), which is another Carmen system. Some documented reports of implementation include CPOS in American Airlines and US air, the Carmen Crew Pairing system in Northwest Airlines and Delta Airlines, and RALPH system in Delta Airlines etc.

A central procedure in pairing generation is path enumeration. A pairing generator can use forward, backward, or interior enumeration. Which version to use is determined by the selection of the so-called seed legs. If a seed leg with departure at some home base city is selected, then a forward enumeration is initiated. If a seed leg with arrival at some home base city is selected, then a backward enumeration is initiated. If a seed leg with both departure and arrival away from any home base city is selected, then an interior enumeration is initiated. An enumeration process is a tree growing process, the root of the tree is the seed flight leg. The tree grows unidirectinally or bidirectionaly, subject to the pairing generation rules. Once a directed cycle that passes the root is formed, then a legal pairing is generated. A pairing generator can starts from many seed legs, deterministically or randomly. Pairing generation can also proceed parallely in parallel computing environment. No matter what a pairing generation algorithm is, book keeping and duplicate detection is required. To conclude this section, we list an illustrating example (shown in Figure 3.1.1) provided in [9].

		City		Time	
	Flight No.	DEP	ARR	DEP	ARR
1	221	ORD	DEN	0800	0934
2	223	ORD	DEN	0900	1039
3	228	DEN	ORD	1100	1423
4	230	DEN	ORD	1200	1521
5	238	DEN	ORD	1800	2121
6	766	LAX	DEN	1600	1912
7	105	ORD	LAX	1100	1314
8	259	ORD	LAX	1400	1609
9	274	LAX	DEN	0800	1116
10	293	DEN	LAX	1400	1510
11	412	LAX	ORD	1400	1959

(a) A partial view of flight table



(b) A legal pairing that spans 3 days

Figure 3.1.1. A example that a legal pairing (in (b)) is generated from a given flight table (in (a))

As Figure 3.1.1 shows, a legal pairing consists of a flight leg (# 223) that departs at 9:00am from DRD and arrives at 10:39 am in DEN, a layover between day 1 and day 2, a flight leg (# 293) that departs at 14:00pm from DEN and arrives at 15:10pm in LAX, a flight leg (# 766) that departs at 16:00pm from LAX and arrives at 19:12pm in DEN, a layover between day 2 and day 3, and a flight leg (# 228) that departs at 11:00am from DEN and arrives at 14:23pm in ORD.

3.2 Methods for Pairing Selection

The methods for the pairing selection problem are roughly divided into two classes: exact algorithms and heuristic algorithms. Due to the many applications of the set covering and set partitioning problems a large number of papers have treated these problems and many algorithms have been proposed. We only focus on the main ideas.

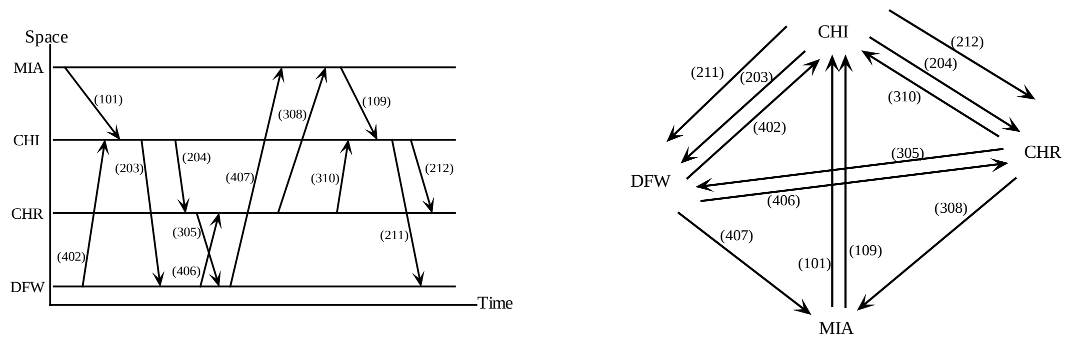
Most exact algorithms for the set covering or set partitioning problem are based on branch-and-Bound procedures. Lower bounds are usually calculated by solving the continuous relaxation version. But Lagrangian relaxation may also be used. Upper bounds are usually calculated by using heuristics. Considerably many computational experiments show that the gap between the optimal objective value of the continuous relaxation version and the optimal objective value of the integer programming is very small. Small problems often have integer solutions to the associated relaxation versions, but there may have a gap of up to a few percent. Large problems rarely have integer solutions to the corresponding relaxation versions, but the gap is always very small. Of the 11 problems with more than 100 constraints considered in [14] the largest gap was 0.7%. In a study of over 100 large problems from several European airlines, the gap was almost always less than 0.5% and for the largest problems (more than one million non-zeros in the coefficient matrix) the typical gap was 0.1% (see [15]). Branching on single variables is generally very inefficient. In [16], some success in using the "follow-on" branching strategy proposed in [17] was reported.

Very recently, Anbil *et al*. reported some advances in crew pairing with CPOS system (see [18]). CPOS uses a combined linear programming and integer programming approach (LP/IP) based on a set partitioning formulation where the rows are flights and the columns are pairings. The problem is to select the least cost partition from a pool of candidate pairings. CPOS first solves the LP using a method called Sprint pioneered at IBM and based on core solvers in OSL. It then employs a branching rule based on how flights connect in the LP solution in order to obtain the integer solution. A key IBM innovation concerns the generation of a very good starting pool of candidate pairings for the LP/IP approach. Here a column generator produces pairings based on approximate LP duals from a new IBM subgradient method called the Volume algorithm. Computational experiences show that the Volume duals are much better seeds for column generation than LP duals, and are also much easier to compute. Additionally, the final Volume duals provide an advanced starting basis for the LP/IP approach.

Many heuristics have been proposed. One possibility is to use a Branch-and-Bound based method but to stop before proven optimality. The simplest implementation of this idea is to fix the largest fractional x_j value from the LP solution to 1. This eliminates the constraints covered by this variable and all other variables covering these constraints. The procedure is applied until the remaining problem is of manageable size. In [16], reasonable results for small problems were reported. The approach is essentially a partial evaluation of the branching tree and it works with any branching strategy. The difficulty of the LP can also be avoided by attacking the problem from the dual side. A natural way to do this is by Lagrangian relaxation, which works by moving hard constraints into the objective function and punishing the objective if the constraints are not satisfied. Lagrangian relaxation techniques nowadays form an important family of methods in combinatorial optimization and have been proven to be effective both practically and theoretically. In [19], Fisher provided a systematic analysis of the Lagrangian relaxation techniques. The analysis forms the basis of many subsequent researches. For the crew pairing problem, the so-called Lagrange heuristics (see [10]) are also applied quite often. One report can be found in [20]. We conclude this section by remarking that today, with an advanced crew pairing system like CPOS or Carmen, pairing optimization from the largest airlines in the world is not a challenge any more. Clients of Carmen system reported (see [21]) in 2005 that they could solve up to 25,000 dated flights, directly from scratch. Daily solutions with 1,500 flights are not a problem.

4. A Case Study

In this section, we go through the crew pairing process by a simple example provided by American Airlines. In this example, a partial flight table is translated into a time space network shown as Figure 4.1(a). A view via directed graph is shown as Figure 4.1 (b). The connection relation is shown in Table 4.2. It's easy to enumerate all possible legal pairings from this small set of flight legs. Figure 4.3 shows how 4 legal pairings are generated. With the search method illustrated through Figure 4.3, we can generate all the legal pairings. They are listed in Table 4.4. Each pairing is associated with a cost calculated by American Airlines.



(a) A partial view of time-space flight network (b) A partial view of directed graph

Figure 4.1. A collection of flight legs for crew pairing

Table 4.2. Feasible flight connections

Flight	Connectable flights
101	203, 204, 211, 212
109	211, 212
203	406, 407
204	305, 308, 310
211	
212	
305	407
308	109
310	211, 212
402	203, 204, 211, 212
406	308, 310
407	109

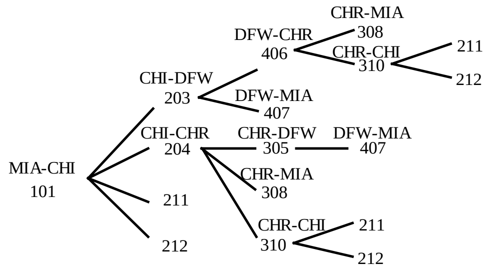


Figure 4.3. Four legal pairings generated by the seed leg 101

Table 4.4. Possible pairings and associated costs

Index	Decision variables	Pairing	Cost (in hundred dollars)
1	x_1	101-203-406-308	2900
2	x_2	101-203-407	2700
3	x_3	101-204-305-407	2600
4	x_4	101-204-308	3000
5	x_5	203-406-310	2600
6	x_6	203-407-109	3150
7	x_7	204-305-407-109	2550
8	x_8	204-308-109	2500
9	x_9	305-407-109-212	2600
10	x_{10}	308-109-212	2050
11	x_{11}	402-204-305	2400
12	x_{12}	402-204-310-211	3600
13	x_{13}	406-308-109-211	2550
14	x_{14}	406-310-211	2650
15	x_{15}	407-109-211	2350

Now, given the data, we are ready to set up a pairing selection model. Since there is no workload information, we first consider whether a partitioning solution can be found. If the answer is “yes”, then each flight leg is covered by exactly one pairing. We aim to find such a solution with minimum cost. The model is given below:

Min $2900x_1 + 2700x_2 + 2600x_3 + 3000x_4 + 2600x_5 + 3150x_6 + 2550x_7 + 2500x_8 + 2600x_9 +$
$2050x_{10} + 2400x_{11} + 3600x_{12} + 2550x_{13} + 2650x_{14} + 2350x_{15}$

Subject to

$x_1 + x_2 + x_3 + x_4 = 1$ (cover flight 101)

$x_6 + x_7 + x_8 + x_9 + x_{10}+ x_{13} + x_{15} = 1$ (cover flight 109)

$x_1 + x_2 + x_5 + x_6 = 1$ (cover flight 203)

$x_3 + x_4 + x_7 + x_8 + x_{11} + x_{12} = 1$ (cover flight 204)

$x_{12} + x_{13} + x_{14} + x_{15} = 1$ (cover flight 211)
$x_9 + x_{10} = 1$ (cover flight 212)
$x_3 + x_7 + x_9 + x_{11} = 1$ (cover flight 305)
$x_1 + x_4 + x_8 + x_{10} + x_{13} = 1$ (cover flight 308)
$x_5 + x_{12} + x_{14} = 1$ (cover flight 310)
$x_{11} + x_{12} = 1$ (cover flight 402)
$x_1 + x_5 + x_{13} + x_{14} = 1$ (cover flight 406)
$x_2 + x_3 + x_6 + x_7 + x_9 + x_{15} = 1$ (cover flight 407)
$x_1, x_2, \cdots, x_{15} \in \{0, 1\}$

An optimal solution given by Lindo is $x_1^* = 1$, $x_9^* = 1$, $x_{12}^* = 1$, and $x_j^* = 0$ for $j = 2, 3, 4, 5, 6, 7, 8, 10, 11, 13, 14, 15$. Hence the selected pairings are

Pairing 1: 101-203-406-308 with cost \$290000 and home base MIA
Pairing 9: 305-407-109-212 with cost \$260000 and home base CHR
Pairing 12: 402-204-310-211 with cost \$360000 and home base DFW

With a total cost \$910000. We can see that as long there is one crew team in each of the three home bases: MIA, CHR, and DFW, then this solution is executable.

5. Future Discussions

So far, we have reviewed the most fundamental versions of crew pairing problem. The case study provided a concrete example for understanding how a crew pairing problem might be solved. However, there is much more to say about crew pairing problem.

Airline operations are also facing never-ending irregular situations or disruptions. A variety of events such as sever weather occurrences, aircraft mechanical failures, air traffic control problems and crew unavailability etc will affect the execution of the published flight schedules. The impact of the disruptions is significant. The cost arises from every stage of the airline resource planning and scheduling: work needs to be done to adjust timetables, aircrafts need to be redeployed, pairings need to be reconstructed, and rosters need to be updated accordingly. The perturbations come continuously and pass through the entire airline operation system very quickly. Statistics had shown that most airlines have delay rates around 20% and cancellation rates around 3%, the cost implication is in millions or tens of millions dollars annually, depending on the size of the airline (see [22]).

Recovery is the process reacting to a disruption: it returns an airline to its original schedule, at the minimum cost, as soon as possible. Optimal recovery decisions are hard to determine since the disruptions come stochastically in real-time fashion and have extensive influences. In old practices, most airlines make recovery decisions manually with little optimization-based decision support (see [23]). However, considerable researches have been done. Some representative work on crew pairing repair can be found in [23, 24, 25, 26]. In later review, we will discuss the problem of crew pairing under uncertainty.

References

[1] E. Andersson, E. Housos, N. Kohl, and D. Wedelin. Crew Pairing Optimizatio. In: OR in Airline Industry – edited by G. Yu, Kluwer Academic Publishers, Boston, 1997.

[2] G. Yu and B. Thengvall. Optimization in the Airline Industry, Handbook of Applied Optimization – edited by P. M. Pardalos and M. G. C. Resende, Oxford University Press, New York.

[3] R. Anbil, E. Gelman, B. Patty, and R. Tanga. Recent advances in crew-pairing optimization at American airlines. Interfaces, 21(1), 62-74. 1991.

[4] C. Barnhart, A. M. Cohn, E. L. Johnson, D. Klabjan, and G. L. Nemhauser, P. H. Vance. Airline Crew Scheduling. In: Handbook of Transportation – edited by R. W. Hall, Kluwer Scientific Publishers, 2003.

[5] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Column Generation For Solving Huge Integer Programs. Operations Research, 46, 316-329, 1998.

[6] G. Desaulniers, J. Desrosiers, Y. Dumas, S. Marc, B. Rioux, M. M. Solomon, and F. Soumis. Crew Pairing at Air France. European Journal of Operational Research, 97, 245-259, Elsevier Science, 1997.

[7] M. Bazargan. Airline Operations And Scheduling. Ashgate Publishing, Ltd., 2004.

[8] C. P. Medard, N. Sawhney. Airline Crew Scheduling from Planning to Operations. European Journal of Operational Research, 183, 1013–1027, 2007.

[9] G. W. Graves, R. D. McBride, I. Gershkoff, D. Anderson, and D. Mahidhara. Flight Crew Scheduling. Management Science, 39(6), 736-735, 1993.

[10] G. L. Nemhauser and L. A. Wolsey. Integer and Combinatorial Optimization. Wiley Interscience, 1988.

[11] http://www-03.ibm.com/industries/travel/doc/content/resource/thought/187977106.html#Section1

[12] http://www.carmen.se/air_products/prod_folders/crew_pairing/index.htm

[13] R. Marsten. Crew Scheduling with RALPH. Presentation given at the workshop "Optimization in Production and Transportation", The Hague, The Netherlands, Nov. 9-11, 1994.

[14] K. L. Hoffman and M.Padberg. Solving Airline Crew Scheduling Problems by Branch-and Cut. Management Science, 39(6), 657-682, 1993.

[15] F. Engel. Summary over Test Runs. Internal Report, Carmen Systems AB, Gothenburg, Sweden.

[16] R. Anbil, R. Tanga, and E. L. Johnson. A Global Approach to Crew Pairing Optimization. IBM system Journal, 31(1), 71-78, 1992.

[17] D. M. Ryan and J.C. Falkner. A Bus Crew Scheduling System Using a Set Partitioning Model. Asia Pacific Journal of Operational Research, 4, 39-56, 1987.

[18] R. Anbil, F. Barahona, L. Ladanyi, R. Rushmeier and J. Snowdon. IBM Makes Advances in Airline Automation. IBM Research Report, IBM T. J. Watson Research Center, 2003.

[19] M. L. Fisher. The Lagrangian Relaxation Method for Solving Integer Programming Problems. Management Science, 27(1), 1-18, 1981.

[20] D. Wedelin. An algorithm for Large Scale 0-1 Integer Programming with Application to Airline Crew Scheduling. Annals of Operations Research, 57, 283-301, 1995.

[21] Carmen Crew Pairing: Questions and Answers. Production Report, Jeppesen Company, Goteborg, Sweden, 2005.

[22] G. Yu, M. Arguello, G. Song, S. M. McCowan, and A. White. A New Era for Crew Recovery at Continental Airlines. Interfaces, 33(1), 5-22, 2003.

[23] W. Guo, G. Yu, and M. Song. Optimization Model and Algorithm for Crew Management During Airlines' Irregular Operations. Journal of Combinatorial Optimization, 1, 305-321, 1997.

[24] J. W. Yen. A Stochastic Programming Formulation of the Stochastic Crew Scheduling Problem. Ph.D. thesis, University of Michigan, Ann Arbor, MI. 2000.

[25] M. Ehrgott, D. M. Ryan. Constructing Robust Crew Schedules with Bi-criteria Optimization. Journal of Multi-Criteria Decision Analysis, 11(3) 139–150, 2002.

[26] A. J. Schaefer, E. L. Johnson, A. J. Kleywegt, G. L. Nemhauser. Airline Crew Scheduling Under Uncertainty. Transportation Science, 39(3), 340-348, 2005.