10th International Conference of the Practice and Theory of Automated Timetabling
PATAT 2014, 26-29 August 2014, York, United Kingdom

Scheduling Air Traffic Controllers

R.Conniss · T.Curtois · S.Petrovic · E.Burke

Keywords Personnel Scheduling, Rostering, Heuristics

1 Introduction

The effective rostering of Air Traffic Controllers is a complex and under researched area of the personnel scheduling literature. An effective method to produce real world rosters for controllers requires the ability to model shifts, breaks, multiple tasks and their associated qualifications, to rotate staff through all the tasks for which they are qualified to maintain skill levels, the requirement to train staff whilst continuing normal operations and an ability to reroster in the event of unexpected events. Examples in the literature that examine some of these components include shift scheduling [5], break planning [2] [6] and multi skilled staff [4] [3].

We shall present an algorithm that can effectively model many of the features of the ATC rostering problem, and produce useful real world rosters for operational use.

R.Conniss
OMIS, NUBS, University of Nottingham, UK
E-mail: psxrc@nottingham.ac.uk

T.Curtois
ASAP Research Group, School of Computer Science, University of Nottingham, UK
E-mail: tim.curtois@nottingham.ac.uk

S.Petrovic
OMIS, NUBS, University of Nottingham, UK
E-mail: Sanja.Petrovic@nottingham.ac.uk

E.Burke
University of Stirling E-mail: e.k.burke@stir.ac.uk

2 Problem Description

Creating a feasible roster for controllers depends on similar hard constraints to other classes of rostering problem. Controllers have limits on their maximum shift length, how long they can work without a break during a given shift and minimum time off between shifts. Although the initial focus of the research was on RAF controllers, the working limits of civilian controllers have been used in the model for two reasons. Firstly, military controllers are not subject to well defined working time rules, due to the nature of military service. Secondly, the Civil Aviation Authority publishes UK wide rules for all civilian controllers in the Scheme for the Regulation of Air Traffic Controller's Hours (SRATCOH) [1].

For a controller to be assigned to a console position, or task, the controller must hold the correct qualification to staff the task. Each controller will hold some subset of all available qualifications and some will be fully qualified for all tasks. Qualifications are obtained by completing on the job (OJT) training under the supervision of an instructor. Whilst this is similar to the use and acquisition of qualifications in other industries, such as Nurse Rostering, the difference here is that controllers must maintain familiarity, or currency, in all the positions for which they are qualified. Currency is measured in days since a controller last worked in a particular position. This restriction requires that controllers must regularly work in all relevant positions, failure to do so results in the suspension of that qualification and leads to an enforced period of retraining, with a final competency check, before the qualification can be reactivated. Once a controller has worked productively in a position i.e. they have controlled a reasonable number of aircraft, the currency value is reset to zero. The objective function for the model is therefore to maximise the sum of currency for all assignments of a controller to a position.

Maintaining currency for all controllers is the central goal for any useful rostering approach. Other considerations such as minimising the cost of staffing a shift can be ignored, as staff are salaried by rank or grade and minimum staffing levels for each unit are determined by the relevant regulating authority. This means that the objective of the solution is to prevent controllers from violating currency limits, over time. This can be best achieved by forcing controllers to vary the positions worked as often as is reasonable.

Our model divides time into 30 minute blocks, and a controller should not work for more than 2 hours, followed by at least a 30 minute break as defined by SRATCOH. Staff can not directly swap positions because for each change of controller in a position a handover briefing is required, which requires that the relieving controller must have been unassigned in the previous time block. Although it would be valid to have controller A work for 2 hours, be relived by controller B for 30 minutes and for A to return to their original position, this increases the number of staff on a given shift . Ideally, B should stay in the position and A should be assigned another task, to maintain their currency.

3 Solution Approach

To produce valid rosters in reasonable time, a constructive, greedy heuristic algorithm has been developed, to meet the above requirements. The process begins with a set of input parameters based on controller availability, qualifications, demand for tasks and currency values. Every assignment in the model is defined as a tuple containing values for controller, position and time (c, p, t).

The algorithm is an iterative procedure which selects the first assignment from a sorted list of valid assignments at each step and appends this to the solution. If the list is empty the algorithm discards the last assignment to the solution, backtracks and restarts. The solution is a list of assignments, and it's index relates to the position and time slot that requires staffing i.e. solution(1) is for position(1), time slot(1), solution(2) is for position(2), time slot(1) etc.

The list of valid assignments is first sorted by the currency value of controllers, for the required position at each index of the solution, to maximise total currency for the roster. However, this ordering alone can cause problems in the final roster. Consider the situation where controller A is the least current in position APP. She is assigned to APP in the first 4 time slots and replaced by controller B in the 5th. The ordering by currency of valid assignments would mean that A would be reassigned to APP in the 6th time slot, to maximise currency. This is a poor approach, as B is not given enough time to reset his currency and would require a larger pool of available controllers to satisfy the staffing requirements. An additional preference rule is applied to the list that attempts to keep a controller in a position for as long as possible, subject to break rules, and reduces the possibility of having 2 controller changes in a position in consecutive time slots.

4 Experimental Results

A series of 240 single day rosters have so far been produced, with progressively more constrained starting conditions (fewer available controllers) and arbitrarily limited to a 60 minute time limit for a solution. Th algorithm is written in Python 2.7 and run on a consumer grade i7 desktop PC. The final 30 experiments with the fewest number of least qualified controllers (14 controllers to 11 positions) managed to produce valid rosters in an average time of 26 minutes. Table 1 shows an example of the algorithm output with controllers {A, B, ..., Z} assigned to positions {SUP, APP, ..., BKN SRA} for each timeslot {1, 2, ..., 18}. The score of 3,557 is the total sum of currency for all assignments, with a theoretical maximum value of 5,940. This upper value could never be achieved as it would require no controller to have worked in any position for 30 days, which is completely unrealistic.

To take controller M as an example, table 1 shows them assigned to four different positions for the day, therefore maximising the chance that they reset their currency in those positions. There are several other examples of con-

Table 1 Example roster: Controllers assigned to positions for given time slot. Currency score = 3557

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUP	V	V	V	V	T	T	T	T	V	V	V	V	T	T	T	T	V	V
APP	S	S	S	X	X	X	K	K	K	K	Y	Y	Y	Y	S	S	S	S
CWL DIR	T	T	T	O	O	O	O	X	X	X	X	K	K	K	K	Y	Y	Y
BKN DIR	O	O	Y	Y	Y	H	H	H	H	S	S	S	S	O	O	O	O	X
CWL DEPS	H	H	H	H	S	S	S	S	O	O	O	O	X	X	X	X	Z	Z
BKH DEPS	X	X	Z	Z	Z	Z	Y	Y	Y	T	T	Z	Z	Z	Z	H	H	H
ADC	M	M	B	B	B	B	E	E	E	E	H	H	H	H	E	E	E	E
GND	Y	E	E	E	E	W	W	W	W	Q	Q	Q	Q	M	M	M	M	T
PAR	W	W	W	W	Q	Q	Q	Q	M	M	M	M	B	B	B	B	K	K
CWL SRA	Q	Q	Q	M	M	M	M	B	B	B	B	E	E	V	V	W	W	W
BKN SRA	B	K	K	K	K	V	V	Z	Z	Z	W	W	W	W	Q	Q	Q	Q

trollers assigned to multiple positions in this roster, implying that the solution is of sufficient quality for use.

5 Future Work

There are several obvious extensions to the model that will require further investigation. Including training plans for new controllers is obvious next step. Other possible areas for research include:

- Allowing for breaks of different length e.g. lunch hours.
- Encoding personal preference for tasks/shifts.
- Embedding fairness into the rosters i.e evenly distributing time assigned to positions.
- Producing extended days rosters with overlapping shifts.
- Generalising the model for use in other industries e.g. airport security.

References

1. Authority, C.A.: Cap 670: Air traffic services safety requirements (2003)
2. Di Gaspero, L., Gärtner, J., Musliu, N.: A hybrid LS-CP solver for the shifts and breaks design problem. Hybrid . . . pp. 46–61 (2010). URL http://link.springer.com/chapter/10.1007/978-3-642-16054-7_4
3. Li, H., Womer, K.: Scheduling projects with multi-skilled personnel by a hybrid MILP/CP benders decomposition algorithm. Journal of Scheduling **12**(3), 281–298 (2008). DOI 10.1007/s10951-008-0079-3. URL http://link.springer.com/10.1007/s10951-008-0079-3
4. Quimper, C.G., Rousseau, L.M.: A large neighbourhood search approach to the multi-activity shift scheduling problem. Journal of Heuristics **16**(3), 373–392 (2009). DOI 10.1007/s10732-009-9106-6. URL http://link.springer.com/10.1007/s10732-009-9106-6
5. Rekik, M., Cordeau, J., Soumis, F.: Implicit shift scheduling with multiple breaks and work stretch duration restrictions. Journal of scheduling (2010). URL http://link.springer.com/article/10.1007/s10951-009-0114-z
6. Widl, M., Musliu, N.: An improved memetic algorithm for break scheduling. Hybrid Metaheuristics pp. 133–147 (2010). URL http://link.springer.com/chapter/10.1007/978-3-642-16054-7_10