Journal of the Operational Research Society (2000) 51, 258–262 0160-5682/00 $15.00

http://www.stockton-press.co.uk/jors

Scheduling charter aircraft

D Ronen

University of Missouri-St. Louis, USA

We present a system that is used for scheduling charter aircraft. At the core of the system is an elastic set partitioning model that is embedded in a decision support system. The model assigns a set of flights that have to be performed to the available fleets of aircraft at minimal cost while satisfying all operational requirements. Flights that cannot be accommodated by the available fleets are sold off to other operators. The minimised costs include the cost of flying the aircraft, the cost of selling off flights, and penalties on violations of soft constraints. The system has been in daily operation for almost a year, and it provides high quality schedules and saves numerous hours to the schedulers.

Keywords: scheduling; air transport; integer programming; decision support systems

Introduction

Every operator of aircraft fleets faces the decision of which specific aircraft to assign to each flight. This problem is generally known as the aircraft (or fleet) routing problem. However, due to differences in operational environments, there is a large variety of aircraft routing problems.

A commercial airline publishes flight schedules and the set of flights to which aircraft have to be assigned is known well in advance, and, due to maintenance requirements, usually has cyclical patterns. Each aircraft has to visit a maintenance base every few days for inspection. In contrast, charter operators, especially those that operate smaller aircraft, often have less predictable flight patterns. For them quick aircraft scheduling decisions are crucial. A corporate fleet operator has somewhat more flexibility in scheduling his aircraft than a charter operator because parties may share flight legs in the same equipment (such sharing is usually not allowed in charter operations).

This paper deals with scheduling several fleets (aircraft types) of small jet aircraft by a charter operator, and describes a computerised decision support aid that we have developed for such an operator.

Although a considerable amount of work has been published regarding scheduling problems associated with operating aircraft, most of that work is focused on commercial airlines. Commercial airlines face a hierarchy of operational decisions concerning the deployment of their fleets. First they have to assign aircraft types to flights (*fleet assignments*), then assign a specific aircraft to each flight (*fleet routing*) and finally assign crews to the aircraft (*crew scheduling*), all these while meeting the aircraft maintenance requirements (*maintenance planning*). These decisions overlap and interact to a large extent, but usually they are too complex to be made simultaneously, and due to operational uncertainties, they have different planning horizons. Proper resolution of these issues may have a significant financial impact on an airline. *Fleet assignment* was recently addressed by Clarke *et al*.[1] *Fleet routing* for commercial airlines was analysed by Barnhart *et al*.[2] *Crew scheduling* has received a lot of attention, see Hoffman and Padberg,[3] Wark *et al*,[4] Day and Ryan,[5] and Beasley and Cao.[6] *Maintenance planning* was recently discussed by Gopalan and Talluri.[7]

We address here a combined *fleet assignment* and *fleet routing* problem that incorporates maintenance activities and crew availability considerations. Due to the nature of the operation the flights that have to be performed by the fleets are known only two or three days in advance, and therefore most of the work cited above is not applicable to our problem. A recent paper by Keskinocak and Tayur[8] analysed a problem almost identical to ours. However, we address the problem from a wider scope by considering crew availability and allowing subcontracting of sequences of flights. In contrast to their custom heuristic algorithm and use of a general purpose solver, we cast the problem into a set-partitioning model and use a customised solver. We solve larger size problems with a smaller integrality gap (our solutions are closer to optimal). The use of a set-partitioning model provides us much more flexibility in conveying scheduling preferences by minimising costs (rather than flying hours). The costs include penalties on violations of soft constraints, and minimal charges for using subcontractor aircraft.

Our aircraft scheduling problem has its counterparts in other transportation modes. In the trucking industry this problem is known as a truckload continuous move, where a

Correspondence: Dr D Ronen, School of Business Administration, University of Missouri-St. Louis, 8001 Natural Bridge Road, St. Louis, MO 63121, USA.

truck is assigned a sequence of full truckloads and after it delivers one it picks up the next one.[9] In ocean shipping a sequence of full loads is assigned to each vessel.[10]

The next section describes the problems that we faced, which is followed by the mathematical model. The schedule generation process which is a key to deriving good solutions is discussed in the next section, followed by the results and their analysis. We close with a conclusion.

Problem description

A fleet of several types of aircraft is available to perform a set of flights. Each aircraft may be available at a different location and at a different time. Due to the short notice of revenue flight orders, the planning horizon spans 24 to 48 hours. During the planning horizon an aircraft may have a maintenance appointment (at a known location). Each revenue flight has its origin, expected time of departure (ETD), destination, and requested aircraft type. However, a flight may be upgraded to a better aircraft type (but it may not be downgraded). In case the available fleet cannot accommodate all the flights, flights may be sold off to other operators at a known cost. The objective is to assign to each aircraft a feasible sequence of revenue flights in a manner that minimises the cost of flying the aircraft and selling off orders (if necessary) while meeting maintenance appointments and all other operational requirements. It should be obvious that ferry (positioning) flights may be necessary between revenue flights, and the cost of these flights must be taken into account. Work rules and limited availability of crews necessitate control over the number of required crew swaps (replacements) in the fleet schedule. Selling off a sequence of flights to another operator requires returning the aircraft to its home base at the end of its assignment and involves a daily minimum charge. The basic data requirements are listed in Figure 1.

We cast this problem into an Elastic Set-Partitioning (ESP) model and use an efficient ESP solver to solve it. The set partitioning approach is very appealing to problems like this where the costs are non-linear and discrete, and complex operational rules are involved. In addition, violation of soft constraints is controlled through penalties on such violations that are incorporated in the cost of the schedules. We generate a large number of feasible candidate schedules for each aircraft and select one schedule for each aircraft in a manner that covers each revenue flight exactly once and minimises the total cost of performing these flights and selling off flights that cannot be accommodated by the available fleets.

Elastic set partitioning model

We cast the problem into the following elastic set partitioning (ESP) model.

Schedule data:
- –Start date & time
- –End date & time
- –Crew work rules parameters
- –Maximal number of crew swaps allowed.

Aircraft (AC) data:
- –ID
- –type
- –home base
- –cost per flying hour
- –location avilable
- –date & time available
- –default ground time
- –crew data
- –(flying) hours remaining till maintenance
- –cycles remaining till maintenance
- –maintenance date, time and location

Revenue flights (orders) data:
- –flight number
- –requested AC type
- –departure airport
- –destination airport
- –ETD date & time
- –number of passengers
- –ground time override
- –cost of selling off

Figure 1 Basic data requirements.

Indices:

$a = 1, \ldots,$ Aircraft

$f = 1, \ldots,$ Flights

$s = 1, \ldots,$ Schedules

$SA(a)$ schedules for aircraft a

$SF(f)$ schedules performing (revenue) flight f.

Data:

Cost_s – cost of schedule s (a function of the aircraft and the set of flights)

SCost_f – cost of selling off flight f.

ICost_a – cost of keeping aircraft a idle.

$\mathrm{Swap}_s = 1$ if schedule s requires a crew swap, 0 otherwise.

MAXSWAP – maximum number of crew swaps permitted in the set of schedules for the aircraft.

Binary decision variables:

$$\mathrm{SCHED}_s = 1 \text{ if schedule } s \text{ is selected.}$$

$$\mathrm{SELLOFF}_f = 1 \text{ if flight } f \text{ is sold off.}$$

$$\mathrm{IDLE}_a = 1 \text{ if aircraft } a \text{ is idle.}$$

ESP formulation:

$$\min\left\{\sum_{s}\mathrm{Cost}_s\mathrm{SCHED}_s + \sum_{f}\mathrm{SCost}_f\mathrm{SELLOFF}_f + \sum_{a}\mathrm{ICost}_a\mathrm{IDLE}_a\right\} \quad (1)$$

subject to:

$$\text{for every flight:} \quad \sum_{s\in \mathrm{SF}(f)} \mathrm{SCHED}_s + \mathrm{SELLOFF}_f = 1 \quad (2)$$

$$\text{for every aircraft:} \quad \sum_{s\in \mathrm{SA}(a)} \mathrm{SCHED}_s + \mathrm{IDLE}_a = 1 \quad (3)$$

$$\text{crew swaps limit:} \quad \sum_{s} \mathrm{Swap}_s\mathrm{SCHED}_s \leqslant \mathrm{MAXSWAP} \quad (4)$$

The objective function (1) includes the cost of performing the selected schedules (including the associated penalties on violations of soft constraints), the cost of selling off single flights, and the cost of idle aircraft. Constraint (2) assures that every flight either appears in the schedule or is sold off. Constraint (3) assures that every aircraft is either assigned a schedule or is idle. Constraint (4) takes care of the crew swaps limit. By allowing violation of constraints at a penalty, the ESP model goes a long way to ensure feasible solution and hence makes the system robust.

We solve this problem using an extremely efficient solver for this type of problem.[11] However, the key to solving the problem lies in the generation of a large (but not too large) set of good feasible candidate schedules for each aircraft. This schedule generation process is described in the following section.

Schedule generation

The ESP model requires the generation of a set of alternate feasible schedules for each aircraft from which it selects the cheapest subset for all aircraft. On the one hand a large number of good alternate schedules for each aircraft is desirable in order not to foreclose good alternatives, but on the other hand if the generated set is too large it may be impossible to solve the problem in reasonable time. Our initial experiments with operational data indicated that the set of all feasible schedules for a single aircraft with a 48 hour planning horizon may consist of over a quarter of a million feasible schedules, a number that is far too large to handle. Therefore we introduced several parameters that control the generation of the feasible schedules and assure that the generated schedules are efficient and acceptable. These parameters, whose value is controlled by the user or the system administrator, are: (a) maximal ferry distance (for each aircraft type); (b) maximal waiting time for ETD (for each aircraft type) between flights; and (c) number of alternate closest revenue flights to consider for an aircraft once it has finished its formerly assigned flight. We call this last number **k** and, to a large extent, it controls the schedule generation process.

A flow chart of the schedule generation process is provided in Figure 2.

Firstly, all the data are read and checked for consistency and completeness. Major problems in the data may cause rejection of orders (revenue flights), aircraft, or even abortion of the run. Minor data problems are corrected and warnings are issued. Secondly the flight orders are sorted by their ETD in ascending order. Thirdly, a set of feasible schedules is generated for each aircraft. For each aircraft all the compatible orders are extracted. Each one of these orders is seeded in a candidate schedule. Then each candidate schedule is considered for appending additional orders. The next candidate schedule is selected and the **k** feasible closest orders (after the completion of the last order in the candidate schedule) are identified (closest in terms of ferry flight time). The candidate schedule is copied **k** times, and to each of these **k** copies a unique one of the **k** closest orders is appended. These **k** new candidate schedules are added to the end of the list of the candidate schedules. The process ends when there are no additional orders to append or no additional candidate schedules to consider for appending. At that point the process moves to the next aircraft. This

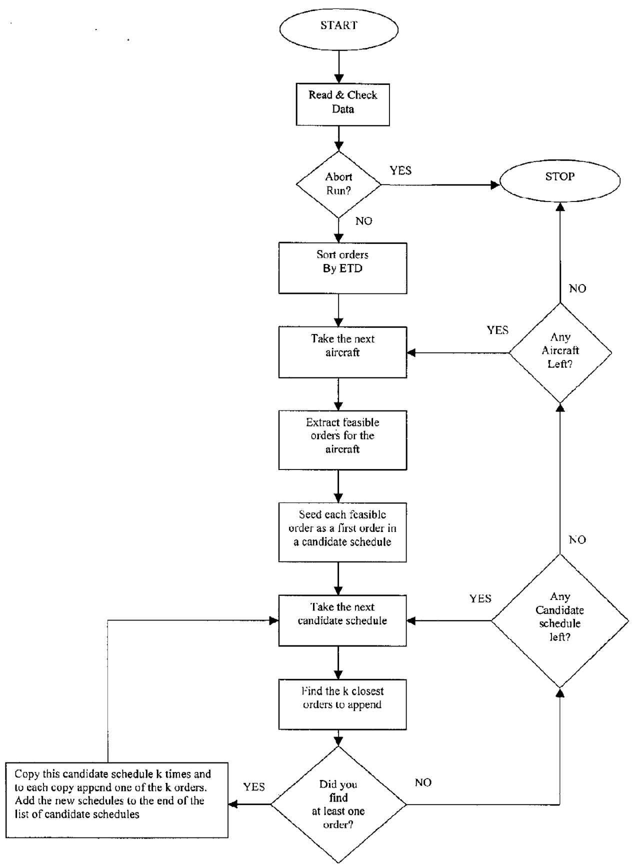

Figure 2 Schedule generation process.

process creates only feasible schedules that satisfy all the operational requirements. Maintenance appointments are incorporated in this process as a special type of orders.

The output of this schedule generation process is a set of (alternate) schedules for the aircraft. Each of these schedules consists of the aircraft ID and a sequence of revenue flights. Each one of these schedules is submitted to a cost calculator that calculates the cost of that specific schedule (that cost includes the associated penalties on violation of soft constraints). Special routines for calculation of flying distance and time between any pair of airports by a specified aircraft type are supporting this generation process. The resulting set of schedules along with their costs is submitted to the ESP model.

Selling off a sequence of revenue flights is often cheaper than selling off each one of these flights separately, but may invoke minimal charges. This alternative is incorporated in the schedule generator by including subcontractor aircraft in the aircraft data file and treating such aircraft according to their specific operating rules and costs.

Results and analysis

The performance of the system is demonstrated using operational data that were used to test the system. A summary of results for large problems is provided in Table 1.

The size of the problem (no. of candidate schedules) grows exponentially with the planning horizon. However, frequent changes in the orders and operational disruptions make schedule planning beyond 48 hours highly speculative, and the emphasis is on the first 24 hours. Late customers, weather problems, equipment problems, air traffic control limitations, and other reasons combined with work rules contribute to schedule disruptions. Therefore, the scheduler focuses on the first 24 hours of the schedule (the next day) with attention to the start of the following day (where will the aircraft spend the night and what will it do the following morning?). The result is that the effective planning horizon is 24 to 36 hours.

We solve the model using the proprietary X-System.[11] These model runs were made on a 400 MHz Pentium II processor with 128 MB of RAM. The optimisation time is below a minute for up to 36 hours planning horizon and definitely acceptable for an interactive decision support system. All the runs were made with a suboptimally tolerance of 1%. Once an integer solution that is within 1% of the lower bound is identified, that solution is adopted and the process stops. For each planning horizon the total cost of the solutions in Table 1 are within 1% of each other, and thus are of the same quality. In these runs a relatively small value to the parameter **k** (the number of closest orders considered for appending to a candidate schedule) provides as good a solution as larger values. Although a larger value of **k** creates a richer set of candidate schedules it does not seem to contribute much to the quality of the solution.

The model described above is embedded in an interactive scheduling system where the model itself is transparent to the user. The user communicates with the system through a windows-based graphical user interface (GUI) that displays flights on an elaborate Gantt chart with full data drill-down and click-and-drag functionality. This is the same GUI that the schedulers used for manual scheduling. The system is connected to operational databases for data extracts and results upload.

The scheduling system that was outlined above has been in daily operational use for almost a year. It provides the scheduler with excellent quality schedules that constitute a solid basis for changes as the hours roll on. The system saves about three hours per scheduler per day (the time it took to assemble a schedule manually with the same GUI). Although the system was designed to be relaunched on

Table 1 Results

No. of aircraft	*No. of orders*	*Planning horizon (h)*	**k**	*No. of candidate schedules*	*Total cost*	*Run time (seconds)*[a]	
						Generation	*Optimisation*
47	50 + (2)	24	5	6446	5952	3	1
		24	7	7817	5952	3	2
		24	9	9066	5952	3	2
		24	11	10 337	5952	4	3
47	64 + (2)	36	5	41 626	7537	6	24
		36	7	56 688	7529	7	22
		36	9	75 285	7529	8	41
		36	11	97 553	7529	10	57
48	92 + (6)	48	5	194 712	10465	17	208
		48	7	251 767	10485	21	282
		48	9	288 799	10528	25	413
		48	11	319 400	10556	27	518

[a] On a 400 MHz Pentium II machine with 128 MB RAM.( ) = maintenance appointments.

request during the day (with the then current data) this has not been practiced yet. Changing schedules that have already been communicated to crews is time consuming, disruptive, and often cannot be done in a timely manner. Therefore once a schedule is adopted the scheduler tries to minimise changes. The model was designed to maintain schedule persistence through penalties on changing formerly assigned schedules[12] but this capability has not been experimented with yet by the schedulers.

Conclusions

We presented a system for scheduling charter aircraft that is being used daily to quickly create high quality schedules. At the core of the system is an Elastic Set Partitioning model that is fed a large set of alternate feasible schedules for each aircraft. These schedules are created by a schedule generator that encompasses the large variety of operational requirements and the associated costs. The fast solver for the ESP model facilitates quick turn around, and the user gets the recommended fleet schedule within a minute or two.

This application provides additional evidence that the fast development of computing power and solver algorithms facilitate the application of OR tools to solving practical resource allocation problems quickly and efficiently.

References

1 Clarke LW, Hane CA, Johnson EL and Nemhauser GL (1996). Maintenance and crew considerations in fleet assignment. *Transport Sci* **30**: 249–260.

2 Barnhart C *et al* (1998). Flight string models for aircraft fleeting and routing. *Transport Sci* **32**: 208–220.

3 Hoffman K and Padberg M (1993). Solving airline crew scheduling problems by branch-and-cut. *Mgmt Sci* **39**: 657–682.

4 Wark P, Holt J, Ronnqvist M and Ryan D (1997). Aircrew schedule generation using repeated matching. *Eur J Opl Res* **102**: 21–35.

5 Day PR and Ryan DM (1997). Flight attendant rostering for short-haul airline operations. *Opns Res* **45**: 649–661.

6 Beasley JE and Cao B (1998). A dynamic programming based algorithm for the crew scheduling problem. *Comp and Opns Res* **25**: 567–582.

7 Gopalan R and Talluri KT (1998). The aircraft maintenance routing problem. *Opns Res* **46**: 260–271.

8 Keskinocak P and Tayur S (1998). Scheduling of time-shared jet aircraft. *Transport Sci* **32**: 277–294.

9 Ferland JA and Michelon P (1988). The vehicle scheduling problem with multiple vehicle types. *J Opl Res Soc* **39**: 577–583.

10 Bausch DO, Brown GG and Ronen D (1998). Scheduling short-term marine transport of bulk products. *Maritime Policy and Mgmt* **25**: 335–348.

11 INSIGHT (1990). *The X- System: A Large-Scale Linear, Integer, and Nonlinear Optimisation System*. Alexandria, VA.

12 Brown GG, Dell RF and Wood RK (1997). Optimisation and persistence. *Interfaces* **27(5)**: 15–37.

Received July 1999;
accepted October 1999 after one revision