不定期便に対応したパイロット乗務スケジューリング

田島 玲　味園 真司
日本アイ・ビー・エム株式会社 東京基礎研究所

Abstract

パイロットの乗務スケジューリングとは、与えられた航空機の運航ダイヤから、それらを過不足なくカバーするパイロットの勤務パターンを作成する問題であり、従来多くの研究がなされている。そのほとんどは全ての便が毎日飛ぶと仮定してコンパクトに表現した問題のみを解くものだった。しかし近年、顧客の満足度向上のため多くの不定期便が運航されており、スケジューリング時に明示的に扱うことが求められつつある。この論文では、整数計画法と補助グラフを用い、そうした不定期便に対応した手法を示す。また、航空会社より提供された現実のデータを使った検証も行なう。

Airline Crew Scheduling Problem with Many Irregular Flights

Akira Tajima　Shinji Misono
IBM Research, Tokyo Research Laboratory

Abstract

The airline crew scheduling problem has been investigated for over twenty years. Previous studies have assumed that every flight is available every day during the scheduling period. But these days, the number of irregular flights is increasing to satisfy the passengers' requirements. This article presents a new approach that reduces the total number of man-days in pilots' round-trip flight patterns (called crew pairings) that cover all the regular and irregular flights. It systematically find ways to merge irregular flights into pairings consisting only of regular flights. The approach is validated by using real-world datasets provided by an airline company.

1 Introduction

We discuss the flight crew scheduling problem under a new circumstance where many flights are scheduled only on certain parts in the given period of time. The flight crew scheduling problem is to find the optimal assignment of flight crews to a set of scheduled flights, or *flight legs*, on aircraft of the same type. More precisely, we consider a *crew pairing* that starts from the crew's base and returns to the same base after consecutive flights while satisfying various regulations and other conditions. It can contain *deadheads*, which represent the repositioning of crews as passengers. Each crew pairing is assigned a cost. The crew scheduling problem is then, given a set of flight legs in a given period of time, to find a set of crew pairings that covers all the flight legs with the minimum cost. It has been attracting practical interest for over twenty years [12, 10, 1, 6, 7, 2, 3, 8, 5, 11], since the crew's cost usually accounts for the main portion of the operational cost of every airline company. In most cases it can be formulated mathematically as the *set partitioning problem* (SPP), and many heuristics and approaches have followed this line of investigation [10, 6, 9, 4, 13].

Recently, the authors had an opportunity to perform a feasibility study of the automation of the crew scheduling procedure for an airline company. The real data sets supplied by the airline company differ significantly from those discussed in published studies, in that each of them contains a significant number of *irregular flights*. By irregular flights we mean those that are scheduled only in certain parts of the given period. We call flight legs that have the same schedule every day in the given period *regular flight legs*. The existing literature on crew scheduling discusses data only for regular flights. Moreover, the cost of crew scheduling in the airline company strongly depends on the total number of *man-days* of crew pairings. Here the *man-day* of a pairing is defined as its duration, or how many days it lasts. The primal objective of the crew scheduling system in this paper is to find the crew pairings with the minimum number of man-days when there are many irregular flights.

The simplest approach for handling many irregular flights and for finding pairings with the minimum number of man-days is to solve the *full-sized* SPP defined over the full range of the given period (usually a month). But the size of the full-sized SPP becomes far too large to handle.

An approach to reducing the problem size is to separate flights into two sets: one of regular flights and the other of irregular flights. For a set consisting only of regular flights we can easily transform a full-sized SPP defined over a month into a *compact* SPP defined over only three days (three is the maximum permissible duration of a crew pairing). This approach is not useful, however, if the total number of man-days is an important measure, as in the present case. In the abovementioned airline company, experienced engineers have hitherto scheduled crew parings manually. Many useful techniques have been developed for reducing the total number of man-days when there are many irregular flight legs. We found that the simple separation approach cannot outperform the manually created schedule.

We therefore propose a new approach to solving the crew scheduling problem with many irregular flight legs, and show that this approach can find a schedule superior to a manually designed one both in terms of the total number of man-days and other costs. Our approach systematically finds ways to merge irregular flight legs into pairings consisting only of regular flights, without solving the set partitioning problem of the full size. It has five steps: (1) We begin by solving the full-sized SPP defined over a month for irregular flights. From the obtained schedule we extract irregular *flight blocks*, or a sequence of flights that is to be used in the insertion step below. (2) For regular flights we enumerate all feasible paths within a day for all pairs of airports. We define each path's cost so that it reflects how well the path accepts irregular flight blocks. (3) We solve the compact SPP defined over three days for regular flights. (4) We insert as many irregular flight blocks as possible into the pairings for regular flights. (5) For irregular flights not inserted into regular pairings, we solve the full-size SPP again.

The organization of this paper is as follows. In section 2 we explain our crew scheduling problem in detail. In section 3 we describe our approach, and discuss its performance through numerical experiments for real-life data sets in section 4. Summary is given in section 5.

2 Problem Statement

2.1 Objective of Crew Scheduling

The objective of the given crew scheduling problem is to find the set of crew pairings with the minimum cost that covers all flight legs in a given period of time (a month). A characteristic of the present case is that the real data we consider contain many *irregular flights*. The number of irregular flights is large in our case. In one month of 1996, it amounted to 3116 out of 5760 flights, over 55% of the total. The airline company expects this proportion to increase year by year.

The airline company's prime goal for the crew scheduling is to minimize *the total number of man-days* (which is to be described below) involved in a crew schedule for the given period. Its secondary goal is to find the one that minimizes a quantity determined by crews' expenses and various preferences about pairings. Examples of factors that determine this quantity are on-duty hours, hours used for deadheads, and the number of flights.

We define the number of *man-days* by the duration of each pairing; a one-day-long pairing occupies one man-day, while a three-day-long pairing occupies three man-days. We then sum up the man-days for all pairings in a crew schedule to obtain the total number of man-days N_{tot}^{MD}. Consider the case in which a crew schedule for a month consists of a two-day-long pairing (pairing p) composed only of regular flights and a one-day-long pairing (pairing q) composed only of irregular flights which are scheduled only for the first ten days of the month. Suppose the paring p is composed of

$$\begin{array}{rl} \text{1st day}: & B_1- < FR_1 > -A_1- < FR_2 > -A_2 \\ \text{2nd day}: & A_2- < FR_3 > -A_3- < FR_4 > -B_1 \end{array}$$

where B_i, A_i, FR_i, are the i-th base airport, airport, and regular flight, respectively. Suppose also that the pairing q is composed of

$$B_1 - (DH_1) - A_1- < FI_1 > -A_4- < FI_2 > -A_1 - (DH_2) - B_1,$$

where FI_i is the i-th irregular flight. The total number of man-days N_{tot}^{MD} for this crew schedule then becomes 70 $(= 2 \times 30 + 10)$.

If we can find a good combination of irregular flights with regular flights, we can reduce the total number of crew pairings. In the above examples of pairings, p and q, if we can merge irregular flights, FI_1 and FI_2, into the pairing p for the first ten days in a month as

$$\begin{array}{rl} \text{1st day}: & B_1- < FR_1 > -\mathbf{A_1}- < \mathbf{FI_1} > -\mathbf{A_4}- < \mathbf{FI_2} > -\mathbf{A_1}- < FR_2 > -A_2 \\ \text{2nd day}: & A_2- < FR_3 > -A_3- < FR_4 > -B_1, \end{array}$$

then we can reduce N_{tot}^{MD} from 70 to 60 $(= (10 + 20) + 1 \times 30)$. In the airline company we worked with, crew scheduling has hitherto been done manually by experienced engineers. Many techniques have been developed for reducing N_{tot}^{MD}. Our main motivation is to find a systematic method to reduce N_{tot}^{MD}.

2.2 Set Partitioning Formulation

We formulate the crew scheduling problem as the SPP as

$$\begin{array}{lll} & \text{minimize } \sum_{j=1}^{n} c_j x_j & \\ \text{s.t.} & \sum_{j=1}^{n} a_{ij} x_j = 1 & \text{for all } i = 1, ..., m \\ & x_j \in \{0, 1\} & j = 1, ..., n \end{array} \qquad (1)$$

where $x_j = \begin{cases} 1 & \text{if a pairing } j \text{ is adapted in the schedule} \\ 0 & \text{otherwise,} \end{cases}$

c_j is the cost of pairing j,

$$a_{ij} = \begin{cases} 1 & \text{if a pairing } j \text{ contains the flight leg } j \\ 0 & \text{otherwise,} \end{cases}$$

n is the total number of possible pairings, and m is the total number of flight legs in a month.

We construct the cost c_j of the pairing j from two parts:

$$c_j = C^{MD} \times (\text{number of man-days of pairing } j) + C_j^{MISC} \tag{2}$$

where C^{MD} is the coefficient and C_j^{MISC} is the function of cost related to the secondary goal described in the previous subsection. We set the coefficient C^{MD} to be large enough compared to C^{MISC} so that the man-day part dominates the cost c_j.

In the full-sized SPP, which is defined over the full range of the given period (a month), the flight leg index i specifies not only its flight number but also its service date; thirty different indices are used for one regular flight number. In the compact SPP defined over three days, on the other hand, the index i specifies only the flight number.

3 Approach

3.1 Outline

First we solve the full-sized SPP for irregular flights and the compact SPP for regular flights separately in order to reduce the problem size and thereby the computation time. We then perform *multi-step insertion* of irregular *flight blocks* into pairings of regular flight legs. Here we define a *flight block* as a sequence of flight legs, or a part of a pairing within a day. Our approach reduces the total number of man-days by using the multi-step insertion as a post-processing to the solution of the compact SPP for regular flights.

Multi-step insertion is a generalization of the merging procedure introduced in Section 2.1. Single-step insertion of an irregular flight block is a procedure for merging it into a pairing consisting only of regular flight legs. Such insertion is possible in two cases: (1) inserting the flight block into an interval in a pairing, or into a period when a crew is staying at an airport, (2) replacing a deadhead in a pairing with the flight block. The previous section contained an example of case 1.

In *multi-step insertion*, we first merge a irregular flight block into a pairing by pushing out a regular flight block in it. We then insert the pushed-out regular flight block into another pairing. We continue these processes until the pushed-out blocks fit into a pairing without creating any more pushed-out blocks. As an example of two-step insertion, let us consider the case in which there is an irregular flight block $B_1 - < FI_5 > - A_{10} - < FI_6 > - B_1$ and two regular pairings

$$\begin{aligned} &\text{Pairing 1} : B_1 - < FR_9 > - A_7 - < FR_{10} > - B_1 - < FR_{11} > - A_8 - < FR_{12} > - B_1 \\ &\text{Pairing 2} : B_1 - < FR_{13} > - A_9 - < FR_{14} > - B_1. \end{aligned}$$

We first insert it into pairing 1 by pushing out the block $B_1 - < FR_{11} > - A_8 - < FR_{12} > - B_1$ as

$$\text{Pairing 1} : B_1 - < FR_9 > - A_7 - < FR_{10} > - B_1 - < \mathbf{FI_5} > - \mathbf{A_{10}} - < \mathbf{FI_6} > - B_1$$

then merging it into pairing 2 to obtain

Pairing 2 : $B_1 - < FR_{13} > - A_9 - < FR_{14} > - B_1 - < \mathbf{FR_{11}} > - \mathbf{A_8} - < \mathbf{FR_{12}} > - B_1$.

The complete procedure in our approach is as follows, and the details of each step are given in following subsections.

Step 1: Generate a crew scheduling for irregular flights by solving the full-sized SPP defined over a month for irregular flights.

Step 2: Construct a set of irregular flight blocks from the crew pairings obtained in Step 1. Enumerate all feasible paths within a day for all pairs of airports, using only regular flights, and define the cost of each path so that it reflects the potential to include irregular flight blocks.

Step 3: Generate a crew scheduling for regular flights by solving the compact SPP defined over three days by the column generation method, using the paths enumerated in Step 2.

Step 4: Apply multi-step insertions of irregular flight blocks into the crew pairings for regular flights.

Step 5: List the irregular flight legs that were not merged into regular pairings in Step 4. Solve the full-sized SPP for these irregular flights again to obtain the crew schedule.

3.2 Solving the SPP

In the feasibility study the airline company limited the computation time for generating a crew schedule to two hours on a workstation. We thus content ourselves with finding approximate solutions to the SPP.

We employ the linear programming relaxation (LPR) of the SPP and the column generation approach [10, 6, 3]. Starting from a limited number of columns selected from candidate pairings, we iteratively solve the LPR, extract a fixed number of promising pairings that have negative reduced cost with respect to the dual potential in the optimal solution of the LPR, and add them to the column set. We repeat this procedure until no further improvement of the LPR is attained. More precisely, we first include all one-day-long pairings, then iteratively add a fixed number of promising two-day-long pairings to the column. After it terminates, we start the iteration for the three-day-long pairings. For the columns used in the last iteration, we solve the SPP itself.

The generation of pairings is one of the most time-consuming parts of the column generation method. Laboie et al.[10] discuss a fast column generation approach based on calculation of the shortest path in a graph. In our case, however, this approach is not applicable, since the constraints and the cost evaluation for a pairing are more complicated. Thus, we enumerate all feasible one-day paths for all pairs of airports, and calculate their costs in advance. Later, during the column generation procedure, we patch two one-day paths to generate a two-day-long pairing. The number of feasible three-day-long pairings explodes, and the candidates are priced out after their first two paths are fixed by checking whether the minimum available reduced cost is negative or not.

An advantage of the path-connecting method lies in the efficiency of checking whether a flight block b can be inserted into a path pt of a pairing p or not, which is required in modifying pairings' cost to reflect their potential to include irregular flight blocks. Such test can be implemented efficiently by using hashing techniques.

3.3 Constructing flight blocks

A flight block, which is a unit of a multi-step insertion operation, can be as small as a single flight leg. However, we consider only flight blocks that either start or end at one of the base airports. We adopt this approximation on the basis of the following observations. First, multiple flight legs are inserted at the same time in case 1 in section 3.1. Second, in the irregular pairings obtained in Step 1, there are many series of irregular flights whose components are tightly connected, which start or end at one of the bases.

We divide the pairings obtained by solving the SPP into flight blocks, using base airports as delimiters. As for Paring 1 in section 3.1, for example, splitting at base B_1 gives us two flight blocks, $B_1 - < FR_9 > - A_7 - FR_{10} - B_1$, and $B_1 - < FR_{11} > - A_8 - < FR_{12} > - B_1$.

3.4 Cost adjustments for regular pairings

We modify the costs of pairings defined in equation (2) for regular flights so that larger number of irregular flight blocks are inserted in regular pairings. More concretely, for each regular path, which is a component of a regular pairing, we check if any irregular flight block can be inserted in it. If such block exits, we decrease the path's cost by a fixed amount C^{FB}. As C^{FB} increases, the number of insertable flight blocks increases. At the same time, the number of man-days of regular pairings increases. We thus need to set C^{CB} so that the best trade-off is realized.

3.5 Multi-step insertions using auxiliary graph

In executing multi-step insertions in Step 4, we use an auxiliary graph G where each node represents a flight block or a pairing. Arcs are defined between a flight block node and a pairing node. A pairing node which has incoming arcs and outgoing arcs can participate in a push-out. A pairing node with only incoming arcs can be the end of a multi-step insertion. In graph G, a multi-step insertion is represented as a path from a irregular flight block node to a pairing node.

To obtain a set of edge disjoint paths in G that maximize the total number of inserted irregular flight blocks, we apply a greedy method whose primal priority is the number of operation dates of irregular flights. We also consider a minimum-cost-flow approach with the same graph G with different cost settings for arcs. The details of G and the both methods will be given in the full paper.

4 Computational experiments

4.1 Setups

We use two datasets, A and B, supplied by the airline company for periods of 30 days, containing the actual flight schedules for two months in 1996. These two differ in the ratio of regular and irregular flights as shown in Table 1. Table 1 also includes the scheduling results obtained with our prototype. Data A has the largest number of irregular flights for any month up to the present, and the airline company predicts that the number of irregular flights will grow in future. We will therefore use Data A as a basis for our discussions in the following sections.

We built our prototype by using the IBM Optimization Subroutine Library to solve set partitioning problems (SPPs) and their linear programming relaxations (LPRs). All runs were made on an IBM RS/6000 model 990.

We add a fixed amount of new pairings to the column of the constraint matrix in the SPP at each iteration of the column generation procedure. We set the fixed size to 4,000 for two-day-long pairings, and to 20,000 for three-day-long pairings. The column size then becomes between 70,000 and 90,000 when the column generation procedure terminates. We adopt, as the final solution, the first feasible solution found in the course of the branch-and-bound, which is available in several minutes. It is well known that feasible solutions found during the execution of the branch-and-bound hae costs quite near to the optimal one [9]. In our case with Data A, the cost of the first feasible solution is 0.6% over the optimal one.

4.2 Results

We compare the output of our prototype system with the schedule generated by the experienced engineers in the airline company. They have several techniques for reducing the number of man-days, as we have described in section 2. Their techniques, however, do not take into account of the miscellaneous cost C_j^{MISC} in equation (2).

As shown in table 2, our result outperforms that of experienced engineers in both the total number of man-days and the total cost: the total man-days are lesser by 1.2% and the cost is smaller by 2.1%. The values of the cost are normalized so that the cost of the schedule output by our prototype is equal to 100.

Table 3 shows the effects of insertions and cost adjustment. The first row indicates that, without the cost adjustment, only a few irregular flights can be inserted, and the result is inferior to that obtained by human experts, shown in Table 2.

In Table 3, C represents the degree of the adjustment. Its value should be determined relative to other cost parameters. Here, we use C^{MD} in equation (2) as a measure for C. As the value of C increases, the number of inserted units tends to increase. This means that the cost adjustment works. However, the number of man-days for regular flights also increases. Thus, the total number of man-days after the insertions first decreases, then increases as the value of C increases. Hence the value of C should be carefully selected reflecting the characteristics of dataset.

Table 1: Specifications of datasets and computational results

Dataset	Data A		Data B	
Specifications				
Regular flights × operation dates (=30)	5760 (=192 ×30)		8730 (=291×30)	
Irregular flights × operation dates	3116		774	
computational results				
CPU time	41 minutes		92 minutes	
Man-days (regular)	2580	(86×30)	3030	(101×30)
Man-days (inserted)	420	(14×30)	90	(3×30)
Man-days (irregular)	188	(6×30+8)	170	(5×30+20)
Man-days (TOTAL)	3188	(106×30+8)	3290	(109×30+20)

We also investigate how far we can reduce the total number of man-days of regular pairings. We add a constraint regarding the total number of man-days to the SPP for regular flights. It turned out that no solution exists with the total number of man-days less than 99×30. It means our approach with the cost setting in equation (2) successfully found the minimum possible value of the total man-days of regular flights.

5 Conclusions

We have addressed the crew scheduling problem with many irregular flights whose primal goal is to minimize the total number of man-days of pairings. We have presented a new approach to the problem: We first solve the SPP for a set of regular flights and that for a set of irregular flights separately, and then merge irregular flight blocks into regular pairings as many as possible through multi-step insertions. We have validated our approach by using real-world datasets provided by an airline company, without any simplifications for constraints. The schedule output by our prototype system was feasible in practice, and outperformed the schedule created by experienced engineers in the airline company, in both the total man-days and other cost.

Not only the airline company we worked with, but also other companies are increasing the number of irregular flights to cope with the various requirements of their passengers. Thus, there are growing needs for heuristics for solving the crew scheduling problem including many irregular flights.

Acknowledgments

The authors wish to thank Koichi Tsuzuki, Yukihiro Fujimoto, and Takeshi Ujikawa, all of the airline company they coped with, for their supplying not only real data sets and manually designed pairings, but also helpful comments and suggestions.

References

[1] Abara, J. (1989). "Applying integer linear programming to the fleet assignment problem," *Interfaces* 19, 20-28.

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

[3] Anbil, R., R. Tanga, and E. L. Johnson. (1992). "A global approach to crew-pairing optimization," *IBM Systems Journal* 31, 71-78.

[4] Atamtürk, A., G. L. Nemhauser, and M. W. P. Savelsbergh. (1995). "A combined Lagrangian, linear programming, and implication heuristic for large-scale set partitioning problems," *Journal of Heuristics* 1, 247-259.

Table 2: Result comparison

	Experienced engineers		Our prototype	
Number of man-days	3226	(107×30+16)	3188	(106×30+8)
Cost		102.1		100.0

Table 3: Effects of insertions and cost adjustment

Value of C	Man-days before insertions (regular+irregular)		Inserted blocks	Man-days After insertions (total)	
0	3242	(99×30+272)	1	3233	(107×30+23)
C^{MD}×1.0	3242	(99×30+272)	1	3226	(107×30+16)
C^{MD}×1.2	3242	(99×30+272)	3	3204	(106×30+24)
C^{MD}×1.4	3242	(99×30+272)	2	3212	(107×30+02)
C^{MD}×1.6	3242	(99×30+272)	3	3196	(106×30+16)
C^{MD}×1.8	**3242**	**(99×30+272)**	**4**	**3188**	**(106×30+08)**
C^{MD}×2.0	3272	(100×30+272)	4	3203	(106×30+23)
C^{MD}×2.2	3272	(100×30+272)	4	3203	(106×30+23)
C^{MD}×2.4	3332	(102×30+272)	7	3233	(107×30+23)
C^{MD}×2.6	3362	(103×30+272)	9	3253	(108×30+13)
C^{MD}×2.8	3362	(103×30+272)	10	3251	(108×30+11)
C^{MD}×3.0	3362	(103×30+272)	9	3240	(108×30+00)

[5] Barnhart, C. L. Hatay, and E. L. Johnson. (1995). "Deadhead selection for the long-haul crew pairing problem," *Operations Research* 43, 491-499.

[6] Desrochers, M., and F. Soumis. (1989). "A column generation approach to the urban transit crew scheduling problem," *Transportation Science* 23, 1-13.

[7] Gershkoff, I. (1989). "Optimizing flight crew schedules," *Interfaces* 19, 29-43.

[8] Graves, G. W., R. D. McBridge, I. Gershkoff, D. Anderson, and D. Mahidhara. (1993). "Flight crew scheduling," *Management Science* 39, 736-745.

[9] Hoffman, K. L., and M. Padberg. (1993). "Solving airline crew scheduling problems by branch-and-cut," *Management Science* 39, 657-682.

[10] Lavoie, S., M. Minoux, and E. Odier. (1988). "A new approach for crew pairing problems by column generation with an application to air transportation," *European Journal of Operational Research* 35, 45-58.

[11] Pavlopoulou, C., A. P. Gionis, P. Stamatopoulos, C. Halatsis. (1996). *Proceedings of the Second International Conference on the Practical Application of Constraint Technology*, London, UK, 24-26.

[12] Rubin, J. (1973). "A technique for the solution of massive set-covering problems with application to airline crew scheduling," *Transportation Science* 7, 34-48.

[13] Wedlin, D. (1995). "An algorithm for large scale 0-1 integer programming with application to airline crew scheduling," *Annals of Operations Research* 57, 283-301.