Rostering Air Traffic Controllers

Richard Conniss [1]

Department of Electronics, Computing and Mathematics
University of Derby
Kedleston Road, Derby, DE22 1GB, United Kingdom

Abstract

Many, if not most, real world scheduling problems fall into the class of NP. Classical, mathematically exact methods when applied to these problems often suffer from scaling issues, that prevent the computation of a solution in reasonable time. Similarly, real world problems can often be characterised by the requirement to dynamically change any prebuilt roster in reaction to unforeseen changes to the resources available, such as short notice staff absence or a change in the requirements of the task to be fulfilled. Metaheuristic methods have been successfully used to make approximations to optimal solutions which are good enough for practical use. In this paper an example of a novel and complex employee scheduling, or rostering, problem will be discussed and a simple metaheuristic method demonstrated to solve the problem.

Keywords: Rostering, Scheduling, Heuristics

1 Introduction

All personnel rostering problems necessarily concern themselves with rostering staff to tasks. Scheduling problems arrise in many industries and as demon-

[1] Email: `r.conniss@derby.ac.uk`

strated by [1], there exist many methodolgies to provide solutions. Recent developments, as surveyed by [2], have focussed on the use of qualifications, or skills, to assign individuals to tasks and the planning of breaks for staff.

ATC schedules have some similar conditions and requirements to other types of rostering problems in other industries. All staff must be allocated to tasks, based on their professional qualifications, to ultimately produce a viable roster to maintain operations. Staff must be given reasonable, or legally mandated rest breaks and time off for both leave and between shifts. The differences in how controllers work lead to an unusual set of conditions and constraints that make this an interesting area of study. Essentially, the important characteristics of the problem depend on two main components, those of qualifications and currency.

2 Problem Statement

Most ATC units have multiple control positions, each with unique demands and training requirements. Ideally, all controllers will eventually become endorsed in all positions and this is where the first main difference with other scheduling problems occurs. If a controller holds an endorsement in a position they are expected to be able to staff that task as required.

This is quite different to the use of qualifications in other rostering problems. As an example, in many nurse rostering problems qualifications denote the seniority of an employee. If a senior, or more qualified nurse is assigned to a task that would more normally be undertaken by a more junior college, often the assignment is penalised in some fashion by the solution method. For controllers, the need to maintain familiarity with all tasks for which they are qualified is safety critical. Skill fade is a significant problem and can induce potentially catastrophic effects on the safe movement of aircraft. Controllers salaries are excluded from rostering decisions as their remuneration has no effect on their ability to execute a task. Currency is defined as the number of days since a controller worked productively in a given position. The current limit is set at 30 days and if a controller were to violate this restriction they would have to undergo a period of retraining and re-qualify for that position. Like any employee controllers require rest breaks throughout the working day. In the civilian ATC world there are very strict and legally binding working rules and conditions to ensure the safety of aviation operations. The rules include maximum durations for a controller to work in a position (usually 2 hours) and frequency of rest breaks and meal breaks.

Given a set of controllers C with $c \in C$, a set of positions P with $p \in P$

and a set of qualifications Q containing tuples $< c, p, f >$ where f denotes a currency value in the range $\{1, 2, \ldots, 30\}$. The shift is divided into a set of fixed interval time slots T with $t \in T$, and a set of assignments A containing tuples $< c, p, t >$ is to be found that satisfies the above criteria.

3 Methodology

An appropriate linear programming model for a single day roster is required to understand the complexities involved. The day can be divided into T time slots of 15 minutes duration to simplify the model. With n controllers $i = 1 \ldots n$, m positions $j = 1 \ldots m$ and $t \in [0, T]$, the following matrices are defined.

$R_{ijt} = 1$, if controller i is in position j at time t
$Q_{ij} = 1$, if controller i is qualified in position j
$D_{jt} = 1$, if position j should be staffed at time t
$A_{ij} = 1$, if controller i is available to work at time t

This leads to the following set of hard constraints: A controller must be qualified to work in a position:

$$R_{i,j,t} \leq Q_{i,j}, \forall i, j, t \tag{1}$$

A controller must be available to work:

$$R_{i,j,t} \leq A_{i,t}, \forall i, j, t \tag{2}$$

If there is demand for a position it must be staffed:

$$\sum_i R_{i,j,t} \geq D_{i,j} \tag{3}$$

Each controller can only be assigned to a single position:

$$\sum_j R_{i,j,t} \leq 1 \tag{4}$$

Controllers cannot instantaneously switch tasks, and must have a break before being assigned to a position. This allows for a formal handover of responsibility as new controllers take on a task.

$$R_{i,j,t} - \sum_j R_{i,j,t} + 1 \geq R_{i,j,(t+1)} \tag{5}$$

A controller must be current in a position to work:

$$C_{i,j} \leq 30 \tag{6}$$

Additionally, controllers will require sufficient rest breaks during their shift.

$$\sum_{t=s}^{s+4} \sum_{j=1}^{m} R_{i,j,t} \leq 4 \tag{7}$$

An effective algorithm to produce valid rosters has to consider all of the above restrictions placed on controlling staff. During conversations with senior ATC staff at RAF Cranwell a number of key requirements for a daily roster have been identified. These are:

(i) All operational demand for the flying program must be met by qualified controllers.

(ii) Controllers must have suitable rest breaks.

(iii) The system must be able to adapt to sudden changes in staffing or the operational flying task

Given the above restrictions and requirements for successfully scheduling controllers, the proposed algorithm constructs a roster and satisfies the requirements of a particular day.The structure of the algorithm is similar to that of the Depth First Search (DFS) algorithm. DFS is a procedure for traversing every node in a given graph or tree, via their connecting vertices. Each level of the tree represents a combination of a position and time value. The first layer is the first position in the first time slot, the second is the second position and first time slot etc. Each node is the assignment of a controller at a particular position and time. Using currency to order the newly generated nodes is equivalent to expressing a preference to assign controllers into positions that they have not worked in for some time, it does not guarantee that the least current controller in a position is always assigned.

To examine the effect of different heuristic sorting methods on performance, four variants of the process were created.

dfs The basic version of the search. No ordering is applied to controllers currency or to the order of assignment of controllers to tasks.

dfsQ The set of assignments required are ordered by the number of controllers qualified for each task.

dfsC The set of controllers that are qualified for each position are ordered by descending currency value and presented for assignment in turn.

dfsQC The final variant is a mix of dfsQ and dfsC. The set of assignments are first ordered as in dfsQ and then controllers are presented in order of currency as in dfsC.

To compare the behaviour of each type of search a shared set of controller and task data was produced.

4 Results

n	20	19	18	17	16	15	14	13
dfs	1.13	19515.40	9441.07	5627.17	17322.13	11050.30	32078.30	187795.23
dfsC	2.43	19495	9447.63	5632.50	17276.27	11047.57	32074.03	187969.6
dfsQ	1.3	80.47	35.17	71.47	1076.13	593.60	884.5	7527.83
dfsQC	1.3	80.47	34.70	71.30	1080.03	593.07	884.40	7495.43

Table 1
Average solution time in ms for different numbers of controllers

Table 1 shows the average solution time in milliseconds for each set of 30 comparisons for n available controllers. The basic dfs approach is clearly the slowest to find solutions of all the variants, although it is still relatively quick to find a feasible solution. The dfsQC algorithm shows significantly better performance for all comparisons. One interesting feature to note is the trend in solution decreases from 20-17 controllers and then spikes at 16. The trend then re-emerges and continues to reduce.

n	20	19	18	17	16	15	14	13
dfs	1968.13	1989.27	1922.67	1887.03	1917.60	1898.03	1851.73	1860.37
dfsC	2408.80	2375.23	2364.87	2303.80	2069.00	2147.30	2122.43	2019.63
dfsQ	1968.13	1490.00	1922.67	1887.03	1917.60	1898.03	1851.73	1860.37
dfsQC	2408.80	2375.23	2364.87	2303.93	2069.00	2147.30	2122343	2019.63

Table 2
Average solution quality (larger is better) for different numbers of controllers

Table 2 shows the average solution quality for each set of comparisons. The solution quality is simply the sum of the currencies of controllers for each assignment. Larger values suggest more high currency controllers have been assigned and therefore will have the opportunity to reset their currency. This has the effect of preventing all controllers form going out of currency over time. It is interesting to note that reductions in solution time are related to improvements in solution quality. Due to the structure of the implementation of the algorithm, it is relatively simple to combine both the above sorting methods simultaneously. Which, as demonstrated, leads to an improvement in both speed and quality of generated rosters.

References

[1] Van den Bergh, Jorne, Jeroen Belin, Philippe De Bruecker, Erik Demeulemeester, and Liesje De Boeck, *Personnel scheduling: A literature review*, European Journal of Operational Research 226, no. 3 (2013), 367–385.

[2] De Bruecker, P., Van den Bergh, J., Belin, J., and Demeulemeester, *Workforce planning incorporating skills: State of the art*, European Journal of Operational Research, 243(1), 1–16.