AIRLINE CREW SCHEDULING

An airline must cover each flight leg with a full complement of cabin crew in a manner consistent with safety regulations and award requirements. Methods are investigated for solving the set partitioning and covering problem. A test example illustrates the problem and the use of heuristics. The Study Group achieved an understanding of the problem and a plan for further work.

1. Introduction

This problem was presented to the 1992 MISG at Macquarie University by The Preston Group Pty Ltd. The Preston Group is a systems development company providing cost effective workstation-based interactive scheduling and simulation products. Its products for the international airline industry include

- Total Airspace and Airport Modeller (TAAM)
- Terminal Management Systems (TMS)
- Air Crew Scheduler (ACS)

The Air Crew Scheduler is an interactive computer software system for air crew scheduling. The system is used by planners to develop and modify crew pairings for flight crews and cabin crews. Currently the Air Crew Scheduler facilitates the production of legal crew schedules. The Preston Group wishes to enhance the product by developing a capability for

- automatic generation of legal trips
- optimisation of crew schedules which cover all flight legs

The input tasks are a set of flight legs. A flight (sector, segment) leg corresponds to a flight between two cities, departing one city at a specified time and arriving at the other city at a specified time. A duty period is made up of a set of flight legs. Safety requirements limit the number of flight legs in a duty period and the duration of the period. A pairing (pattern) is a sequence of required duty periods that a crew must complete that starts and ends at the same domicile. A regulation specifies the maximum duration of a pairing and the minimum rest period between duty periods. In long haul operation, the duration of a pairing may be as long as 21 days and a minimum rest period may be 46 hours. A rest period between duties is called a slip. A bid line (roster line) is a set of pairings that represent the work schedule for the crew over the planning horizon.

2. Background

The various airlines have developed different methods of operation which are described as hub and spoke, short haul and long haul. Long haul operations are characterised by factors such as

- days away from base
- multi-base operations
- language requirements of crew
- paxing (passengering crew between operating flights)
- complex duty and rest rules
- crew splitting due to different size aircraft.

Figure 1 illustrates the processes involved in the planning of airline operations. Estimates of passenger demand are used to develop a timetable and schedule the aircraft. Each flight leg of the aircraft has requirements for technical crew and cabin crew. We limit our work to the development of cabin crew schedules. Many airlines plan their crewing via a three stage process:

- development of pairings or crewing patterns which specify a crew schedule from base until return to base
- development of bid lines which link pairings and specify a crew schedule over the planning horizon
- development of a roster from the preferences expressed by the crew for the bid lines.

The Study Group restricted its attention to the development of pairings which efficiently cover the required flight legs over the planning horizon. The group did not consider real time operation where disruption can affect the schedule.

The costs of flight crew and cabin crew form a major component of the direct operating costs of an airline. Air crew planners continually strive for more effective use of available crews to reduce the total number of crews needed and to minimise costs of crew stopovers and allowances. However, the planners' tasks usually involve much time-consuming and laborious manual work. This reduces the time available to research various crewing options which may produce much more effective crewing pairings.

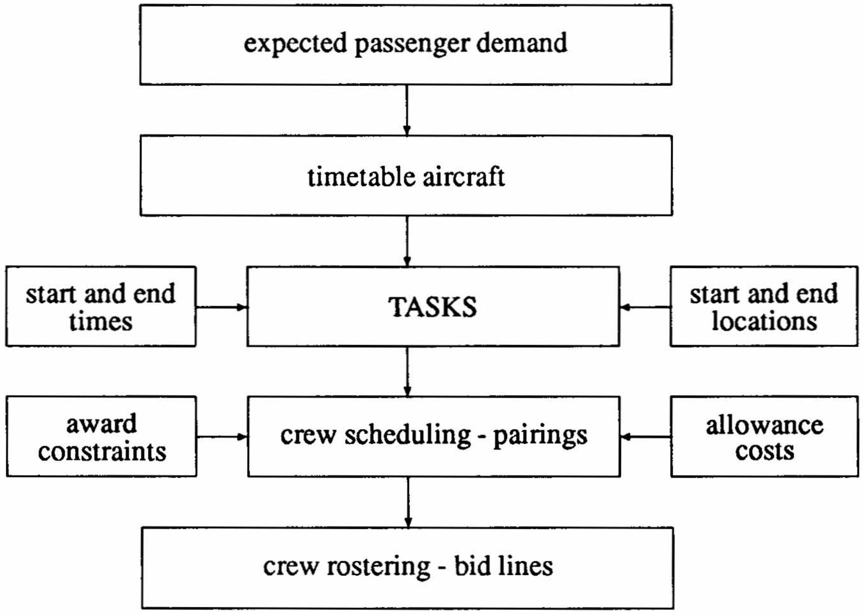

Figure 1: Processes involved in the planning of airline operations.

The planner starts with a schedule or plan of aircraft movements for a period of time. The planner then constructs pairings of air crew movements from any crew home base, along a number of sectors, with suitable rest or slip periods, and return to home base, usually some days later. In developing these pairings, the planner must ensure that numerous government regulations, union/management agreements and safety rules are met.

The rules which specify duty and rest periods are determined by government and management. Some rules are hard (must) while others are soft (should). Examples of a duty rule and a rest rule follow.

- Given that Duty is Paxing and Duty is Operating and Last Sector in Duty is not Operating and Duty is not To Home then:
  Operating time in Duty must be less than 12 hours

- Given that Previous Duty is valid and Slip Time in Previous Duty < 46 hours then:
  Slip Time in Duty should be > 46 hours

Planners have traditionally used spider graphs to assist in the manual process of crew scheduling. The planner creates pairings by selecting a flight from the home base and then adding legal flight legs until the planner selects a flight leg to home base. Skill is required to select cost effective pairing which cover all flight legs over the planning horizon, say 8 weeks.

The Preston Group has developed a computer assisted system which incorporates the rule base for determining legal trips and graphics to display spider graphs. The system

- selects trips from the mainframe
- partitions the trips in different geographical areas
- re-uses previous pairings
- produces multiple solutions
- compares the ratio of flight time to total time (premium)
- returns the pairings to the mainframe

The sets of resulting pairings for a planning period are the basis for assigning actual crew names and are made available to a preferential bidding system, assembled into pairing strings in a bid line system, or assigned to specific crew members in an assignment system.

The Study Group was asked to address the particular issues of

- long haul operations where crews may be away from home base for as long as 21 days
- crew splitting in which a crew of 17 may finish a flight leg and then start new flight legs as crews of 8 and 9 on smaller planes.

3. Set partitioning problem

The airline crew scheduling problem is to find a minimum cost set of pairings which cover all flight legs.

The problem can be formulated as a set partitioning problem (SPP). The input data is an A matrix whose rows are the flight legs and columns are the pairings. Let the input data be represented by

m = number of flight legs
n = number of pairings
c_j = cost of pairing j

$$a_{ij} = \begin{cases} 1 & \text{flight leg } i \text{ contained in pairing } j \\ 0 & \text{otherwise} \end{cases}$$

The decision variables are

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

The set partitioning problem (SPP) is

$$\text{minimise} \sum_j c_j x_j \tag{1}$$

subject to

$$\sum_j a_{ij} x_j = 1 \quad \text{for } i = 1, 2, \ldots, m \tag{2}$$

$$x_j = \{0, 1\} \quad \text{for } j = 1, 2, \ldots, n \tag{3}$$

The integer programming problem (1–3), can be solved by first solving the relaxed linear programming problem where (3) is replaced by

$$x_j \geq 0 \quad \text{for } j = 1, 2, \ldots, n \tag{4}$$

and then using branch and bound to remove fractional solutions. Experience shows that constraint branching is more efficient than variable branching for these problems, and it is usually the extent of the branch and bound process that limits the size of the problems that can be solved this way. We take a closer look at branch and bound approaches in Section 5 of this report.

4. Column generation

For realistically sized problems, it is not possible to include all possible columns that represent legal pairings. The problem is to generate a "good" set of pairings. A file containing flight information was obtained from The Preston Group. Part of this file, with flights numbered for reference, is shown below.

```
  1  QFA0009  744   SYD(31/03/91 13:15)->MEL(31/03/91 14:45) 01:30
  2  QFA0001  744   MEL(31/03/91 13:45)->SYD(31/03/91 15:05) 01:20
  3  QFA0009  744   MEL(31/03/91 16:00)->SIN(31/03/91 21:40) 07:40
  4  QFA0001  744   SYD(31/03/91 16:30)->BKK(31/03/91 22:55) 09:25
  5  QFA0009  744   SIN(31/03/91 23:05)->LHR( 1/04/91 05:50) 13:45
 19  QFA0001  744   BKK( 2/04/91 00:30)->LHR( 2/04/91 06:55) 12:25
 98  QFA0010  744   MEL( 9/04/91 06:45)->SYD( 9/04/91 08:05) 01:20
 99  QFA0010  744   LHR( 8/04/91 22:30)->SIN( 9/04/91 18:40) 13:10
100  QFA0002  744   BKK( 9/04/91 08:20)->SYD( 9/04/91 20:05) 08:45
```

A program has been written to generate pairings from the information in this datafile. The program generates the pairings by constructing a tree in which a given flight destination is linked to all subsequent legal flights whose source is the same as this location. Once a tree is constructed for a given starting flight, the pairings are extracted by traversing the paths from the initial node to each of the final nodes and extracting the flight numbers at each node.

The program is at present linked to a rule base consisting of a subset of the full Preston Group rule base and has been used to generate relatively small sets of pairings using only 100 flights in a 10 day interval. These pairings provide reasonably sized data matrices for testing set partitioning and set covering algorithms to find the optimal schedules.

A legal pairing generated by the program is shown below. The slip and duty times are in hours and the cost is calculated as the ratio of total time to duty time.

```
 4  SYD-BKK : slip    0.00 :  duty    9.42 : cost   1.00
19  BKK-LHR : slip   25.58 :  duty   12.42 : cost   2.18
44  LHR-MAN : slip   48.00 :  duty    1.00 : cost   4.22
50  MAN-LHR : slip   10.50 :  duty    0.92 : cost   4.54
74  LHR-SIN : slip   51.17 :  duty   13.17 : cost   4.66
92  SIN-MEL : slip   25.58 :  duty    7.00 : cost   4.66
98  MEL-SYD : slip    1.50 :  duty    1.33 : cost   4.59
```

This pairing gives a column j with

$$c_j = 459$$

$$a_{ij} = \begin{cases} 1 & \text{for } i = 4, 19, 44, 50, 74, 92, 98 \\ 0 & \text{for all other } i = 1, \dots, 100 \end{cases}$$

This pairing (solid line) and a number of other legal pairings are shown in figure 2.

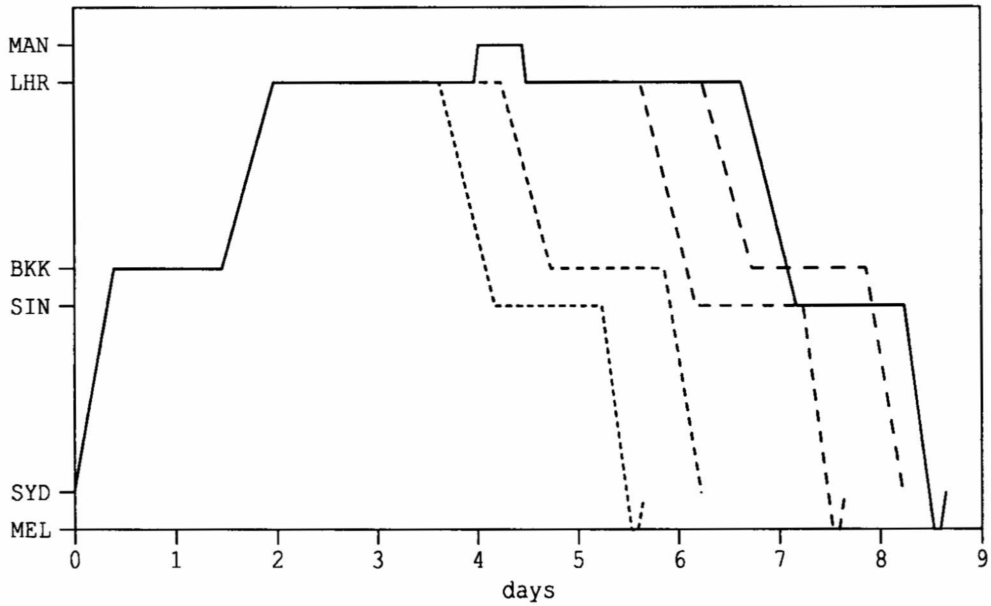

Figure 2: Spider graph showing some legal pairings from Sydney

To achieve integer programming problems of a size that can be solved it is necessary to reduce the number of columns, and this is done by choosing columns, *i.e.* potential pairings, that are likely to be in the final optimal solution. It is also desirable to choose columns that are likely to lead to fewer fractional solutions of the relaxed LP. There is an art in making appropriate choices here and the method then becomes heuristic.

There is also a method of generating further columns once the solution of the LP is found. To do this, a shortest path problem is solved where the nodes in the network are the tasks to be done; in this context the flights that are to be covered, and the distances represent the reduced costs of the flights in the current solution of the LP. If a sequence of flights can be found with a negative total reduced cost, then that represents a suitable variable to enter the basis of the LP. If no such sequence exists, then the solution of the LP is optimal over all possible pairings. It is not clear that this method is practical, however, since there is still the difficulty of finding the solution of the integer programming problem.

5. Branch and bound approaches

Consider the formulation of the Crew Scheduling Problem given by (1–3). It has been pointed out earlier in the report that this formulation is similar to the formulation of the general set partitioning problem (SPP). By replacing the equality constraints in the constraint-set (2) with the $\leq$ inequalities, the resulting formulation is the related set covering problem (SCP).

The formulation, given by (1–3) can be used, in conjunction with a constrained branching strategy, and a good (lower) bounding scheme to generate an algorithm for the SPP.

Bounds on the optimal solution of the subproblem at each node of the tree search can be computed from an LP relaxation (of the 0–1 binary integer restrictions on the variables x_j), thereby producing the formulation of the relaxed problem, given by (1), (2) and (4). This bounding strategy has proved to be computationally expensive, particularly for large problems. Furthermore LP relaxations of the SPP can be highly degenerate. However the LP lower bounds can be approximated via an approach that includes Lagrangian relaxation and a subgradient optimisation procedure to improve the Lagrangian bounds. Initial results by Etcheberry (1972), although for the set covering problem, indicated that small instances of the SPP could be solved using this method.

A Lagrangian relaxation of the SPP is the Problem LSPP, obtained by relaxing the constraint set (2) and by including it in the Lagrangian objective function with the help of Lagrangian multipliers $\lambda_i, i = 1, \ldots, m$:

$$\text{LSPP}_\lambda \equiv \min_{x_j \in \{0,1\}} \sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} \lambda_i (1 - \sum_{j=1}^{n} a_{ij} x_j)$$

That is

$$\text{LSPP}_\lambda \equiv \min_{x_j \in \{0,1\}} \sum_{j=1}^{n} (c_j - \sum_{i=1}^{m} \lambda_i a_{ij}) x_j + \sum_{i=1}^{m} \lambda_i$$

The solution of LSPP_λ for a given λ is trivial. If C'_j is defined by

$$c'_j = \sum_{j=1}^{n} (c_j - \sum_{i=1}^{m} \lambda_i a_{ij})$$

then the Lagrangian problem above is

$$\text{LSPP}_\lambda \equiv \min_{x_j \in \{0,1\}} \sum_{j=1}^{n} c'_j x_j + \sum_{i=1}^{n} \lambda_i \qquad (5)$$

It is easily verified that

$$x_j = \begin{cases} 1 & \text{if } c'_j < 0 \\ 0 & \text{otherwise} \end{cases}$$

Then

$$Z(\text{LSPP}_\lambda) = \sum_{j=1}^{n} c'_j x_j + \sum_{i=1}^{m} \lambda_i$$

forms a valid lower bound on $Z(\text{SPP})$, the value of the optimal solution of Problem SPP.

The best lower bound of this type is $l^* = \max_\lambda Z(\text{LSPP}_\lambda) \leq Z(\text{SPP})$, the Lagrangian dual problem, which is solved by finding λ^*, the best set of multipliers that maximizes the Lagrangian lower bound. This is obtained via the iterative subgradient optimisation approach.

Starting with an initial vector of Lagrange multipliers $\lambda = \lambda^0$, the values of the Lagrange multipliers $\lambda = \lambda^{k+1}$ at any iteration $(k + 1)$ are given by

$$\lambda_i^{k+1} = \lambda_i^k + p^k \left(\frac{Z_{UB} - Z(\text{LSPP}_{\lambda^k})}{\sum_{i=1}^{m} \|\delta_i^k\|^2} \right) \delta_i^k$$

where

p^k is a scalar multiplier such that $0 < p^k \leq 2$, and $p^k \to 0$ as $k \to \infty$
Z_{UB} is an upper bound
$Z(\text{LSPP}_{\lambda^k})$ is the lower bound at iteration k, using $\lambda = \lambda^k$
δ_i^k is the subgradient such that, $\delta_i^k = (1 - \sum_{j=1}^{n} a_{ij} x_j^k)$, $i = 1, 2, \ldots, m$

where x_j^k is the solution of (5) using $\lambda = \lambda^k$

The scalar multiplier is initialized, at iteration 0, to $p^0 = 2$. Then p^k at any iteration k is halved periodically (under certain conditions) until stopping criteria terminate the iterative procedure.

Although easy to implement, this algorithm has been shown to be ineffective, especially for large SPPs, since (a) the algorithm is sensitive to the initial choice of multipliers, $\lambda = \lambda^0$; (b) convergence to the optimal solution is slow, due to the oscillatory nature of the Lagrangian lower bound sequence; (c) the requirement for a good upper bound; and so on.

Recently, algorithms for set covering problems and mixed covering/partitioning problems by Balas and Ho (1980), Beasley (1987) and Fisher & Kedia (1990) overcome these difficulties by adopting continuous adaptive algorithms that apply a sequence of heuristic and exact procedures to obtain good solutions to medium-sized SCPs and SPPs. The basic idea in all these approaches is to employ various primal and dual heuristics and subgradient optimisation, similar to that explained above, in a continuous manner. The primal (greedy) heuristic produces upper bounds on the value of $Z(\text{SPP})$. A dual heuristic obtains a lower bound on the optimal solution of the LP dual of the SPP. Apart from giving a lower bound on the primal solution, the dual heuristic procedure also generates an initial value for the dual variables, associated with each of partitioning constraints. A subgradient optimisation method then uses the primal upper bound and the dual heuristic solution (as a starting value for the Lagrangian multipliers) to find the values of x_j. This cycle of adaptive procedures is applied continuously until stopping criteria are met. Each of these approaches make extensive use of logical tests for checking primal-dual feasibility and to reduce the problem size.

Although there is no conclusive evidence of the performance of this genre of algorithms on very large SPPs, there are reports of the solution of some problems of size 200 rows (flights) and 10,000 variables (pairings) in reasonable computing time.

Our belief is that these recent methods can be used effectively to solve large-scale SPPs of the type that was presented to the Study Group.

6. Heuristic solution methods

In the related problem of bus and crew scheduling, one method which is known to give good heuristic solutions is repeated matching. This method has been developed largely by Michael Forbes and a commercial code has been developed by OPCOM Pty Ltd. It has been applied successfully to problems of a size at least comparable with the air crew scheduling one presented here and with greater complexity. This is almost certain to work for air crew scheduling but a test on a full data set would be needed to confirm that.

In general terms, the method consists of first pairing the tasks optimally by using a matching (*i.e.* non-bipartite matching) algorithm. These then become potential pairings. The process is then iterated by continually breaking up the potential pairings, or a specified fraction of them, and recombining, again with the use of the matching algorithm.

The success of this method depends on having a fast matching code capable of solving very large scale matching problems. It also requires that the feasibility and cost of any potential pairings can be computed efficiently. As with the standard integer programming formulation, it also needs some user skill in setting up appropriate objective functions for the matching process and in choosing the appropriate breaking up of the existing pairings.

7. Further work

The Study Group achieved

- an understanding of the airline crew scheduling problem
- a set of test problems for evaluation of solution methods
- hands on experience in the generation of pairings
- a literature search of exact and heuristic solution methods for set partitioning

The requirements for further work on this problem are

- data sets of flight leg information and award rules
- costing information
- working pairings used by airlines
- documentation on the bidding process

The plan is to decouple the award rule base from the airline crew scheduler and use the rules to generate aircrew pairings. The next stage will incorporate heuristic algorithms to provide improvement capability.

Acknowledgements

The moderator for this problem (Graham Mills) would like to acknowledge the assistance of The Preston Group representatives, Myles Harding and Ken Greer, and the contributions of Natashia Boland, Mohan Krishnamoorthy, David Panton, Leslie Parcell, David Sier and Tony Watts.

References

J.P. Arabeyre *et al.*, "The airline crew scheduling problem: a survey", *Transportation Science* 3 (1969), 140–163.

E. Balas and A. Ho, "Set covering algorithms using cutting planes, heuristics and subgradient optimization: a computational study", *Math Programming* 12 (1980), 37–60.

J. Barutt and T. Hull, "Airline crew scheduling: supercomputers and algorithms", *SIAM News* 23 No 6, (1990).

J.E. Beasley, "An algorithm for set covering problems", *European Journal of Operational Research* 31 (1987), 85–93.

L. Bodin, *et al.*, "Routing and scheduling of vehicles and crews", *Computers and Operations Research* 10 (1983), 63–211.

T.G. Crainic, and J–M. Rousseau, "The column generation principle and the airline crew scheduling problem", *INFOR* 25 (1987), 136-151.

J. Etcheberry (1977), "The set covering problem: a new implicit enumeration algorithm", *Operations Research* 25 (1977), 760–772.

M.L. Fisher and P. Kedia, "Optimal solution of set covering/set partitioning problems using dual heuristics", *Management Science* 36 (1990), 674.

M.A. Forbes, J.N. Holt, P.J. Kilby and A.M. Watts, "A heuristic algorithm for the production and vehicle routing problem with deadlines" (preprint, 1991).

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

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

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