On-line decision support for take-off runway scheduling with uncertain taxi times at London Heathrow airport.

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

September, 2006

[1] {jaa,ekb}@cs.nott.ac.uk, School of Computer Science and Information Technology,
University Of Nottingham, Jubilee Campus, Wollaton Road, Nottingham, NG8 2BB
[2] NATS CTC, 4000 Parkway, Whiteley, Fareham, Hampshire, PO15 7FL
[3] National Air Traffic Services, Heathrow Airport, Hounslow, Middlesex, TW6 1JJ

Abstract:

This paper addresses the challenge of building an automated decision support methodology to tackle the complex problem faced every day by runway controllers at London Heathrow Airport. Aircraft taxi from stands to holding areas at the end of the current take-off runway where they wait in queues for permission to take off. A runway controller attempts to find the best order for aircraft to take off. Sequence-dependent separation rules that depend upon aircraft size, departure route and speed group ensure that this is not a simple problem to solve. Take-off time slots on some aircraft and the need to avoid excessive delay for any aircraft make this an even more complicated problem. Making this decision at the holding area helps to avoid the problems of unpredictable push-back and taxi times, but introduces a number of complex spatial constraints that would not otherwise exist. The holding area allows some flexibility for interchange of aircraft between queues, but this is limited by the physical layout of the current holding area. These physical constraints are not usually included in academic models of the departure problem However, any decision support system for the take-off runway controller must include them. We show, in this paper, that a decision support system could help the controllers to significantly improve the departure sequence at busy times of the day, by considering the taxiing aircraft in addition to those already at the holding area. However, undertaking this re-introduces the issue of taxi time uncertainty, the effect of which we explicitly measure in these experiments. Empirical results are presented for experiments using real data from different times of the day, showing how the performance of the system varies depending upon the volume of traffic and the accuracy of the provided taxi time estimations. We conclude that the development of a good taxi time prediction system is key to maximising the benefits, although benefits can be observed even without this.

1 Introduction

Our research considers the role performed by a departure runway controller at London Heathrow airport and the potential benefits of producing a decision support system to help in the complicated task performed. London Heathrow airport is the busiest international airport in the world but it only has two runways available for use at any time.

With two parallel runways it is more efficient to employ them both for arrivals and departures, [21]. By alternating arrivals and departures on each runway the wake vortex constraints can effectively be eliminated, with consequent benefits to the throughput and delay for Heathrow, shown in [6]. However, Heathrow airport is situated very close to London, with some flight paths over highly populated areas, so a number of noise control measures are in place. The main effect of these restrictions is that the runways have to run in segregated mode (for either landings or take-offs but not both) for most of the day. Details of the precise rules and the reasons for them can be found in [9].

When ready for departure, aircraft request permission to push-back from stands on to taxiways. Aircraft then taxi along fixed taxiways to a holding area at the end of the departure runway. There, a runway controller attempts to re-sequence them to obtain a better take-off order. The runway controller spends most of his or her time communicating with pilots and monitoring the situation so there is very little time available for deciding upon the take-off order. However, a good take-off order is vital for maintaining a high throughput for the runway, and consequent low delays for waiting aircraft. It is, therefore, worth investigating the feasibility of a decision support system to help them with this task and we have developed the search algorithms for such a system.

When deciding upon a take-off sequence there are many things a controller must consider. Sequence-dependent separations have to be imposed between aircraft at take-off. These separation rules ensure both safety at take-off, by allowing time for wake vortices to dissipate to safe levels, and that required in-flight separation distances will be attained by ensuring minimum separations between the take-off times for aircraft which will depart along similar routes. Separations are also applied to reduce the frequency of flights into busy airspace and may be increased at times of congestion to further reduce the flow of aircraft into the congested airspace. Separation rules ensure that it is better, in general, to group multiple aircraft together by weight class and to avoid having aircraft which follow similar departure routes in adjacent positions in the take-off sequence.

The separation rules for the runway mean that the throughput of the runway is often much less than the throughput of the taxiways to the holding areas, so it is imperative to maintain as high a runway throughput as possible, to keep delays down for airlines and passengers. This is also true at many other airports, for example Boston Logan, [19].

Around thirty to forty percent of aircraft taking off from Heathrow have to adhere to a fifteen minute take-off time slot. Departure system load and delays can mean that these timeslots are not always achievable for all aircraft by the time they reach the holding area, so a small number of five minute extensions are permitted. As few extensions as possible should be used.

The algorithms we have developed for a decision support system to aid the runway controller are described in this paper, along with details of the simulation used to verify their efficacy. The basic system was described in [7] for the situation where there was perfect knowledge of the taxiing aircraft. The performance of the system is considered in more detail in this paper, as are the various improvements to the system, and to the simulation used to evaluate it. The effects of taxi time uncertainty are considered, in order to determine whether the system would be of use amidst the uncertainty intrinsic in the real-world situation. Although it is important to consider the long-term effects of decisions made in such a dynamic system, it was found that the long duration of the datasets used in [7], which covered half-day periods, was masking some of the detail of the performance of the system over shorter time periods. Experimental results presented in this paper show how the performance of the system varies over time, and an explanation for the results is provided.

In section 2, we discuss the previous research in this area, and evaluate its relevance to the departure problem at London Heathrow airport. In section 3, we explain the take-off problem in detail and the various constraints. We introduce our decision support system in section 4 and detail two particular parts, the handling of the holding area constraints and the evaluation of a solution value in sections 5 and 6. The dynamic departure system simulation used to evaluate the decision support system is detailed in section 7, including the important improvements over that used in [7]. In section 7.2 we explain the experiments we performed and the uncertainty involved in the taxi times from stands to holding area, then we present the experimental results in section 8. We end this paper by drawing conclusions from the results, in section 9.

2 Previous research

A significant amount of research has been published on the various aspects of air traffic control and air travel. In this paper, we restrict the discussion to take-off scheduling and refer the interested reader to overview papers such as [10] or [24] to better understand the wider context within which the take-off scheduling takes place.

The take-off sequencing problem has been considered by a number of researchers in the past. However, none of them considered all of the constraints we describe here; all of which are present at Heathrow. Work which considers the aircraft sequencing usually ignores the necessary aircraft movement, and research concentrating upon ground movement does not usually consider all of the sequencing constraints. For example, only wake vortex separation rules are considered in [18], making the sequencing problem far simpler.

Various search techniques have been applied to the departure problem in the past. These usually consider the sequencing of departures, without consideration of any constraints imposed when the overtaking has to be performed within the holding area. Constraint satisfaction techniques were successfully applied to the departure problem by van Leeuwen et al. in [23]. However, the problem was much smaller than the one faced at Heathrow. In [22], Trivizas applied dynamic programming to solve the problem, applying a maximum position shift constraint to reduce the problem size. Anagnostakis et al. used a search tree with a branch and bound or A* algorithm to solve the problem in [2]. Anagnostakis and Clarke proposed a two-stage solution method in [3] and [4], where the first stage ignored the downstream constraints. However, Heathrow is heavily affected by the downstream constraints, as we showed in [6], so this sort of approach is less appropriate there.

If the separations are small enough so that only the separation from the immediately preceding aircraft has to be considered, the problem can be seen to be a variant of the cumulative asymmetric travelling salesman problem (ATSP), with ready times, as explained in [14]. However, at Heathrow, the separation rules do not obey the triangle inequality so it is not sufficient just to check the separation from the immediately preceding aircraft and the equivalency to the cumulative ATSP does not hold. Furthermore, the introduction of take-off timeslots to the problem means that not only ready times but also due dates need to be introduced into the problem. This is further complicated by the fact that timeslots cannot always be achieved, so cannot be treated as hard constraints.

Arrival scheduling has many similarities to take-off scheduling as both have sequence-dependent separations. This has led researchers in the past to state that their approach to arrivals scheduling is also appropriate for departure scheduling. However, this does not hold when the holding area constrains the overtaking. Additionally, the objective is usually different for arrival and departure scheduling. The arrivals problem is often formulated as having target times for aircraft arrivals rather than aiming for earliest take-off times, and the separation rules that are considered are usually simpler.

Beasley, Sonander and Havelock used a population heuristic to assign landing times to arriving aircraft in [12]. The results compared favourably with those from a simple improvement heuristic and with the real controller-produced schedule. Abela et al presented a genetic algorithm to solve the problem in [1], gave an exact formulation of the problem and used a branch and bound approach to solve this, with a revised simplex algorithm to solve the Linear Program sub problems. Ernst et al presented a network simplex algorithm for determining take-off times given a specified take-off order in [15] and used heuristic and branch and bound methods to determine the take-off order.

Beasley et al. presented mixed integer zero-one formulations in [11] for the landing problem and provided an extensive review of the relevant literature at the time. Although they suggest that the formulation is equally appropriate for take-offs, this is not the case for Heathrow as the holding areas are more flexible than the simple queues which are common at many airports. This means that the interactions between precedence constraints become too complex to model in this way.

The main advantage of performing the take-off sequencing at the holding area is that the runway controller doing the scheduling knows precisely which aircraft are available and has a good idea of when they would be able to line up for take-off. Scheduling the take-offs any earlier introduces a lot more uncertainty into the problem, as variations in taxi times and taxiway congestion need to be taken into account. However, having multiple aircraft in a holding area means that the controller has to consider how any overtaking will be achieved, and indeed, some may not be possible at all. The physical layout of the holding area and the current positions of aircraft within it are therefore key to the decision-making process.

The only work we are aware of which currently considers the effects of the holding areas is [20], which used a simplified version of one of the simpler holding areas. The holding area in [20] is simple enough that the search space is reduced sufficiently to make a dynamic programming approach feasible. Applying this sort of approach to a real holding area at Heathrow is not feasible, however, as the number of positions aircraft can assume is far greater than in the example used. The state space for the dynamic program is therefore prohibitively large.

Simpler versions of the departure system simulation and decision support system presented here were introduced in [7]. The earlier versions lacked any method for predicting holding area positions for aircraft, so issues such as positional prediction accuracy and taxiway congestion, considered in section 5 of this paper, did not occur. The improved version described in this paper includes additional elements in the solution evaluation to penalise deviation from previous schedules, an element which became more important when taxi times were no longer predictable. The chosen decomposition method has also allowed the development of a solution cache, as described in section 4.4, which has in turn allowed the time for deterministic follow-on searches to be performed to give some guarantee about the quality of the presented solution.

3 Problem description

This paper considers the problem faced by a runway controller or decision support system intending to sequence aircraft for take-off at London Heathrow airport. This is a real problem, solved manually hundreds of times per day by runway controllers in the control tower at Heathrow. However, the take-off scheduling problem is a difficult problem with many different and complex constraints. It may be thought of as a combination of a sequencing problem, determining a take-off order, and a control problem. The control problem is concerned with determining whether a take-off order is achievable given the current positions of the aircraft and how it should be achieved.

The complete take-off problem is an on-line, dynamic problem. A runway controller is faced with an ongoing set of decisions, having to determine the order in which aircraft should take off and facing the consequences of these decisions later as they affect later sequencing possibilities. The complete take-off problem at the holding area can be defined as follows: ‘Given the current positions of the aircraft within the holding area and the expected arrival points and times of aircraft taxiing towards the holding area, determine a take-off order that is easily achievable by the pilots, meets as many take-off timeslots as possible, has a low delay for aircraft and is as fair as possible.’

Very little time is actually available to the runway controller to consider the take-off problem and various mental pattern-matching techniques appear to be used. The time constraints often require that the controllers ignore information that would be relatively time consuming to obtain, for instance information about taxiing aircraft. The various constraints upon the problem have been grouped below.

As all of the overtaking is performed within the holding area the structure of this and the current positions of the aircraft within it are key to determining what overtaking is possible. Decisions made for some aircraft may have later effects upon other aircraft. For example, aircraft A may only be able to overtake aircraft B if B moves out of the way, but doing so may then block another aircraft C from reaching the runway on time. The holding area constraints are complicated and are explained in more detailed in section 5, where the use of a directed graph model of the holding area to determine whether any required overtaking is achievable is considered.

Separations must be enforced between aircraft taking off. Aircraft leave wake vortices behind them as they take off; the strength being dependent upon the size of the aircraft. The following aircraft cannot take off until the wake vortices have dissipated to a point where safety will not be compromised. Normally aircraft can take off with one-minute separations. However, whenever a smaller category aircraft follows a larger aircraft a two-minute separation is needed. An inevitable consequence of this is that it is usually better to group aircraft together by weight category, to reduce the number of times that larger separations are needed due to weight category decreases between consecutive take-offs.

In addition to meeting wake vortex separation rules, aircraft must also meet additional separation rules which are applied to control the workload for air traffic controllers managing the nearby airspace and to ensure that aircraft attain the mandatory in-flight separation distances. A minimum separation of two or more minutes is imposed between flights following the same or similar departure route. As aircraft taking-off from Heathrow follow pre-planned departure routes, called Standard Instrument Departure (SID) routes, we refer to this route-dependent separation as a SID-separation. At times of congestion in the airspace these mandatory separations may be temporarily increased, changing the constraints upon the problem for a while. Furthermore, if the aircraft taking off are in different speed groups, the SID-separation may be increased or decreased depending upon whether the following aircraft is faster or slower than the preceding aircraft and how close the departure routes are to each other. Finally, these separations need to be maintained from all previous aircraft, not just the immediately preceding aircraft. In normal practice, with separations of between one and three minutes, this means checking separations from the previous two aircraft (as the minimum separation is one minute). When larger separations are imposed then departures from longer in the past may also need to be taken into account.

If a schedule is moved earlier or later the mandated separations between aircraft will not change. This is not the case with all of the constraints. Some aircraft have a target take-off time, called a Calculated Time of Take-off (CTOT). These are calculated to smooth congestion in busy airspace sectors and at busy destination airports. Aircraft are permitted to take-off up to five minutes before the target time, or up to ten minutes after the target time, so it effectively specifies a fifteen minute take-off window. Missing this means a renegotiation of a new CTOT for the aircraft and possibly large delays. As delays can sometimes mean that not all aircraft can achieve these CTOTs, to reduce the number that need to be renegotiated (hopefully to zero) a limited number of five minute extensions are permitted to the controllers. It is important to use as few of these as possible and ideally to send the aircraft as close to the CTOT as possible.

Before take-off, an aircraft has to taxi from its stand to the holding area, then taxi through the holding area and line up for take-off. This takes time and will limit how early an aircraft can take off. Similarly, even after push-back it will take the flight crew some time to perform all of the pre-flight checks. If a large aircraft pushes back from a stand near to the holding area it is possible that the pre-flight checks will not have been performed by the time it reaches the holding area. It is, therefore, important that a decisions support system does not schedule larger aircraft to take off too soon after push-back, even if they are close to the holding area. Any decision support system needs to account for these two limitations to the earliest take-off time.

All other considerations being equal, schedules should be as fair as they can be. The fairest schedule could be thought of as the first-come-first-served schedule. However, that is usually an extremely bad schedule from the point of view of delay and CTOT compliance. It is important not to cause a pilot unnecessary work taxiing an aircraft through a holding area. This means that certain paths through the holding area will not be used in practice, even though they are actually possible. It also means that longer paths or paths which involve more manoeuvring should only be used if really necessary.

Additionally, over time the suggested take-off order will change as new aircraft enter the system or more information becomes available. It is important to control the amount by which a schedule changes and the way in which it changes. This is a common effect in a dynamic system and was, for example, considered by Beasley et al. for the arrival problem in [13]. Changing the take-off position of an aircraft will be harder the closer it is to its take-off time. To avoid problems, we fix the position of an aircraft in the take-off schedule two minutes before the planned take-off time. We also penalise changes to the position of the aircraft in the take-off schedule, where the penalty is higher if aircraft are moved further in the schedule. Finally, we remember that a controller has to give instructions to pilots from the point at which they enter the holding area and that doing so is time-consuming. For this reason we fix the path an aircraft will use to traverse the holding area at the point it enters so that the system avoids the situation where complicated re-direction of aircraft is necessary. The design of the system does not actually require this, and the path allocation system could be permitted to re-allocate paths, if that was desirable, until the aircraft passes the point at which the available paths diverge, but doing so would risk introducing additional workload for controllers so was not permitted in the experiments performed for this paper.

4 Decision Support System

A decision support system for the runway controller has to take some input information, make a decision about a desired take-off order and present the decision to the runway controller. The inputs, outputs and decision-making element are all explained below. One of the most important objectives for the decision support system is to return the results very quickly. It is imperative that the system reacts to changes to the situation very quickly, for instance to the addition of a new aircraft or to updated timings for taxiing aircraft. Our system is designed to return a suggested take-off order within one second.

4.1 Inputs - the current departure system state

The decision support system needs information about the current state of the departure system in order to make a decision about which take-off order to suggest. This input state is comprised of the following information:

- The weight class, speed group and departure route of all aircraft currently under consideration.
- Any CTOTs which apply to the aircraft.
- The current position (node number) of any aircraft in the holding area and the time at which the aircraft arrived at the holding area.
- The predicted (possibly inaccurate) arrival time and holding area entrance for any taxiing aircraft.
- The take-off times of aircraft which recently took off.
- Any previously selected take-off order. This allows a preference to keeping the schedule similar to any previously planned schedule.
- Any previously allocated paths through the holding area for aircraft already within the holding area, to ensure they are not changed.

The aircraft in the system at any time consist of those within the holding area, those which took off recently and those which are still on the taxiways. All of this information could be made available to a live decision support system, through interfaces with existing systems such as the electronic strips system the controllers use to visualise the take-off order and the ground radar aircraft-tracking systems. If taxiing aircraft are included in the system, then the predicted holding area arrival times would need a separate system to consider the current position of any taxiing aircraft, the taxiway congestion and positions of other taxiing aircraft. One purpose of this paper is to examine how accurate a prediction this system would need to provide.

4.2 Output results

The decision support system returns information about the desired take-off order and how it will be achieved. It is imperative that the controller is not overloaded with unnecessary information. The intention is to give the controller only the desired take-off order, perhaps by annotating their existing displays. If desired, the method in which it was achieved could also be made available, perhaps upon controller request. Providing only a take-off sequence should be feasible due to the simplicity of the per-entrance path allocation method and is another reason for the sequencing to be achievable in an easy to understand manner. When working with a simulation of the decision support system rather than a real controller, the simulation is also given the predicted take-off times for aircraft, the holding area paths that were allocated and the predicted holding area positions for aircraft within the holding area, as described in section 5.5.

4.3 Seeking a take-off order

The decision support system has to find a high quality take-off order, and has to do so very quickly. This means seeking a take-off order which is easy to achieve (i.e. limited workload for the controllers and pilots), sensible to a controller (otherwise any suggestion would be ignored), has a low total delay for aircraft, does not penalise any aircraft too much and achieves as high a CTOT compliance as possible.

In order to simplify the task of meeting so many different objectives, the problem of finding a good take-off order is decomposed into two sub-problems. The first sub-problem is concerned with identifying a take-off order that has low delay, high CTOT compliance and is as equitable as possible. This sub-problem considers only a take-off order rather than how it will be achieved and whether it is possible. The second sub-problem is concerned with verifying that the take-off order is achievable, ensuring that the work to attain any given schedule is kept as low as possible and that this workload is not excessive. In practice this means attaining any target schedule in the way that is easiest for pilots and controllers and rejecting any that are difficult, thus ensuring that the relevant workload objectives are met.

Our problem decomposition has a number of key advantages for the solution of both sub-problems. Our decision support system solves the first sub-problem using a tabu search methodology to investigate possible take-off orders, then applies two follow-on searches to avoid obvious problems, as described in sections 4.4 and 4.5. The second sub-problem is solved heuristically and is detailed in sections 5 and 6. Our problem decomposition relies on being able to solve the second sub-problem very quickly (as it must be solved for each solution to the first sub-problem) and the first sub-problem having a well structured search space (to reduce the number of solutions which need to be examined).

The first sub-problem, the sequencing problem, is much easier to solve than the original problem. Since sequences can usually be achieved in multiple ways, ignoring the method by which the re-sequencing is achieved, while restricting the tabu search to the feasible solutions, results in a large reduction in the size of the solution space that must be considered. More importantly, however, the high quality solutions are clustered closely together when using the moves that we have provided to the tabu search. This is possible because the value of a solution in terms of delay and CTOT compliance depends only upon the take-off order, and consequent take-off times, not upon how the order is attained. There are still local optima, however, as was shown in [5] where a first descent search was outperformed by searches with the ability to escape local optima. Our search is explained in section 4.4. An important extra benefit of the selected approach is that solution caching becomes feasible for the sequencing problem as the number of sequences that need to be evaluated is relatively few.

Solving the second sub-problem means verifying the feasibility of required overtaking and determining the value of a take-off order. The first stage in evaluating a schedule is to determine whether it is feasible, and if so then how it is achieved. Although there will usually be multiple ways to achieve the same changes to the take-off order, there is usually a preferred way for the runway controller to achieve it. Other methods will often involve more work or may risk unnecessary congestion of the holding area. As our feasibility check knows the target take-off order we can ensure that the preferred method of overtaking is used to achieve the re-ordering of the aircraft. This is a major advantage as it avoids having to investigate many possible solutions that would not seem sensible to a human, so would have to be rejected later. This feasibility check is complex and is explained in section 5.

Once a schedule is known to be achievable, a determination of its worth has to be made. Take-off times are predicted for aircraft in the schedule, as described in section 6.2, and then a cost is calculated based upon the delay aircraft are predicted to experience and the number of CTOT extensions that are predicted to be needed, as explained in section 6.3.

4.4 Tabu search

We use a tabu search [16, 17] for the initial search of schedules. Our tabu search algorithm uses a memory of the moves it has recently made, in order to avoid undoing recent moves. This forces it to keep moving away from any local optimum it recently found and, hopefully, discover better local optima or even the global optimum.

Full details of the search can be found in [7], including details of how the moves help to reduce the number of local optima. We briefly summarise the algorithm below for completeness:

```
Loop for 100 iterations
   Generate 50 random neighbouring solutions from the current solution
   For each of the newly generated solutions:
      If the solution is tabu
         Then reject the solution
      Else
         Evaluate the solution to determine feasibility and a cost
      End If
   Select the lowest cost non-tabu solution:
      Make it the new current solution
      Add details of the move made to the tabu list
End loop
```

The tabu search always starts from the last suggested take-off sequence, or the first-come-first-served order if there is no previous suggestion. After this, the last solution suggested by the decision support system is used, modified by the removal of any aircraft that took off so long before that they can no longer affect current take-off times. If aircraft entered the system since the last schedule was produced they are added at the end of the schedule in predicted arrival order. This ensures that the initial schedule will be feasible as no extra overtaking is required than was achieved in the previous solution.

A solution for the tabu search is a take-off order. Generating each neighbouring solution involves first selecting a move to use and then applying it. The shift move is selected 50% of the time. This move selects 1-5 sequential aircraft in the schedule and moves them forwards or backwards to a new random position in the schedule. As good sequences often alternate departure routes, inserting a single aircraft in a new position is often not a good idea. Moving multiple aircraft avoids this problem. The swap move is used 30% of the time. This selects two random aircraft in the schedule and swaps their positions. This is especially useful for moving between good solutions by swapping similar aircraft, seeking the most equitable solution. The randomise move is used the remaining 20% of the time. This selects from 2-5 sequential aircraft in a schedule and randomly reorders them. The limit of five aircraft is a compromise between flexibility and controlling the number of possible moves, and is based on the fact that aircraft rarely move far from the first-come-first-served position. These moves, and the probability of using each move, were selected based upon empirical results obtained using the system considered in [5]. Experiments showed that it was possible to evaluate at least 5000 solutions in the one second search time, given the twenty aircraft problems found in [5]. The number of candidate solutions was chosen as a compromise between the size of the candidate list and the number of iterations of the search.

Evaluating a solution involves assigning paths through the holding area, verifying feasibility, predicting take-off times and evaluating a cost for the schedule. Each of these steps is deterministic, so a given schedule will always return the same result. For this reason, the results for a given schedule are cached in a solution tree. Naturally, this tree will usually be very sparse. However, the tree can be searched for solutions very quickly. Experiments have shown that the tree usually has around 2500 solutions in it at the end of the tabu search, providing that sufficient aircraft are in the system, so around half of the evaluations in the tabu search find a cached solution. Use of the cache significantly increases the speed of the search and it is also made available to the follow-on searches described in the next section, improving the search time there as well.

The tabu list represents the (banned) moves. Whenever a move is made, details of the aircraft which were moved, and where they were moved from, are added as a single entry to a tabu list. Future moves which put all of the moved aircraft back into the positions from which they were moved will be declared tabu. Moves are permitted to move some of the aircraft back to the old positions as long as they are not all in the original positions concurrently. The tabu list has a tenure of ten moves, so it is possible to reverse the effects of a move eleven or more iterations in the past but not one of the previous ten moves.

4.5 Follow-on rolling window search

The tabu search has been shown to perform well in our experiments. However, it makes no guarantees about the quality of the schedules returned. Examination of the schedules produced has shown that they do not usually move many aircraft more than two or three places away from the first-come-first-served schedule. However, at times some aircraft have needed to be moved up to ten places in the schedule. In each case where this has happened in our experiments it has been in order to move an aircraft into its CTOT timeslot. As any schedule where aircraft miss CTOTs is given a very high penalty, as will be seen in section 6.3, the tabu search very quickly moves towards schedules where aircraft are in CTOT, if possible. It then spends most of the search time moving the remaining aircraft around, slowly improving the delay and/or equity of the schedule.

Given the addition of the solution cache, experiments were performed with the number of iterations for the tabu search increased to 200. Although the cache made it possible to achieve this within the one second limit, we observed that the tabu search rarely found improving solutions beyond the first 100 iterations, and when it did so the aircraft that moved were usually close together. We also observed that swapping the positions in the take-off sequence of two aircraft with similar departure routes, weight classes and speed groups can often give a very similar costing schedule. As the tabu search makes no guarantee to consider specific sequences it is theoretically possible, although rare in practice, that it will alternate between two sequences of very similar cost, changing the advised solution at each iteration, merely by not investigating the alternative solution. Ensuring that each search is seeded with the best solution of the previous search, appropriately modified for aircraft which enter or leave the system, helps to prevent this alternating between sequences but presents no guarantee, especially in the presence of uncertain taxi times.

The previous observations led to the development of two follow-on searches. Rather than increasing the number of iterations of the tabu search to 200, the extra time created by the use of the solution cache is instead used by the follow-on searches. These start from the best schedule found by the tabu search and attempt to improve it further.

The first follow-on search checks all possible swaps of aircraft in the take-off sequence, ignoring aircraft for which the position has been fixed. As the evaluated sequences have up to twenty aircraft, this means a maximum of only 380 extra sequences need to be checked. This solves the 'two identical aircraft' problem mentioned previously by always considering the aircraft in the reverse positions, guaranteeing that the alternative schedule will be examined. A bias in the objective function (detailed in section 6.3) towards first-come-first-served schedules and a further bias towards the previous schedule found ensure that the swapping will occur at most once.

A second follow-on search is also performed to investigate any obvious local improvements that can be made to a schedule. A five-aircraft rolling window search is performed, based upon the fact that re-sequencing five aircraft can be performed very quickly and the fact that aircraft should by complying with CTOTs by this point (where possible) so should not need to move far in the sequence. The search starts by considering the first five aircraft in the schedule that are free to be re-ordered. It exhaustively checks all possible take-off orders for these aircraft, with the rest of the schedule fixed. Any improving schedule is adopted as the new best schedule. The search then moves forward one place in the schedule and investigates the next five aircraft, attempting to improve the take-off order of these aircraft. Each search must evaluate 120 different solutions. With a maximum of twenty aircraft free for scheduling a maximum of sixteen of these searches will be performed, for 1920 different solutions. The number needed can be further reduced by noting that in all searches beyond the first, any solution where the last aircraft's position is not changed was evaluated in the previous search so does not need to be re-evaluated, so only 96 (=120-24) different solutions need to be evaluated at each iteration.

Together these searches ensure that the schedules produced have a certain level of quality. In particular, they ensure that a controller will not look at a schedule and see an obvious improvement that could be made - a situation which would greatly undermine any confidence in a decision support system. The total number of solutions that need to be evaluated is at worst 6940 (= 5000 + 380 + 120 + (15 x 96)), but the performance in experiments was slightly improved by the addition of the follow-on searches, and the occasional alternation between similar schedules was eliminated.

It is important to note that the combination of tabu search and follow-on search have been shown to perform significantly better in our experiments than a rolling window search alone; even a rolling window search with a window size of seven aircraft. The tabu search has the advantage of being holistic, whereas the performance of the exhaustive search of the window at any time is partially reliant upon the rest of the schedule which it is not modifying.

5 Testing the feasibility of the take-off order

The feasibility test is similar to that presented in [7], with some additions. The test is summarised here for clarity and the extensions are explained in detail.

Testing whether it is feasible to achieve a desired take-off order is performed in two stages. The first stage involves heuristically assigning a path through the holding area to each aircraft. This heuristic path assignment ensures that only sensible path assignments are made and that the minimum work will be required from pilots to achieve the overtaking. The second stage involves determining whether the required overtaking is achievable given the allocated paths.

5.1 Holding area model

The research presented in this paper used a directed graph to represent each holding area. Each node in the graph represents a position at which an aircraft can be held. Arcs represent valid movements between nodes.

These graphs differ between the ends of the runways. On the southern runway the graph is actually formed by two disjoint subgraphs, one for the holding area north of the runway and one for the holding area south of the runway. An example graph is given in figure 1 for the eastern holding area on the

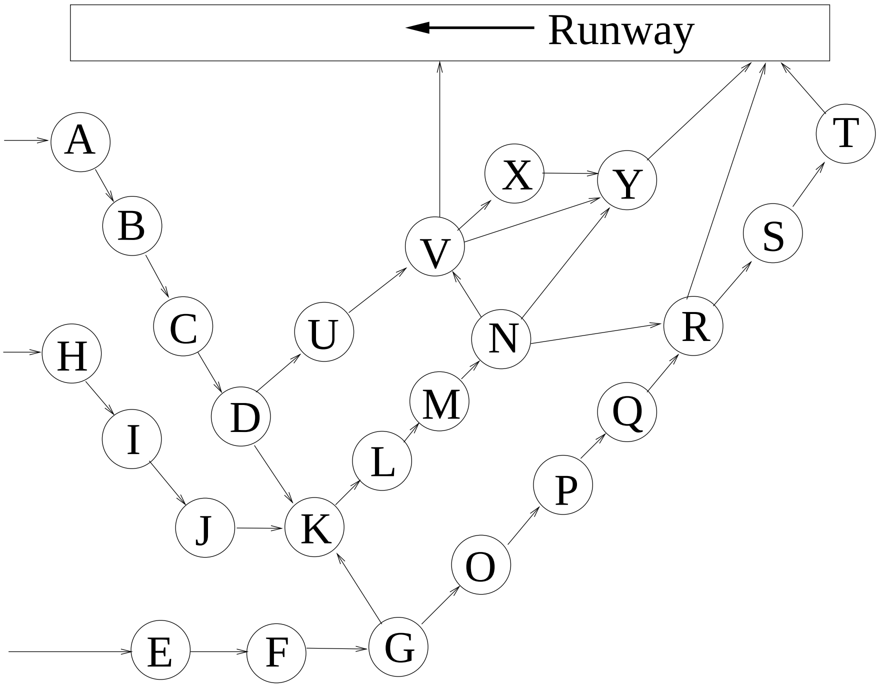

Figure 1: An example holding area network structure.

north runway.

There are different types of nodes in the holding area graph. Some nodes represent the taxiway near to the holding area, forming the input for the holding area. These are labelled A to K in figure 1. Some nodes represent the entrances to the holding area - the point at which aircraft actually enter the holding area. These are labelled D, G and K in figure 1. Some nodes represent exits from the holding area onto the runway. These are labelled R, T, V and Y in figure 1. V enters the runway part of the way along, but still near enough to the end to not affect the separation rules. The other three exits enter the runway at the end.

5.2 The path assignment problem

The path assignment heuristic has four considerations:

1) Ensure that the required overtaking is not prevented by the path assignment. The path assignment should consider the required overtaking and assign paths appropriately.

2) Control the workload. This means ensuring that longer paths are only ever used when necessary, and only by those aircraft which have the time to traverse them.

3) Appear sensible to controllers. For example, a sensible path allocation ensures that the overtak-

ing aircraft have shorter paths than the overtaken aircraft.

4) Limit how long the overtaking takes. For example, do not force an overtaking aircraft to go around an overtaken aircraft where such a path would be prohibitively long.

Full details of the path allocation heuristic can be found in [7]. The separation of the path allocation heuristic from the take-off sequencing and feasibility check aspects results in an extremely flexible system. For example, it is easy to change the heuristic according to circumstance, for example, to use different paths in reaction to the blockage of a path.

The holding area can be thought of as a number of queues of aircraft, each starting at a holding area entrance and ending at the runway, with some interchange permitted between queues at various points. The path assignment heuristic considers the aircraft at each entrance in turn. It assumes that the holding area movement algorithm will be able to ensure that aircraft can be reordered if different entrances are used, by appropriately interleaving the queues of aircraft. It, therefore, only has to ensure that the overtaking can be achieved for aircraft arriving at the same entrance. It performs this by ensuring that any two aircraft that must change order (i.e. one must overtake the other) are assigned different paths through the holding area.

The heuristic used for this paper has the following rules (where letters refer to nodes in figure 1):

1) If an aircraft is not overtaken then assign it the easiest path through the holding area. This will often be the shortest path. If there are two paths which are equally (or almost equally, in a controller's judgement) easy then assign it the most flexible of the two paths. For example, assign DUVXY in preference to DUVY to an aircraft at D.

2) If an aircraft is to be overtaken then assign it a path that will allow this to happen. For example, assign GOPQRST to an aircraft at G.

3) If an aircraft must overtake, then assign it a shorter path than the aircraft it must overtake. For example assign GOPQR to an aircraft which must overtake GOPQRST. (It is also possible to assign DUV to overtake DUVY if necessary, but we prohibit heavy aircraft from doing this in case they need the full runway for the take-off.)

Note that, in this context, overtaking means overtaking another aircraft from the same entrance as the path heuristic only considers one entrance at a time. This heuristic ensures that aircraft will take no longer than necessary to traverse the holding area, pilots will not have unnecessary workload and any overtaking aircraft will have shorter paths than the aircraft they overtake. This is sensible as the overtaking one obviously has less time to get to the runway than the overtaken one (it arrived later but should take off before the other).

5.3 Testing the feasibility using the holding area graph

A feasibility check is performed for each suggested take-off order using the holding area graph. Aircraft already within the holding area are placed at the node representing their current physical location, or the node they are travelling towards if they are between nodes. Aircraft on the taxiways are placed in virtual queues. A record is kept of the current test time throughout the feasibility test.

At any time, each node may have at most one occupying aircraft. A take-off order is feasible if, and only if, it is possible for the aircraft to reach the runway in the desired take-off order by moving the aircraft one node at a time without ever having multiple occupancy of any node. The algorithm can be outlined as follows:

```
Repeat until all aircraft have left system
   Clear movement flag
   Iterate for each aircraft currently in the system
      If the aircraft can move to the next node on its path (see conditions below)
         then do so and set the movement flag
      If the aircraft can exit to the runway and is the next in the take-off order
         then remove the aircraft from the system and set the movement flag
   End iteration
   Iterate for each entrance queue
      If the entrance node is empty
         and the arrival time of front aircraft in queue is before current time
         then start aircraft at the queue entrance and set movement flag
   End iteration
   If flag not set
      then increment time to the time the next aircraft will arrive
   If flag not set and no aircraft are pending
      then declare schedule infeasible
End repeat
```

The conditions under which an aircraft is permitted to move to the next node on its path are complicated, as it is important that moving an aircraft must not block another aircraft from exiting the system in the required order. This feasibility check was detailed in [7], including information about how to check whether any given aircraft can move to the next node. We summarise this information here and provide details of the new elements.

5.4 The current time value

During the feasibility check, a current test time is maintained. The current test time limits how early aircraft are released from the entrance queues into the feasibility check. In this way, it simplifies the complexity of the feasibility check by reducing the number of aircraft under consideration at any time. At any time, only the aircraft that would actually be in the holding area by the current test time can be in the nodes of the holding area graph. The limitation is only upon release into the feasibility check, not upon exiting onto the runway, so aircraft will leave as soon as the entrance is clear. This is necessary as the take-off times cannot be calculated prior to the feasibility check, as one of the outputs of the feasibility check is an earliest take-off time for the taxiing involved.

To explain the second use of the current time we introduce the concept of an earliest taxi time to the runway. Although it would be possible to estimate the amount of time an aircraft would need to reach the runway given its holding area position and taxi characteristics, this estimate would be expected to be uncertain. It would be possible to add this taxi time to the current time of the feasibility check to determine an earliest possible take-off time for the aircraft. However, the imprecision of taxi time values would make the accuracy questionable.

Rather than add complexity or require precise taxi time estimates, we ensure that aircraft always have plenty of time to traverse the holding area. Aircraft usually need less than one minute to traverse a holding area. To ensure that any schedule we derive is easily achieved we allow a minimum of two minutes to traverse the holding area, so we limit the take-off time to be at least two minutes after holding area arrival.

If an aircraft is overtaken, then it may need longer in the holding area as it will spend some time waiting. However, in this case, as the overtaking aircraft will be allowed at least two minutes to traverse the holding area, the overtaken aircraft will have at least two minutes plus the mandatory separation between the aircraft (as it cannot take off before the separation time has expired) from the time the other aircraft reached the holding area.

The remaining consideration is to ensure that aircraft that have to wait for some reason other than being overtaken have sufficient time to do so. The problem is that it is theoretically possible for aircraft A to have an earlier take-off time than aircraft B and C, but for the overtaking to only be feasible if B is allowed to overtake C prior to A moving. In this case the arrival times of B and C and the time taken to overtake limit how early A can take off.

To model this situation, as soon as the feasibility check time is advanced, the earliest take-off time for all aircraft still in the holding area is limited to be the standard traversal time (two minutes) plus the current feasibility check time. As the removal of aircraft from the system is not limited by the test time, all aircraft in the system when the time is advanced must be unable to move without the addition of the new aircraft. It is, therefore, sensible to limit their take-off time according to the arrival time of the new aircraft. In the example above, A would still be in the system, unable to move, until the check time has advanced sufficiently to allow both B and C to enter the holding area, so the arrival times would limit the take-off time of A.

This earliest take-off time constraint, that comes from the feasibility check, is modelled as the value f_i in the take-off time prediction model detailed in section 6. Its value is set during the feasibility check to be 120 seconds after the holding area arrival time for aircraft which can exit the holding area before the first time advancement, or 120 seconds after the current feasibility check time at the point the aircraft exit the graph for other aircraft. These taxi times are much higher than actually required, but serve to build surplus taxi-time into the schedule to avoid aircraft having time pressures. The aim is to ensure that the taxiing required to achieve the schedule is possible rather than to try to accurately model exact taxi times.

5.5 Predicted positions and taxiway congestion

When the decision support system is working with a departure system simulator it is responsible for providing updated predicted positions of aircraft to the simulator. This avoids the simulator having to do so itself. It does this by taking a snap-shot of the holding area positions of the aircraft at the point at which the first aircraft in the holding area takes off, or when the time has to be advanced, whichever comes first. Although this is an output of the decision support system, and so should really be considered to be a part of the departure system simulation, the method in which this snapshot is taken is described in this section as it is obtained by re-executing the feasibility test on the final desired take-off sequence.

When taking a snapshot of the positions of the aircraft it is important to consider whether the positions are realistic. Consideration of the predicted positions of the aircraft within the holding area at the point at which the snapshot was taken showed that there were cases where the positions deviated from those the aircraft would be expected to attain in reality. It was observed that predicted aircraft positions were often further back in the holding area than they would be in practice. This was usually a consequence of not having given the aircraft sufficient opportunity to move forward by the time the snapshot was taken. In practice, aircraft would be moved further into the holding area when it did not reduce re-sequencing possibilities. Sometimes, aircraft were positioned so far back at the time the snapshot was taken that unnecessary holding area or taxiway congestion would have occured behind them. In practice, controllers would not leave aircraft congesting the taxiways if it is at all possible to fit them into the holding area. As this situation would not be permitted to occur in practice, any simulation should also not permit it.

To simulate this expected behaviour of a real system, some further checks are made immediately prior to taking the snapshot of positions, attempting to move aircraft as far forward as they can. Each aircraft is tested in turn to see if it can move forwards without either leaving the holding area or affecting the feasibility of the desired take-off order. The same rules about blocking are used as in the normal movement in the feasibility check. This is repeated until either no more aircraft can move or the taxiways are clear. The effect of this is to move aircraft as far forward as possible in the holding areas, making room for aircraft on the taxiways to enter the holding area. This accounts for the situation where aircraft were too far back merely due to the order in which aircraft were to move.

If there are still aircraft on the taxiways after this, a second, more time consuming test is performed. The normal blocking system uses a simple rule: ‘Do not enter a node before a higher priority aircraft that is entering from a different node, unless there is room to immediately move out of the way.’ In this second stage we use a slightly modified rule: ‘Do not enter a node before a higher priority aircraft that is entering from a different node, unless it will be possible to move out of the way by the time that aircraft needs to take off.’ To do this, each of the nodes it would be possible to move into is examined. If all aircraft which will use the node before the blocking aircraft have take-off times earlier than the aircraft being blocked then these nodes are assumed to be empty for the purposes of the movement check. In this case, the aircraft will have taken off by the time the blocking becomes a problem, at which point the blocking aircraft would be able to move out of the way. This is a more time consuming test and possibly leads to aircraft temporarily blocking other aircraft which take off before them so is not desirable from the point of view of providing schedule flexibility and is normally avoided. This test means that aircraft can sometimes move further into the holding area, possibly freeing nodes to move aircraft off the taxiways.

If this still fails to move aircraft off the taxiways there are two choices. If there are empty nodes in the holding area then it is possible to force aircraft forward even when it will mean the target take-off order is unachievable, in the hope that future searches will be able to recover low cost schedules. Alternatively, sometimes it is possible to reallocate holding area entrances to aircraft. This is something the ground movement controller is likely to do if part of the holding area is congested but a part near another entrance is clear. The final option available to a system is to accept taxiway congestion in these cases. In our experiments, the first of these two checks usually cleared taxiway congestion and the second always did so.

6 Solution Evaluation

The decision support system we have developed was detailed in section 4. A two stage process is used to determine the value of a prospective take-off order. First the take-off times must be predicted for the aircraft, then the cost is calculated based upon the predicted take-off times.

6.1 Definitions

For each aircraft i, let c_i represent its position in the intended take-off order, so $c_i = 1$ for the first aircraft to take off, and let a_i represent its position in the arrival order at the holding area, so $a_i = 1$ for the aircraft which arrived first. As the decision support system is usually given a set of problems to solve, and should slightly favour the previously chosen take-off order to aid schedule stability, let o_i denote the position of the aircraft i in the previously generated take-off sequence, if there is one, or set o_i to the same value as a_i if there was no previous sequence.

Given any two aircraft, i and j, where i takes off before j, the minimum permitted separation between them, S_{ij}, may be calculated by taking the maximum of the required wake vortex and route separations. These depend upon the weight classes, speed groups and departure routes of the two aircraft. As explained in [7], the relationship represented by S_{ij} is asymmetric, does not obey the triangle rule and can change in situations of congestion.

Let f_i be the take-off time limitation applied to aircraft i by the feasibility check, which is used to ensure that aircraft can reach the runway in time for take-off time, as described in section 5.4. Let h_i denote the time at which aircraft i entered the holding area, or is predicted to do so if it is still taxiing. Let d_j denote the (known or predicted) take-off time for aircraft j.

If aircraft i has a take-off timeslot then let b_i denote the start time of the timeslot and let l_i denote the latest time that aircraft i can take off and still be within its timeslot. All times are given in seconds, so in normal operation, $l_i = b_i + 900$ for the fifteen minute CTOT timeslot. For any aircraft i which does not have a CTOT, set b_i to a large negative value and l_i to a large positive value, so that the timeslot effectively covers the entire of the time period of the experiments.

6.2 Take-off time prediction

Take-off time prediction is very similar to the method explained in [7], with the important addition of the factor f_i. The earliest take-off time, e_i, for aircraft i, that abides by all separation rules, may be calculated using equation 1. At busy times, aircraft take off as early as they can, so the take-off time prediction assumes that aircraft take-off as soon as the various take-off time constraints have been met. The take-off time, d_i, can therefore be predicted using equation 2.

$$e_i = \begin{cases} 0 & \text{if } c_i = 1 \\ \max\limits_{j \in \{1,\ldots,n\} | c_j < c_i} (d_j + S_{ij}) & \text{if } c_i \geq 2 \end{cases} \tag{1}$$

$$d_i = \max(e_i, f_i, b_i) \tag{2}$$

6.3 Objective function, the solution cost

As there are multiple objectives for this problem it could be formulated as a multi-objective problem and a number of different solutions could be presented to the controller. However, time pressures on the controller mean it is inadvisable to present him or her with multiple schedules to choose from as doing so risks complicating the job rather than simplifying it. In order to produce a single objective for the problem, we have prioritised some of the objectives by weights and applied an ‘as easy as possible’ constraint for others, such as path assignment. In this way, all of the real-world objectives are guaranteed to be considered.

The objective function used for these experiments is given by equation 3 and the objectives are detailed below. It has a number of parts, which are added together, with appropriate weights, to obtain the final objective function to be minimised. W_1 to W_{11} are constant weights used to combine the multiple objectives together into a single objective function. For this paper, we used $W_1 = 1$, $W_2 = 3$, $W_3 = 10000$, $W_4 = 10000000$, $W_5 = 2000$, $W_6 = 100000$, $W_7 = 1$, $W_{10} = 1$, $W_{11} = 1$. The weights W_3 to W_7 were chosen such that complying with CTOTs was of primary concern, so that CTOTs were always met where possible. In fact, the decision support system puts such a upon CTOT compliance that they will often be complied with regardless of how many aircraft this means the CTOT aircraft has to overtake. The values of W_1, W_2, W_{10} and W_{11} were chosen to balance limiting excessive positional delay on a single aircraft against limiting the total delay. We note that W_8 and W_9 have been omitted to simplify comparison with [7], and that, when comparing W_1 and W_2, a single position shift will often change the delay by multiples of minutes but the values of d_i are in seconds.

$$\begin{gathered} \sum_{i=1}^{n} (C(d_i, b_i, l_i, h_i) + W_1(d_i - h_i) + W_2(\max(0, c_i - a_i))^2 + P(c_i, a_i) \\ + W_{10}((\texttt{abs}(c_i - a_i)(\texttt{abs}(c_i - a_i) + 1)/2) + W_{11}((\texttt{abs}(c_i - o_i)(\texttt{abs}(c_i - o_i) + 1)/2)) \end{gathered} \tag{3}$$

The function $C(d_i, b_i, l_i, h_i)$, in the first term in formula 3, penalises any schedule where CTOT extensions are needed, or where even an extension is missed. The function $C(d_i, b_i, l_i, h_i)$ is defined by equation 4, is summarised below and is fully described in [7]. The primary aim is to heavily penalise any schedule where CTOT extensions are needed (i.e. $(l_i + 300) > d_i > l_i$) and to apply a prohibitively large penalty whenever an aircraft is scheduled too late for even a CTOT extension (i.e. $d_i \geq (l_i + 300)$). No penalty will be associated with an aircraft within its CTOT, or which does not have a CTOT, since $l_i \geq d_i \geq b_i$ holds in that case. Finally, if an aircraft arrives at the holding area very close to the end of its CTOT, then the large penalty associated with CTOT compliance could force the system to schedule it as early as possible, regardless of the effect on other aircraft. To account for this case the constant F_H is applied to reduce the penalty for slightly delaying these aircraft. For these experiments we used $F_H = 240$, reducing the penalty for small delays to aircraft which take off within two minutes of their earliest take-off time.

$$C(d_i, b_i, l_i, h_i) = \begin{cases} W_3((d_i - l_i)^{1.1}) + W_4 & \text{if } d_i \geq (l_i + 300) \quad \text{(i)} \\ W_5((d_i - l_i)^{1.1}) + W_6 & \text{if } (l_i + 300) > d_i > max((h_i + F_H), l_i) \quad \text{(ii)} \\ W_7(d_i - l_i) + W_6 & \text{if } (h_i + F_H) \geq d_i > l_i \quad \text{(iii)} \\ 0 & \text{if } l_i \geq d_i \geq b_i \quad \text{(iv)} \end{cases} \tag{4}$$

The secondary objective is to reduce the delay in the holding area as this is of great importance for passengers, airlines and everyone concerned with pollution. The second term in formula 3, $d_i - h_i$, relates to the holding area delay and is proportional to the time spent in the holding area. The third objective is to ensure equity of delay, where possible, and the remaining terms in formula 3 serve this purpose.

If an aircraft is delayed from the first-come-first-served schedule then it is already being penalised, so it is important to ensure that this penalty is as small as possible. No aircraft should be excessively delayed, and it is better to delay two aircraft by a small amount than one aircraft by a large amount. The factor $(\max(0, c_i - a_i))^2$ is used to apply a penalty proportional to the square of the positional delay each aircraft suffers.

The factor, $P(c_i, a_i)$, in the objective function is added to avoid penalising certain aircraft types. If an aircraft which will always have a large separation associated with it (such as a light or slow aircraft) is delayed to be scheduled near to the end of the take-off schedule this function will apply a penalty to penalise the schedule. Without this factor, there is a tendency for the system to schedule the aircraft later and later, to avoid the separation, then to suddenly move it a long way back into the schedule at the point where the positional delay cost finally surpasses the cost of the larger separation. The presence of this function prevents this fragility in the schedule and examination of the resulting sequences proved its efficacy. In the experiments described here the function was implemented to apply an extra penalty equivalent to a two minute delay to any schedule which delayed a light or a very slow aircraft more than one position in the take-off sequence.

An additional factor is also applied to favour the first-come-first-served order. Given two aircraft with similar characteristics, they should take off in the order they arrived at the holding area when doing so makes no difference to delay or CTOT compliance. If neither aircraft has a positional delay then $(\max(0, c_i - a_i))^2$ will evaluate to zero regardless of the take-off order, so will not ensure that the take-off order favours the arrival order. A factor $(\texttt{abs}(c_i - a_i)(\texttt{abs}(c_i - a_i) + 1)$ is added to the objective function to ensure that the searches favour this preference even when no delay is involved, although with a lower weight than if a positional delay was involved.

There are often cases where there are two very similar schedules. It is better to favour a previously used schedule rather than allow schedule change that will have little benefit. This is especially important if there is some uncertainty in the data used to make the decisions since small perceived benefits may be purely down to data errors. To provide for this, a factor $(\texttt{abs}(c_i - o_i)(\texttt{abs}(c_i - o_i) + 1)$ is added to the objective function.

7 Experimental details and departure system simulation

In order to realistically test a decision support system, it is important to have a realistic simulation of the departure system. Our departure system simulation forms the test harness for the decision support system and is detailed in this section.

The departure system simulation is responsible for providing an input state for the decision support system, as described in section 4.1. The simulation must take the outputs of the decision support system, update the problem appropriately and present the updated problem back to the decision support system.

7.1 Real, historic data

Our experiments used recorded historic data provided by National Air Traffic Services. This helped to ensure that the simulations accurately reflected the real world situation. We have been provided with datasets which cover entire days and have divided them into thirty five smaller datasets containing sixty aircraft each. Although the datasets used in the tests each consisted of sixty aircraft, the take-off positions and take-off times for the first ten aircraft were fixed to be the same as they were historically. This provided a history for the search, which is important as the datasets do not necessarily start at quiet times of the day. These datasets were created from the supplied datasets by taking samples of sixty consecutive aircraft starting from indices 0, 25, 50, 75, and so on, in 25 aircraft increments. Datasets overlap to reduce any bias introduced by imposing artificial boundaries upon the datasets.

Using smaller datasets makes it much easier to see how the accuracy of the take-off time prediction model differs over time. It also allows a comparison of the performance of the system under different conditions. Over short periods of time, especially at busy times, the take-off time prediction model is very accurate. However, over longer periods of time the cumulative differences between real and predicted times can increase. Fifty aircraft is small enough to be able to easily see the effects of any re-sequencing, but large enough that the effects of earlier decisions will be seen in later decisions.

The datasets include the weight class, departure route and speed group for each aircraft. These allow wake vortex and route separations to be calculated between any pair of aircraft. Various timings are also provided for each aircraft, including the time at which each aircraft left its stand, reached the holding area and took off. The allocated stand was also specified for each aircraft. From this an 'easiest' taxi route to the holding area and the entrance at which the aircraft would enter the holding area were determined.

7.2 Taxiing aircraft

Unlike aircraft within the holding area, for which positions were maintained, taxiing aircraft were dealt with at a relatively high level of abstraction. Experiments have shown that our decision support system performs better if it is given knowledge of the taxiing aircraft, enabling the search to have visibility of the effects of current decisions upon these taxiing aircraft.

One common case that including taxiing aircraft help with is when there is a taxiing aircraft with a tight take-off timeslot. In this case, the system needs to leave room for it to overtake aircraft in the holding area. Another example is where there are multiple taxiing aircraft with similar departure routes. It is often useful to fit aircraft already in the holding area between these taxiing aircraft, avoiding the large separations. To do this, it may again be necessary for the first of these aircraft to overtake aircraft in the holding area.

Taxiing aircraft have two important attributes: an expected arrival time and an expected arrival entrance. The simulation assumes that aircraft will leave their stand and arrive at the holding area at the times at which they did so historically. At any time, the decision support system will only have incomplete knowledge of these times. This is simulated by adding an error to the predicted holding area arrival time. The aim of this paper is to examine the effect upon the performance of the decision support system of introducing various levels of uncertainty errors into the taxi times. It has been identified in [8] that knowledge of the taxiing aircraft can improve the performance of a decision support system. Here we examine whether this is still the case when the taxi times are not known precisely, and at what point the knowledge of taxiing aircraft ceases to be useful. Prior to push-back from the stand there will be much more uncertainty in the holding area arrival time than after the aircraft has pushed back. For this reason, aircraft are only added to the simulation at the point at which they push back.

7.3 Simulation steps

The holding area model is used to maintain information about where an aircraft actually is, and what re-scheduling is still possible within the holding area. The use of the holding area model in testing the feasibility of any re-scheduling is discussed in section 5.3.

The departure system simulation is responsible for maintaining and updating the current positions of each aircraft within the holding area. As aircraft arrive at the holding area, they are placed at A, E or H, depending upon their point of arrival. Controllers will give pilots instructions to direct them through the holding area. In our simulations, the decision support system is responsible for assigning routes to aircraft as well as determining a final take-off order so the simulation retrieves this information from the decision support system. Over time, the aircraft move forwards through the holding area along their assigned paths, eventually reaching the runway and taking off.

The simulation is initialised by setting the current time to the start time of the dataset and generating a random uncertainty factor for each aircraft. For these experiments we generated a random percentage taxi time error for each aircraft. A linear distribution was used for the simulated prediction error and the limits were varied from +/-10% to +/-90%. The calculated error was applied to the remaining taxi time to create a taxi time prediction to give to the decision support system, as discussed in section 8. The aim was to identify how the potential benefits from a decision support system vary with the accuracy of the taxi time prediction system. In particular, it is desirable to know the tolerance that would be required of a taxi time prediction system in order to obtain a given level of benefit. Since minimum separations between take-offs are in multiples of minutes, lower levels of error can often be insignificant. The selected linear distribution ensures that the prediction errors will not be as closely clustered around the mean as when a distribution such as the Gaussian one is used, and thus means that the prediction errors are more likely to have an effect upon the system.

The departure system simulation then repeats the following steps:

1) Any aircraft which has a stand leaving time earlier than the current time but is not already in the simulation is added to the simulation now.

2) Update predicted taxi times for all aircraft on the taxiways. Given the current simulation time

and the known holding area arrival time, a predicted holding area arrival time is generated by applying the percentage taxi time error.

3) Pass the current problem to the decision support system and retrieve the suggested take-off order. The current node and path is specified for all aircraft within the holding area. The predicted arrival time and arrival position is specified for aircraft on the taxiways. The take-off time is remembered for aircraft which have taken off but are still in the simulation.

4) Request predicted positions of aircraft in the holding area from the decision support system, using the suggested take-off order. This is explained in section 5.5.

5) Request details of assigned holding area traversal paths from the decision support system.

6) Update internal model of the departure system. Store the positions (nodes) and fix the traversal paths for all aircraft within the holding area.

7) Increment the simulation time by 60 seconds.

8) Remove from the simulation any aircraft which have taken off sufficiently long ago that their take-off time can no longer affect the take-off times of current aircraft.

9) If any aircraft remain in the simulation (or have not yet entered the simulation) then return to step 1.

The results obtained when using this simulation to test the decision support system are presented in section 8.

8 Results

In this paper, we present results showing the effectiveness of our system in scheduling aircraft for take-off at the holding area. The value of predictions for the taxi times of aircraft on the taxiways was seen in [6]. The experiments presented here aim to determine the effects of uncertainty in the taxi times.

Tests were performed on thirty five datasets of sixty aircraft each, the first ten aircraft in each dataset being immovable. Three different half-day periods are covered, the first being datasets 1 to 11, The next 12 to 23 and the final one 24 to 35. The datasets overlap. The first dataset covers aircraft 1 to 60, the next 26 to 85, the third 51 to 110, etc. In previous research, [7], we have seen that, judged over an entire half-day period, the system does very well compared with manually produced schedules. Here we examine the performance of the system over smaller time periods. Each test was performed 100 times and the mean results are shown in Tables 1 and 2. Each row in the table represents a different dataset. Each of the other columns shows the results of one of the tests, or of the real, manual results.

Table 1: Mean CTOT compliance for real and automatically generated schedules

DS	FCFS	Real	Predict	Full	20%	40%	60%	None
1	8	5	4	1.00	1.00	1.00	1.01	1.00
2	6	4	4	0.00	0.00	0.00	0.00	0.00
3	13	2	2	1.00	1.00	1.00	1.05	2.00
4	13	6	7	1.00	1.18	1.47	1.45	3.09
5	12	6	7	2.00	2.02	2.00	2.02	2.00
6	7	1	2	0.00	0.00	0.01	0.02	0.00
7	3	0	1	0.00	0.00	0.01	0.02	0.00
8	1	1	1	0.00	0.00	0.00	0.04	1.00
9	1	1	1	0.00	0.13	0.12	0.15	1.00
10	0	0	0	0.00	0.00	0.00	0.00	0.00
11	0	0	0	0.00	0.00	0.00	0.00	0.00
12	1	0	0	0.00	0.00	0.00	0.00	0.00
13	4	0	0	0.00	0.00	0.00	0.00	0.00
14	1	0	0	0.00	0.00	0.00	0.00	0.00
15	2	2	2	1.00	1.00	1.04	1.08	1.00
16	2	3	2	1.00	1.00	1.03	1.17	1.00
17	8	2	2	0.00	0.00	0.00	0.00	0.00
18	1	1	1	0.00	0.00	0.00	0.00	0.00
19	0	0	0	0.00	0.00	0.00	0.00	0.00
20	4	1	1	0.00	0.00	0.01	0.05	0.00
21	3	1	2	0.00	0.01	0.00	0.03	0.00
22	2	0	1	0.00	0.00	0.00	0.00	0.00
23	0	0	0	0.00	0.00	0.00	0.00	0.00
24	8	1	2	1.00	1.00	1.00	1.07	1.00
25	7	1	2	1.00	1.00	1.02	1.04	1.00
26	1	0	0	0.00	0.00	0.00	0.00	0.00
27	1	0	1	0.00	0.00	0.00	0.00	0.00
28	3	1	3	1.00	1.00	1.00	1.02	1.00
29	7	1	2	1.00	1.00	1.00	1.00	1.00
30	7	0	0	0.00	0.00	0.00	0.00	0.00
31	1	0	0	0.00	0.00	0.00	0.00	0.00
32	0	0	0	0.00	0.00	0.00	0.00	0.00
33	5	3	5	2.00	2.00	2.12	2.24	3.00
34	9	3	5	0.00	0.28	0.60	0.80	1.00
35	6	1	3	0.00	0.03	0.28	0.36	0.00

The dataset number is shown in the first column of the table. Given a take-off order, our take-off time prediction method can be used to predict take-off times for aircraft and obtain a prediction for the number of CTOT extensions missed and the total delay of the schedule. The second column of the tables, labelled ‘FCFS’, shows the predicted performance if the controllers had allowed aircraft to take off in the first-come-first-served sequence. These figures were determined by applying the take-off

Table 2: Mean delay for real and automatically generated schedules

DS	FCFS	Real	Predict	Full	R%	M%	20%	40%	60%	None	R%	M%
1	26235	21868	20253	13094	40	35	13120	13276	13619	16053	27	21
2	25668	27509	25587	13715	50	46	14015	14124	14394	15757	43	38
3	44627	28123	27238	16127	43	41	16970	17589	18224	19627	30	28
4	54576	27541	32616	25506	7	22	26586	28107	28196	28989	-5	11
5	44864	26906	30204	19179	29	37	19252	19335	19686	21224	21	30
6	41745	24230	27625	17359	28	37	17687	17928	18293	17859	26	35
7	34950	17046	22410	18253	-7	19	18290	18526	18894	19804	-16	12
8	14269	12945	14738	10716	17	27	10922	11111	11416	11661	10	21
9	29237	22631	22177	15549	31	30	16029	16692	17156	18735	17	16
10	45848	24591	24853	16266	34	35	17034	17069	17215	17751	28	29
11	24331	15951	15772	12842	19	19	12890	13083	13371	13247	17	16
12	25322	12572	14497	14147	-13	2	14295	14605	14849	15625	-24	-8
13	36028	14452	21388	14042	3	34	14062	14188	14687	15657	-8	27
14	23836	13770	21016	11999	13	43	12054	12173	12614	12921	6	39
15	17643	12714	16355	10916	14	33	10616	10644	10828	10511	17	36
16	22967	19159	18913	13389	30	29	13160	13279	13825	14354	25	24
17	50517	20068	23577	15554	22	34	15966	16289	16708	18206	9	23
18	29421	13813	14736	12038	13	18	12152	12305	12546	12954	6	12
19	16603	11499	11748	10793	6	8	10839	10911	11164	11698	-2	0
20	32176	18235	21510	13788	24	36	13881	13995	14260	17590	4	18
21	55552	23644	30532	17096	28	44	17312	17608	18033	22084	7	28
22	44174	18279	25104	15080	18	40	15132	15247	15493	18348	-0	27
23	26650	14996	19180	10679	29	44	10741	10878	11224	11891	21	38
24	23563	17209	20291	14951	13	26	15041	15095	15241	15089	12	26
25	37445	17314	22116	16487	5	25	16878	17238	17623	18067	-4	18
26	24418	12038	14005	11677	3	17	11718	11941	12179	12403	-3	11
27	13598	11285	13549	11396	-1	16	11410	11427	11599	11492	-2	15
28	23357	16199	20042	15166	6	24	15227	15453	15898	15641	3	22
29	40550	20319	20917	15837	22	24	15839	16045	16327	17338	15	17
30	34123	20028	18973	13839	31	27	13868	14034	14360	15517	23	18
31	20691	15607	15803	13218	15	16	13240	13363	13585	13670	12	13
32	17032	16262	14822	12155	25	18	12174	12295	12520	13178	19	11
33	25533	23335	24347	17793	24	27	18076	18074	18525	18904	19	22
34	38327	22161	25679	18010	19	30	18363	18620	18815	19629	11	24
35	33799	18582	23524	14779	20	37	14985	15236	15545	17080	8	27

time prediction method to the first-come-first-served sequence. The third and fourth columns show the performance of the schedule the controllers actually used on the day. The third column, labelled ‘real’, shows the number of CTOT extensions the controllers really needed and the total delay for aircraft, given the real take-off times. The fourth column, labelled ‘predict’, shows the predicted performance when the take-off time prediction method is applied to the take-off sequence the controllers actually used.

The remaining columns in the table show the performance of the decision support system. Each experiment was repeated 100 times since there is a random element to both the search and the random taxi time error determination. Each execution used a different random seed so the taxi time errors were different for the different experiments. Each of the presented results is the mean of 100 executions of the system. For this reason the results in Table 1 have been provided to two decimal places, to show that not all executions obtain the same CTOT compliance.

In Table 1, the fourth column, labelled 'Full', shows the number of CTOTs missed by the decision support system when it has full, perfect knowledge of the aircraft on the taxiways. To investigate the effect of take-off time uncertainty we injected errors into the taxi times given to the system and evaluated how the performance of the system changed. The next three columns show how the performance changed when errors of up to 20%, 40% and 60% were included in the taxi times. In each case an error percentage was randomly determined for each aircraft before the test, using a linear distribution. For the 20% error each aircraft was assigned a random error between -20% and +20%. Similarly, errors were between +/-40% or +/-60% for the other tests. Every time a predicted arrival time had to be provided to the decision support system the remaining taxi time was modified according to the specified prediction error for that aircraft. This meant that the system was usually working with incorrect taxi times and that the error for that aircraft was proportional to the remaining taxi time. This simulates the effect that predictions become more accurate as aircraft approach the holding area. The final column of Table 1 shows the performance of the system when there is no knowledge of taxiing aircraft until they get within thirty seconds of the holding area.

Table 2 is similar to Table 1 but has four extra columns labelled 'R%' or 'M%'. Table 2 shows the total delay for the aircraft, defined here as the total time in seconds that the aircraft spend in the holding area. The R% column shows the percentage improvement of the automated schedule over the real controller schedule. The M% column shows the percentage improvement over the manual schedule with predicted take-off times. These are shown both for the automated schedules with full taxi time knowledge and with no taxi time knowledge. The uncertain taxi time results usually lie somewhere between these extremes. Together, these two columns can be used to indicate the performance of the system.

The first thing to note from Tables 1 and 2 is the massive benefit that the runway controllers have upon the take-off sequence and illustrates why controllers change the take-off order. The delay is almost always higher when the aircraft are not re-sequenced (dataset 2 being the exception), despite the relatively short time period over which the delay can accumulate. In dataset 2 the controllers must have had a reason for deviating from the first-come-first-served sequence, but it is impossible to determine what that was. Not only is the delay decreased by the re-sequencing, but fewer CTOT extensions are also often required. Consideration of the results shown later, in Tables 3 to 7, will also reveal that the schedule duration is also often much higher, which may lead to possible cumulative delays for later aircraft.

Considering the reliability of the take-off time prediction system, from the third and fourth columns, there are a number of reasons for discrepancies between the real and predicted take-off times. Firstly, take-off time prediction is usually pessimistic. It always assumes aircraft will need at least two minutes in the holding area. This high minimum taxi time is useful for ensuring that automatically created schedules allow plenty of time for aircraft, but is sometimes not so useful when assessing manual schedules. For example, if an aircraft actually only takes one minute then the predicted take-off time for that aircraft would be a minute later than the real take-off time. Additionally, separation rules mean that this prediction error can also be propagated through many of the predictions for later take-off times.

The second reason for discrepancies is that controllers can sometimes reduce certain separations, where they are not related to safety. Thirdly, although the take-off time prediction system assumes that aircraft always take-off as quickly as they can, sometimes real separations are slightly higher than necessary. For example a pilot may be slow to take-off or something may happen which delays the pilot. Finally, in the real-world take-offs rarely happen to the exact second that they could. Slight prediction errors can accumulate over a schedule.

Considering CTOT compliance and delay, in all cases the automated system, with full knowledge of the aircraft on the taxiways, beats the real schedules created by the controllers when they are evaluated using predicted take-off times. In all but three cases, it beats even the real take-off times.

Performance was poor on dataset 12. This dataset represented a quiet time in the morning. As the situation was very quiet aircraft did not need to take two minutes to cross the holding area, they could taxi through line up and take-off very quickly. The predicted take-off times were later than the real take-off times and both the real and predicted delays were very low, i.e. aircraft spent very little time in the holding area. In this circumstance, any scheduling under a two-minute minimum traversal time would be unable to out-perform the real take-off times. The fact that the performance on dataset 13 is much better shows that the problem is with the first 25 aircraft, i.e. the ones from when it is quietest.

In dataset 27, the total delay for aircraft is again very low. This is an indication that aircraft can take off without having a long wait. Again, the restrictions imposed on the system to control pilot workload have made it impossible for the system to match the performance of the real controllers in this case.

The decision support system was designed to help controllers cope with busy periods of the day. In both of these cases the airport was quiet enough that the controllers could cope without a decision support system anyway. The system could be designed to work in the same way: i.e. to allow less time for aircraft to cross the holding area and line up at quiet times of the day. The system may then appear to perform better in the quieter periods, but it is questionable whether this would be better in practice than suggesting easier to achieve schedules and letting the controller improve them manually, as these are the times the controller has time to do that.

Performance was also poor for dataset 7. Here, there is a large discrepancy between the real and predicted take-off times. It is obvious from an examination of the actual flight times that the controller had negotiated a temporary reduction in one of the separation rules back down to the level required for safety rather than the higher level that is normally imposed on this route to reduce downstream congestion. In this case, the controller obviously identified that there was going to be a problem with high congestion at the holding area and arranged to have the separation rules modified to account for it. There is no way the system could attain these levels of delay without altering the separation rules.

Our decision support system allows separation rules to be changed while running. So, a controller who negotiated a reduced separation rule for a while could inform the system of the change and allow it to take advantage of this fact. However, this was not assumed for these tests.

Table 2 also shows how the performance of the schedule decreases as the level of taxi time uncertainty increases. The high cost which the objective function applies to schedules which miss CTOTs means that the system has a large bias towards avoiding having to use CTOT extensions. Dataset 4 is a particularly good example of the behaviour when there are many aircraft with CTOTs. The system misses three CTOTs when there is no knowledge of the taxiing aircraft and only one when it is aware of taxiing aircraft. This means that something was being done in the holding area, in advance of the taxiing aircraft arriving, to allow the aircraft to meet the assigned CTOTs.

It is common if there are aircraft with tight CTOTs for the system to have to move other aircraft out of the way to let these past. The main way to do this is to assign the overtaken aircraft a longer, slower taxi path. The earlier the system is aware of having to clear a path through the holding area, the more opportunity there is to still attain a good schedule with the aircraft that are moved. As the level of uncertainty in taxi times is increased the system becomes aware of these aircraft later, so there is less warning time about tight CTOTs and the system may have to change the schedule later to accommodate them. In this case it is possible that the better schedules are no longer available and there is an intrinsic trade-off between having the low delay schedule or the schedule which meets the CTOT.

The system was designed assuming that taxiing aircraft are included and that fairly accurate taxi time predictions are available for aircraft. One constraint applied to the system is that holding area traversal paths will not be changed after aircraft have entered the holding area. Obviously, if taxiing aircraft are not included in the system then the slower paths will not get assigned to aircraft which will later need to be overtaken as the system will be unaware of the later aircraft that need to overtake them. If taxiing aircraft are not to be considered it would make sense to allow holding area paths to be changed even for aircraft within the holding area, providing the aircraft has not passed the node at which the paths diverge. This would be a simple change to the path allocation heuristic. It would, however, cause more work for controllers as they would then have to communicate the desired changes so we have not assumed it is possible in this work.

In our experiments, we observe that often, with tight CTOTs, the introduction of errors in taxi time prediction means that the delay is affected in preference to the CTOT compliance, as there is a lower penalty for doing this in the objective function. It is questionable whether controllers would be quite so keen on attaining CTOTs regardless of throughput loss or delay. This is especially the case if they know they have a number of permitted CTOT extensions still available. For this paper, however, we assume that CTOT compliance is our primary objective and note that the system usually manages to improve delay as well. In some datasets, such as 2 and 5 for example, the delay was little affected, so this behaviour is not uniform and appears to depend a lot on the number of CTOTs and how tight they are.

Datasets 26 and 27 are again from relatively quiet periods, at the start of a day, and show the advantages that can be gained from allowing aircraft to take less than two minutes in the holding area. Of course, a live decision support system would be able to take advantage of decisions the controller makes to spend less time in the holding area and have aircraft take off earlier.

In general, the system can cope with a relatively high error rate in the taxi times of aircraft and still benefit from knowledge of taxiing aircraft. However, better predictions usually mean better results so a good taxi time prediction system would be key for the success of the decision support system in a live environment.

Although the summary results tables only show the results for errors of 20%, 40% and 60%, errors of 10% to 90% were actually tested, in 10% increments. In order to see the full effects of the introduction of the taxi time prediction errors, Tables 3 to 7 show more information for sample datasets. In these tables, the rows represent the experiments that were performed and the columns give the results.

The ‘Manual, real’ column presents the results for the manual schedule using the real take-off times. The ‘Manual, predicted’ column presents the results for the manual schedule using the take-off time prediction system to predict take-off times. The ‘FCFS, predicted’ column presents results for the first-come-first-served schedule using the take-off time prediction system.

The next row shows the results for the system when it has full knowledge of taxiing aircraft. Following this are nine rows showing the results when various levels of errors are included in the taxi times. The final row shows the performance when no knowledge is given to the system about taxiing aircraft until they are almost at the holding area.

The columns of the table show the number of CTOT extensions required, the total delay for aircraft and the schedule duration. The duration of the schedule is the number of seconds between the first take-off and the last take-off, so is some measure of throughput. For each test the mean, minimum and maximum results for the one hundred executions of the system are shown.

Table 3: CTOT compliance, total delay and schedule duration for dataset 2

Run	CTOTs missed			Total Delay (s)			Schedule duration (s)		
type	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
Manual, real	4	-	-	27509	-	-	5236	-	-
Manual, predicted	4	-	-	25587	-	-	5050	-	-
FCFS, predicted	6	-	-	25668	-	-	5239	-	-
Full knowledge	0.00	0	0	13715	13715	13715	4870	4870	4870
10% uncertainty	0.00	0	0	13976	13715	14580	4870	4870	4870
20% uncertainty	0.00	0	0	14015	13715	14899	4870	4870	4870
30% uncertainty	0.00	0	0	14055	13715	14947	4870	4870	4891
40% uncertainty	0.00	0	0	14124	13715	15052	4871	4870	4902
50% uncertainty	0.00	0	0	14210	13756	15147	4871	4870	4902
60% uncertainty	0.00	0	0	14394	13715	16056	4871	4870	4902
70% uncertainty	0.01	0	1	14533	13715	16674	4874	4870	4902
80% uncertainty	0.00	0	0	14843	13756	17972	4878	4870	4943
90% uncertainty	0.04	0	2	15887	13809	21907	4893	4870	5063
No knowledge	0.00	0	0	15757	15757	15757	4870	4870	4870

Dataset 2 is interesting because the system finds schedules which do so much better than the manually produced ones. Contrasting datasets 9 and 24 shows how the difference in performance as the taxi time errors change is highly dependent upon the dataset. In dataset 9 the mean delay increases rapidly as the uncertainty increases, whereas in dataset 24 is stays relatively stable until high levels of uncertainty are applied. Datasets 4 and 34 are noteworthy as the CTOT compliances differ the most.

In most cases, the performance of the system decreases as the level of uncertainty increases. Not only does the mean performance decrease, but the worst case performance decreases extremely rapidly.

Table 4: CTOT compliance, total delay and schedule duration for dataset 4

Run	CTOTs missed			Total Delay (s)			Schedule duration (s)		
type	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
Manual, real	6	-	-	27541	-	-	4764	-	-
Manual, predicted	7	-	-	32616	-	-	4936	-	-
FCFS, predicted	13	-	-	54576	-	-	5776	-	-
Full knowledge	1.00	1	1	25506	23448	26370	4600	4540	4618
10% uncertainty	1.16	1	2	26287	23424	30449	4606	4540	4738
20% uncertainty	1.18	1	2	26586	25006	29874	4611	4540	4690
30% uncertainty	1.30	1	3	26996	23424	31119	4617	4540	4738
40% uncertainty	1.47	1	3	28107	25188	33060	4655	4540	4900
50% uncertainty	1.41	1	3	28036	25133	34880	4656	4540	4900
60% uncertainty	1.45	1	3	28196	24523	31847	4660	4540	4795
70% uncertainty	1.53	1	3	28724	24777	35495	4691	4540	4905
80% uncertainty	1.82	1	5	29153	24720	33300	4711	4540	4918
90% uncertainty	2.04	1	4	29741	23640	35118	4740	4540	5200
No knowledge	3.09	3	4	28989	28662	29022	4714	4714	4714

Table 5: CTOT compliance, total delay and schedule duration for dataset 9

Run	CTOTs missed			Total Delay (s)			Schedule duration (s)		
type	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
Manual, real	1	-	-	22631	-	-	4964	-	-
Manual, predicted	1	-	-	22177	-	-	5002	-	-
FCFS, predicted	1	-	-	29237	-	-	5420	-	-
Full knowledge	0.00	0	0	15549	15515	15635	4613	4613	4613
10% uncertainty	0.05	0	1	15712	15515	16533	4625	4613	4702
20% uncertainty	0.13	0	1	16029	15515	17733	4642	4613	4762
30% uncertainty	0.14	0	1	16405	15515	18507	4655	4613	4762
40% uncertainty	0.12	0	1	16692	15515	18078	4667	4613	4760
50% uncertainty	0.14	0	1	16949	15635	18841	4675	4613	4762
60% uncertainty	0.15	0	1	17156	15635	18946	4677	4613	4761
70% uncertainty	0.21	0	1	17413	15659	19341	4683	4613	4748
80% uncertainty	0.25	0	1	18029	15995	22432	4697	4613	4821
90% uncertainty	0.33	0	1	19200	15959	24484	4730	4613	4995
No knowledge	1.00	1	1	18735	18735	18735	4794	4794	4794

With sufficient error in the taxi time predictions the system actually performs worse than if taxiing aircraft were not considered at all.

The prediction error on each aircraft differed between executions of the experiments. The variation in performance across the different executions of each experiment indicates that the system is not only sensitive to the degree of uncertainty in the taxi times but also to which aircraft are affected at the time. It is not obvious from the results of these experiments whether it is possible to categorise the

Table 6: CTOT compliance, total delay and schedule duration for dataset 24

Run	CTOTs missed			Total Delay (s)			Schedule duration (s)		
type	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
Manual, real	1	-	-	17209	-	-	5665	-	-
Manual, predicted	2	-	-	20291	-	-	5752	-	-
FCFS, predicted	8	-	-	23563	-	-	6093	-	-
Full knowledge	1.00	1	1	14951	14951	14951	5661	5661	5661
10% uncertainty	1.00	1	1	15045	14951	15361	5677	5661	5692
20% uncertainty	1.00	1	1	15041	14951	15397	5674	5661	5692
30% uncertainty	1.00	1	1	15044	14953	15409	5677	5661	5692
40% uncertainty	1.00	1	1	15095	14953	15696	5682	5661	5718
50% uncertainty	1.03	1	2	15152	14953	16032	5684	5661	5718
60% uncertainty	1.07	1	2	15241	14963	16247	5683	5661	5718
70% uncertainty	1.08	1	2	15511	14965	17426	5688	5661	5718
80% uncertainty	1.14	1	3	16272	14981	20067	5704	5661	5888
90% uncertainty	1.19	1	4	17060	15145	23005	5714	5661	5888
No knowledge	1.00	1	1	15089	15089	15089	5661	5661	5661

Table 7: CTOT compliance, total delay and schedule duration for dataset 34

Run	CTOTs missed			Total Delay (s)			Schedule duration (s)		
type	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
Manual, real	3	-	-	22161	-	-	4446	-	-
Manual, predicted	5	-	-	25679	-	-	4624	-	-
FCFS, predicted	9	-	-	38327	-	-	5215	-	-
Full knowledge	0.00	0	0	18010	17262	18748	4348	4348	4348
10% uncertainty	0.15	0	1	18095	17262	18931	4354	4348	4408
20% uncertainty	0.28	0	2	18363	17535	19621	4360	4348	4430
30% uncertainty	0.57	0	2	18458	17262	20021	4380	4348	4528
40% uncertainty	0.60	0	3	18620	17682	20053	4381	4348	4468
50% uncertainty	0.82	0	2	18568	17595	20623	4391	4348	4490
60% uncertainty	0.80	0	3	18815	17682	21188	4396	4348	4476
70% uncertainty	0.93	0	3	19018	17611	21874	4402	4348	4536
80% uncertainty	0.97	0	2	19338	17622	22683	4416	4348	4648
90% uncertainty	1.04	0	3	19943	17262	24964	4442	4348	4768
No knowledge	1.00	1	1	19629	19629	19629	4490	4490	4490

areas where the system is most sensitive to errors in the taxi times. Further experimentation and investigation will be necessary to determine which the system is most sensitive to having accurate taxi time information for. It may be aircraft with tight CTOTs, aircraft of specific weight classes, aircraft with specific departure routes, or aircraft types which are in short supply at the holding area at the time. All of these are possibilities, or possibly a combination of these factors. If this could be determined, then it may be possible to prioritise these aircraft during the taxiing, improve the predictability of their taxi times and improve the performance of the system as a whole.

Together, Tables 1 and 2 allow an assessment of the value of the automatically produced sequences in terms of total delay and compliance with CTOTs. In order to ensure that the system meets all of the objectives, the workload involved to achieve the re-sequencing and the equity of the sequencing must also be considered. The deterministic path allocation method ensures that the re-sequencing is always performed in the simplest way possible. The positions of aircraft that were predicted using the holding area position prediction system were stored and played back using a graphical display. This allowed visual confirmation that the paths used and sequencing decisions made were acceptable.

The last objective to consider relates to the equity of delay. The major problem with trying to determine whether a schedule is equitable enough is that there are so many factors to consider. The positional delay, defined here as the number of positions that the aircraft is later in the take-off sequence than it was in the arrival sequence at the holding area, is considered in Table 8. The positional delay of an aircraft is related to the equity of the sequencing since it approximates the number of other aircraft which overtake it. For each dataset the maximum positional delay, number of aircraft which have a positional delay, the total positional delay and the sum of the square of the positional delay for each aircraft is given in Table 8 for both the manual and automated sequences. (The values given for the automated sequences are again the mean values of 100 executions.)

In general, a more equitably sequencing will have a lower total positional delay for aircraft. (The first-come-first-served sequences is the most equitable and has zero positional delay for all aircraft.) The sum of the square of the positional delay is useful as such a sum applies a higher weighting to larger positional delays, so, for two sequences with the same total positional delay, the one with the lower total sequared positional delay could be said to share the delay out more equitably. The maximum positional delay is another good measure, as it explicitly shows whether extreme penalties have been applied to some aircraft. However, the problem is that a large positional delay may actually be acceptable, if there is a reason for it.

Space prohibits an individual consideration of the equity of delay for each individual aircraft but it is worth making some observations. Firstly, the large positional delays in some of the datasets were due to aircraft having to wait for the start of a CTOT slot, and other aircraft overtaking them while they were waiting. It is obvious from the real take-off times that some of these CTOTs were actually re-negotiated, allowing the aircraft to take off earlier. Improved sequencing of aircraft can mean that more can take off before the start of a CTOT, leading to a better sequence but a higher positional delay for the overtaken aircraft even if the actual seconds delay was unaffected.

A second observation is that the system often delays less aircraft but for slightly longer each than

Table 8: Maximum positional delay, total positional delay, number of aircraft delayed and total squared positional delay for real and automatically generated schedules

		Manual	Results			Automated	Results	
DS	Maximum	Number	Delay	Squared	Maximum	Number	Delay	Squared
1	5	17	34	86	5	12	25	72
2	5	23	45	111	5	12	25	75
3	7	25	59	195	7	15	36	138
4	7	25	62	212	10	19	64	370
5	4	23	48	118	6	14	38	146
6	8	20	43	147	7	17	41	170
7	3	16	25	47	5	12	32	108
8	2	13	18	28	3	12	19	37
9	6	18	33	85	8	12	32	164
10	6	23	40	96	9	17	44	210
11	2	13	19	31	5	13	19	43
12	4	15	29	75	7	15	32	108
13	4	15	25	59	3	17	27	51
14	3	12	20	40	3	12	22	48
15	3	11	15	25	6	10	17	51
16	4	16	27	61	8	13	32	138
17	4	15	27	63	9	15	43	244
18	5	9	16	42	3	13	20	40
19	1	10	10	10	2	7	10	16
20	5	17	32	80	5	15	26	64
21	5	20	44	124	6	16	33	106
22	5	18	30	68	7	15	33	132
23	3	21	27	41	2	12	16	24
24	6	16	36	116	5	15	31	83
25	5	15	30	88	4	14	27	65
26	2	10	14	22	2	9	12	18
27	2	9	12	18	2	7	8	10
28	6	14	31	101	5	13	30	112
29	6	20	38	106	10	16	38	164
30	4	17	32	74	6	11	23	80
31	4	15	25	69	6	13	23	71
32	3	21	25	77	6	11	29	111
33	5	23	48	140	9	15	41	237
34	5	20	46	140	9	16	44	218
35	4	19	26	46	5	17	29	67

the controllers do. However, the values are not as disparate as might have been expected given the greater knowledge the system has about taxiing aircraft. It is possible that controllers would sometimes adopt less equitable sequences, in exchange for throughput or CTOT compliance benefits if they were given a greater knowledge of the taxiing aircraft, especially if the benefits of doing so had already been proven. The controllers are limited to some degree in the amount of re-sequencing they can perform at present, due to the limited visibility of the taxiing aircraft.

A third observation is related to CTOTs. The high cost associated with a CTOT miss in our objective function means that the system will effectively prioritise CTOTs. This can, on occasions, lead to aircraft without CTOTs getting a slightly higher positional delay than may otherwise be the case, when they are overtaken by those with tight CTOTs.

A fourth observation is that, any aircraft which was consistently delayed until the end of the dataset would have to have a positional delay of at least 25 in at least one experiment as the overlapping nature of the datasets means it would be in the first half of at least one dataset. However, the maximum positional delay is only ten places, and those were for the reasons of CTOT compliance, considered above.

In conclusion, even with the occasional large delay awaiting the start of a CTOT slot, the positional delay in the automated sequences is still not excessive. In particular, the algorithm is not indefinitely delaying problematic aircraft, as may be feared without a consideration of individual delay or equity of delay. The results are roughly comparable with those the controllers actually use, once larger delays waiting for CTOT slots are taken into consideration.

9 Conclusions

The job of the runway controller at Heathrow is an extremely busy and complicated one. It is, obviously, useful to examine the possibility of providing a decision support system to aid the controller in this demanding task. We outlined the algorithms for a decision support system to aid the controller in [7]. In this paper, we have presented a number of improvements to the decision support system and evaluated its performance when working with imprecise information.

The improvements included provide some guarantee of the value of the schedule. For example, this helps to avoid the situation where the heuristic search misses obvious improvements to the schedule. The effects of holding area movement, or delays caused by waiting for other aircraft to move, have been explicitly included in the model and produced schedules. The possibility of swapping between two identical or similar cost schedules over time has been explicitly accounted for, increasing the stability of the schedules. Additionally, the simulation used to evaluate the system has also been improved. Taxiway congestion is now considered and controlled, allowing a guarantee that the taxiways will not be unnecessarily congested - if there is room in the holding area for aircraft then it will be utilised.

Performing experiments upon smaller datasets has allowed analysis of the differing system performance under different levels of system load. Previous results, for example in [7], have shown that the decision support system out-performs controllers when used on half-day periods. The results in this paper show that the overall improvement of the automatically produced schedules over the manually produced schedules is far greater at some times than at others. Along with significant CTOT compliance improvements, the tests predict up to a 50% reduction in delay in one of the tests (46% vs the predicted times in that case). However, at other, usually quieter, times the controllers can out perform the automated system as they can adopt behaviour which the system will not permit. This is a valuable result as it shows that a better overall performance should be achieved by using the decision support system to help controllers at busy times and the controller utilising extra flexibility at quieter times, when the workload permits.

If the decision support system is to be used at quieter times, there would need to be an option for a controller to specify that higher workload activities could be permitted or that the minimum holding area traversal times could be reduced. The value of doing this is questionable, however, as the controllers already perform very well at quieter times and the time pressures are less so the controller would be far less likely to refer to a decision support system anyway.

At times, controllers can safely reduce certain separation rules. As these are at the discretion of the controllers the system does not assume this will be done. It would be a simple matter to provide controllers with an interface to inform the system that they are permitting reduced separations so that the system could take advantage of these when scheduling. As this was not done in the tests presented in this paper, in some cases this has made a large difference between the actual and predicted take-off times of the aircraft.

The results presented here indicate that there is an advantage to be gained from considering taxiing aircraft when determining a take-off order. They also show that the system we have designed is capable of using this information quickly enough to provide real-time decision support for controllers. This is so even if the taxi times are relatively inaccurate. However, to gain the full benefit of a decision support system, an accurate taxi time prediction system is vital. The more accurate the predictions, the better the system can perform.

Comparing the behaviour of the decision support system and controllers it is obvious that the system puts a much higher priority on CTOT compliance than the controllers, as the controllers miss more CTOTs despite the importance of meeting them. In fact, without knowing about the CTOTs on taxiing aircraft, the controllers could only hope to meet this level of CTOT compliance if they ensured that there was always a clear path through the holding area, just in case an aircraft with a tight CTOT arrived and needed to pass those already there. This is not practical given the lack of space, and would severely restrict the re-sequencing possibilities. By knowing about tight CTOTs in advance (while aircraft are still taxiing) the decision support system can ensure that CTOTs are met while also keeping the delay low. The controller view is sensible given the constraints they have to work under. Since they are permitted a number of extensions, they are happy to use them if it will help to keep the throughput high and delay low. The danger with unnecessarily missing CTOTs, however, is that there can never be visibility of how many extensions will be needed later in the day. Although a controller can decide to take the risk, it is not wise for a decision support system to do so, so CTOTs compliance has to be prioritised by a decision support system.

The results in this paper show that the system is able to apply a very high penalty to CTOT non-compliance, but still obtain a much lower delay for aircraft, providing that the taxi time predictions are relatively accurate. If the information about taxiing aircraft is missing or less precise the high penalty the system applies to missing CTOTs starts to detrimentally affect the delay. If taxi times could not be predicted accurately enough, then the higher priority that has been given to CTOT compliance should perhaps be reconsidered. Future work could investigate the benefits of changing the CTOT/delay trade-off.

We conclude from these results that there is a major benefit to be gained at Heathrow from implementing a decision support system for the runway controllers if the system continues to operate the way it currently does, with the runway controllers performing just-in-time scheduling at the holding areas. The main benefits are, however, dependent upon the development of an accurate taxi time assessment system.

10 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] J. Abela, D. Abramson, M. Krishnamoorthy, A. de Silva, and G. Mills, “Computing optimal schedules for landing aircraft,” in *proceedings of the 12th National Conference of the Australian Society for Operations Research, Adelaide, July 7-9*, pp. 71–90, 1993.

[2] I. Anagnostakis, J.-P. Clarke, D. Böhme, and U. Völckers, “Runway operations planning and control, sequencing and scheduling,” in *proceedings of the 34th Hawaii International Conference on System Sciences (HICSS-34), Maui, Hawaii, January 3-6*, 2001.

[3] I. Anagnostakis and J.-P. Clarke, “Runway operations planning, a two-stage heuristic algorithm,” in *proceedings of the AIAA Aircraft, technology, Integration and Operations Forum, Los Angeles, CA*, October 2002.

[4] I. Anagnostakis and J.-P. Clarke, “Runway operations planning: a two-stage solution methodology,” in *proceedings of 36th Hawaii International Conference on System Sciences (HICSS-36), Hawaii*, January 2003.

[5] J. A. D. Atkin, E. K. Burke, J. Greenwood, and D. Reeson, “A meta-heuristic approach to aircraft departure scheduling at London Heathrow airport,” presented at *the 9th International Conference on Computer-Aided Scheduling of Public Transport (CASPT04), San Diego, USA*, August 2004.

[6] J. A. D. Atkin, E. K. Burke, J. Greenwood, and D. Reeson, “An examination of take-off scheduling constraints at London Heathrow airport,” presented at *the 10th International Conference on Computer-Aided Scheduling of Public Transport (CASPT06), Leeds, UK*, June 2006.

[7] J. A. D. Atkin, E. K. Burke, J. Greenwood, and D. Reeson, “Hybrid meta-heuristics to aid runway scheduling at London Heathrow airport,” *Transportation Science*, vol. 41, no. 1, pp. 90–106, 2007.

[8] J. A. D. Atkin, E. K. Burke, J. Greenwood, and D. Reeson, “The effect of the planning horizon and the freezing time on take-off sequencing,” in *proceedings of Second International Conference on Research in Air Transportation (ICRAT2006), Belgrade, Serbia and Montenegro*, June 2006.

[9] BAA Heathrow, “Flight evaluation report 2004/2005.” Available at : `http://www.heathrowairport.com/assets/B2CPortal/Static%20Files/New2005Booklet.pdf`.

[10] C. Barnhart, P. Belobaba, and A. R. Odoni, “Applications of operations research in the air transport industry,” *Transportation Science*, vol. 37, no. 4, pp. 368–391, 2003.

[11] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, “Scheduling aircraft landings - the static case,” *Transportation Science*, vol. 34, pp. 180–197, 2000.

[12] J. E. Beasley, J. Sonander, and P. Havelock, “Scheduling aircraft landings at London Heathrow using a population heuristic,” *Journal of the Operational Research Society*, vol. 52, pp. 483–493, 2001.

[13] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson, “Displacement problem and dynamically scheduling aircraft landings,” *Journal of the Operational Research Society*, vol. 55, pp. 54–64, 2004.

[14] L. Bianco, P. Dell’Olmo, and S. Giordani, “Minimizing total completion time subject to release dates and sequence-dependent processing times,” *Annals of Operations Research*, vol. 86, pp. 393–416, 1999.

[15] A. T. Ernst, M. Krishnamoorthy, and R. H. Storer, “Heuristic and exact algorithms for scheduling aircraft landings,” *Networks*, vol. 34, pp. 229–241, 1999.

[16] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Computers and Operations Research*, vol. 13, pp. 533–549, 1986.

[17] F. Glover and M. Laguna, *Tabu Search.* Kluwer Academic Publishers, Boston/Dordrecht/London, 1997.

[18] J.-B. Gotteland and N. Durand, “Genetic algorithms applied to airport ground traffic optimization,” in *proceedings of the 2003 Congress on evolutionary computation*, (Canberra, A.C.T), 2003.

[19] H. R. Idris, B. Delcaire, I. Anagnostakis, W. D. Hall, J.-P. Clarke, R. J. Hansman, E. Feron, and A. R. Odoni, “Observations of departure processes at Logan airport to support the development of departure planning tools,” in *proceedings of the 2nd USA/Europe Air Traffic Management R&D Seminar*, 1998.

[20] R. A. Leese, A. Craig, R. Ketzscer, S. D. Noble, K. Parrott, J. Preater, R. E. Wilson, and D. A. Wood, “The sequencing of aircraft departures,” study group report from the *40th European Study Group with Industry, Keele*, 2001.

[21] G. F. Newell, “Airport capacity and delays,” *Transportation Science*, vol. 13, no. 3, pp. 201, 41p,, 1979.

[22] D. A. Trivizas, “Optimal scheduling with Maximum Position Shift (MPS) constraints: A runway scheduling application,” *Journal of Navigation*, vol. 51, pp. 250–266, 1998.

[23] P. van Leeuwen, H. Hesselink, and J. Rohling, “Scheduling aircraft using constraint satisfaction,” *Electronic notes in theoretical computer science*, vol. 76, 2002.

[24] C.-L. Wu and R. E. Caves, “Research review of air traffic management,” *Transport Reviews*, vol. 22, no. 1, pp. 115–132, 2002.