A meta-heuristic approach to aircraft departure scheduling at London Heathrow airport.

Jason A. D. Atkin[1], Edmund K. Burke[1], John S. Greenwood[2], and Dale Reeson[3]

[1] School of Computer Science and Information Technology, University Of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, NG8 2BB
`{jaa,ekb}@cs.nott.ac.uk`
[2] Analysis & Research, National Air Traffic Services Ltd, Spectrum House, Gatwick, West Sussex, RH6 OLG
[3] National Air Traffic Services, Heathrow Airport, Hounslow, Middlesex, TW6 1JJ

Abstract: London Heathrow airport is one of the busiest airports in the world. Moreover, it is unusual among the world's leading airports in that it only has two runways. At many airports the runway throughput is the bottleneck to the departure process and as such it is vital to schedule departures effectively and efficiently. For reasons of safety, separations need to be enforced between departing aircraft. The minimum separation between any pair of departing aircraft is determined not only by those aircraft but also by the flight paths and speeds of aircraft that have previously departed. Departures from London Heathrow are subject to physical constraints that are not usually modelled in departure runway scheduling models. There are many constraints which impact upon the orders of aircraft that are possible and we will show how these constraints either have already been included in the model we present or can be included in future. The runway controllers are responsible for the sequencing of the aircraft for the departure runway. This is currently carried out manually. In this paper we propose a metaheuristic-based solution for determining good sequences of aircraft in order to aid the runway controller in this difficult and demanding task. Finally some results are given to show the effectiveness of this system and we evaluate those results against manually produced real world schedules.

1 Introduction

London Heathrow airport is a busy two-runway airport which, due to its popularity with both airlines and passengers, suffers severe aircraft congestion at certain times. Traffic in airports is not evenly spread, for obvious reasons which pertain to airline and passenger preferences. There are inevitably times when the departures process is congested but the arrivals are sparse, vice versa, and times when both are congested. London Heathrow airport is actually situated on an extremely small plot of land in comparison both to how busy the airport is and to other airports around the world.

The airport capacity problem is concerned with estimating the capacity of an airport in terms of arrivals and departures. It has been examined for a number of years. Newell [14] provided a model and showed that the capacity of the airport is increased when arrivals and departures can be alternated on both runways. Although mixed mode, where arrivals and departures are intermixed on a runway, is preferable for increasing the throughput, this is not currently possible at Heathrow due to the proximity of the surrounding residences, although it may begin to be considered for peak times.

The departure flow at Logan airport was analysed in [11] and [12] and Logan airport was compared to other major airports. Runway scheduling was seen to be a bottleneck upon the departure process and the authors concluded that it is vital to increase the throughput of the departure runway.

There are some similarities between the arrival and departure processes for the runways at an airport. Both processes are subject to sequence-dependent separation times between aircraft. Previous research has looked at the arrivals problem with the goal being to order arriving aircraft for a single runway so as to either minimise the total completion time or to minimise the total deviation from an ideal arrival time for each aircraft. Mixed integer zero-one formulations were presented in [6] and Genetic Algorithms were shown to be effective in [7].

Abela et al [1] looked at the arrivals problem for a set of aircraft with landing time windows. They presented a genetic algorithm to give an approximate solution and branch and bound algorithm for solving the problem when formulated as a 0-1 mixed integer programming problem to give an exact solution.

A heuristic approach for an upper bound and a branch and bound algorithm for the arrivals problem were given in [10]. A network simplex method was used to assign arrival times given any partial ordering of aircraft.

The arrivals problem, as it is presented in the literature, however, does not address the major constraints upon the departures problem at London Heathrow airport.

A constraint satisfaction based model for the departure problem was presented in [13] for solution by ILOG Solver and Scheduler. A fifteen minute time slot was assigned to each aircraft and separations were assigned based upon the size and speed of the aircraft and upon the exit point that the departing aircraft were going to use.

The departure process was analysed and a departure planner proposed by Anagnostakis et. al. in [3], [4] and [5]. A search tree was described and branch and bound techniques or an A* algorithm were recommended for solving the departure problem in [2]. A dynamic program was suggested in [15] to solve the departure order problem by limiting the possible number of aircraft that are considered for any place in the schedule, reducing the search space dramatically.

If only considering separations between adjacent aircraft and ignoring the physical constraints from the holding points, the departure problem can be seen to be a variant of the single machine job sequencing problem where jobs have sequence-dependent processing or set-up times. Substantial research has been undertaken into this problem. For example Bianco et. al. [8] looked at the generalised problem with release dates as well as sequence-dependent processing times, showing the equivalency to the cumulative asymmetric travelling salesman problem with release dates. To ensure safety in the departure process, however, it is not possible to only consider adjacent pairs of aircraft and it is easy to produce schedules where all adjacent pairs have the required separations but other aircraft pairs do not.

Craig et. al. [9] did look at the effects of one holding point structure and gave a dynamic programming solution for scheduling take-offs. In practice, however, the holding point structures are more flexible than the one described here and a more general solution needs to be developed.

There are important constraints at London Heathrow airport that are not normally considered in the departure problem as it is presented in the current scientific literature. These are identified in the problem description below.

2 Problem description

The objective of this paper is to increase the throughput of the departure runway subject to various constraints, with safety being paramount.

There are currently only two runways in normal use at Heathrow, however, if environmental targets are met, there may be a possibility to add a third, parallel runway in the future. At any time of the day only one runway can currently be used for departures.

The direction of the wind determines the direction in which the runways are used. The runways are labelled according to the direction in which they are used and whether they are on the right or the left when facing that direction. The four runway configurations have been labelled in Fig. 1. For example when arriving or departing heading west, the northern runway is referred to as 27R as it has a direction of 270 degrees and is the runway on the right.

There is actually a third runway already but this can only ever be used for arrivals. It is shorter than the other two and not long enough for many Heathrow departures. It is used no more than twice per year. It also intersects both of the other runways so it is not practical to use it if either of the other two runways is in use, indeed it is usually used as a taxiway.

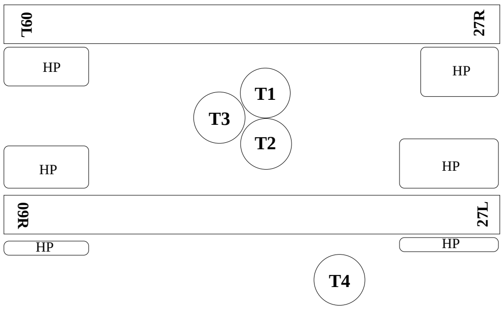

Fig. 1. The layout of London Heathrow Airport

There are currently four terminals at London Heathrow, labelled T1 to 4 in Fig. 1. Three terminals are situated between the runways but the fourth is to the south of the southern runway.

When a flight is ready to depart a delivery controller has to give permission for engine start up. A ground controller then instructs the pilot in order to control the movement of the aircraft around the taxiways. Once an aircraft approaches the runway end and is no longer in conflict with any other aircraft the ground controller will relinquish control of the aircraft to the runway controller.

In this paper, we are concerned only with the operations of the runway controller. We assume that the ground controller and delivery controller are currently outside of the system and merely feed aircraft into the start of the system. Later research will look to include these roles into the model.

There are holding points, labelled HP in Fig. 1 at each end of each of the runways, and both north and south of the southern runway. Within these physical holding point structures the runway controller can reorder the aircraft before they reach the runway.

2.1 Holding point constraints

Aircraft go through holding points to get to the runways. Holding points can be considered to be one or more entrance queues to some maneuvering space then finally to a single take-off order on the runway. Where there are different entrance queues available, the ground controller will usually send an aircraft into the most convenient queue. The runway controller can request aircraft to be sent to specific queues but in practice, as the runway controller is very busy with the aircraft already in the holding points, there is rarely sufficient time to also consider the aircraft the ground controller has.

As mentioned before, Heathrow has very limited space so the holding point and taxi space is limited. Given the initial order of aircraft in the input queues to the holding points the runway controller has to decide how to sequence the take-offs in order to maximise the throughput at the runway. This can be a very difficult task at times.

Only limited amounts of reordering are possible at these holding points. The configuration of the holding points varies greatly between runway ends and will determine what reordering operations can take place and the costs involved in each operation.

2.2 Minimum separations

To ensure safety, minimum separation times are imposed between aircraft taking off. The order of the aircraft for take-off can make a significant difference to the total delay that needs to be imposed upon the aircraft.

The minimum separation between aircraft is determined by:

- Wake Vortex. Large aircraft leave a stronger wake vortex than smaller/lighter aircraft and are also less affected by wake vortex. Every aircraft has a weight category and the wake vortex separation for any pair of aircraft can be determined by comparing their weight categories.
- Departure Routes. Aircraft will usually have a Standard Instrument Departure (SID) route assigned to them, giving a pilot a known departure route to follow. The relative SID routes of any two aircraft will impose a minimum departure interval between them. This ensures that safe minimum separation distances are kept while in flight. At times of congestion in the airspace a larger than normal separation may be required between certain SID routes, in order to increase the separation between flights heading into the congestion. These separations differ depending upon the runway in use at the time.
- Speed Group. The relative flight speeds of the aircraft can also make a difference to the separations which must be imposed upon aircraft flying the same or similar routes. The relative speed groups of the two aircraft modify the separation required for the relative SID routes. If the following aircraft will close the distance then a larger initial separation is necessary. Conversely if the following aircraft is slower then a lower separation can sometimes be applied.

The runway controller will aim for minimum separations between aircraft wherever possible. It should be noted here that a controller has some discretion as far as some separations are concerned. In particular some of the SID route based separations can be reduced in good visibility.

2.3 Other constraints

Calculated Time of Take-off (CTOT) is the name given to the fifteen minute take-off time slot that is assigned to some aircraft in order to avoid congestion en-route and at busy destination airports. It is important that such aircraft take off within this window. For the results in this paper we have no CTOT information so we assume no CTOT limitations.

The departure process is a dynamic system where aircraft are added to and removed from the system over time. The runway controller will have only limited knowledge about the aircraft that are not currently at the holding points.

The runway controller has a lot of information that is very hard to capture as hard data. In many cases a controller will be weighing the effects of contradictory constraints such as maximising throughput while minimising overtaking, to ensure fairness and minimising maneuvering, to reduce workload.

2.4 Overall objective

The objective is to find candidate solutions for which the runway throughput is maximised and all constraints are met. We were told by one air traffic controller that the best figure obtained for Heathrow was 54 aircraft in an hour and that this figure is so good that it is extremely unusual.

3 Model description

In this model we aim to maximise the throughput of the runway by minimising the total delay, D, suffered by the aircraft at the holding points. Let h_i be the arrival time for aircraft i at the holding point, where i is an integer ≥ 1. The integer i represents the position of the aircraft in the take-off order. If d_i is the take-off time for aircraft i from the runway, then we can calculate the total delay at the holding points using equation 1 where n is the total number of aircraft departing.

Call e'_i the earliest possible take-off time for aircraft i such that all required separations from earlier aircraft take-offs are maintained and e''_i the earliest time at which aircraft i could physically taxi to the departure runway.

The earliest take-off time, e_i, for aircraft i cannot be any earlier than either the physical taxi time or the separations require, equation 2.

For the model we assume that all aircraft take off at the earliest possible time, so the actual take-off time, d_i, is equal to the earliest possible take-off time, e_i, equation 3.

We define a function $S(j,i)$ to give the minimum separation necessary between leading aircraft j and (not necessarily immediately) following aircraft i to meet all separation requirements. Function $S(j,i)$ incorporates all separation rules for weight classes, SID routes and speed groups. Then e'_i, the earliest take-off time for which all separations are maintained, can be calculated, equation 4.

Function $S(j,i)$ can be decomposed into two parts, a function $W(w_j,w_i)$ which will calculate the required wake vortex separation from the weight categories w_i and w_j of aircraft i and j and a function $R(r_j,s_j,r_i,s_i)$ which will calculate the required separation based upon the SID routes, r_i and r_j, and the speed groups, s_i and s_j, of the aircraft i and j, equation 5.

Both functions $W(w_j,w_i)$ and $R(r_j,s_j,r_i,s_i)$ are defined to return standard separation values in accordance with current regulations. It should be noted that the runway controller has some flexibility in good weather to reduce the separations given by $R(r_j,s_j,r_i,s_i)$ and a fully operational decision support system would allow the controller to do just that.

If we assign each aircraft a route through the holding point structure then, given a holding point entry time, h_i, and a suitable function, $T(t_i)$, for the traversal time through the holding points along a traversal route t_i for aircraft i, the earliest time the aircraft can reach the runway, e''_i, can be calculated, equation 6.

It should be noted at this point that the separations for SID routes differ depending on which runway the aircraft are departing from. This means that both $T(t_i)$ and $R(r_j,s_j,r_i,s_i)$ are runway specific.

3.1 Formal description of the mathematical model

We can express this model as follows:

Minimize

$$D = \sum_{i=1}^{n}(d_i - h_i) \tag{1}$$

Subject to

$$e_i = \max(e'_i, e''_i) \tag{2}$$

$$d_i = e_i \tag{3}$$

$$e'_i = \max_{j=1..(i-1)}(d_j + S(j,i)) \tag{4}$$

$$S(j,i) = \max(W(w_j,w_i), R(r_j,s_j,r_i,s_i)) \tag{5}$$

$$e''_i = h_i + T(t_i) \tag{6}$$

3.2 Holding point constraints

Any practical model must incorporate the holding point constraints. There is no point in presenting candidate solutions to a runway controller if the controller cannot actually achieve the order due to the physical constraints.

An example of a holding point structure can be seen in Fig. 2. The nodes are the valid positions for aircraft and the arcs show moves that aircraft could make. This network is more restrictive than the actual network at the associated holding point at Heathrow and is deliberately so. Any solution which is feasible for this network should be both feasible and sensible for the real network.

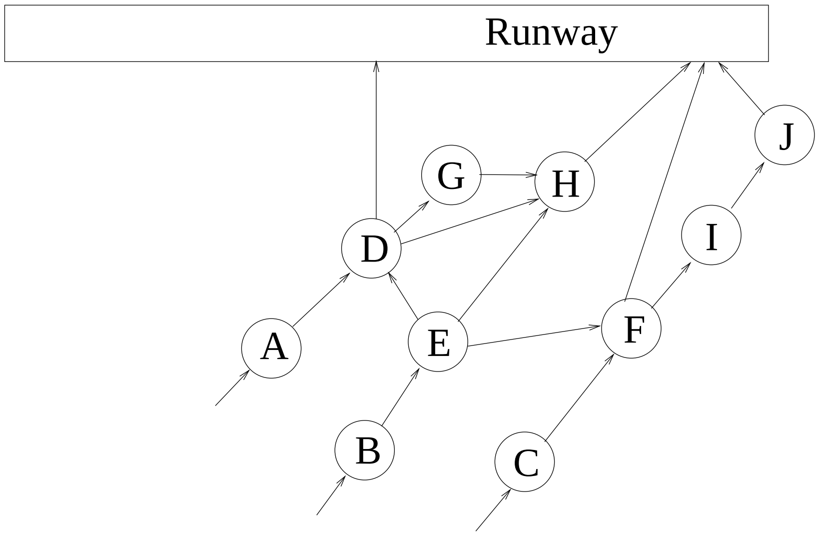

Fig. 2. An example holding point network structure.

We will be investigating metaheuristic local search, as specified in section 4. This means that the search will move from one solution to the next. A solution could consist of just a final take-off order or it could give details about all of the taxi movement within the holding points and a take-off order could be derived from this.

Modelling the path of each aircraft through the holding point structure as a part of the solutions would make the search space extremely large. However all final schedules would be known to be achievable within the limitations of the holding point structure. Many solutions would give the same take-off order but different routes through the holding points. Some routes take longer to traverse than others, so some solutions will be much better than others that have the same take-off order.

Rather than modelling the movement within the holding points, the selected model instead looks at solutions which specify only a take-off order. Not all orders of take-off will be achievable however. Given an order for take-off, routes through the holding point structure are assigned heuristically to aircraft so that aircraft which overtake are assigned faster routes and aircraft which are overtaken are assigned slower routes. A feasibility check is then performed afterwards to verify that the solution is achievable, given the holding point structure.

The feasibility of the schedule is checked by feeding aircraft into the start nodes and testing that it is possible for them to exit in the correct order at the runway. Pre-processing of the nodes based upon the take-off order provides knowledge about whether any aircraft can move without blocking another aircraft, so this check can be made deterministically.

4 Departure Scheduling Algorithms

4.1 The Basic Search Algorithm

All of the search heuristics that we investigated had the same basic format but differed in the details. The full algorithm for the basic search is as follows:

1. Obtain initial solution. An initial solution will usually be a solution where the aircraft are in the order at which they arrived at the holding points. This solution has the advantage that it will always be feasible as no re-ordering is necessary within the holding points.
2. Heuristically assign holding point routes to each aircraft.
3. Check the feasibility at the holding point structure to ensure that the order of take-off is possible.
4. Evaluate the cost of the solution.
5. Accept or reject the candidate solution. This is the main place in which the metaheuristic searches differ. If the solution is accepted then it becomes the new current solution.
6. If the given number of evaluations have been completed then stop the algorithm and report the best result so far. Otherwise select a solution is the neighbourhood of the current solution and return to step 2.

4.2 Search Algorithms

The following local search approaches are described in this paper:

First descent

The first descent algorithm is the most simplistic algorithm. Each new solution is accepted only if it is better than the current solution.

Steeper descent

The steeper descent algorithm selects fifty candidate solutions at a time. Each candidate is evaluated and the best of the feasible candidates is adopted. The best candidate is adopted even if it is worse than the current solution, which means this is more than a strict descent algorithm. This gives the algorithm a limited ability to move out of local optima but no method to avoid it moving straight back to the local optima it just left.

Evaluations of candidates are expensive so the searches are limited to a number of evaluations rather than a number of iterations. This means that the first descent algorithm runs for fifty times as many iterations as the steeper descent algorithm.

Tabu search

The tabu search algorithm is similar to the steeper descent algorithm except that it maintains a list of tabu moves. When a move is made, the reverse move is added to the tabu list to ensure that the search does not go back to where it came from. The reverse move that is recorded will stop any move which would put all of the aircraft that moved back into the absolute positions they previously occupied.

Simulated Annealing

The simulated annealing algorithm is similar in structure to the first descent algorithm. It will, however, sometimes accept moves to worse solutions. If the cost of the new solution is less than the cost of the current solution then the new solution will always be accepted. If the cost of the new solution is more than the cost of the current solution then there is a small chance to still accept the new solution.

Let D_{curr} be the cost of the current solution and D_{cand} be the cost of the candidate solution.

The candidate solution will be accepted if:

$$D_{cand} < D_{curr} \tag{7}$$

or

$$R < e^{-\delta/T} \tag{8}$$

where $\delta = D_{cand} - D_{curr}$ is the difference between the current and candidate solutions, R represents a uniform random variable in the range [0..1] and T is a temperature which is initially large but decreases over time.

4.3 Neighbourhood design

The following moves were investigated.

Swap single aircraft

The swap single aircraft move takes two aircraft from the schedule and swaps the positions of the aircraft in the final take-off order.

Shift aircraft

The shift multiple aircraft move selects a set of one or more aircraft that are currently scheduled for consecutive take-offs and moves them forwards or backwards in the schedule, shifting the other aircraft they are moved past forwards or backwards to make room.

Randomise a set of aircraft

The randomise a set of aircraft move selects a consecutive set of aircraft as the target. Each aircraft within this set is then moved to a random position in the set. This move may emulate a shift, swap or a reversal in the order in some cases but some of the schedules attainable through this move are not attainable otherwise. In experimental results this move has shown a valuable contribution in finding good schedules, when not overused.

4.4 Objective function

It is advisable to limit the amount of deviation from the holding point arrival order as well as to limit the delay. Reducing the number of 'swaps' of aircraft in the take-off order will aid in reducing workload for the pilots and controllers and it will also make it easier for the next iteration to build a feasible schedule.

With this goal in mind, the following objective function is used by the search algorithms:

$$D = \sum_{i=1}^{n} (A_i - i)^2 + 5 \sum_{i=1}^{n} (d_i - h_i) \tag{9}$$

Where n is the number of aircraft in the take-off schedule, d_i is the take-off time and h_i is the holding point arrival time of the ith aircraft in the take-off queue. A_i is the position, 1, 2 ... n, in the initial holding point arrival order, of the ith aircraft in the take-off queue.

5 Results

5.1 Input data

Historical recorded data was used for the evaluation. Three datasets were used with different numbers of aircraft (123, 189 and 299 respectively).

The most convenient holding point entrance for the allocated stand was assigned to each aircraft. There were no CTOT restrictions so only simple holding point traversal routes were necessary. The real holding point arrival times from the historic data were used. In a real system, precise arrival times would not be known until the aircraft actually arrived at the holding points and estimated arrival times would have to be used until then.

Recorded data shows that it takes a minimum of just over a minute for an aircraft to traverse the holding point structure and get airborne but this time can vary widely. For this paper all holding point traversal times were assumed to be equal and independent of the route taken, as only good routes were used. Two values for this time were tested, one and two minutes. A traversal time of one minute has the advantage of allowing aircraft to arrive, enter the runway and take-off very quickly, which is what often happens in practice at quiet periods. A two-minute traversal time, although no longer allowing fast entry at times when this is possible, seems better suited for the model in many ways as it can be assumed to account for some of the uncertainty in arrival time or traversal time in real life.

It is important to attempt to automate the system, so that it can be tested in an objective rather than subjective manner, even though this is not how it would be used in practice. In a real system not all suggested reorderings will be accepted, as the controller has a number of other objectives to keep in mind. Here we are assuming that the metaheuristic order will always be accepted.

A two-stage test was used and the final resulting schedule was examined.

5.2 Stage 1 - Initial schedule

Starting from an initial case where there are no aircraft in the system an initial schedule is built.

1. Add the first 20 aircraft.
2. Run the search algorithms for 10000 evaluations. Keep the best result found.
3. Fix the take-off order, take-off time and taxi routes of the first 5 aircraft to take off. Taxi routes for aircraft overtaken by these aircraft were also fixed.
4. Add the next 5 aircraft to the system.
5. Run the algorithms for 5000 evaluations. Keep the best result found.

5.3 Stage 2 - Repeated rescheduling

This is the stage that more closely emulates what will happen in practice, with some aircraft having take-off slots or taxi routes already assigned. Each iteration takes between 0.4 and 0.8 seconds.

1. Fix the take-off order, take-off times and taxi routes of the first 10 aircraft to take off. Again this also fixes the taxi routes of all aircraft they overtake.
2. Add the next aircraft to the system.
3. Remove the first aircraft from the system.

4. Run the search algorithms for 5000 evaluations. Keep the best result.
5. If there are no more aircraft to add then stop, otherwise return to step 1.

As aircraft are removed from the system the take-off order is recorded and at the end, the combined schedule of all of the departures is built and evaluated.

5.4 Total delay on aircraft

The test schedule was executed ten times for each of the search approaches, on each set of data, for both one and two minute holding point traversal times. The mean values of the total delay in seconds for the ten runs are shown in the tables below. The best figures are presented in bold.

Table 1. Comparison of mean delays - 1 minute traversal time

Metaheuristic	Dataset 1	Dataset 2	Dataset 3
Manual schedule	55140	136168	103692
First Descent	23548	49966	51438
Steeper Descent	**23511**	49158	50977
Simulated Annealing	**23511**	**48613**	50788
Tabu Search	23516	48767	**50661**

Table 2. Comparison of mean delays - 2 minute traversal time

Metaheuristic	Dataset 1	Dataset 2	Dataset 3
Manual	62244	142828	121632
First Descent	**30831**	59170	69377
Steeper Descent	**30831**	58275	68916
Simulated Annealing	**30831**	57815	68728
Tabu Search	**30831**	**57504**	**68601**

5.5 Evaluation of the results

The metaheuristic solutions provide much lower total delays than the manual solution and this provides significant evidence for the high value of such approaches. However, there are a number of reasons why our automated solutions are so superior (in terms of delay). In fact, the manual solutions are very good, with very few separations above the minimum.

1. This is a multi-objective problem and minimising delay only looks at one objective. Many conflicting objectives need to be satisfied and this is one reason why an automated solution can only ever be advisory.
2. Maximising throughput is not the same as minimising delay. The controller is trying to maximise throughput not minimise total delay. Minimising delay will have the effect of moving larger separations as late as possible in the schedule. Minimising the delay will maximise the throughput but the converse is not true. For example assume a six minute period with only three aircraft available to take off. Two minute separations would give the same throughput as one minute separations but a lot larger delay. Where larger separations will be necessary, a runway controller may sometimes wish to have them earlier to avoid delaying aircraft which take advantage of these to cross the runway.
3. Good orders suggested by the metaheuristics may have been impossible due to constraints not currently modelled, such as CTOT limitations on aircraft.
4. Taxi times are not actually identical or predictable. We have no way of knowing whether certain aircraft were exceptionally slow or fast in practice.
5. The metaheuristics have more knowledge about the future than the runway controller did. Sometimes a good order from the metaheuristics has been a result of knowing which aircraft are going to be arriving later. Reducing the load on the runway controller via an advisory system should allow a runway controller to take account of these later arrivals themselves; something they do not currently have the time to do.

Minimising the delay is a good way to try to ensure maximal throughput of the runway as it makes it easier to reschedule as new aircraft enter the system.

The fact that the metaheuristics give better delays than the manual solution means that they hold significant promise for forming the basis of an advisory system. By reducing the work load of the runway controller and allowing more aircraft to be considered than are currently in the holding point structure it should be possible to reduce the delay and increase throughput in practice.

Dataset 1 was from a less busy time of the day than the other two datasets. There were less possibilities to reorder aircraft as there were less aircraft in the holding points at any time. All but the first descent metaheuristic found the same good schedule for the aircraft in this dataset, the mean values of 23511 and 30831 were also the minimum values found for this dataset, by any of the algorithms. The tabu search failed on one execution to find this good schedule hence the slightly higher mean for the tabu search with one minute traversal time.

Datasets 2 and 3 were from busier times of the day. For both traversal times, for both datasets 2 and 3, student t-tests showed that tabu search performed significantly better than the steeper descent algorithm and that both simulated annealing and tabu search performed significantly better than the first descent algorithm, with a confidence level of 99% in each case.

The simulated annealing algorithm gave good results across the datasets. It got the best results for dataset 2 on table 1 and equal best on dataset 1 for both tables. Student t-tests performed on the results, however, failed to show a significance in the difference between the results for simulated annealing and tabu search, for either of the traversal times for dataset 2, despite the difference in the mean values of the results.

With ten executions of the algorithms on each dataset for each traversal time, there are forty executions that can be compared for these datasets. Tabu search gave better results that the steeper descent algorithm on 39 of the executions and the same result on the other execution. The only difference between the two approaches is the presence of the tabu list so we conclude that the tabu list is contributing to the success of the search.

Tabu search produced the best result for dataset 3 on table 1 and the best results for all three datasets on table 2, although all of the automated results got equal best results for dataset 1. Student t-tests showed that tabu search performed significantly better than simulated annealing for both traversal times for dataset 3, with a confidence level of 99%.

6 Conclusions

The departure problem is a complicated one due to the many constraints upon the schedule and the sequence-dependent separations between aircraft. Most of the existing research has looked at the arrivals problem rather than the departure problem and it is common to check only the separations between adjacent aircraft. However, it is not sufficient merely to look at adjacent pairs of aircraft for the departure problem as a schedule that provides safe separations for all adjacent pairs of aircraft will not necessarily provide safe separations for other aircraft pairs.

Many different techniques have previously been applied to this problem yet none account for the physical constraints upon reordering that exist at an airport like London Heathrow. There are many constraints upon a departure system that are not normally modelled and any solution should also aim to minimise other aspects such as controller and pilot workload and fairness.

This paper has presented a model for the system that can take account of the real life constraints. The initial results presented here include some of the constraints that are particularly important at Heathrow.

The results show that it is feasible to check the effects of the holding points after schedules have been generated and that the metaheuristics will still perform well in the limited time that they have.

From the experiments carried out here we can conclude that Tabu search performed best although it was the worst performer on dataset 1 in table

1. Simulated Annealing performed well across all the experiments but not always as well as tabu search. Further research will include much more experimentation to see whether these results apply in general for the Heathrow problem.

We can determine from the results that the metaheuristic searches form a promising basis for an advisory system for a controller as they are suggesting schedules which improve on the delay in the schedules the controllers are currently implementing.

Further research will add to this model and evaluate the effects of the constraints that have not yet been included. Implementation using genetic algorithms and hybridised metaheuristics are also planned.

7 Acknowledgements

This work was supported by EPSRC (The Engineering and Physical Sciences Research Council) and NATS (National Air Traffic Services) Ltd. from a grant awarded via The Smith Institute for Industrial Mathematics and Systems Engineering.

References

1. Abela J, Abramson D, Krishnamoothy M, de Silva A, Mills G (1993) Computing optimal schedules for landing aircraft. Proceedings of the 12th National Conference of the Australian Society for Operations Research, Adelaide, July 7-9, 1993, p71-90 Available at: http://www.csse.monash.edu.au/davida/papers/asorpaper.pdf [30 March 2004].
2. Anagnostakis I, Clarke J-P, Böhme D, Völckers Uwe (2001) Runway operations planning and control, sequencing and scheduling. Proceedings of the 34th Hawaii International Conference on System Sciences (HICSS-34), Hawaii, January 3-6, 2001.
3. Anagnostakis I, Idris HR, Clarke J-P, Feron E, Hansman RJ, Odoni AR, Hall WD (2000) A conceptual design of a departure planner decision aid. 3rd FAA/Eurocontrol International Air Traffic Management R & D seminar, ATM-2000, Naples, Italy, June 13-16, 2000. Available at: http://atm-seminar-2000.eurocontrol.fr/acceptedpapers/pdf/paper68.pdf [30 March 2004]
4. Anagnostakis I, Clarke J-P (2003) Runway operations planning, a two-stage methodology. Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS-36), Hawaii, January 6-9, 2003.
5. Anagnostakis I, Clarke, J-P (2002) Runway operations planning, a two-stage heuristic algorithm. AIAA Aircraft, technology, Integration and Operations Forum, Los Angeles, CA, October 1st-3rd, 2002. Available at: http://icat-server.mit.edu/ Library/Download/167_paper0024.pdf [30 March 2004]
6. Beasley JE, Krishnamoorthy M, Sharaiha YM, Abramson D (2000) Scheduling aircraft landings - the static case. Transportation Science 34:180-197

7. Beasley JE, Sonander J, Havelok P (2001) Scheduling aircraft landings at London Heathrow using a population heuristic. Journal of the Operational Research Society 52:483-493
8. Bianco L, Dell'Olma P, Giordani S (1999) Minimizing total completion time subject to release dates and sequence-dependent processing times. Annals of Operations Research 86:393-416
9. Craig A, Ketzscer R, Leese R A, Noble S D, Parrott K, Preater J, Wilson R E, Wood D A (2001) The sequencing of aircraft departures. 40th European Study Group with Industry, Keele 2001. Available at: http://www.smithinst.ac.uk/Projects/ESGI40-NATS/Report/AircraftSequencing.pdf [30 March 2004]
10. Ernst A T, Krishnamoorthy M, Storer R H (1999) Heuristic and Exact Algorithms for Scheduling Aircraft Landings Networks, Vol 34, Number 3, p229-241
11. Idris HR, Delcaire B, Anagnostakis I, Hall WD, Clarke JP, Hansman RJ, Feron E, Odoni AR (1998a) Observations of departure processes at Logan airport to support the development of departure planning tools. Presented at the 2nd USA/Europe Air Traffic Management R&D Seminar ATM-98, Orlando, Florida, Dec 1st-4th 1998. Available at: http://atm-seminar-98.eurocontrol.fr/finalpapers/track2/idris1.pdf [15 December 2003]
12. Idris HR, Delcaire B, Anagnostakis I, Hall WD, Pujet N, Feron E, Hansman RJ, Clarke JP, Odoni A (1998b) Identification of Flow Constraint and Control Points in Departure Operations at Airport Systems. Proceedings of the AIAA Guidance, Navigation and Control conference, Boston, MA, August 1998.
13. van Leeuwen P, Hesselink H, Rohling J (2002) Scheduling Aircraft Using Constraint Satisfaction. Electronic Notes in Theoretical Computer Science 76.
14. Newell GF (1979) Airport Capacity and Delays. Transportation Science 13:201-241
15. Trivizas DA (1998) Optimal Scheduling with Maximum Position Shift (MPS) Constraints: A Runway Scheduling Application. Journal of Navigation 51:250-266