ON-LINE DECISION SUPPORT FOR TAKE-OFF RUNWAY SCHEDULING AT LONDON HEATHROW AIRPORT

Jason Adam David Atkin, BSc, PGCE

Thesis submitted to The University of Nottingham for the degree of Doctor of Philosophy

March 2008

Abstract

The research problem considered in this thesis was presented by NATS, who are responsible for the take-off runway scheduling at London Heathrow airport. The sequence in which aircraft take off is very important and can have a huge effect upon the throughput of the runway and the consequent delay for aircraft awaiting take-off. Sequence-dependent separations apply between aircraft at take-off, some aircraft have time-slots within which they must take-off and all re-sequencing performed by the runway controller has to take place within restrictive areas of the airport surface called holding areas.

Despite the complexity of the task and the short decision time available, take-off sequencing is performed manually by runway controllers. In such a rapidly changing environment, with much communication and observation demanded of the busy controller, it is hardly surprising that sub-optimal mental heuristics are currently used. The task presented by NATS was to develop the decision-making algorithms for a decision support tool to aid a runway controller to solve this complex real-world problem.

A design for such a system is presented in this thesis. Although the decision support system presents only a take-off sequence to controllers, it is vitally important that the movement within the holding area that is required in order to achieve the re-sequencing is both easy to identify and acceptable to controllers. A key objective of the selected design is to ensure that this will always be the case. Both regulatory information and details of controller working methods and preferences were utilised to ensure that the presented sequences will not only be achievable but will also be acceptable to controllers.

A simulation was developed to test the system and permit an evaluation of the potential benefits. Experiments showed that the decision support system found take-off sequences which significantly reduced the delay compared with those that the runway controllers actually used. These sequences had an equity of delay comparable with that in the sequences the controllers generated, and were achieved in a very similar way. Much of the benefit that was gained was a result of the decision support system having visibility of the taxiing aircraft in addition to those already queueing for the runway. The effects of uncertainty in taxi times and differing planning horizons are explicitly considered in this thesis. The limited decision time available ensures that it is not practical for a runway controller to consider as many aircraft as the decision support algorithms can.

The results presented in this thesis indicate that huge benefits may be possible from the development of a system to simplify the sequencing task for the controllers while simultaneously giving them greater visibility of taxiing aircraft. Even beyond these benefits, however, the system described here will also be seen to have further potential benefits, such as for evaluating the effects of constraints upon the departure system or the flexibility of holding area structures.

Acknowledgements

My thanks and appreciation go to my supervisor, Professor Edmund Burke, for his help, guidance and support throughout this work, for obtaining the funding for the project and for giving me the opportunity to join the research group. My gratitude is also deserved by the various members of the ASAP research group who have provided help, general advice and often a sounding-board during my time in the group. My particular thanks go to Steven Gustafson, Dario Landa-Silva and Graham Kendall for their help in understanding the research process and the academic world.

My thanks and appreciation must go to John Greenwood of NATS (formerly National Air Traffic Services) for his help and guidance throughout this project. Not only did he ensure that I received the information necessary to bring the project to a successful conclusion, but he was always happy to give feedback. His knowledge of both the operational research side and the air traffic control side of the problem were invaluable.

My thanks go to the helpful and friendly controllers at Heathrow, and in particular to Dale Reeson. Without their help I would have been unlikely to understand the problem in sufficient depth and could not have understood the controller point of view sufficiently to be able to design a system to produce acceptable take-off sequences. Further thanks are deserved by the runway controllers at Gatwick, who were kind enough to give me an insight into how mixed mode runway usage was handled there.

My thanks also go to those at EPSRC and NATS who were responsible for funding this project, and to the Smith Institute for Industrial Mathematics and Systems Engineering for bringing NATS and The University of Nottingham together for this project. Without any of these organisations this project would not have existed.

Finally, but by no means least, my thanks go to my wife, Rachel, for her unending patience and support through the entirety of this work, but particularly through the writing up period.

Dedicated to my wife, Rachel, for her constant and unwavering support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background And Motivation

London Heathrow airport is one of the busiest international airports in the world[1] but only has two runways available for use at the moment[2]. Restrictions upon the airport mean that only one of the runways can be used for departures at any one time. Which runway is used, and in which direction, depends upon the time of day and the wind direction. This severely restricts the take-off rate at this busy airport and means that good take-off sequencing is vitally important.

The departure system at an airport can be considered to consist of a number of distinct stages. Once loaded with passengers, luggage and fuel, a pilot will perform final checks with a ground movement planner. Once ready to push back from the stand, control will be handed over to a ground movement controller (GMC) who will grant permission to do so at the appropriate time. There are currently two GMCs at Heathrow, controlling the taxiing of aircraft on different parts of the taxiways. The GMCs aim to safely move the aircraft around the airport, to get them to their destinations as quickly and easily as possible. Departing aircraft will usually be heading from their stands towards the current departure runway and arriving aircraft from the current arrival runway to their allocated stands. There may also be aircraft being towed around the airport, congesting the taxiways and complicating the ground movement problem.

At the end of each runway is a holding area, within which the waiting aircraft are sequenced for take-off. Departing aircraft will be directed around the taxiways to one of the entrances of the holding area for the current departure runway. At this point, control of these aircraft will be handed over to the runway controller for that runway. The runway controller has to determine the sequence in which the aircraft will take off. The constraints upon and objectives of the take-off sequencing are complex and are detailed in chapter 2. Briefly, the controller has to consider the current positions of the aircraft in the holding area, the sequence-dependent separations required between aircraft at take-off and the constraints imposed to enforce down-

[1] According to Airports Council International (http://www.airports.org) figures, it was third in the rankings for total number of passengers in each of the years 2002 to 2006.

[2] In contrast, the two airports with more passengers per year are Hartsfield-Jackson Atlanta, with five runways, followed by Chicago O'Hare airport with six runways, although the runways at O'Hare intersect so cannot all be used at once.

stream flow control measures. Taking all of this into consideration, a desired take-off sequence is determined and enacted.

The runway controller is extremely busy since he/she not only has to sequence the aircraft but also has responsibility for them until they leave the local airspace. The controller currently has to solve the take-off sequencing problem manually. As he/she has to spend a lot of the time observing aircraft through the windows of the control tower while talking to pilots, there is very little time available for actually considering the sequencing of aircraft. Controllers tend to use pattern matching skills to perform the task and know from much practice which sub-sequences are likely to give good take-off sequences and which will not. Given the tight time constraints upon controllers and the complexity of the problem, requiring a manual solution to the problem is highly undesirable and the methods used can result in lower quality solutions in the long term.

1.2 Aims

NATS (formerly National Air Traffic Services) Ltd are responsible for the ground and air traffic control at Heathrow. Together with the Engineering and Physical Sciences Research Council (EPSRC) they funded the research presented in this thesis, through the Smith Institute for Mathematics and Systems Engineering. Two related questions were posed. The first question was whether it was possible for a computer to solve the take-off sequencing problem, given all of the constraints imposed upon the sequencing in the real situation. The second, dependent question was to determine how this could be done if it was possible.

This thesis answers both of these questions by presenting the algorithms for a decision support system which can solve the take-off sequencing problem fast enough to be of use for real controllers. A key aim of a decision support system is to make suggestions that will be adopted by the controllers rather than automating the job that the runway controller is currently performing. This means that the algorithms must present solutions which a controller will consider to be sensible rather that just possible. This is a key motivation for the solution method that has been developed.

The aim of this research is not to change the working methods of the controllers but is instead to aid them in the performance of the re-sequencing task by suggesting improved take-off sequences which will be acceptable to them. The effects of these aims can be observed in section 1.3 below. An improved take-off sequence in this case is a sequence which will improve the predicted delay for aircraft, while complying with as many take-off time-slots as possible, as discussed in section 2.10. An acceptable sequence can be summarised as one where both the method of achieving the sequencing will be obvious to the controllers, and the way in which it is achieved will be acceptable to them.

1.3 Scope

The details of the problem presented by NATS have the following effects upon the scope of the research considered in this thesis:

1.3.1 Only the take-off sequencing problem is considered in this thesis

Air transportation is an extremely varied and complex subject area. Each of the stages of movement for passengers may involve multiple organisations and much research has been performed into the associated problems. These include the flow management problem for the regional or local airspace [62], and ground handling issues at the airport such as the staffing and positioning of check-in desks [167] or the allocation of stands to aircraft in order to reduce passenger movement or airline inconvenience [70]. Although the application of optimisation methods to the terminal design and operations [137, 165], the turn-around process [172], or gate allocation [70] can be expected to bring benefits in terms of lower costs or lower resource requirements (for example, a reduction in the turn-around time or the number of tugs required), these areas are beyond the scope of the problem considered in this thesis. By necessity, this thesis focuses upon a very small portion of the air transportation subject area, being related solely to the take-off sequencing problem.

Where the take-off sequencing problem relates to other parts of the air transportation system, there may be a brief discussion since it is sometimes necessary to understand the surrounding situation in order to better understand the reasons for various constraints and characteristics of the problem. For example, the ground movement of the airport is important as it determines when and where aircraft will arrive at the holding area. Similarly, the structure of the local airspace affects the take-off sequencing problem since various constraints are applied to the take-off sequencing problem to help with flow control and maintaining inter-aircraft separations in the airspace.

1.3.2 Only decision support for the runway controller is considered

The re-sequencing is currently performed at the holding areas rather than at the stands due to the level of uncertainty in ready times and taxi times for aircraft. The aim of this thesis is to design the decision making element of a decision support system for the runway controller, allowing him/her to better re-sequence the aircraft, while keeping the current working methods and preferences.

It may eventually be possible to further improve the take-off sequences by re-sequencing aircraft while they are at the stands, thus avoiding many of the issues related to holding area congestion. Although this could potentially provide further improvements and is discussed further in section 3.9, this is outside of the scope of the problem considered in this thesis.

1.3.3 The ground controller will not help with the re-sequencing

The ground movement controller is responsible for directing the aircraft from the stands along the taxiways to the current holding area(s). There are usually multiple holding area entrances to which a ground movement controller can deliver an aircraft. When time and workload permits, it is possible for a ground movement controller to perform intelligent pre-sequencing of aircraft to deliver them to an appropriate entrance so that the re-sequencing possibilities within the holding area are improved. For this research it is assumed that the runway controller performs the entire re-sequencing task within the holding areas, and that the ground movement controller does not pre-sequence aircraft. Instead, it is assumed that the ground movement controller always delivers aircraft to the most convenient holding area entrance for the stand that the aircraft has taxied from.

1.3.4 The arrival and departure systems will be treated as if they are independent systems

Since the re-sequencing is performed in the holding area, it is assumed that the departure system and arrival systems have been de-coupled. This means that the take-off sequencing can be performed without consideration of the arrival stream. At Heathrow, the coupling between the processes takes place at the stands, on the taxiways and during runway crossings. Both arrivals and departures use the same taxiways so the presence of arrivals could affect the taxi times for departures. However, in this research, the details of the aircraft on the taxiways are considered to be inputs to the system, rather than as a part of the problem to be solved. It is assumed that any taxi time prediction system to provide taxi time estimations for aircraft will take account of arrivals on the taxiways. By the time the aircraft reach the holding area they have passed all points of contention with arrivals. If the scheduling were performed earlier than at the holding area, then there would be a need to also consider the movement of arrivals around the airport.

It is necessary for aircraft to cross the southern runway to reach terminal four. If the southern runway is being used for arrivals, then this may have the effect of increasing the uncertainty in the taxi time for the departures from terminal four, as they may be delayed awaiting a suitable time for crossing the runway. This is assumed to be considered by the taxi time prediction system. If the southern runway is being used for departures then it is assumed that runway crossings can occur in the natural gaps in the take-off sequence. This assumption is believed to be valid as it can be observed that two minute gaps are frequent in the departure schedules. Two-minute gaps are often required due to wake-vortex separations or departure route separations, as described in section 2.3. Larger wake vortex separations can never be totally avoided when there are aircraft of different weight categories taking off, and larger departure route separations also occur quite frequently due to the uneven distribution of departures across the various departure routes.

1.3.5 Runways and departure routes will have been pre-allocated to aircraft

Since London Heathrow usually has a single take-off runway in operation at any time of the day, it is assumed for this research that that segregated mode is in operation, so that all aircraft take-off from the same runway, and that the runway in use is pre-determined by the time of day at which the take-off occurs. The runway allocation problem that is sometimes considered in arrival or take-off sequencing, [26], is therefore not relevant for this research.

1.3.6 Only the take-off sequence will be presented to controllers

It was determined, as a part of the problem presented by NATS to be solved, that only a take-off sequence will be presented to a controller, rather than presenting the details of the method by which it can be achieved. This decision ensures that the effects of the user interface issues are reduced. In this case the user interface design could be as simple as annotating the electronic strips which the controllers use to plan a take-off sequence with a suggested take-off position number. The alternative would be to somehow present the controllers with a diagram of how to perform the re-sequencing, in terms of the paths that aircraft should take and the sequence in which movement should occur. This would introduce complex user interface issues, and risk actually complicating the sequencing task for the runway controller. Such issues are beyond the scope of the problem considered in this thesis.

1.3.7 The method to achieve the re-sequencing must be obvious

As a consequence of only presenting the take-off sequence to the controllers, it is extremely important that the method by which any presented take-off sequence can be achieved must be obvious to the controller. This means that take-off sequences for which this does not hold can and should be rejected.

1.3.8 Only a single take-off sequence should be presented

The decision support system for the runway controller is designed to aid in a real-time situation. It has been decided that, for the purposes of this research, it is inadvisable to present a controller with multiple schedules to choose from, since the controller is already working under very tight time pressures. The aim of this research is to simplify the task for the runway controller and presenting multiple schedules would instead risk complicating it. This decision support system should present the runway controller with a single advised take-off order. This means that the objectives have to be combined into a single compound objective. This decision prevents the need for multi-objective solution methods such as those described in [65].

1.3.9 All required holding area movement must be acceptable to controllers

The movement which is required within the holding area has a strong influence upon whether any desired re-sequencing is seen as acceptable or sensible by a controller. Although the decision support system is not intended to specify the way in which the re-sequencing is to be achieved, it is important to verify that it is achievable in a manner which would be acceptable to a controller. This is an important scoping decision since controllers have strong preferences for avoiding what they perceive as excessive workload for pilots. The effects of this scoping decision are discussed further in section 2.8.

1.4 Non-disclosure Agreement

There is a non-disclosure agreement covering this work such that release of information must be approved by NATS. This means that some material cannot be reported due to confidentiality issues. In particular, the real data that was used for the experiments performed for this thesis cannot be made available.

1.5 The Importance Of The Research

With cheaper and cheaper flights year on year, air travel is becoming ever more popular, both for business and leisure. According to the International Air Transport Association[3] (IATA) annual report for 2006, [112], passenger numbers have risen by one quarter in the two years up to 2005. The BAA[4] flight evaluation report for Heathrow (2004/2005), [20], also shows the increasing trend in the number of air transport movements year on year. Control of pollution, both in terms of noise and fuel usage, is important at Heathrow and reducing the time that aircraft are awaiting take-off, with their engines running, is a key motivator for the research presented in this thesis.

Passengers have preferences for specific flight times, which often depend upon the destination distance and timezone difference. Competition between airlines ensures that flights are not, therefore, evenly spread throughout the day. Furthermore, the departure route assigned to an aircraft is often dependent upon the destination of the aircraft, so the use of specific departure routes is more common at certain times of the day than at others. As the separations required between aircraft at take-off are larger for aircraft on similar routes than for aircraft on dissimilar routes, this predominance of aircraft for specific departure directions can reduce the overall throughput and make the take-off sequencing even more important than might otherwise be the case.

Not only do passenger preferences tend to prevent an even spread of departures, but so do airline preferences. Airlines often operate hub-and-spoke networks, with one or more main

[3] IATA web site: http://www.iata.org/
[4] BAA web site: `http://www.baa.com/`

airports from which they fly to a number of other airports. This kind of schedule structure allows airlines to provide flights between many different airports by a single transfer between flights at the hub airport. In order to reduce the inconvenience for passengers, it is common to schedule aircraft such that multiple aircraft arrive at the hub at similar times, are on the ground at the same time, then take off very soon after each other. This helps to get a more even spread of transfer times, limiting the time passengers have to wait for connections without unnecessarily increasing the risk of missing connections. The consequent grouping or clustering of flights creates a less even departure flow, with peaks and troughs, which can result in an increased delay for aircraft leaving in the departure peaks. Increasing demand for flights, the increasing cooperation between airlines and the increasing use of automation in schedule generation is only likely to increase the size of departure peaks, making the departure flow even less smooth than it currently is.

At the moment there are only two runways available for use at Heathrow, although a government white paper, [66], discusses the possibility of a third runway in the future. The main concerns for building a third runway are, however, the noise and pollution issues. The possibility of a third runway depends upon being able to manage both noise and air pollutants (for example, Nitrogen Dioxide emissions), hence the importance of limiting the delay for aircraft. Furthermore, as a third runway would not be expected to come into operation until 2015-2020, better utilisation of the runways would be useful in the meantime for helping to cope with the increasing demands for air travel.

London Heathrow airport is situated close to residential areas and is affected by a number of noise control agreements, such as the Cranford Agreement, a westerly preference and Runway Alternation. All of these are explained in [20] and are implemented in order to control the noise level for residents who live near the flight paths. The effect of many of these is to limit the throughput of the airport and/or increase the complexity of the sequencing problem faced by runway controllers.

The primary effects of the agreements upon departures are to prevent simultaneous usage of both runways for westerly departures and to prevent the northern runway from normally being used at all for easterly departures. This is detrimental for departure delay as separation rules for aircraft at take-off and landing ensure that mixed mode, where both runways are used for both arrivals and departures, is usually more efficient than segregated mode, where a runway is used for only arrivals or departures.

Given the various constraints upon the departure system and the increasing demand for air travel, the importance of the decisions made by the runway controller can only be expected to increase. The results presented in this thesis predict that a decision support system could enable a busy runway controller to perform even better than at present. The results further show that this benefit is made possible by giving controllers increased visibility of future departures, enabling the current sequencing at any time to help alleviate delay for future departures. The importance of this research will be found in a predicted reduction in delay for aircraft and the consequent reduction in pollution, cost and passenger dissatisfaction.

For the first time, all of the major constraints upon the take-off sequencing problem at the holding area are considered. At Heathrow, the high throughput means that a bad sequencing, that introduces unnecessary delays into the take-off sequence, can unnecessarily delay a high number of aircraft. In addition, the location of the airport, near to busy airspace, means that the departure sequencing is unusually constrained, as observed in sections 2.3 to 2.5. There is usually very little, if any, slack to allow for bad sequencing, so the sequencing is vitally important.

The results in this thesis illustrate the fact that, although the sequencing and physical movement problems are both difficult problems to solve independently, the heuristic and meta-heuristic methods used here allow the combined problem to be solved quickly, even fast enough to be used in an online decision support system.

Ultimately, the potential delay benefits from implementing such a decision support system as described in this thesis are so great that it is easy to justify the necessity of providing an automated tool to aid the controller, to reduce the complexity for the controller in some other way, or to find a method by which the controller can gain the time to consider the taxiing aircraft in the take-off sequence.

1.6 Overview Of The Thesis

Following this introduction, the take-off sequencing problem at London Heathrow airport is introduced in chapter 2. The reasons for re-sequencing the aircraft taking off are explained and the objectives for the controller are considered. There are various constraints upon the sequences that can be created and upon the take-off times that are achievable for aircraft. Each constraint is considered and detailed, including the important ones that result from the structure of the physical holding areas, which are usually ignored in the take-off sequencing problems.

The problem considered in this thesis is a combination of a sequencing and a physical movement problem (usually called the *'control problem'*) and any solution method has to solve both problems. Some associated problems have been considered previously and a summary of the most relevant related research is presented in chapter 3.

The aim of this thesis is to specify the underlying algorithms for a decision support system to aid a runway controller. The triplet and alternative graph models are presented in chapter 4, to aid in understanding the complexity of the problem. The chapter continues with a consideration of various solution approaches which coult be applied. Chapter 4 ends with an overview of the designed decision support system, introducing a problem decomposition between the take-off sequencing and holding area movement sub-problems. Although the two sub-problems are not easy to solve, the decomposition makes the problem amenable to the selected heuristic solution methods.

The first sub-problem, the sequencing and scheduling problem, is considered in chapter 5. The specifics of the problem mean that exact methods are not applicable, as discussed in section 4.5, so heuristic and meta-heuristic methods are necessary. A tabu search algorithm for solving the problem is detailed and the chapter continues with a consideration of the method in which the sequences are evaluated and of the objective function used. The chapter ends with an explanation of the enhancements that were made to improve the performance.

The ground movement, or control problem, is considered in chapter 6. By consideration of the controller's aims, and the value of traversal paths in particular, a heuristic solution approach was developed. The traversal paths are allocated first, using a heuristic based upon controller preferences, before the feasibility of re-sequencing is tested, hence ensuring that the allocated paths and workload will always be sensible and justifiable from the controller's point of view, as discussed in section 2.8. The fixed path control problem is then similar to the blocking job shop problem. Although this problem is known to be difficult to solve, the restricted size and complexity of the holding areas, together with considerations of the timing of re-sequencing, makes a heuristic solution method both possible and effective here as well. The solution method is explained in detail in chapter 6.

The take-off sequencing problem is a dynamic problem, changing over time. The effects of decisions made at one instant in time will not necessarily be observed until later. Any evaluation of the system must be performed in a realistic manner, but it is impractical to test the developed system in a real situation, with real controllers. A simulation of the departure system at the airport had to be developed and is described in chapter 7.

It is important to tune the decision support system to produce sequences which will be acceptable to runway controllers. The validity of the simulation was verified by comparison of the behaviour of the simulation against playbacks of recorded historical data. In particular it is important to model the rules that controllers would use to determine whether any additional workload is justified. The real-time graphical user interface and playback tool that were used to verify the validity of the decision support system and simulation are also described in chapter 7. This tuning must be performed on a holding area specific basis and was performed using the departure system simulation. Once tuned, the system continued to provide acceptable sequences.

The tuned decision support system and simulation were evaluated using historic recorded data provided by NATS. This data was provided to the simulation which then produced a set of realistic partial problems for the decision support system to solve. These problems were formed by providing the decision support system only with information that would be available at the time, and modifying later problems according to the decisions made earlier. The overall take-off sequence, which resulted from the sequencing requested by the decision support system for each partial problem, was evaluated under a number of different conditions. The experiments undertaken to evaluate the performance are explained in chapter 8 and results are presented to show the effectiveness of the system.

The thesis ends in chapter 9 with a number of conclusions being drawn about the effectiveness of the designed system and the possible benefits of implementing such a system for runway controllers. Future extensions are also discussed, showing how later research could build upon the research presented here.

1.7 Contributions Of The Thesis

The importance of the runway sequencing role at Heathrow has already been discussed. Although both arrival and departure sequencing have both been considered in the past, this work is the first to consider all of the constraints upon the take-off sequencing problem, including the presentation of realistic models for the holding areas where the re-sequencing is performed.

1.7.1 The first research to consider the real problem

The previous approaches to similar problems are considered and evaluated in this thesis. These approaches have considered only a subset of the constraints or considered an unrealistic simplified problem. The ground movement and sequencing problems are usually considered separately. Even where studies previously considered holding area structures and sequencing constraints, the constraints were much simpler than those considered here.

The effects of the take-off time-slots, described in section 2.4, are also usually ignored. Even where take-off time-slots are applied to the problems, they are usually modelled as hard constraints limiting the movement from the ideal take-off time, rather than presenting an accurate model of the real-world constraints and preferences.

This thesis presents a combined method which considers both the sequencing and ground movement problems. In this way a good take-off sequence can be found which can be achieved using only simple ground movement.

1.7.2 A solution system which considers the controller preferences

Controller preferences for how the aircraft will move through the holding areas are important when creating a decision support system. These are explicitly considered by the system described in this thesis. This is an important aspect of this approach and one which is impossible to achieve without access to information about these controller preferences.

NATS made historic recorded movement data available for this research, together with a playback tool that provides a visualisation of what actually happened. This was invaluable for understanding the ways in which holding areas are used by controllers. Furthermore, access was given to real controllers, to aid in correcting any misunderstandings and to understand the reasons why the problem is solved in the way it currently is.

1.7.3 A solution method for a complex problem

Although both the sequencing and control problems are complex, the presented approach utilises the controller preferences and working methods to simplify the problem by using an appropriate problem decomposition, thereby enabling the design of an effective solution system. The problem decomposition that is used produces sub-problems which are amenable to heuristic and meta-heuristic solution methods.

1.7.4 Identification of the importance of traversal paths

The key to the selected approach was the identification of the importance of the paths the aircraft use to traverse the holding area and the independence of the queues at the holding area entrances in the good solutions. It is important to assign the simplest paths possible to the aircraft and the independence of the entrance queues means that this can be performed based upon the overtaking required within each entrance queue.

1.7.5 A useful departure system simulation

In addition to the algorithms for a decision support system, this thesis presents a simulation for the departure system, from the point of view of the runway controller. This allows the dynamic problem to be considered, rather than considering an unrealistic static problem.

It is possible to use the departure system simulation to evaluate the effects of different aspects of the departure system. An evaluation of the effects of the different sequencing, take-off slot and holding area constraints was performed using the system described in this thesis and the results are presented in chapter 8. Similarly, the sequencing that can be performed with the different holding areas is also considered.

Future work could use the simulation to consider the effects of modifications that could be made to the holding areas. This kind of simulation is important as the expense involved in changing the structure of the taxiways means that the effects of doing so should be evaluated beforehand. This system is ideal for this as it works at a macroscopic scale. It allows the ways in which controllers will use holding areas to be modelled, rather than merely considering the theoretical movement that could take place. Importantly, although most simulations that can be used for this kind of research rely upon deterministic decision making rules about the movement to be performed according to the aircraft characteristics, this system performs an optimisation of the sequencing and uses this to determine the movement to be performed, in a way which is far more similar to the method the controllers use.

The simulation can also be used to determine the effects upon the sequencing of the decisions that the ground controller makes about holding area entrance allocation, to evaluate the benefits of allowing different traversal paths to be used, or the consequences of restricting the use of intersection runway entrances.

1.7.6 A realistic model of holding area movement

The validity of the holding area movement has been verified using specially developed playback tools and compared against the historically recorded holding area movement. The decision support system has to solve a sequence of static problems over time. As the later problems that are considered may change in response to the earlier sequencing decisions, it is important to be able to examine the problems and the decisions made. The developed simulation allows the static problems to be examined visually, displaying the positions of all aircraft in the holding area and the predicted take-off sequence in a way similar to the ground radar display and flight strips the controllers are used to.

1.7.7 A predicted performance which significantly reduces the delay for aircraft

The decision support system in this thesis shows that, although the runway controller performs the sequencing function very well, there are possibilities for the sequencing to be improved if the controller could be given foresight about what is happening on the taxiways at the time and enough time to consider the overall take-off sequence rather than only considering a few of the aircraft at once. The simulation shows results which significantly reduce the predicted delay for aircraft awaiting take-off, with consequence cost, pollution and satisfaction benefits. This is possibly the most significant contribution of this thesis.

1.8 Summary

The take-off sequencing problem can be solved quickly enough for use in an on-line decision support system. The decision support system design presented in this thesis shows how this could be done. The implemented design and simulation predict that substantial benefits could be obtained if the taxiing aircraft could be taken into consideration. Providing a decision support system to the runway controller is one way in which this could be achieved.

Chapter 2

Take-off Runway Scheduling

2.1 Introduction

This thesis describes the underlying search algorithms for a decision support system for runway controllers at London Heathrow airport. It is important to begin with an understanding of the problem that the runway controller faces before introducing a potential solution method for the problem.

The runway controllers are responsible for sequencing and scheduling take-offs from the airport. Although complex, the problem is currently solved manually by the runway controller, originally using paper strips containing the aircraft information as a type of short-term memory and more recently using a representation of these flight strips on a computer screen. The controllers use pattern matching skills to identify good and bad partial take-off sequences by looking at the information strips. The controller will have a number of objectives in mind when sequencing the aircraft. Some of these objectives cooperate and some conflict with each other, leaving the controller with an explicit or implicit decision to make about how to balance the various objectives. The various objectives for the take-off sequencing and the constraints upon it are explained in this chapter.

The runway controller will usually be very busy directing the aircraft and communicating with pilots, leaving very little time to consider the desired take-off sequence. This will often mean that there is insufficient opportunity to consider all of the elements which may affect the take-off sequence. Importantly, it may not be possible to explicitly consider the effects of current decisions upon aircraft which are still taxiing or at their stands. The manually produced schedules can suffer due to this myopia forced upon controllers by the high workload at busy times. It is quite common for a schedule which appears to be optimal for the subset of aircraft which are currently within the holding area to be a sub-optimal sub-sequence when the longer term scheduling problem is considered. In order to avoid this problem, it is necessary to consider the effect of short term decisions upon the longer term sequencing problem.

The aim of this thesis is to present the algorithms for a decision support system to aid the runway controller in this difficult task. By taking into consideration more information than the controller could hope to consider and then presenting the controller with suggested take-off sequences, the aim is to provide solutions that are globally better than the schedules that would otherwise be created.

This chapter begins with a summary of the situation at London Heathrow Airport, putting the problem faced by the runway controller into context. The chapter continues with an explanation of the various constraints upon the sequencing that is possible, followed by a discussion of the objectives of the runway controller. The full formulation of the problem is reserved for chapters 4, 5 and 6 as it is dependent upon the solution method used. For example, in the solution method proposed in this thesis, certain objectives are handled externally to the objective function. The effects of the scoping decisions (described in section 1.3) that have been made are then considered, such as the need for any re-sequencing to be acceptable to controllers. The chapter ends with a summary of the objectives, a consideration of why delay is measured instead of throughput, and a summary of the inputs and outputs of the system.

2.2 London Heathrow Airport

London Heathrow Airport is a very busy two-runway airport. The popularity of Heathrow can cause severe congestion at certain times of the day and the capacity of the airport has to be limited in order to reduce the delays to acceptable levels.

When first designed, the airport had three pairs of parallel runways. As each pair intersects the other pairs, only one pair could be in use at once, this being determined by the wind direction. Over time, most have been built over and the two running (roughly) East-West have been extended to cope with modern aircraft. Another of the original runways still existed, and was occasionally used, until recently. It has now been decommissioned, however, and is used as a taxiway. It had not been extended (so it was not long enough for many of the Heathrow departures) and intersected both of the other runways (so using it prevented use of either of the other runways).

A symbolic representation of the layout of the airport is provided in figure 2.1. The two physical runways can be used in either direction, depending upon the direction of the wind, and are named differently depending upon the direction of use. The runway names are labelled in figure 2.1. For example, the northern runway is called 27R when used by aircraft taking off or landing facing west and 09L when used in the other direction. A colour-coded plan of London Heathrow Airport is also provided in figure 2.2. The taxi-ways are coloured red, the holding areas green and the runways, terminals and other buildings are white.

Heathrow airport is situated very close to London, with many 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 airport has to run in segregated mode, where each runway can be used for either landings or take-offs but not both, for most of the day. Details of the precise rules and

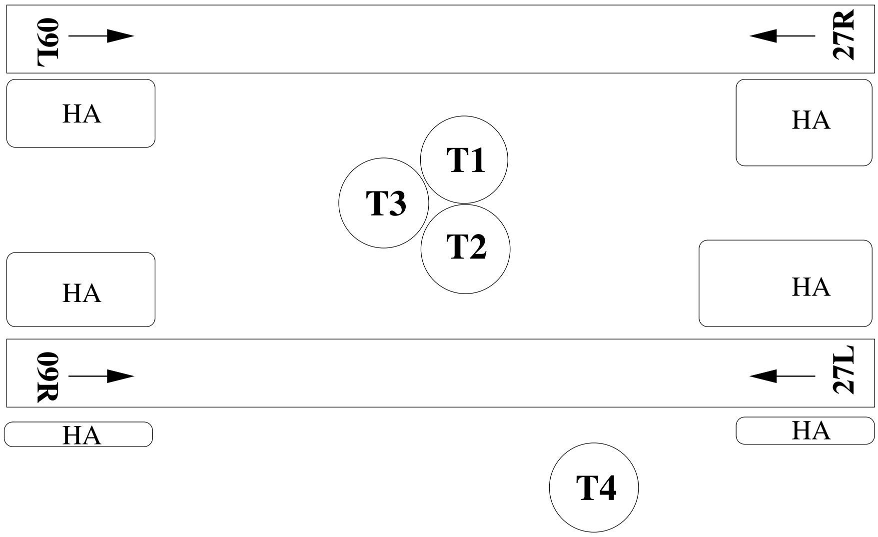


FIGURE 2.1: The layout of London Heathrow Airport

the reasons for them can be found in the BAA flight evaluation report, [20].

It is well known, and was analytically shown in papers such as [135], that running a two-runway airport like Heathrow in mixed mode (using both runways for both arrivals and departures) is more efficient than running it in segregated mode. Mixed mode allows the larger wake vortex separations between arrivals to be utilised by departures, and vice versa. As mixed mode is not usually possible at Heathrow, the controller has to consider the wake vortex separations when solving the take-off sequencing problem. This not only complicates the sequencing but also adds some additional delay beyond that which may be necessary if mixed mode was being used. The overall effects of these wake vortex separations upon the total delay can be seen in the results presented in section 8.5.

The various separation rules for the runway mean that the throughput of the runway is much less than the throughput of the taxiways to the holding areas, so the runway usually represents the bottleneck for the departure system. This is also true for many other airports. For example, Idris et al. reported this for Boston Logan airport [110]. It is therefore imperative that the throughput of the runway be maintained at as high a level as possible, in order to keep delays down for airlines and passengers.

There are currently four terminals at London Heathrow, labelled T1 to T4 in figure 2.1, although a fifth is due to open in March 2008. Three terminals are situated between the runways but the fourth is to the south of the southern runway. When a flight is ready to depart, a ground movement controller will direct the pilot around the taxiways towards the current departure runway.

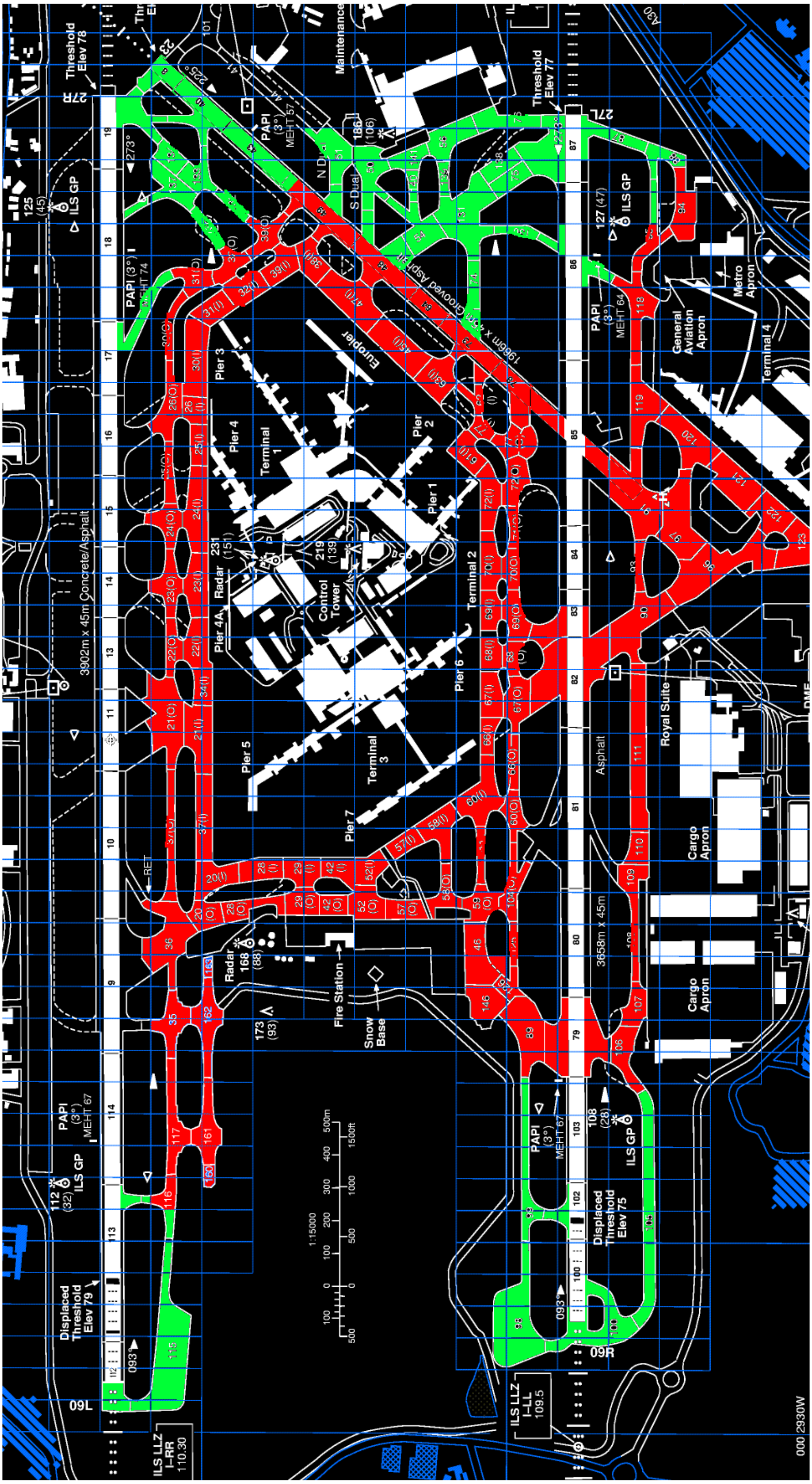


Figure 2.2: Colour-coded plan of London Heathrow Airport, used with permission of NATS Ltd

Once an aircraft approaches the runway end and is no longer in conflict with any other aircraft the ground movement controller will give the pilot instructions about where to taxi to and will relinquish control of the aircraft to the runway controller. This usually happens as the aircraft enters a holding area near to the end of the departure runway, labelled HA on figure 2.1. The ground movement controller would typically instruct the pilot to taxi to a holding point just inside the holding area, or to join the end of a queue in the holding area, so that it is not necessary for the runway controller to immediately give further instruction.

At Heathrow, it is usually most practical to re-sequence the aircraft within the holding areas rather than earlier in the departure system. This is not as true at some other airports, so the literature for this type of problem usually ignores the physical constraints that are present at the holding areas of airports like Heathrow. Unfortunately, at Heathrow the sequencing constraints that result from the holding area structures are key and not only affect which sequences are achievable but also how undesirable any re-sequencing is.

2.3 Separation Rules

Sequence-dependent separations must be enforced between aircraft at take-off. Appropriate modification of the take-off sequence can therefore greatly affect the throughput of the runway. These separation rules are of two types, and both sets of constraints must be met.

2.3.1 Wake vortex separations

When aircraft take off they leave wake vortices behind them which can affect the take-off of later aircraft. In order to ensure safety at take-off, a separation is imposed between aircraft to allow the wake vortices of the previous take-off to dissipate to safe levels. Heavier aircraft create larger wake vortices and lighter aircraft are more affected so the separation depends upon the weight classes of the aircraft.

The wake vortex separations are the major constraint at many airports. These are so important that they have been used to estimate total airport capacity, [68], by assuming that departure capacity is higher than arrival capacity and that the latter is primarily determined by the wake vortex separations. The research presented in this thesis (in section 8.5) will show that the wake vortex separations are not actually the most constraining of the sequencing constraints at Heathrow.

Now that Concorde is no longer operational, wake vortex separations are simpler than they used to be. At the moment, a two-minute separation is required whenever a smaller category aircraft follows a larger category aircraft. Otherwise, a one-minute separation can be used. The new airbus A380 is expected to complicate the rules again, requiring larger wake vortex separations to be required.

For departures there are only three weight classes that need to be considered: ‘Heavy’, 'Medium' and 'Light', since both the 'Upper Medium' and 'Small' weight classes are subsumed into the 'Medium' class for departures. As weight classes cover such a large variety of aircraft, a two minute separation could be excessive in some cases, but must still be applied. For this reason, it has been suggested that wake vortex separation rules may change in the future, [53, 69, 153].

It should be noted that the separation rules can change if aircraft do not use the full runway for take-off. If an aircraft enters the runway part of the way along, it will be closer to the preceeding aircraft than if it had entered at the end. When this kind of *'intermediate take-off point'* is used the separations may need to be increased. In the experiments performed for this thesis, even those runway entrances that were used that were not at the end of the runway were close enough to the end to not count as intermediate take-off points for the purpose of wake vortex separation rules.

2.3.2 Route-based separations

Departing aircraft will usually have a Standard Instrument Departure (SID) route assigned to them, identified by a three letter name. SID routes are pre-determined routes that enable a pilot to depart the local airspace of an airport without having to confirm every manoeuvre with the air traffic controller. Both the pilot and controller will know in advance the route the aircraft will take so the workload of both the pilots and controllers is reduced.

Aircraft must maintain adequate separation distances when in flight. Wake vortex separations are not always large enough to ensure that in-flight separations are attained for aircraft which are departing along similar departure routes. A minimum separation time is required for any two aircraft to ensure that they attain a safe in-flight separation. This depends upon the departure routes of the aircraft.

The standard separations for the different combinations of departure routes and take-off runways can be found in tables 2.1 to 2.4, that were produced from data and documentation supplied by NATS. To find the standard separation time, in seconds, that must be applied between aircraft taking off along the indicated combination of SID routes, cross reference the row for the SID of the leading aircraft with the column for the SID of the following aircraft. Standard separations vary from 60 to 180 seconds.

These separation rules do not always obey the triangle inequality. For example, consider the sequence DVR-BPK-DVR for the 27L runway. The minimum separation distances can be read from table 2.1. A three minute separation is required between the two DVR aircraft, but only a one-minute separation between adjacent take-offs. As all separation rules must be obeyed it is not sufficient to consider only separations between adjacent take-offs.

If the two aircraft that are taking off are of different speed groups then the route-based separation must often be modified according to the speed groups of the two aircraft. If a faster aircraft follows a slower aircraft then the initial separation will normally need to be increased to allow for the fact that the gap will decrease over time. If a slower aircraft follows a faster aircraft then the initial separation can sometimes be decreased as the aircraft will naturally separate in flight. The rules for modifying the separations are different for the different combinations of departure routes and are detailed in appendix D. The final minimum route and speed separation that is required can therefore be determined from the routes and speed groups of the two aircraft involved. Obviously, once speed modifications have been applied to separations, even those separations which were previously symmetric are no longer so.

TABLE 2.1: Basic separation times runway 27R, by SID routes of the leading (rows) and following (columns) aircraft

SID	BPK	WOB	DET	DVR	CPT	SAM	MID	MAY
BPK	120	120	60	60	60	120	60	60
WOB	120	120	60	60	60	120	60	60
DET	60	60	180	180	60	60	120	180
DVR	60	60	180	180	60	60	120	180
CPT	60	60	60	60	120	120	120	60
SAM	120	120	60	60	120	120	120	60
MID	60	60	120	120	120	120	120	120
MAY	60	60	180	180	60	60	120	180

TABLE 2.2: Basic separation times runway 27L, by SID routes of the leading (rows) and following (columns) aircraft

SID	BPK	WOB	DET	DVR	CPT	SAM	MID	MAY
BPK	120	120	60	60	120	120	60	60
WOB	120	120	60	60	120	120	60	60
DET	60	60	180	180	60	60	120	180
DVR	60	60	180	180	60	60	120	180
CPT	120	120	60	60	120	120	120	60
SAM	120	120	60	60	120	120	120	60
MID	60	60	120	120	120	120	120	120
MAY	60	60	180	180	60	60	120	180

TABLE 2.3: Basic separation times runway 09R, by SID routes of the leading (rows) and following (columns) aircraft

SID	BPK	BUZ	DET	DVR	CPT	SAM	MID	MAY
BPK	120	120	60	60	60	60	60	120
BUZ	120	120	60	60	60	60	60	120
DET	60	60	120	120	120	120	120	120
DVR	60	60	120	120	120	120	120	120
CPT	60	60	120	120	120	120	120	120
SAM	60	60	120	120	120	120	180	180
MID	60	60	120	120	120	120	120	120
MAY	120	120	120	120	120	180	180	120

2.3.3 Reducing separations

The Civil Aviation Authority[1] (CAA), provide a Manual of Air Traffic Services (MATS), within which are documented the air traffic control regulations. MATS part 1 applies nationally and

[1]Civil Aviation Authority (CAA), web site: `http://www.caa.co.uk`

TABLE 2.4: Basic separation times runway 09L, by SID routes of the leading (rows) and following (columns) aircraft

SID	BPK	BUZ	DET	DVR	CPT	SAM	MID	MAY
BPK	120	120	60	60	60	60	60	120
BUZ	120	120	60	60	60	60	60	120
DET	60	60	120	120	120	120	120	120
DVR	60	60	120	120	120	120	120	120
CPT	60	60	120	120	120	120	120	120
SAM	60	60	120	120	120	120	180	180
MID	60	60	120	120	120	120	120	120
MAY	120	120	120	120	120	180	180	120

is freely available[2]. MATS part 2 applies to the local airport and is not publically available. The specific departure route-based separation rules for Heathrow that were described above are detailed in the MATS part 2 for Heathrow.

The circumstances under which reduced separations can be used in the vicinity of an airport are detailed in MATS part 1. In particular, the standard separation minima may be reduced at the controller's discretion when he/she has continuous visibility of aircraft. Where separation rules are higher than actually required for safety, for instance when they have been applied to control the rate of flow along departure routes to reduce downstream workload or congestion, it is sometimes possible for a controller to reduce such separations back down to the level required for safety, if the controller can verify that this will not cause a problem downstream at the time. This will always be at the controller's discretion, so it is perhaps unwise for any decision support system to assume that this reduction will necessarily be performed.

2.3.4 Summary of separation rules

The separation that needs to be imposed between aircraft depends upon the size, speed and departure direction of the aircraft. These separation rules are not symmetric and do not obey the triangle inequality. Re-sequencing aircraft can have a large effect upon the required separations and can greatly affect the throughput of the runway and the total delay for aircraft taking off.

2.4 Calculated Time Of Take-off (CTOT)

The movement of aircraft can be thought of as a flow between airports. In addition to the congestion at many airports, the European airspace is congested at a number of points. Eurocontrol aims to control this congestion using flow management techniques, [61].

The Central Flow Management Unit of Eurocontrol in Brussels assigns a *'Calculated Time Of Take-off'* (CTOT) to aircraft which will enter congested airspace or go to congested airports. The intention is to smooth the traffic flow by limiting the times at which an aircraft can enter these congested areas. Aircraft cannot take off more than five minutes before this

[2]Manual of Air Traffic Services, Part 1, accessible from the CAA web site: `http://www.caa.co.uk/application.aspx?catid=33&pagetype=65&appid=11&mode=detail&id=222`

calculated time or more than ten minutes after the calculated time, giving the fifteen minute take-off window or time-slot. Around thirty to forty percent of aircraft taking off from Heathrow have to adhere to one of these fifteen minute take-off time slots.

Despite the fact that these CTOTs should be hard constraints, it is not uncommon for at least some of the take-offs from airports to miss the allocated CTOTs, [95]. Although it is always possible to delay an early aircraft in order for it to take off late enough to comply with the CTOT (providing that there is somewhere at which to hold the aircraft), airport congestion and delays can mean that aircraft will sometimes not arrive at the holding area in time to take off before the end of the allocated CTOT. In the worst case this can mean that a new CTOT needs to be arranged for such aircraft, which may involve a high delay. To reduce the number of these re-arranged CTOTs, rules are in place at Heathrow to allow extensions of up to five minutes to be arranged for up to four aircraft per hour and no more than twenty aircraft in a day. These extensions are to be avoided wherever possible, but it will be obvious from the results presented in chapter 8 of this thesis that this is not always achievable.

2.5 Minimum Departure Intervals (MDIs)

For safety reasons, it is important to control the workload for en-route controllers (those controlling the airspace that the aircraft will later pass through) and this usually means reducing the number of aircraft that these busy controllers have to control at the same time. There is little that can be done about the aircraft already under the control of the controllers, so the primary way to reduce the workload over time is to reduce the rate of flow of aircraft into the busy sectors.

Most separation rules for aircraft taking off ensure that safe distances are maintained between aircraft, whether at take-off or in the air. A *'Minimum Departure Interval'* (MDI) is a larger separation that must be met for aircraft departing on specific routes. The MDI is the fastest way to control the rate of in-feed into the airspace along these routes and is used to control workload and ease congestion downstream.

Some departure routes into busy airspace have MDIs in effect all of the time. For example, three-minute MDIs are in permanent operation along some departure routes from Heathrow, and are in operation at specific times of the day along other routes. This is not surprising given the proximity of the airport to the city of London and to other UK airports such as Gatwick, Stanstead, London City airport and Luton.

At times the capacity of airspace sectors may be decreased, such as in times of very bad weather, [164], in which case the load on surrounding sectors may be increased. In these cases the MDI applies a short term strategy for handling the problem. A slightly longer term strategy for flow control is performed by the CTOT system, described earlier.

The airspace is a highly complex and dynamic system so any control system must react quickly to changing circumstances. This means that a controller will, in general, not know exactly when the MDI will commence or end. When an MDI is in operation, CTOT compliance is unnecessary on those routes for the duration of the MDI, however, as controllers do not necessarily know the duration of an MDI, they must still try to plan take-offs within CTOT where possible in case the MDI ends before the aircraft takes off.

2.6 Earliest Take-off Time

There are various constraints upon how early an aircraft should be scheduled to take off. For example, an aircraft should not be scheduled to take off before it can reach the runway, before it has been prepared and all passengers are seated, or before the start of a take-off time-slot if it has one.

Before take-off, an aircraft has to taxi from its stand to the holding area, taxi through the holding area and line up for take-off. This process takes time and will limit how early an aircraft can take off so it is important for a planned take-off sequence to allow sufficient time for an aircraft to actually reach the runway. Given the time at which an aircraft enters the holding area, the time at which it could be ready for take-off depends both upon the path taken through the holding area and upon whether the aircraft has to wait for any other aircraft to manoeuvre before it can enter the runway.

Similarly, when a large aircraft pushes back from a stand near 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 decision support system does not schedule aircraft to take off too soon after push-back, even if they are close to the holding area. The required time between push-back and take-off will often be higher for larger aircraft than for smaller aircraft.

The calculated time of take-off (CTOT) determines both an earliest and a latest take-off time. As aircraft should not take off more than five minutes before the CTOT time, the decision-support system should not schedule the take-off before then.

2.7 Sequencing Within The Holding Area

Some overtaking may be needed in order to achieve a desired take-off sequence. At present, this overtaking is performed within the holding areas, near the ends of the runways. Doing this has two distinct and important advantages compared with sequencing at the stands.

The first advantage is that it means that the ground movement controllers can ignore the desired take-off sequence when directing aircraft from the stands to the holding area. The departure system can therefore be easily decomposed into mutually independent sub-problems, with consequent workload benefits for the ground movement controller.

The second advantage of sequencing within the holding areas is that the uncertainties intrinsic to push-back times, taxi times and the contention between aircraft, (arrivals or departures) on the taxiways can effectively be ignored. The runway controller has a much more manageable problem to handle at the holding area than a controller would have if trying to schedule the take-offs earlier, for example when aircraft were still at the stands.

However, the overtaking that can be performed within a holding area is limited by the physical structure of the holding area. The holding area structure has a major effect upon the take-off sequencing problem at Heathrow and can greatly affect which sequences can be achieved.

The holding areas are shown in green in the plan of London Heathrow airport given in figure 2.2 and can be thought of as one or more entrance queues to some manoeuvring space before one or more exits to the runway. As all of the overtaking is performed within the holding area, the structure of the holding area in use and the current positions of the aircraft within it, are key to determining the overtaking that is possible and the cost of any overtaking. For example, if aircraft A has to overtake aircraft B then there must either be room for aircraft A to go around aircraft B, or there must be somewhere aircraft B can move to in order for aircraft A to go past. The problem is that it is not usually possible to consider just two aircraft in isolation. Decisions made for these may affect other aircraft. For example, moving an aircraft aside may block a useful taxi route needed by another aircraft. Alternatively, the path assigned to an aircraft A, to allow it to overtake an aircraft B may prevent an aircraft C from in turn overtaking A.

It is obvious from figure 2.2 that the configuration of the holding areas varies greatly between the runway ends. The southern holding areas allow aircraft to enter the runway from either the north or south side. Although the structures will change with the advent of terminal 5, only the 09R holding area is expected to change significantly. The holding area structures and the possible overtaking within them are explained in detail in section 6.2.

Where there are different entrance queues available, the ground movement controller will often send an aircraft into the most convenient queue, although some pre-sequencing may be performed or there may be a heuristic allocation method being used based upon the departure routes of aircraft. The runway controller can request aircraft to be sent to specific queues but, at the times when it most matters, is usually too busy with the aircraft already in the holding area and local airspace to have sufficient time to also consider the aircraft the ground movement controller has.

2.7.1 Directed Graph Models Of The Holding Areas

Directed graph models for each of the holding area structures were developed for this research, taking into consideration how the holding areas are used as well as the physical layouts. These models can be used to illustrate the problem of re-sequencing within the holding area. The models for the Heathrow holding areas are given in chapter 6, together with details of how they were produced and how they can be used to determine the feasibility of re-sequencing. An example graph is presented in figure 2.3 for the 27R holding area.

Each node in the holding area graph represents a position that an aircraft can hold at and is uniquely labelled. The arcs of the graph represent the valid movements that aircraft can make between nodes. The structure of the holding areas and the number of holding points in the holding areas differ between the holding areas. When using the southern runway (27L/09R), the holding area actually comprises two separate holding areas, one south of and one north of the runway.

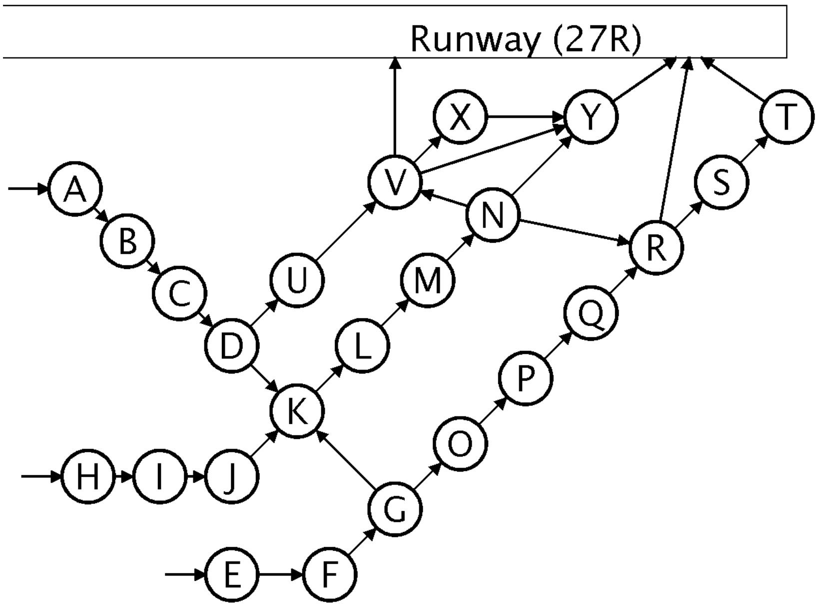


FIGURE 2.3: An example holding area graph, for the 27R holding area

The graph can be used to determine which re-sequencing is possible and which is not by feeding aircraft into the entrance nodes (A, E or H) of the holding area graph, in the order that they are predicted to arrive at the holding area. The re-sequencing possibilities can be investigated by considering the ways in which aircraft can move through the holding area graph and the sequences in which they can reach the runway, given the restrictions that aircraft can only move between nodes along the arcs of the graph and each node can contain only a single aircraft at any one time.

2.8 Re-sequencing Must Be Acceptable To Controllers

The scope of this project, described in section 1.3, is such that there is no intention to ever enforce any decision upon a runway controller. For this reason, it is vitally important that the re-sequencing within the holding area that is required in order to achieve the target take-off sequence is acceptable to the controllers to whom the suggestion is made. If this is not so then the re-sequencing is extremely unlikely to be enacted. This means that any decision support system has to consider the question of how the re-sequencing will be achieved, even though this information is not intended to be transmitted to the controller.

2.8.1 Traversal paths

Each aircraft will follow a specific path through the holding area. These paths are referred to as *'Traversal Paths'* throughout this thesis. The path will be chosen by the runway controller and may be refined over time. For example, a controller may ask a pilot to taxi an aircraft forward to a specific position and hold there, then later developments may allow the controller to make further decisions about the path the pilot should follow from that point.

When a holding area graph is available, a traversal path can be denoted by the ordered list of nodes through which the path passes. In the 27R holding area graph in figure 2.3, the six theoretical paths which run from node K to the runway (here named as node Z) are $KLMNYZ$, $KLMNVZ$, $KLMNVYZ$, $KLMNVXYZ$, $KLMNRZ$ and $KLMNRSTZ$.

2.8.2 The path preferences of controllers

Discussion with air traffic controllers and observation of behaviour showed that there is a strong relationship between the paths which are allocated to the aircraft and the acceptability of the re-sequencing to the controllers. Some paths are perceived to be faster or slower than other paths, or more or less difficult than other paths. These perceptions (which are supported by anecdotal evidence from controller experience) drive the controller path preferences. It is important that the decision support system ensures that any required re-sequencing is achievable in a method which is acceptable to controllers and will appear to be sensible to them.

General path preference rules are illustrated below using the 27R holding area graph given in figure 2.3, where the paths are again named by the nodes the path passes through. The specific preferences for each of the holding areas at Heathrow are discussed in chapter 6.

Observation 1: There is a preferred default path from a node to the runway

There are usually multiple entrances to the holding area but there is a single preferred path through the holding area from each entrance. As long as the re-sequencing can be performed using only such paths, these paths will be the only ones allocated. For example, the $KLMNYZ$ path is the easiest path to navigate from node K to the runway and so is the preferred path for aircraft to take from node K to the runway. Any decision support system which aims to suggest an acceptable take-off sequence to a controller should assume that the preferred paths will be used when possible.

Observation 2: Some paths are more difficult to traverse than others

Traversing some paths is perceived to require more work than traversing others. Controllers have very strong preferences for paths which they perceive as requiring less work from themselves or pilots. Deviation from these paths would need to be justifiable, for instance as being necessary to achieve the required re-sequencing.

Some (theoretically possible) paths are perceived by controllers to require so much work from a pilot that they would not usually ask a pilot to traverse them. If a controller would not use the path then it is important that a decision support system also does not do so.

Observation 3: Longer paths can be used to allow aircraft to be overtaken

Some paths take longer to traverse than others. For example, the path $KLMNRSTZ$ is relatively simple for a pilot to navigate, but is much longer than the $KLMNRZ$ path. It will not always be possible for the re-sequencing to be achieved using the preferred paths for each aircraft. Where this is not possible, at least one of the aircraft will need to be given a less preferred path through the holding area. Longer paths will only be used when the controller is sure that the aircraft has the time available to traverse the path and still attain the predicted take-off time. When an aircraft is overtaken it will have more time available to traverse a longer path so, if longer paths must be used in order to achieve the re-sequencing, it is sensible to allocate overtaken aircraft to such paths. In particular, if two aircraft need to be re-sequenced such that one overtakes the other, and at least one of them would need to be allocated to the longer path, then the sensible path allocation is for the one which is overtaken to be allocated to the longer path, since it will spend longer in the holding area (it arrived prior to the overtaking aircraft but takes off later than it). In other words, it is not sensible for an aircraft to overtake using a path which will take longer to traverse, since this increases the risk of delays due to it not reaching the runway in time.

Observation 4: The runway entrances at the end of the runway are preferred

Where paths enter the runway away from the end, the aircraft will not have the benefit of the full runway for take-off. Additionally, it is often more difficult for aircraft to turn and line up from these runway entrances. It is, therefore, desirable to use the entrances at the end of the runway wherever possible. For example, although the $KLMNVZ$ is a short path, it is harder to navigate and enters the runway away from the end so it less desirable for these reasons.

2.8.3 Implications of the path preferences

Re-sequencing should be suggested only by a decision support system if it is possible using controller preferences for paths. Such a re-sequencing would then be acceptable to controllers. Three implications can be concluded from the above observations about path preferences. These observations and implications were important in the design of the path allocation system described in this thesis.

Implication 1: The holding area entrance at which the aircraft arrives is important

Controllers have a strong preference for using the easiest path through the holding area. Paths from different entrances converge at some point, either at the runway or within the holding area. Aircraft from different entrances can be re-sequenced relative to each other by altering the order in which aircraft pass the point at which the paths converge. As long as the only overtaking to take place is between aircraft from different entrances (so that aircraft from the same entrance have a fixed take-off order relative to each other), all aircraft can be assigned the simplest path for their entrance and the problem can be considered to be one of interleaving queues of aircraft. Aircraft from the same entrance can only overtake each other if at least one of them is allocated to a less preferable path through the holding area.

Implication 2: The preferred path allocation depends upon the take-off sequence

Given two aircraft, *1* and *2*, that arrive at the same holding area entrance (for example at the entrance related to node K in the 27R holding area), if they take off in the sequence *1* then *2*, then both should be allocated to the straightforward path (for example, the path $KLMNYZ$). If they take off in the sequence *2* then *1*, then the overtaken aircraft (*1*) should be allocated to a slower path ($KLMNRSTZ$), to allow it to be overtaken, and the overtaking aircraft (*2*) should be allocated to the faster path ($KLMNYZ$), since it has less time available to traverse the holding area. The path to allocate to aircraft *1* can thus be seen to depend upon whether or not it is overtaken by another aircraft from the same entrance.

Implication 3: There are many possible path allocations that are not preferred

Even in the two aircraft example, given above, there are 6 different paths which could be allocated to each aircraft, for a combination of 36 different path allocations between them. However, there is a single preferred path allocation, shown above, for any given take-off sequence, and any solution which is suggested by a decision support system should ensure that this is used. Even with this small problem, it is highly unlikely that an arbitrary path assignment will result in the preferred paths being assigned. With more realistic sized problems, it is increasingly unlikely that the preferred paths will be assigned by chance, but any other allocation will be perceived to cause unnecessary or less sensible workload for pilots, controllers or both. It is vital to assess the value of the paths assigned to aircraft when considering whether any re-sequencing will be acceptable to a controller.

2.8.4 Summary of the acceptability of re-sequencing

It is important for a decision support system to ensure that the re-sequencing can be performed in an way that will be acceptable to controllers. An important aspect of this is to ensure that the paths that are allocated to aircraft will appear sensible to the controllers. However, the sensible path allocation depends upon the target take-off sequence. This is an important characteristic of the problem as it means that the value of a path assignment for an aircraft cannot be determined until the details of which other aircraft will overtake it, and which aircraft it will itself overtake, are known.

2.9 Equity Of Delay

Allowing aircraft to overtake other aircraft is intrinsically inequitable, and penalises those aircraft which are overtaken. It is therefore important to keep the take-off sequence as close to first-come-first-served as possible, given the other objectives. This helps to ensure fairness for airlines, passengers and crew and helps to avoid large delays for any passengers. However, it will be seen in the results presented in chapter 8 that a first-come-first-served take-off sequence is extremely poor in terms of both CTOT compliance and delay. It is important to balance the individual delays imposed upon any specific aircraft against the total delay for all aircraft combined.

2.10 Objectives Of The Decision Support System

Given the previous description of the problem and the constraints which apply to it, the objectives of this research can be expressed as finding a take-off sequence which:

- Uses as few of the CTOT extensions as possible, and does not miss CTOTs by more than the permissible five minute extension.
- Keeps the runway throughput as high as possible.
- Keeps the sequencing as equitable as possible.

While ensuring that:

- The required re-sequencing can be performed within the holding area.
- The method of re-sequencing will be both obvious to and acceptable to controllers.
- All separation rules are obeyed.
- All aircraft have sufficient time to reach the runway and perform pre-flight checks prior to the predicted take-off time.

The first two objectives are obvious given the previous explanation of the constraints upon the take-off sequencing and to some degree these objectives co-operate, as shown by the results in section 8.9.3. For example, if throughput is reduced then delay will increase for aircraft and aircraft are less likely to be able to achieve take-off within CTOT time-slots. However, within the best take-off sequences there is often a trade-off between CTOT compliance and delay. CTOT compliance may force the prioritisation or delay of specific aircraft, increasing the overall delay in the take-off sequence. This can be clearly seen in the results in section 8.5, where lower delay sequences are obtained when CTOTs are removed from the problem.

In fact, the decision support system presented in this thesis aims to reduce delay rather than to explicitly improve the throughput of the runway. The reasons for this are given in section 2.13.

2.11 Solution Time

While considering the objectives for a decision support system it is imperative to keep in mind the real-time nature of the problem. Decisions must be up to date, taking account of the latest situational changes. The intention is that the decision support system has a suggested sequence ready whenever the controller decides to consult it. The time delay between the situation changing and the decision support system updating its advice is critical and must be as short as possible. If not then the system could give out of date, and hence possibly inappropriate, advice to the runway controller. At the very least this will undermine the controller's confidence in the system.

The responsiveness of the decision support system to a changing situation is determined by the solution time for the take-off problem. A response time of a second was decided upon. The decision support system described in this thesis returns results within this time limit when running on a 2.4GHz Pentium IV desktop PC and running under the simulation described in chapter 7.

2.12 A Dynamic Problem

The problem considered by the runway controllers, and any decision support system, is to determine a take-off sequence which will obtain a good overall performance for the day. However, the take-off sequencing problem is an on-line, dynamic problem. A runway controller is faced with an ongoing set of decisions to make, rather than a single decision. The consequences of these decisions will be faced later as they affect subsequent sequencing possibilities.

The runway controller or decision support system will have only limited knowledge about the aircraft that are not currently within the holding areas. Although a computerised system may be expected to be able to take more information into account, the uncertainty in the situation still limits the knowledge.

2.12.1 The cost of changing a decision

Over time, the suggested take-off sequence may change, as new aircraft enter the system or more information becomes available. It is important to control the amount by which a suggested schedule changes for aircraft which are already in the system, and also the way in which it changes. The position of the aircraft in the departure system can influence the cost of changes. Prior to an aircraft arriving at the holding area, the runway controller will not have issued any instructions to the pilot, since the ground movement controller has control at that point. The position of these aircraft in any predicted take-off sequence can therefore be freely changed with no cost. However, changing the position in the take-off sequence of an aircraft in the holding area will become more expensive as it gets closer to its take-off time. It is important that any decision support system must allow for a preference towards aircraft maintaining the same position in any new take-off sequence as in the previously suggested take-off sequence.

2.12.2 Fixed traversal paths

Until an aircraft reaches the holding area, no instructions will have been given to the pilot about how to traverse the holding area. Once aircraft are within the holding area, instructions may have been given and changing such instructions may be costly, in terms of additional time the controller must spend communicating with the pilot. The instructions given to a pilot are related to the path through the holding area (discussed in section 2.8) that the aircraft is intended to follow. Changing the planned path for an aircraft will usually involve communicating changed instructions to pilots. In order to control the amount of time the system requires that the controller spend communicating with pilots, it is assumed in this research that the paths that the system allocates to aircraft are fixed at the point at which the aircraft enters the holding area.

Despite this assumption being made for the research presented in this thesis, successful application of the decision support system described in this thesis does not rely upon this assumption being made. It is possible to use the same system and allow path changes up until the point at which the aircraft has passed the point at which the current and new paths diverge. Doing so would have no adverse effect upon the speed of the algorithms described in this thesis. (This can be seen by observing that the path allocation heuristics described in sections 6.4 and 6.5 can still be applied if the check for the allocated path having been fixed is replaced by a check for whether a path is still available to an aircraft or whether it has moved past the point at which the path diverges from the currently allocated path.) It is important for any decision support system to ensure that the controller is not given unnecessary work (in terms of communicating changes of instructions) and this decision was made purely on that basis. An additional consideration is that, without this assumption, the validity of a path allocation would be dependent upon the position of the aircraft in the holding area. The validity of any results obtained with a simulator would therefore be dependent upon the reliability of the holding area position prediction system. A useful side-effect of this decision to fix the paths was that it removed this dependency.

2.12.3 A frozen take-off position

Once pilots know their planned position in the take-off sequence, it is highly undesirable to change the take-off sequence. Changing instructions that have been given to the pilots is costly, in terms of the time taken by the busy controller to do so. Consequently, at some point in time, the position of an aircraft in the take-off sequence should be assumed to be fixed. This assumption ensures that all pilots will have adequate notice of their take-off positions to ensure that line-up can occur without difficulties. It is important that any decision support system must allow the position of some aircraft in the take-off sequence to be frozen.

There are sequencing benefits to keeping the freezing time low, but sometimes workload benefits to be gained from increasing it. For example, conditional clearances can be given to pilots if the take-off sequence is fixed at that point. These inform a pilot that they can take off following another aircraft's take-off, preventing the controller from needing to give individual take-off clearance just before take-off. Once these have been given, there is a workload cost associated with changing them.

For this research, the position of an aircraft in the take-off sequence is assumed to be *'frozen'* a specified time before its planned take-off time. The desired duration prior to take-off at which the take-off position for an aircraft is frozen is here called the *'freezing time'*. The experiments performed for this thesis were performed with a two minute freezing time. This means that the position of an aircraft in the take-off queue was frozen two minutes before take-off.

It should be noted that the concepts of a fixed path and a frozen position in the take-off sequence differ. It is possible to have a fixed traversal path allocated to an aircraft and still to allow the decision support system the flexibility to change the position of an aircraft in the take-off sequence.

2.13 Delay, Throughput And Schedule Duration

The aim of the take-off sequencing is to keep the runway throughput high and total aircraft delay low. These two factors are obviously linked but are not identical. There are important differences and different sequences may result from considering different objectives. This raises the question of which factor a decision support system should use to evaluate take-off sequences.

The first possibility is to measure the delay for the aircraft in the sequence. This involves summing the time for each aircraft between the arrival at the holding area and take-off. The second possibility is to measure the throughput of the runway, in terms of movements per unit time. The easiest way to do this for a fixed set of aircraft is to measure the time until the last take-off, the schedule duration. This is what many throughput improving algorithms do, where the aim is to find sequences where the last aircraft takes off as early as possible.

Some example take-off schedules will illustrate the difference between the two objectives. Assume there are eight aircraft, four of which are southbound, two westbound and two northbound. Simplified separation rules will be assumed for these examples, so one-minute separations will be permitted as long as aircraft depart in different directions, but two minute separations are required when aircraft depart in the same direction. A three minute MDI is assumed on the

Table 2.5: Example schedules comparing throughput and delay

Schedule, aircraft taking off at time:										Duration	Delay	
0	1	2	3	4	5	6	7	8	9	(mins)	(mins)	
S	-	-	S	N	W	S	N	W	S	9	3+4+5+6+7+8+9	= 42
S	N	-	S	N	W	S	-	W	S	9	1+3+4+5+6+8+9	= 36
S	N	W	S	N	W	S	-	-	S	9	1+2+3+4+5+6+9	= 30

southern departure route, so aircraft that use it have a three-minute separation.

Some of the possible take-off schedules are shown in figure 2.5. Aircraft departure routes are labelled N for north, S for south and W for west, and the horizontal position shows the take-off time of the aircraft. In all cases the duration is nine minutes, as the southbound aircraft ensure that the duration has a lower bound of nine minutes. However, the delay is different for each schedule.

Aiming to reduce the delay has benefits for the dynamic problem as it penalises bad separations at the start of the sequence more than at the end of the sequence. Aiming to reduce the duration has an obvious problem whenever the take-off time for an aircraft is delayed beyond the natural end of the take-off sequence. For example, if a CTOT applies to one of the aircraft in the above sequences such that the aircraft cannot take off before twelve minutes into the schedule then this aircraft will put a lower bound of twelve minutes on the schedule duration. It will then be irrelevant how badly sequenced the earlier aircraft are, as far as minimising duration is concerned.

The important difference between the duration reduction and delay reduction objectives is that the delay reduction objective will move earlier aircraft as early as possible in the schedule, rather than being so dependent upon the take-off time of the last aircraft. As the take-off problem is a dynamic one, large gaps are not necessarily a problem as long as they occur late enough in the take-off sequence to stand a chance of inserting aircraft which arrive later into them.

For example, given the schedules in table 2.5, consider the case where two more aircraft arrive at the holding area, one northbound and one westbound. These aircraft could naturally fit into the gaps in the existing schedules. This is obviously easier to do if the gaps in the schedule are closer to the end of the schedule, since the new arrivals will have to overtake less aircraft.

The emphasis upon the start of the sequence, which a delay-based objective implicitly has, does introduce a problem. A system to minimise delay will naturally tend to move larger separations later in the sequence, but some aircraft have characteristics which will necessarily incur large separations either before them (if they are fast or light) or after them (if they are slow or heavy). These aircraft would naturally be moved later in the sequence, moving any delay they would cause later in the sequence and therefore affecting less aircraft. Unless controlled, this can unfairly penalise these aircraft. The use of a delay-based rather than throughput-based objective makes the control of inequitable sequences even more important.

2.14 Decision Support System Inputs

A decision support system needs to know the current state of the departure system in order to make a decision about the re-sequencing. Aircraft in the holding area should be included. Aircraft on the taxiways can optionally be included. Due to the separation rules, aircraft that have recently taken off need to be kept in the system until they can no longer affect the take-off times of the aircraft awaiting take-off. When large minimum departure intervals are in force on some departure routes this may mean keeping aircraft within the system for up to ten minutes or more after departure.

It is envisaged that all of the data below will be available to a real decision support system, for instance through an interface with existing systems such as the electronic strips the controllers use to record the chosen take-off sequence or the ground radar aircraft tracking systems. There may be a degree of uncertainty associated with some data, such as taxi times. The accuracy of taxi time predictions could be improved by implementing an intermediate prediction system between the ground positional data system and the decision support system. Such a system could consider taxiway congestion and the current positions and taxi speeds of aircraft in order to improve the accuracy. The design of such a system is beyond the scope of this thesis.

The available data should include:

- The weight class of the aircraft, so that wake vortex separations can be determined.
- The speed group and allocated departure route of the aircraft, so that route-based separations can be determined.
- Any CTOT take-off time-slot associated with the aircraft.
- The take-off time of the aircraft if it has already taken off.
- The arrival time at the holding area if the aircraft is within the holding area. So that the delay at the holding area can be predicted.
- The predicted (possibly inaccurate) arrival time and holding area entrance for any taxiing aircraft, if taxiing aircraft are being included in the system.
- The position of the aircraft in any previously chosen take-off sequence, enabling sequences that are closer to previous suggestions or closer to a controller's current plans to be preferred.
- The current position of any aircraft that is within the holding area. The manoeuvring available to these aircraft is, obviously, more restricted than that available to aircraft still on the taxiways as aircraft within the holding area may have already passed decision points where paths through the holding area diverge.

- Optionally, any fixed paths through the holding area. The simulation used to test the system (described in chapter 7) fixes the paths at the point aircraft reach the holding area and provides this information to the decision support system. A live system need not be so inflexible but could estimate the allocated paths from the movement of the aircraft or could allow it to be specified by controllers if this is a desirable feature.

The push-back time was not included in the list above but could be made available and used to favour take-off sequences which are more equitable in terms of time from stand to take-off rather than aiming for equity of holding area delay. This was not used by the decision support system presented in this thesis, although the simulation used to evaluate the performance of the system (described in chapter 7) does require the push-back time information. It is used to determine when to add aircraft into the system and for simulating a taxi time prediction system with specific levels of uncertainty.

2.15 System Outputs

In a live situation, with a real runway controller present, the decision support system would display to the controller only the suggested take-off sequence, and possibly an estimated take-off time. One approach could be to annotate the electronic strips that the controllers currently use with a suggested take-off sequence number. If a separate display is used it may be useful to also show the allocated CTOT and the characteristics of the aircraft so that the controller can immediately judge the value of the suggested sequence without having to cross-reference two different displays.

When running with the simulation described in chapter 7, the decision support system has to provide more information as it is effectively acting as a runway controller simulation. This means that it has to provide information about the instructions to be given to pilots, and it even predicts the movement of the aircraft within the holding area in response to the sequencing decisions.

2.16 Other Issues

The runway controller has to perform a difficult task under tight time constraints. In many cases, the controller will be weighing the effects of contradictory constraints in an attempt to find the best take-off sequence. Objectives such as improving throughput or delay must often be weighed against complying with the CTOTs and against controlling the amount of overtaking.

The system is (and can only be) intended to be advisory. The runway controller has a lot of information available and some of it may be hard to capture for input into an automated system. A decision support system will also need to accept some kind of feedback from the controller so that the controller can reject or correct schedules with which they disagree.

Finally, the departure process is a dynamic system where aircraft are removed from the system at some point after take-off and more aircraft are added as they are released from the stands. Any system will need to respond to changing situations, for instance aircraft being delayed longer than expected or developing problems.

Chapter 3

Literature Review

3.1 Introduction

This thesis concerns the design of a decision support system for take-off runway scheduling at London Heathrow Airport. This chapter discusses the research that has already been performed into the investigation and solution of similar problems.

The chapter starts with a brief overview of air transportation research and the air traffic control problem in general. The importance of the runway is then considered in section 3.2, followed by a discussion of the measurement of airport and runway capacity in section 3.3. Ultimately, a decision support system should aim to help the runway controller to keep the throughput of the runway as high as possible.

Solving a real-world problem involves first building a model of the problem then solving the model. By necessity the model will only provide an abstraction of the real world problem, but the task is to build a model for the problem which is at the correct level of abstraction so that results obtained using the model are relevant to the underlying problem. Having an appropriate model is as important as the optimality of the solution obtained for the problem solved using the model. There is often a trade-off between accuracy and simplicity when determining the level of abstraction for the model. More complex models may provide more accurate answers but will often be much harder to solve and require much more tuning and calibration before they can be used.

There are a variety of common models that have been applied to the different problems considered in this chapter. Many of the problems can be modelled as machine scheduling problems. For example, aspects of the take-off sequencing problem have important similarities to the classical single machine and job shop scheduling problems. Because of this, an overview of machine scheduling problems is given in 3.4. The job shop problem is re-considered in section 4.3 where disjunctive and alternative graphs are introduced.

The take-off problem has three distinct aspects although it will be seen in chapter 4 that the problems are highly coupled. The first is the sequencing problem; determining a take-off order. The second is the scheduling problem; assigning take-off times to aircraft. The third problem is the control problem; determining whether the required overtaking is achievable and how to achieve it.

The control, sequencing and scheduling problems can be solved individually or concurrently, but any solution to individual sub-problems is likely to be sub-optimal when the whole problem is considered. The sequencing and scheduling problems are combined in most of the current research, however, the control problem is usually solved separately.

The arrivals and departures sequencing and scheduling problems have many similarities to one another and to the classical travelling salesman problem. Due to these similarities, the travelling salesman problem is reviewed in section 3.5, then the arrival scheduling problem is reviewed in section 3.6 and finally the departure sequencing problem is considered in section 3.7.

The ground movement problem at an airport is a larger version of the control aspect of the take-off problem. The ground movement literature is examined in section 3.8 and the relevance to the Heathrow problem is assessed. The chapter ends with a consideration of the ways in which the departure system may change in future, followed by a summary of the chapter.

3.2 Air Transportation

Optimisation techniques have been applied to a wide variety of problems within the air transportation industry. The applications range from the airline side, creating flight, crew or maintenance schedules, [93, 79, 150], to issues of airport layout and ground operations, [130] and flow management, [125]. A summary of the various problems and models can be found in [174]. Wu and Caves gave a summary of the research in the area of air traffic management in [171], showing the relationships between the different areas. More recently, Barnhart, Belobaba and Odoni, [24], reported on the various applications of operational research in the air transport industry. As the demand for air transport is increasing faster than the system capacity, the potential role for optimisation techniques seems set to increase.

To understand the importance of, and constraints upon, both arrival and departure scheduling at an airport, it is useful to understand the greater context of air traffic control as a whole. A good and simple introduction to the subject is given by Graham Duke in [72].

The air transportation system can be thought of as a cyclic flow of aircraft through the airspace from departure to destination airports, along the landing runway, through the taxiways to the stands. At a stand the aircraft conceptually flows through the stages of a turnround process, unloading, preparing and re-loading the aircraft before the flow continues through the taxiways, the departure runway, the local airspace and eventually back into the regional airspace to the start of the cycle.

Depending upon the role of the viewer, the cycle can be considered to start at different places. For example, those considering the ground operations may consider the cycle to start when aircraft enter the local airspace, and end when they leave the airport. On the other hand, from the point of view of the air traffic controllers controlling the airspace, the cycle starts as they become aware of aircraft desiring entry to the airspace and ends when aircraft leave their airspace.

Considering the system as a cyclic flow helps in understanding the connections between the various elements of the air transportation system. In particular, it is important to be aware of the fact that some of the constraints applied to the departure problem are in fact a consequence of flow (or capacity) control measures to smooth the flow for later elements of the cycle. For obvious safety reasons it is better to apply any delay to aircraft on the ground rather than those in the air. The consequence of this is that the output flow or capacity of the airport may be reduced in order to control the flow through the airspace. Of course, a single airport supplies aircraft to a number of different routes and airports, each of which may have different flow restrictions. The aim for the departure runway controller can be considered to be to attempt to maximise the total flow out of the airport while meeting all of the flow constraints applied by the limitations of the destination airspace and airports as well as the local limitations imposed by the runway capacity and airport infrastructure.

Only a small part of the air transportation cycle is considered in depth in this thesis: the departure system. Starting from the airport, the aircraft is readied and loaded for departure. When ready, a pilot requests push-back clearance before leaving the stand. Aircraft then taxi around the airport towards a take-off runway. During this process various control points are passed where the sequence in which aircraft actually enter the runway can be altered. The final part of the ground operations occurs when aircraft take off along a runway, where a controller is responsible for ensuring a good take-off sequence for the runway.

3.3 Airport And Runway Capacity

The airport capacity problem is concerned with estimating the capacity of an airport, usually in terms of the number of arrivals and departures per hour. Measuring airport capacity has long been considered to be important. Accurate capacity estimation models can aid in planning new airports or planning changes to existing airports, ensuring that the design is relatively efficient.

Many of the models that have been used are high-level models, modelling various capacity constraints by considering the rules that apply to the departures and arrivals systems at airports. Low-level models can also be of use, for example modelling the movement of each aircraft around the airport. These low-level models are, however, usually much more complicated and much harder to calibrate.

3.3.1 High-level (macroscopic) models

In 1959, Blumstein presented a method for analytically estimating the capacity of an arrivals runway, [36]. He concluded that, at that time, the required separation at the start of the glide path was more prohibitive on the throughput than the required separation at the point of landing.

Hockaday and Kanafani presented models to cover the usage of a runway by arrivals and/or departures in [103]. Their model included the detailed operating rules and the stochastic nature of the system and was found to compare adequately with the observed capacities.

An introduction to runway usage was given by Newell in [135], with an explanation of why the separations between aircraft are needed. Newell defined the rate of flow of aircraft and passengers and developed a model to determine the capacity based upon categorising pairs of aircraft types. Considering departures with an equal split between heavy and medium aircraft, he drew the conclusion that the improved sequencing of the departures beyond first-come-first-served yielded little benefit. However, this conclusion was drawn from figures which showed an increased departure rate from 43.6 to 48 departures per hour. When considering a departure system as busy as that at Heathrow, a reduction in the departure rate of four take-offs an hour is actually very significant. Newell's results indicate that aircraft accumulate significant take-off delays in the absence of re-sequencing, so such a situation is extremely unlikely to be acceptable at Heathrow. Furthermore, there are more constraints an the take-off sequencing at Heathrow than were considered by Newell, some of which will be shown in section 8.5 to be even more significant than wake vortex separations, so the re-sequencing at Heathrow is even more important.

The MANTEA Airfield Capacity And Delays (MACAD) model[1], [161], was designed for high level capacity planning. It was designed for speed rather than accuracy and includes elements to estimate runway capacity and delay.

3.3.2 Queueing Models

In [110], Idris et al. identified the runway system as a key constraint for the departure system and suggested that improvement in departure operations should target runway system inefficiencies. The authors identified the control points for the departure sequencing and the various constraints which affect the departure rate. A simple model of the departure system was presented, where buffers were used to represent the re-sequencing that was possible at different control points. Idris et al. presented similar queueing models in [109] and [107], discussing the departure system in more detail.

A simple, easy to calibrate, queueing model for the departure system was also considered by Pujet, Delcaire and Feron in [149]. The intention was to create a model which was easy to calibrate but which would allow the evaluation of taxi-out-time reduction schemes. Results showed that taxi-out-times were highly affected by departure system congestion. The aim was to reduce congestion by predicting queueing times and holding the aircraft on the stand, which was shown to be much cheaper. As aircraft must queue for take-off, it is also intuitive, if the runway forms a bottleneck for the departure system, that more aircraft mean a longer queue which leads

[1] MANTEA project web site: `http://cordis.europa.eu/telematics/tap_transport/research/projects/mantea.html`

to longer waiting times. At Heathrow, however, the separation rules mean that the delay can be highly dependent upon the type of aircraft and the allocated routes as well as the number of aircraft.

Another queueing model, this time for the combined arrival, turnround and departure process, was developed by Andersson et al., [8]. This was designed for the consideration of the ground movement at various airports, such as Hartsfield-Jackson Atlanta International (ATL), Dallas-Fort Worth Airport (DFW), Boston Logan (BOS) and George Bush Intercontinental Airport (IAH, at Houston). The departure rate was observed to be related to both the weather and airport congestion. This queueing model was further developed by Idris et al. in [108].

In [51], Carr et al. extended the Andersson et al. model of [8] for predicting taxi delay by including an additional expected delay for aircraft assigned to departure routes that would be closed. Controllers reported that delays occurred when departure routes were closed, but observed that the effect of increases to many of the separation rules could be absorbed by re-sequencing the departing aircraft or by changing the flight paths in the local airspace. The implication was that increased separations need not be considered when estimating delay but that closed routes should be. Results showed general success but a discrepancy in predictions occurred for aircraft with large downstream separation constraints leading the authors to suggest that the model should include factors to account for these large separations as well as complete route closures. These results imply that re-sequencing can account for smaller changes to separation rules but larger changes cannot be handled so easily.

Preliminary experiments with applying MDIs (increases in required separations, see section 2.5) using the departure system simulation described in this thesis were inconclusive. The effect of an MDI was found to vary widely depending upon the route(s) it was applied to and the number of aircraft queued for that route at that time but usually resulted in at least a small increase in the total delay for aircraft. London Heathrow is much more constrained by the downstream constraints (as shown in section 8.5) than many of the US airports are, and this can explain why the addition of small MDIs at those airports can be much more easily absorbed by re-sequencing than they can at Heathrow. Further research in this area is suggested in order to fully understand the effect of MDIs upon the delay for aircraft.

The Center-TRACON Automation System[2] (CTAS) is currently under ongoing development by NASA Ames research centre[3]. The list of publications[4] shows that the group have performed a lot of research for different parts of the system.

One part of the system, called FAST (Final Approach Spacing Tool), is responsible for solving the arrival control problem and is considered in section 3.6.1. Another part is responsible for the surface movement. The Surface Management System is not a true ground movement simulator but helps to reduce congestion by predicting queues and delays on the surface. Jung and

[2]NASA Ames Research Center, CTAS: `http://ctas.arc.nasa.gov/index.html`
[3]NASA Ames Research Center, official web site: `http://www.nasa.gov/centers/ames/home/index.html`
[4]NASA Ames Research Center, CTAS publication list: `http://ctas.arc.nasa.gov/publications/index.html`

Monroe discussed the development of the system in [118]. Atkins used the surface management system to evaluate runway efficiency in [17] by estimating what was done and comparing this with estimates of what could be done. In [19], Atkins and Brinton suggested the benefits of integrating the Surface Management System with the Traffic Management Advisor element of the system to obtain better arrival time estimates, [18].

3.3.3 Low-level capacity analysis

The layout of the airport is important for determining the maximum throughput and tools exist for simulating the running of an airport. It is possible to use low-level simulation of the airport to measure the capacity of the airport. Polak reported on a study into the capacity of Amsterdam Schipol airport in [113] using a fast-time simulation model called TAAM, described in section 3.8.3. Bazargan, Fleming and Subramanian, [25], also used TAAM to investigate the way in which the runway capacity of Philadelphia International Airport would be changed by changes to the airport layout.

3.3.4 Statistical approximation

Pitfield and Jerrard, [146] and Pitfield, Brooke and Jerrard, [145], used monte-carlo techniques to consider the effects of changes to the ground movement without having to implement a full simulator. In [145], statistical methods were employed to estimate the effects of towing an aircraft around a proposed new airport during peak arrival/departure times. Although the proposed airport design was cheaper than some alternatives, there were concerns that the effect of towing aircraft at busy times would be too great. Monte-carlo simulation showed that the effect was less than expected and the cheaper design could be adopted. In [146], similar methods were used to investigate the consequent capacity of different runway configurations under varying arriving/departing aircraft mixes.

3.3.5 Arrival Capacity versus Departure Capacity

There is a trade-off between the arrival capacity and departure capacity of a runway with both arrivals and departures. Newell considered this analytically in [135]. In terms of the sum of arrivals and departures, the total throughput is higher when two runways are used in mixed mode (using runways for both arrivals and departures) rather than segregated mode. Additionally, if the number of arrivals and departures is equal then alternating arrivals and departures on the runway achieved the optimal throughput.

The arrivals/departures trade-off curve was shown in [108] and was explicitly examined in [85] with the intention of using management of the capacities as the basis of a method to help ease congestion at busy times. Clayton and Capozzi also presented a model for measuring the arrival and departure capacity of an airport, and the trade-off between them, in [55]. The airport was modelled as a graph, with flow constraints on each arc and the capacity was approximated to the maximum flow through the arcs in the graph.

London Heathrow has to operate in segregated mode for most of the day due to regulations to reduce the noise for residents on flight paths, so a reduced capacity compared with mixed mode operations is only to be expected. However, the arrival/departure rate trade-off is explicitly performed at Heathrow by allowing the departure runway to be used in mixed mode for a few arrivals per hour if the arrival delay reaches a given threshold value. This temporarily increases the arrival rate, while decreasing the departure rate, preventing the arrival delay from rising to unsafe levels, [20].

3.3.6 A summary of capacity analysis

From the point of view of aircraft departure scheduling, the layout of an airport appears to be important in two main ways. Firstly, the layout determines the control points for the sequencing; those points where limited or full re-sequencing of the departing aircraft can take place. Secondly, it may impose additional constraints upon the schedules being built, for example requiring additional time to be built into a departure schedule to allow aircraft that have landed on another runway to cross the departure runway.

The runway is the interface between the airport and the airspace. The runway represents a bottleneck to the departure system for two reasons. Firstly, the wake vortex separations between aircraft must be attained in order to ensure safety at take-off, and secondly, flow constraints applied to the airport to control airspace congestion are usually applied at the runway. Take-off re-sequencing is used by the airport to try to alleviate the throughput effects of the capacity reductions imposed by these constraints upon the runway. The presence of these constraints on the runway has meant that the runway capacity is often used as a measure of the capacity of an airport as a whole.

3.4 Machine Scheduling Problems

Many scheduling problems have a relationship to machine scheduling problems. These problems consist of one or more jobs to process, one or more machines to do the processing, and a set of constraints and objectives. The processing of a job on a machine is called an operation. A job therefore consists of one or more operations and a machine will process zero or more of these operations.

A standard categorisation for machine scheduling problems, presented by Graham et al., [97], consists of a triple, $\alpha|\beta|\gamma$. The values α, β and γ are summarised below. This terminology has since been extended by later researchers such as Pinedo, [144]. Brucker and Knust maintain a list of the known complexities of various machine scheduling problems[5].

[5]Complexity results for scheduling problems, Peter Brucker and Sigrid Knust : `http://www.mathematik.uni-osnabrueck.de/research/OR/class/`

The α value specifies the machine configuration. For example a single machine problem is denoted by the value 1 for α. The single machine problem is important in the context of this thesis as it can be used to model the separation rules on the runway. The job shop problem is also important in the context of this thesis. In a job shop there are a number of jobs to be processed on a set of machines. Each job follows a specific path through the machines. The aim is to determine the sequence in which the jobs should be processed on each machine. A job shop problem of m machines is denoted by the value Jm.

The β value specifies the characteristics or constraints upon the problem. For example, the value s_{jk} (or *seq-dep*) indicates that there is a sequence dependent set-up time for machines. In this case, the time to set a machine up before processing can start on job k will depend upon the type of the previous job j that the machine was processing.

The value of r_j for β indicates a ready time for each job. Processing cannot start on job j until time r_j or later. A value of *prec* is used to indicate the presence of precedence constraints. These indicate relationships between pairs of operations such that the processing of one operation cannot begin until after the processing of another operation. Multiple values can be provided in the β field.

In some problems, there is limited (or no) possibility for jobs to wait between machines. In this case the problems are called blocking (denoted *block*) as the previous machine will be blocked whenever there is no intermediate space left between machines. The ultimate form of the blocking problem is where there are no buffers at all between machines, so jobs block the previous machine until they can be moved to the next machine. In a no-wait problem (denoted *nwt* or *nowait*), not only are there no intermediate buffers between machines but jobs must not be held on a machine beyond the end of processing.

The γ field indicates the objectives of the scheduling. Possible values include the minimisation of the makespan, the time to the completion of the last operation (denoted C_{MAX}), or the sum of the completion times (denoted $\sum C_j$).

The most important problems in the context of this thesis are:

The $1|s_{jk}, r_j| \sum w_j C_j$ problem : single machine models for runway sequencing

A single machine scheduling problem with sequence dependent set-up times and ready times. The objective is to reduce the total weighted completion time. This problem is related to the arrival scheduling problem and is strongly NP-hard, [31].

The $Jm|s_{jk}, r_j|C_{MAX}$ problem : job shop models for holding area movement

A job shop problem, with sequence dependent set-up times and ready times for jobs. Here the objective is to reduce the total makespan; the time at which the last job finishes. The total weighted completion time ($\sum w_j C_j$) is an equally applicable objective. This problem incorporates some of the characteristics of the holding area problem which is discussed in chapters 4 and 6.

3.4.1 Blocking job-shop problems

Blocking job shop problems are particularly important for the holding area problem considered in this thesis. Hall and Sriskandarajah presented a survey of the research into the blocking and no-wait machine scheduling problems in [100]. The complexity is reported for many different problems although the majority are flowshop rather than jobshop scheduling problems. Mascis and Pacciarelli presented further results in [129] and explained how these problems can be expressed using the alternative graph introduced in [128].

The disjunctive graph model for the job shop problem is a powerful and useful representation of the problem. An introduction to disjunctive graph models for job shop problems can be found in [42] and [144]. The alternative graph is an extension of the disjunctive graph, also called the generalised disjunctive graph. The alternative and disjunctive graphs are explained in more detail in section 4.3 due to the applicability to the holding area movement problem.

Pacciarelli and Pranzo, in [140], elaborated upon the equivalence between a railway scheduling problem and the blocking job shop problem and modelled the problem as an alternative graph. A tabu search algorithm was presented for the problem. As the solution space was not connected, the algorithm strives for feasibility from infeasible solutions and optimality from feasible solutions.

3.5 The Travelling Salesman Problem

The travelling salesman problem (TSP) involves finding a sequence in which a salesman should visit a set of cities and return to the starting point such that the total distance travelled is a minimum. Let the distance from city i to city j be denoted by d_{ij}. In the symmetric travelling salesman problem, $d_{ij} = d_{ji}$ for all cities i and j. If this equality does not hold, then the problem is called the asymmetric travelling salesman problem (ATSP).

The travelling salesman problem has been widely studied for many years, [123, 29]. The general problem is *NP-Hard*, [114], so there is no known algorithm which will solve all instances in polynomial time. Many special cases of the TSP can be solved more easily, [86, 43, 23], and many heuristic algorithms work well in practice, [116].

The ATSP is often harder to solve than the symmetric TSP and has also been well studied. Many different techniques have been applied, [45, 175] and comparisons of methods have been made, [115, 54].

As long as the set-up times obey the triangle inequality, the single machine scheduling problem, with sequence dependent set-up times is equivalent to the ATSP, where the salesman is represented by the single machine, the cities visited are represented by the jobs and the distance d_{ij} can be represented by a variable setup time, s_{ij}, necessary to set up the machine to accept job j after job i. The aim is then to reduce the makespan, the completion time of the last job to finish.

Bianco et al. discussed the equivalence between the ATSP with time windows and the $1|r_j, s_{jk}|C_{MAX}$ single machine scheduling problem in [34] and presented a branch and bound algorithm for it. As the arrival scheduling problem, with an objective of minimising the schedule duration rather than delay, can be formulated as the $1|r_j, s_{jk}|C_{MAX}$, the ATSP can be considered an appropriate model for some of the arrival scheduling problems. This equivalence is not valid for the take-off sequencing problem at Heathrow, due to the structure of the separation rules. The introduction of take-off time-slots into the problem and the fact that these cannot be treated as hard constraints further complicates the problem.

3.5.1 Cumulative Travelling Salesman Problem

A variant of the TSP, the cumulative TSP, [33], is relevant for the arrival scheduling problem, [31]. The cumulative TSP uses the same model as the TSP, however the objective is different. For the cumulative TSP, the cost of a solution is not the sum of the distances but the cumulative sum of the distances from the source city to each city in turn. The objective for the equivalent single machine scheduling problem is therefore the minimisation of the sum of the completion times of the jobs rather than minimising the maximum completion time of the jobs. In [33], Bianco, Mingozzi and Ricciardelli discussed the equivalence of the $1|s_{jk}|\sum C_j$ problem to the cumulative asymmetric travelling salesman problem (cumulative ATSP). Bianco, Dell'Olmo and Giordani showed the equivalence of the $1|r_j, s_{jk}|\sum C_j$ machine scheduling problem to the cumulative asymmetric travelling salesman problem (ATSP) with ready times or time windows in [31]. The authors made clear that this is exactly the formulation usually used for the arrivals sequencing and scheduling problem.

3.6 Arrivals Sequencing

The arrivals problem and departures problem have many similarities, although they also have important differences. Both the arrivals and departures problems are often considered to have separate sequencing and control problems. The sequencing problem consists of finding a good landing or take-off sequence for aircraft. The control problem consists of identifying how the desired landing or take-off sequence will be achieved and whether it is actually possible.

The sequencing problem is the part of the arrivals problem that is most relevant to departure scheduling. Indeed, researchers who have developed solutions for the arrival sequencing problem have stated that the solution will work equally well for the departure sequencing problem. In practice, however, this would mean ignoring the important downstream constraints at airports such as Heathrow. Even so, a review of the research into the arrivals problem potentially offers insight into solution methods for the departure problem.

3.6.1 The control problem for arrivals

The control problem for arrivals usually involves allocating paths to arriving aircraft so that they will arrive at the runway in the desired landing sequence. This may involve changing the length of the path taken, so as to manipulate the landing time, or holding for different amounts of time at intermediate points. Existing tools usually take a target landing sequence and calculate the paths to achieve it, rather than solving both the control and sequencing problems simultaneously. For example, the Final Approach Spacing Tool (FAST) (introduced by Colin Smith in [159] developed by NATS (formerly National Air Traffic Services Ltd) is responsible for advising controllers about when aircraft should make turns in order that the approach legs are the correct length to ensure the correct separations on approach and landing.

A Final Approach Spacing Tool[6] (FAST), is also an element of the Center-TRACON Automation System[7] (CTAS) from the NASA Ames Research Center[8]. The CTAS FAST was described by Davis, Krzeczowski and Bergh in [60] and aims to merge arriving traffic flows. Although the merge order is partially determined by the current speeds of aircraft and the distance from the airport, the primary influences upon the merge order are the existing sequencing decisions.

Even when running in passive mode, providing only runway assignment and sequencing advice, operational results showed substantial throughput increases, [59]. However, the main gain was considered to be the reduction in controller workload by making the controller aware of the possibilities for slipping later aircraft into the existing schedule, [151]. A major expected benefit of the decision support system presented in this thesis would also be that of making the runway controller aware of aircraft that are not currently visible at the holding area. The results presented in section 8.4 of this thesis imply that there is considerable value in considering aircraft while they are still taxiing.

Fuzzy reasoning was used in the CTAS FAST to introduce expert knowledge of arrival sequencing into the system, [151] and the design of the system required considerable controller input, [101]. Without this input a developed system is far less likely to be of practical use. Similarly, in the problem considered in this thesis, controller feedback, consideration of controller behaviour and examination of actual aircraft movement was essential in areas such as path selection (described in section 2.8) and in understanding what a controller would and would not accept.

Bianco, Dell'Olmo and Giordani presented a model for scheduling the arriving and departing aircraft in the near-airport airspace (the Terminal Manoeuvring Area or TMA) in [32] that resembles the ground movement models described in section 3.8. Traversal paths were defined through the local airspace in the same way that paths are defined along the taxiways in the

[6] CTAS FAST, web site: `http://www.ctas.arc.nasa.gov/project_description/fast.html`
[7] NASA Ames Research Center, CTAS web site: `http://ctas.arc.nasa.gov/index.html`
[8] NASA Ames Research Center, official web site: `http://www.nasa.gov/centers/ames/home/index.html`

ground movement. In a similar way to the ground movement models presented in section 3.8, the paths were pre-allocated to aircraft and the system was responsible for determining a combined landing and take-off schedule that would maintain all required separations. The authors considered that the decomposition of the TMA handling into separate control and sequencing problems was not conducive to finding an overall optimal solution, as discussed above. This consideration holds equally well for the departure problem, which is why the control and sequencing problems are both considered in the decision support system described in this thesis.

3.6.2 The arrival sequencing and scheduling problem

The sequencing and scheduling aspects of the arrival problem may be solved simultaneously or separately. It is worth considering the previous research into the sequencing and scheduling of arrivals in case it is also applicable to the sequencing problem for the departures.

Brinton presented a branch and bound algorithm to solve the sequencing problem in [40] and showed how precedence constraints could be incorporated into the model. The chosen objective was to minimise the delay between the actual and earliest landing times for aircraft. It was always better to schedule an aircraft as early as possible, so it is easy to allocate landing times to aircraft once a landing sequence has been determined. The analogous situation will be seen in the departure problem. In order to ensure a fast solution for the problem, a maximum position shift constraint was added (as described in section 3.6.3), limiting the size of the search tree. The problems of applying a maximum position shift to the departure sequencing problem, due to the presence of CTOT take-off time-slots, are discussed in section 3.6.3 and apply equally well to applying this method. The general problems of applying a branch and bound method to the departure problem are discussed in section 4.5.1.

Despite the objective in [40] to land aircraft as early as possible, the more common formulation of the arrivals problem penalises aircraft landing before as well as after a target landing time. Given a landing sequence and a convex piecewise linear function for the cost of an aircraft deviating from its ideal landing time, it is possible to formulate the landing time problem as a linear program (LP) for solving by any LP solver. This piecewise linear cost function is exactly the form of the objective function used in [2, 26, 27, 74]. These papers all use a similar model for the arrival scheduling problem where the objective of these problems is usually to minimise the deviation from target landing times, with some additional factors considered in some cases.

Fahle et al. give an overview of some of the major formulations for the arrival sequencing and scheduling problem in [78]. Although the departure sequencing problem is also mentioned, the experimentation was performed only on the arrival sequencing problem. Written after many of the other papers mentioned in this section, it summarises and compares the most common methods used to model and solve the arrival sequencing problem.

The first model was a mixed integer program (MIP) where landing times are given by continuous variables and zero-one integer variables were used to represent the landing order. Beasley et al. used a MIP formulation similar to this in [26], simplifying the problem by noting that the values of the zero-one integer variables can be predetermined for aircraft where there is no overlap in landing time-slots. The model was applied to the multiple as well as single runway situation.

The second representation was produced by discretising the landing times. In this way the continuous variables for the landing times can be eliminated and an integer program results. The cost of a deviation from the ideal landing time then need not be linear, so different cost functions could be modelled. This was obviously at the expense of efficiency.

The third problem representation considered the problem as a series of constraints and uses a constraint programming approach. An introduction to constraint programming can be found in [80]. Although formulated in a different way, the model is very similar to that given in the MIP formulation. The objective was to find take-off times that minimise the weighted sum of the time before and after the target time. A successor variable was used to denote the landing order and was exactly analogous to the ordering variables of the MIP. Special fixing procedures were applied to determine and fix the optimal landing time for an aircraft once it could no longer be affected by other aircraft and transitivity constraints were added to allow the constraint programming solver to reduce the domains of redundant variables.

The fourth problem representation presented the arrival sequencing problem as a satisfiability problem. In this case, the objective was to find a feasible rather than optimal solution. Time was again discretised and boolean variables were used to represent the proposition that an aircraft had landed at a certain time.

The fifth formulation was provided for use by local search methods. A solution was represented by a landing sequence alone. Landing times were heuristically determined for each landing sequence in order to obtain a cost, by landing each aircraft as early as it can, given the landing window and separation rules. Although ensuring landing times can always be assigned if a feasible assignment exists, the obvious weakness with this method is that the earliest time may be far from the optimal time so this method is unlikely to give optimal results.

Abela et al. presented a genetic algorithm to investigate the arrivals problem in [2]. An introduction to genetic algorithms can be found in [155]. The potential solutions consisted of a set of landing times for aircraft and the landing sequence was determined implicitly from the landing times. Two heuristics were applied to the solutions before the cost was evaluated. The first forced the schedule to be feasible by delaying aircraft to ensure the separation rules were obeyed. The second shortened unnecessarily long separations between aircraft, attempting a local optimisation on the schedule. The authors also presented a branch and bound algorithm for solving the problem exactly by formulating it as a 0-1 mixed integer program and then using heuristic upper bounds and a linear relaxation of the problem to obtain a lower bound. It is important to note that only four categories of aircraft had to be considered in the arrival sequencing problem. Once combinations of departure routes, speeds and weight classes are considered, there are far more than four categories of aircraft to consider in the departure problem.

Ernst, Krishnamoorthy and Storer proposed a branch and bound method for solving the sequencing problem in [74] and used a network simplex variant to allocate landing times for a given landing sequence. A problem space search heuristic consisting of a genetic algorithm to generate the weights used in a constructive heuristic which was then used to obtain a good upper bound for the branch and bound. The branch and bound algorithm was extended to the multiple runway problem, introducing the runway assignment problem, where aircraft have to be assigned to runways as well as sequenced.

Beasley, Sonander and Havelock used a population heuristic to assign landing times to arriving aircraft in [28]. In this case the population members represented the position of each aircraft within its landing window. The fitness of each solution was evaluated based upon the square of the deviation from the target landing time. An 'unfitness' was also evaluated for solutions where separation rules were broken. The population replacement scheme used both the fitness and unfitness to move the population towards both the feasible landing orders and those which land aircraft closer to the target landing times. The results compared favourably with those from a simple improvement heuristic and with the real controller-produced schedule.

Beasley et al. adapted the population heuristic approach of [28] to the dynamic scheduling case in [27]. Here the value of proposed landing times depended not only upon the ideal landing times for aircraft but also upon the previously proposed landing times. Three solution methods were tested. The first was an adapted version of the exact method used in [26]. The second was an adapted version of the heuristic method used in [26]. The third was a version of the population heuristic described in [28], modified to work with multiple runways. The methodology was tested using a rolling window approach, simulating the way in which a real system becomes aware of aircraft. Results for the dynamic problem were not as good as those for the static problem and the exact method often outperformed the heuristic methods, usually doing at least as well. However, importantly, in some cases the heuristic methods outperformed the exact method on the dynamic problem. This shows that the best solution for the dynamic problem at each stage (which would always be found by the exact method) is not always the best solution for the overall problem. An improved solution method for the sub-problems did not always benefit the final solution for the whole problem. The same thing was found occasionally for the departure problem considered in this thesis, as described in section 8.6.1.

Caprì and Ignaccolo also considered the dynamic arrivals problem, in [49]. The problem changed over time as new aircraft enter the system so relative position shift constraints, similar to those in [27], were added to keep solutions similar to those previously identified.

Cheng, Crawford and Menon presented three encodings for the arrival sequencing and runway allocation problem, for solution by genetic algorithms, [52]. The first formulation included runway allocation in the chromosomal structure, the second and third assigned the runway according to a priority scheme. They also presented an encoding which has similarities to a genetic programming approach, [122]. Hansen applied the same solution methods to test problems in [102] and discovered that the genetic programming-like approach consistently performed better in his experiments, although it performed the worst in [52].

A number of elements of the arrival sequencing research could potentially be applied to the departure sequencing problem. In particular, it is important to consider whether the approaches to solving the sequencing problem could be applied to the departure problem. The approaches to the sequencing problem can be divided into two categories: exact and heuristic. The problems with applying the various exact approaches are considered in section 4.5, following the discussion of the characteristics of the departure problem in sections 4.2 to 4.4. Although a heuristic approach is used in the designed decision support system, a tabu search was selected instead of the population heuristic/genetic algorithm that was used in research such as [28, 52, 102]. The developed tabu search algorithm performs well on the problem, and avoids the issues of regaining a feasible take-off schedule which had to be performed by the algorithms used in [28, 52, 102]. The consideration of the holding area movement makes the issue of recovering a feasible take-off schedule both more important but also more difficult to achieve.

The observations that were made in [27, 49] about penalising deviation from a previous schedule are directly relevant to a dynamic departure scheduling problem. In section 5.6.5, a similar element will be observed in the design of the decision support system described in this thesis.

3.6.3 Reducing the complexity of the problem

To reduce the size of the sequencing problem, methods have been applied to reduce the number of aircraft considered at any given time. The most common options are to decompose the problem according to time windows, for instance using a rolling time window through the aircraft and solving a sequence of sub-problems, [117], grouping the aircraft into categories and solving for these groups, [147], or limiting the number of possibilities for any position in the sequence, [63, 64].

Jung and Laguna, in [117], applied a rolling time window to a problem like that in [26], creating sub-problems that could be solved exactly. Given the advantages of increasing the planning horizon that can be observed from the results in section 8.4, it is perhaps unwise to reduce the size of the departure sequencing problem considered in this thesis in a way which would reduce the effective planning horizon. On the contrary, it may actually be useful to increase the effective planning horizon.

Hu and Chen used a receding horizon approach in [105] and [106] and evaluated the effects of different horizon lengths. This method involves solving the problem for a rolling window of aircraft and applying a terminal penalty to reflect the expected effect of the current schedule upon those beyond the horizon. The decision support system presented in this thesis uses a similar method to solve the problem although the horizon is determined implicitly by the amount of knowledge the system has at the time rather than by an explicit selection. The complexities of the departure system scheduling constraints make it impractical to apply a terminal function as the decision of what to do with the last aircraft in the sequence depends heavily upon what aircraft will appear next. Instead the system is given (possibly inexact) knowledge of the taxiing aircraft so that the sequence will be fixed by the time it has to be enacted. This inexact knowledge of the taxiing aircraft could be considered to be performing the role of the terminal function in a receding horizon approach.

Maximum position shift

Psaraftis presented a dynamic programming approach for scheduling groups of identical jobs in [147]. The approach was suggested for arrival sequencing and scheduling by grouping aircraft by weight class. Constrained Position Shift (CPS) rules, [63, 64], were applied to reduce the size of the problem and incorporate a concept of priority. A maximum position shift (MPS) value was applied, limiting the positions which each aircraft can take off to be within a maximum distance from the position of the aircraft in the first-come-first-served sequence.

A local improvement heuristic with maximum position shift was also presented by Luenberger in [126] and compared favourably against an approach which used an exhaustive enumeration with maximum position shift for a rolling window through the aircraft. A major advantage of the MPS method is that it allows a simple trade off between solution quality and search time by varying the value of the MPS parameter.

Trivizas also used a maximum position shift constraint in [166] and suggested a dynamic programming approach to solve the sequencing problem for both arrivals and departures. Balakrishnan presented a dynamic programming based approach which could support precedence constraints in [22] (which were missing in [166]) but this method required that the separation rules obey the triangle inequality, so would be inapplicable for departure sequencing at Heathrow.

Although useful for the arrivals problem, where the deviation of the take-off position of any aircraft from its position in first-come-first-served sequence is highly restricted, the CTOT time-slot constraints cause a problem for applying this approach to the departure sequencing problem. The take-off time-slot may require an arbitrarily long positional advance or delay for an aircraft, as observed in the example results from this system given in chapter 8.8. In particular, it should be noted that the best schedule found would have required a maximum position shift of 25 to be applied. Such a large value for the maximum position shift would mean that the approach did not actually reduce the size of the solution space.

3.7 Departure Sequencing

The departure sequencing problem is analogous to the arrival sequencing problem, but there is no advantage for departures from taking off any later than the earliest take-off time. Take-off time prediction, given a take-off sequence, is therefore much easier than landing time prediction given a landing sequence. Indeed, at Heathrow, the contention for the runway is such that, for the majority of the day, applying a delay to any aircraft would usually delay so many following aircraft that it would be extremely inadvisable even if there were a benefit for a single aircraft.

A conceptual architecture for a departure planner was specified by Anagnostakis et al. in [7], with two different levels of planning. A strategic planner would schedule departures with a three to four hour window and a tactical planner would then provide some rescheduling with a 15-30 minute window. Members of the same research group then considered the problem in more detail.

Anagnostakis et al. gave a summary in [6] of the sometimes conflicting criteria that affect departure scheduling. They presented a model for a departure planner which took the various types of runway usage into account, including arrivals, departures and runway crossings. A search tree was suggested for solving the problem in conjunction with a branch and bound or an A* algorithm.

The departure process was further analysed by Anagnostakis and Clarke in [4, 5], and a two-stage departure planner was proposed. The research builds upon the earlier work of the group, such as that presented in [111] and [148]. The work is accessibly documented in the report based upon Anagnostaskis' thesis, [3].

The two-stage departure planner in [4, 5] works by applying a problem decomposition. The first stage sequences departure class slots and ignores downstream constraints. The second stage assigns aircraft to slots to fulfil the remaining constraints. The problem with applying this type of approach at Heathrow is that the position of the airport means that the downstream constraints have more effect upon the sequencing than any others, as will be shown in section 8.5, so it is inadvisable to leave these for later consideration.

3.7.1 Constraint satisfaction

Van Leeuwen, Hesselink and Rohling presented a constraint satisfaction based model for the departure problem in [168], scheduling up to 12 aircraft in a 50 minute time interval. Here the aim was not to obtain an optimal delay or throughput, but to assign a runway and departure route to each aircraft such that all aircraft can take off within the required time slots. This problem differs significantly from that considered in this thesis, as does the problem size. Firstly, as discussed in section 1.3.5, it is assumed that the departure routes and runways are pre-allocated to aircraft for the problem under consideration in this thesis, since this is the case from the point of view of the runway controller. Secondly, since Heathrow has typical departure rates of between 30 and 50 aircraft per hour, the size of the solution space which needs to be considered is significantly increased. Thirdly, there are problems with allocating take-off time-slots to aircraft. Aircraft departing from Heathrow sometimes need to utilise extentions to CTOT take-off time-slots, so these cannot be treated as hard constraints. On the other hand, many aircraft do not have mandatory take-off time-slots, and adding such is not feasible due to the contradictory pressures towards equity of delay (implying that time-slots should be tight) and allowing for the large delays that can accumulate at busy times (implying that time-slots must be large enough to allow for this delay). In summary, while the selected model in [168] was obviously appropriate for the problem considered therein, it is less appropriate for the Heathrow situation.

3.7.2 Holding area constraints

As will be seen later, the holding areas at Heathrow inflict considerable constraints upon the departure problem since re-sequencing of the aircraft is usually achieved within the limited space at the holding area.

In 2001, John Greenwood of NATS proposed a problem at the 40th European Study Group with Industry. The report, [124], has a good overview of the problem at Heathrow with regards to the relative problems of sequencing at the stands compared to within the holding areas. A simple branched holding area is considered within the report, where up to three aircraft may be held on a side branch, letting later aircraft overtake. This is a simplified model of one of the holding areas at Heathrow. Using this model, a dynamic programming solution was created that could solve the problem to optimality in reasonable time. The problems of applying this sort of approach to the far more flexible real Heathrow holding areas are considered in section 4.5.

Bolender and Slater considered the runway scheduling at Dallas-Fort Worth (DFW) International Airport in [39]. The departure sequencing consists of selecting a runway for aircraft, determining a sequence in which to release aircraft onto taxiways and then creating a take-off sequence by interleaving two runway queues. The various problems were considered separately, although experiments performed by the authors indicated that the sequence in which aircraft were released to the stands could make a difference to the queueing time for the runway.

Sequencing by interleaving two queues can be viewed as a simplification of the Heathrow holding area problem. Indeed, this was a simplification of even the DFW problem, as the authors explained that there was actually a third queue, used for aircraft with take-off time-slots, which was not considered in the problem. The initial sequencing from the stands is comparable to the ground movement and push-back sequencing at Heathrow.

Bolender addressed the same problem in his thesis, [38], along with the additional, separate problem of merging departing aircraft with arrivals in the same airspace while maintaining or attaining all required inter-aircraft separations. This latter problem is beyond the scope of this thesis as the departure separation rules are present at Heathrow to ensure that the inter-aircraft separations will be achievable.

A number of greedy heuristics for the queue interleaving problem were evaluated in [38] and the heuristic that was presented in [39] was found to perform the best. Evaluation of queue allocation strategies showed that the best performance required both a balance of aircraft between queues and a consideration of the take-off constraints. The queue allocation problem can be considered to be analogous to the holding area entrance allocation problem at Heathrow.

Even though the research presented in [38, 39, 124] considers a similar problem to that considered in this thesis, the solution methods utilised relied upon the holding area structures being far more restrictive than those at Heathrow. The solution methods utilised do not scale well as the complexity of the holding area structure is increased. The application of a dynamic programming solution approach using the Heathrow holding areas is discussed in more detail in section 4.5.2.

3.7.3 Further mixed-mode research

In his thesis, [162], Teixeira examined a mixed-mode problem. Given the predicted gaps in an arrival stream that may be utilised for a take-off, the taxi time predictions and the separation rules, a greedy algorithm was produced to determine the sequencing of departures within time windows using a rolling horizon approach. A local improvement algorithm was then applied to the initial solutions from the greedy search in order to improve the sequence. Despite the fact that the problems differ, there are some similarities between the approach in [162] and the one used in this thesis, since both involved a heuristic search (the tabu search used in this research is described in section 5.3) followed by a local improvement method (the follow-on searches are described in section 5.7.3). The tabu search algorithm is more powerful than a greedy algorithm, however. The results for the first-descent algorithm in appendix C show that a greedy method worked well for the problem considered in this thesis, but that the selected tabu search algorithm worked significantly better.

3.8 Ground Movement

The control problem for take-offs can be viewed as a ground movement problem. The applicability of previous ground movement research should, therefore, be considered. In general the previous ground movement research considers a larger ground movement problem and has more emphasis on getting to the runway as quickly as possible or by target times rather than in achieving better take-off sequencing.

The ground movement at an airport is dynamic as the situation changes over time and rarely evolves in exactly the way that was predicted. It can be extremely difficult to predict future positions for taxiing aircraft due to the uncertainties involved in taxi times, the time spent taxiing from the stand to the runway. For this reason, ground movement research usually involves a ground movement simulator, responsible for simulating the uncertainty to allow a better evaluation of the worth of the decisions being made.

3.8.1 Simulation

Odoni et al. assessed the many simulations that were available at the time [136]. More recently, the THENA (THEmatic Network on Airport Activities) Consortium position paper on airport simulation and modelling issues, [163], lists various simulations.

Some of the simulations available for ground movement are discussed in more detail below. The ATOS (Airport Traffic Optimization Simulator) is discussed as it is a simpler simulation and the methods used are of relevance to the design of the decision support system described in this thesis. TAAM is then discussed as it is widely used in airport ground movement research and NATS use TAAM for ground movement research. Despite the availability of TAAM, NATS use an in-house simulation called HERMES (HEuristic Runway Movement Event Simulation) for more detailed modelling of the runway operations.

3.8.2 The Airport Traffic Optimization Simulator (ATOS)

The Airport Traffic Optimization Simulator[9] (ATOS), was developed by the Global Optimisation Laboratory of ENAC[10] (Ecole Nationale de l'aviation civile) and the Airports, Towers and Terminal zones (ATT) division of CENA[11] (Centre des Etudes de la Navigation arienne). The simulator was described by Pesic, Durand and Alliot in [143], Gotteland et al. in [96] and later by Gotteland and Durand in [94] and includes an optimisation component to reduce the taxi times or delay for aircraft.

The aim of the early research was to develop a ground movement optimiser to allocate paths, holding times and holding positions to aircraft in order to reduce the ground movement time and enable target take-off schedules to be achieved without causing conflicts between aircraft. Two aircraft were said to be in conflict if they were too close together on the taxiways, or attempted multiple occupancy of the runway. Later research included the wake vortex separation rules for the runway and used these to determine take-off times for aircraft. The airport was modelled as a graph, where nodes represented positions on the taxiways and arc lengths the distances between these positions. Each aircraft was assigned a path to use to reach the runway, which could change over time. Aircraft could hold at various points on these paths. The solver was responsible for assigning paths to aircraft and determining where and for how long the aircraft should hold. The simulator was responsible for moving aircraft around the taxiways according to the paths that had been allocated to them. The amount moved depended upon the taxi speeds, which have a degree of uncertainty added to them, and the holding rules that have been specified.

[9]ATOS : Airport Traffic Optimization Simulator, web site: `http://www.recherche.enac.fr/opti/ATOS/`
[10]ENAC, Ecole Nationale de l'Aviation Civile (ENAC), web site: `http://www.recherche.enac.fr/`
[11]CENA, Centre des Etudes de la Navigation aérienne, web site: `http://www.cena.fr/english/index.html`

The addition of an extra delay, as a function of the turning angle required, was discussed in [143]. In [94], the cost of a link was related to the time needed to traverse it plus some penalty for the undesirability of the link. The simulator used a rolling window approach, considering at any time only the aircraft that would be taxing within a specified time window.

Path pre-allocation was used to ensure that only paths with a cost or length within a given distance of the optimal path would ever be assigned. The complexity of the problem was reduced both by using this path pre-allocation and by an additional decomposition into sub-problems consisting of aircraft in contention which each other. A graphical illustration can be found on the ATOS web site[12]. This sort of path pre-allocation will also be seen in the decision support system presented in this thesis, although a stronger condition will be applied to the path allocation such that aircraft will always be given the shortest paths necessary to achieve the desired take-off re-sequencing.

A number of solution methods were described including a fixed priority scheme for moving aircraft, an approach which used a genetic algorithm to assign priorities and path allocations and another which used a genetic algorithm to evolve path allocation, holding positions and holding times. The first genetic algorithm performed the best.

A custom crossover function was used in [143, 94], attempting to ensure that the child of the crossover between two parents only inherited conflicts between aircraft if the conflicts appeared in both parents. As the conflict matrix can be considered as a partially separable function, each conflict was effectively treated separately and the approach described by Durant and Alliot in [73] was applied. The guided mutation operation also used the concept of determining the worst aircraft, in terms of conflicts, and applying a mutation that would change the parameters for the aircraft with the worst local fitness.

Gotteland and Durand concluded in [94] that the runway was not the only thing that affects the delay for aircraft. Their experiments showed that the ground movement could have a significant effect upon the delay that aircraft suffered. However, the authors stated that a lack of real data meant it was impossible to compare their results to the real situation. Whether the ground movement has a significant effect in practice depends upon how well the real ground movement controllers are able to solve it and the situation appears to be slightly different at Heathrow. At Heathrow there are often queues in the holding area, which provide a buffer to limit the effects of taxi time uncertainty. In that case it is possible for the ground movement problem to be easily solved well enough by controllers to ensure that the runway contention is the only factor that will actually affect the delay. For example, there is little difference between spending three minutes queueing or spending two minutes queueing and an extra minute taxiing.

[12]ATOS web site: `http://www.recherche.enac.fr/opti/ATOS/`

3.8.3 The Total Airspace and Airport Modeller, TAAM

The Total Airspace and Airport Modeller (TAAM) by Preston Aviation Solutions and the Australian Civil Aviation Authority is a comprehensive simulator covering both the airport and airspace. According to Preston Solutions[13], TAAM "is the most sophisticated simulation tool of its type available in the aviation industry". TAAM is well known in the industry and can be used for studying the effects of changes to the airports or airspace. The simulation works at a detailed level, modelling the characteristics of aircraft and using these to determine what can be done.

The Eurocontrol Experimental Centre (EEC) of the European Organisation for the Safety of Air Navigation (Eurocontrol[14]), performed an evaluation of the accuracy of TAAM in 1999, [75]. One of the sub-projects was concerned with the accuracy of airport modelling and Brussels airport was selected for the test. Although the experts identified some areas where improvement would be useful, one of the conclusions in [75] was that "TAAM has demonstrated a significant capability to simulate an airport and the vicinity, in a manner that can be very close to reality".

The two main problems with any low-level simulation are the complexity of the model and ensuring that is it correctly calibrated. The complexity of TAAM led to the MITRE Corporation's Center for Advanced Aviation System Development (CAASD) being asked by the U.S. Federal Aviation Administration (FAA) to document some of the techniques which were being passed around informally between users, [37].

In order to calibrate a simulation, it is common to run it with historical data, with aircraft performing as they would have done historically and to then examine the performance against the real situation. A close correlation between the predicted and actual performance would indicate a better calibrated system. The effect of changes to the situation can then be assessed.

3.8.4 Simulation for real-time decision making

Holden and Wieland discussed the use of a simulation for real time optimisation, [104]. The key requirements are that the system be fast and accurate and that the optimisation procedure be reactive to real time changes. Accurate real-time information would also be needed about not only initial flight plans, but also about any changes such as cancellations or delays.

There are two important problems with using a simulation like TAAM for real-time decision support for the runway controller. Firstly, there is the issue of speed. When discussing the importance of parallelising part of the TAAM algorithm, [160], Sood and Wieland stated that the length of run-time limited TAAM's use at the time.

[13]Preston Solutions, Outline of TAAM, at `http://www.preston.net/products/TAAMoutline.htm`

[14]European Organisation for the Safety of Air Navigation, EUROCONTROL, web site: `http://www.eurocontrol.int`

Secondly, but equally importantly, is the issue of obtaining real time data of the accuracy required by a simulation such as TAAM. This seems to be overly optimistic and is unlikely to be possible at the moment. It is far easier to perform an off-line simulation than to use one for on-line decision support since historic data is much more easily available than real-time data. For this reason, real-time decision support tools tend to use simpler models of the airport than that provided by TAAM and usually assume that less data will be available at the time.

Dalal, Groel and Prieditis presented a real time decision making system that used simulation in [58]. The 'Simulation-based Real time Decision Making' system proposed by them used simulation to evaluate the effects of potential real-time decisions in order to guide the choice of decision. This was effectively building a simulation into the decision making system, in order to get a better evaluation of the worth of potential decisions. The decision making element of the decision support system presented in this thesis performs the same sort of operation in that a simple simulation of the movement within the holding area is performed to evaluate each solution, with the predicted movement of aircraft which are currently still on the taxiways also being taken into account.

3.8.5 Other Ground Movement Simulations

Confessore, Liotta and Grieco discussed another simulation based architecture for a decision support system in [56]. They put forth a case against using SIMMOD or The Airspace Machine (two detailed simulations that include ground movement) for decision support for apron operations. The main objections were the level of training required, as discussed in [161] and the requirement for a significant effort to validate the airport model, discussed in [8].

It should be noted that the authors of [56] use the term 'apron' to include all of the taxiways, from the gates to the runways, rather than the usual meaning of the area where aircraft park, [157]. Again, a graph model was used for the ground movement, although in this case the arcs represented the path segments and had occupancy rules and the nodes represented the positions at which aircraft could start or stop their travel or at which they had a choice of direction. Arcs were given a weight (related to their capacity) and direction as well as a length. Each aircraft had a pre-determined preferred path and some alternative path(s), simplifying the path allocation problem. A shortest path algorithm was used to determine the paths that aircraft would actually take, according to the current weights of the arcs. The arc weights were changed dynamically according to the number of aircraft using the arc, so the actual path used would depend upon the load at the time. The authors compared the paths that were allocated against the situation where only the preferential paths were used. A substantial reduction in taxi time resulted from allowing these alternative paths to be used in order to reduce the load on the preferential paths.

Smeltink et al. considered the ground movement along taxiways in [157, 158]. The airport taxiways were modelled as a graph, with the arcs representing the taxiways and the nodes the points at which taxiways join. The path assigned to each aircraft was determined by the start and end points of the journey. The problem was formulated in terms of the times at which aircraft should pass different nodes in the graph, ensuring that separation rules were obeyed during taxiing and that aircraft reached the runway on time. By appropriate formulation of the various constraints, a mixed integer problem was created which was then solved using CPLEX, using a rolling horizon model to reduce the problem size for each window.

In [169], Visser and Roling also formulated the ground movement problem as a mixed integer linear program. Both time and space were discretised in their model to enable the constraint formulation. A graph of the taxiways was again used, with the nodes representing the valid positions for aircraft. Nodes were defined so that aircraft in adjacent nodes maintained the required separation distances. Constraints enforce a one-aircraft-per-node rule, to ensure separations. Aircraft were pre-allocated a specified number of paths, including the path with the least nodes. The objective was to minimise a linear combination of the total taxi time and total holding time. A maximum delay was specified per aircraft, in order to limit the inequity of the schedule. Two speeds were used; a fast and a slow speed. Arcs between nodes were labelled by the number of discrete time units an aircraft of each taxi speed would be expected to take to traverse it. Flow conservation constraints during the optimisation ensured that each flight had one path and that delays on path segments were sequential. The results showed success on small problems.

Garcia et al. considered the ground movement problem in [81]. They modelled the airport resources as a graph with flow limits and simplified the problem by assuming that all delay for aircraft on a path around the airport was taken as initial waiting at the gate and that all aircraft taxi with identical (known) taxi speeds. The justification for the simplification was that it was desirable for the delay to occur at the gate. This is true, but no measure was provided of how much this simplification affected the solution of the problem. It would probably have more effect if variable taxi speeds were assumed as there would be less possibility for a slow taxiing aircraft to slow down a fast taxiing aircraft. The problem presented to the genetic algorithm was to find a path and a gate holding time for each aircraft. The fitness function penalised solutions which used the same route at the same time or which were infeasible (for example the wrong source or destination for the assigned path). The authors stated that the reason for allowing movement through these infeasible solutions was that the solution space of feasible solutions was not connected. Improve performance was obtained from the genetic algorithm when a modified Minimum Cost Maximum Flow algorithm was used to provide initial solutions for the problems, when the flow was not close to saturation.

Àngel Marìn considered the taxiing problem as a multi-commodity network flow problem with link capacities, side constraints and binary variables in [127]. A graph for the spacial problem was first built, where nodes represented positions for aircraft and arcs represented transitions aircraft could make between nodes. By discretising time, a space-time graph was built, in which each node of the space-time represented a node of the physical graph at a specific instant in time. The path of an aircraft over time was then given by a path through the nodes of the space-time graph and the flow through the graph represented the movement of aircraft. Capacity constraints ensured that only one aircraft could move at once and that nodes could only have one aircraft each. Flow conservation constraints ensured that the number of aircraft in a node at time t equalled the number at time $t-1$, plus the number entering, less the number leaving between time $t-1$ and t. A rolling window approach was used to decompose the problem. Within the window aircraft may appear to start part of the way along a path and some will not reach the destination before the period expires. For aircraft that did not reach the destination, a remaining taxi time was estimated using a shortest path algorithm. The objective function consisted of minimising a weighted sum of the taxi time taken by aircraft plus the estimated taxi time for those aircraft which did not reach their final destination nodes within the time period. Given this formulation of the problem, a *branch and bound* or *fix and relax* approach was suggested.

Like much of the ground movement research, this problem was formulated in terms of keeping the taxi times as low as possible. However, unlike many other approaches, the paths were assigned dynamically rather than selecting good paths to choose from in advance. It should be noted that there are advantages from pre-determining the paths to use, in terms of ensuring that unacceptable or less preferable path allocation is avoided, as well as in reducing the size of the search space that has to be considered. Furthermore, the approach makes no attempt to find optimal sequences for aircraft, or even to attain a desirable arrival order at the destination node. An aggregate lowest taxi time may not always be the best objective in practice.

3.8.6 Summary of ground movement research

Most ground movement models use a directed graph to represent the movement that is possible. In some cases, the nodes represent the valid positions and the arcs the transitions. In others, the arcs represent the taxi segments that are occupied and the nodes are merely decision points. In either case the node or arc representing the positions will have capacity constraints and the arcs will have lengths or weights reflecting the desirability or length of the associated path.

The level of detail of the ground movement models can vary widely, depending upon the purpose for which they have been designed. Many simulations are designed for off-line use, for example for evaluating capacity effects from potential layout changes. On-line decision support usually requires a less detailed simulation in order to return results within reasonable search times. More detailed models may not always give better results and the accuracy of the results can often be highly dependent upon the accuracy of the calibration performed. If the simulator works at too detailed a level then the problem will take an unnecessarily long time to simulate and calibration may be harder. Conversely, if the simulator works at too high a level of abstraction, important constraints upon the problem may be ignored.

Due to the complexity of the ground movement problem, it is common to consider minimising taxi time or attaining given target times at destination points as the objective of any optimisation. It is rare to see constraints such as the runway sequencing rules appear in the models. Research usually considers either the sequencing and scheduling problems or the control (ground movement) problem, but not both. Indeed, this is almost the case at Heathrow as the majority of the ground movement is performed by Ground Movement Controllers and the Runway Controller for the take-off runway is responsible for the sequencing and scheduling within a very small part of the taxiways. However, within these small holding areas, the control problem is a vital component of the problem the runway controller must solve and, as such, a decision support system for the runway controller must consider at least limited ground movement as a part of the sequencing problem. The relationships between the ground movement research and the decision support system described in this thesis are considered in section 6.7.

3.9 Collaborative Decision Making (CDM)

One of the reasons for performing the take-off sequencing at the holding area is the limited reliability of predictions of ready times for aircraft and the contention between aircraft at push-back, resulting in apparently unpredictable taxi times. Carr studied the stochasticity of the airline turn process (the various procedures involved in unloading and re-loading an aircraft and preparing it for departure) for airports and presented the results in [50]. His conclusions support the anecdotal evidence from Heathrow that there is a large degree of uncertainty associated with the push-back times. This is the reason that the simulation used in this thesis usually includes aircraft only at the point where a stand push-back has occurred. It was considered unrealistic for this research to assume that push-back times would be accurately known very long in advance of push-back. Indeed, it was reported in [77], that "Analysis of estimated time of departure (ETD) in June 2004 showed that only 50% of flights departed within +/-5 minutes of their ETD." The problem this causes for departures is obvious, since only a few minutes deviation in the time at which the aircraft is ready to push back can make a significant difference to the position of the aircraft in the take-off sequence.

The European Collaborative Decision Making (CDM) initiative[15], sponsored by IATA (the International Air Transport Association), ACI (the Airports Council International) and EEC (the Eurocontrol Experimental Centre, part of Eurocontrol) is an attempt to obtain more reliable information before push-back, by sharing information between all interested parties. The application of CDM to Heathrow was considered in 2005, [76, 77], and is currently being implemented.

The CDM initiative involves all partners in the turn-around process sharing information, including airlines, controllers ground handling services and other airport facility operators. The

[15]European Collaborative Decision Making initiative, web site: `http://www.euro-cdm.org/`

increase in information sharing should reduce the perceived uncertainty in the completion times of different tasks, for example in the times at which aircraft are ready for push-back. Push-back times may become available further in advance in future, with much reduced uncertainty. Even if take-off sequencing continued to be performed entirely by a runway controller, the benefits of having accurate taxi times and predicted push-back times from stands are considerable, as will be seen in section 8.4 where the total delay for aircraft decreases as the planning horizon is increased. When the taxiing aircraft are included in the input problem, the designed system will be observed in chapter 7 to (at least partially) sequence aircraft while they are still taxiing, even though the runway controller will not have to enact this sequence until the aircraft reach the holding area.

One problem in predicting accurate push-back times is the congestion in the cul-de-sacs, which can mean that aircraft cannot commence push-back until other aircraft have moved. This means that aircraft cannot be considered in isolation so the prediction of a push-back time is not simple. If reliable push-back time predictions could be made then it may be possible to consider the sequencing of take-offs earlier in the departure process, for example when aircraft were still at the stands. Sequencing at the stands has a number of potential benefits even though it would require accurate taxi time predictions, and introduce a coupling between the arrival and departure processes, since both need access to the taxiways, the stands, and the cul-de-sacs upon which the majority of stands at Heathrow are situated. Firstly, aircraft for which long delays were expected could sometimes be held at the stands, thus absorbing some of the delay prior to starting the engines, with consequent reductions in the amount of fuel burnt. Secondly, more accurate push-back times could potentially be used by airlines or ground handling agencies to better prioritise the allocation of limited resources such as tugs and baggage handlers. Cost benefits could be expected from such a system in the same way as they are expected from systems which allocate gates/stands to aircraft in order to reduce the use of such resources [70]. Thirdly, by ensuring that aircraft pushed back in a sequence closer to the expected take-off sequence, such a system could potentially ignore the constraints imposed upon the sequencing by the holding area structure, ensuring that aircraft arrive at the holding area in the correct order for take-off, or at least close enough to the take-off sequence to be easily re-sequenced within even a small, relatively inflexible, holding area. However, it is likely that at least some re-sequencing would still be needed at the holding area in order to account for the taxi time variability.

As described in section 1.3, the scope of the presented problem makes any sequencing by anyone other than the runway controller, and any changing of the working methods of controllers, beyond the scope of this research, but it is important to be aware of potential benefits from altering the way in which operations are currently performed, if possible.

3.10 Summary

The take-off problem at Heathrow has been considered before, but no research has previously considered all of the constraints upon the take-off sequencing. The runway is well known to represent a bottleneck for the departure system, so much so that the capacity of an airport is often estimated by considering the number of runway movements possible per hour.

The take-off problem has two components: a sequencing problem and a control problem. This is also the case for the arrivals problem. Much of the arrivals and departure research considers either the sequencing problem or the control problem but not both. Due to the position at which the take-off sequencing is performed at Heathrow, the control problem must be considered when solving the sequencing problem.

The problem faced by a decision support system is considered in more detail in the next chapter, which ends with an overview of the designed system. The designed decision support system is then considered in more detail in the following chapters.

Chapter 4

Designing A Decision Support System

4.1 Introduction

The problem faced by runway controllers was explained in chapter 2. Any decision support system designed to aid them must solve the same problem. The purpose of this chapter is to explain the reason for the selected solution approach. This will be achieved by explaining the characteristics of the problem in more detail and illustrating the ways in which its structure prevents the successful application of fast exact solution methods.

Despite the intention to only suggest a take-off sequence, as discussed in section 1.3.6, the decision support system has to consider how the take-off sequence can be attained. This means considering how the aircraft will move through the holding area. It was observed in section 2.8 that the paths that are assigned to aircraft are extremely important.

The chapter starts with the presentation of the triplet model for the holding area movement in section 4.2. This is used to illustrate the solution space for the problem that the decision support system has to solve. The alternative graph model for the holding area movement problem is then introduced in section 4.3. This has advantages over the triplet model in that it reduces redundant information from the search space of the problem but is dependent upon the paths that aircraft follow to have been pre-allocated.

The take-off problem is an example of a combinatorial problem, as explained in section 4.4, so the applicability of common exact solution methods for combinatorial problems is discussed in section 4.5.

The chapter ends with an overview of the selected solution approach in section 4.6. The sequencing aspect of the problem is then considered in more detail in chapter 5 and the control aspect of the problem is detailed in chapter 6.

4.2 The Triplet Model For The Holding Area Problem

A triplet notation for describing the holding area movement is introduced here. Each triplet relates to the movement of one aircraft along one of the arcs of the physical holding area graph.

The sequence of the triplets indicates the sequence in which the movement should occur.

Each triplet provides information about a single move by a single aircraft. The first value in the triplet is the source node for the move; the node the aircraft is moving from. This may be a '-' to represent the aircraft entering the holding area. The second value is an identifier for the aircraft. The third value is the destination node for the move. A value of 'R' refers to entering the runway, thus leaving the holding area. The source node is really redundant information, since a record can be maintained of the current location of each aircraft. However, including the source node allows a simple verification of the feasibility of the re-sequencing, as will be seen below.

Example strings of triplets are shown in figure 4.1 for the movement of three aircraft, 1, 2 and 3 through a combination of nodes A, B, C, D, E and R. The triplets that exist for each aircraft are determined by the path the aircraft takes through the holding area. Aircraft 1 and 2 enter the holding area at A, move to node C, then to node E and finally out to the runway. Aircraft 3 enters the holding area at B, moves to node C, then to node D and finally out to the runway.

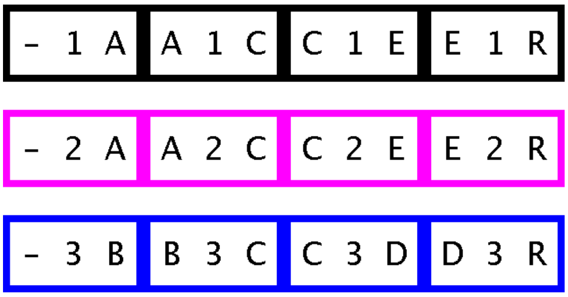


FIGURE 4.1: Example triplets

The triplet model is useful for investigating the sequencing options as it provides a visualisation of the exact sequence of movement in the holding area. The sequencing of the first triplets for each aircraft is determined by the sequence in which aircraft arrive at the holding area. The take-off sequence is determined by the order in which the aircraft enter the runway, which is the sequence in which the last triplets for each aircraft occur.

Some examples of holding area movement and consequent take-off sequences are shown in figure 4.2. Each row represents a different sequence of movement within the holding area. The first sequence in figure 4.2 (the top row) represents the situation where aircraft 1 arrives at the holding area, makes all of its movement then takes off. Aircraft 2 then arrives, makes all of its movement and takes off. Finally, aircraft 3 arrives, makes its movement and also takes off.

The next three sequences in figure 4.2 represent the situation where aircraft 1 does not exit the holding area until aircraft 2 has entered. In all cases the take-off sequence is the same, as the last triplets for each aircraft remain in the same relative order. It is obvious that these three different sequences are identical from the point of view of a take-off sequence. The last three rows in figure 4.2 show some other sample sequences, where the arrival order at the holding area changes.

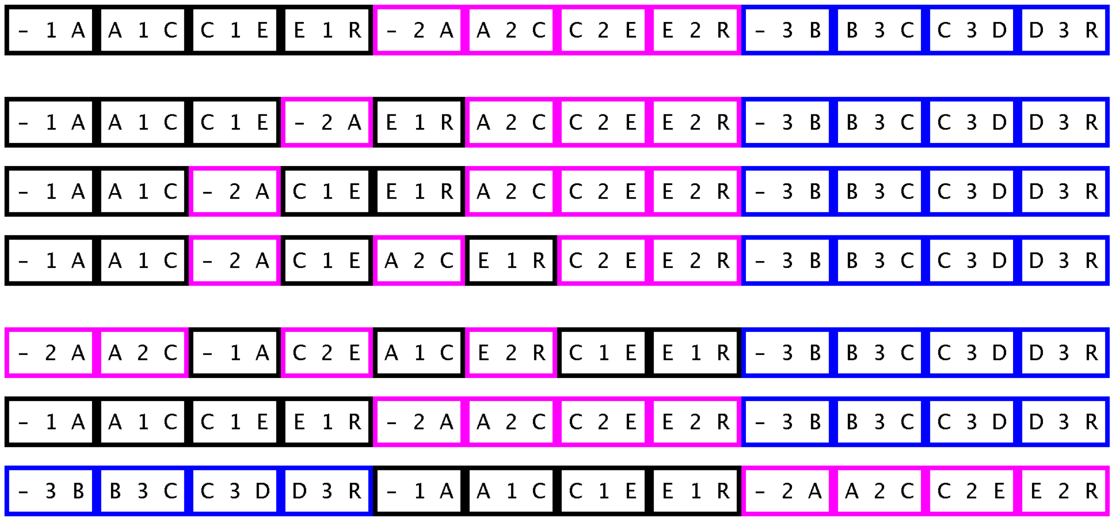


FIGURE 4.2: Triplet representation of some possible take-off sequences

A sequence of triplets is valid if, and only if, the triplets for each aircraft are in sequence (so the source node represents the node the aircraft is currently at) and the current destination node does not currently have an occupant. The validity of a sequence of triplets can be checked in linear time by processing triplets one at a time and tracking the current occupants of each node and the expected position of each aircraft.

The triplet representation is used for the output of the feasibility test performed by the developed system (described in chapter 6). This detailed sequencing can be used to illustrate exactly what is done when and is ideal for demonstrating the sequencing to controllers and for comparing the movement against that performed under the control of a real runway controller. The holding area movement viewer (described in section 7.6.2) can be used to replay these sequences visually.

The triplet representation has a weakness, however, as much re-sequencing of triplets can be performed without affecting the take-off sequence, or even affecting the sequence in which aircraft pass any node. The triplet representation enforces an ordering upon movement which can actually take place simultaneously so there are often multiple ways to achieve the same re-sequencing. The alternative graph, presented in the next section, is better for investigating re-sequencing as it eliminates this redundancy, but it is reliant upon pre-determining paths for aircraft so may be less useful for understanding the full control problem.

4.3 The Alternative Graph Model

The disjunctive graph is useful for visualising and solving job shop problems and is described in [42] and [144]. The alternative graph is an extension of the disjunctive graph, with applications to blocking job shop problems, and is detailed in [128]. The alternative graph is useful for understanding the control problem for take-offs and is explained below in the context of that problem. This is useful as the take-off problem can be formulated as a type of job shop problem.

4.3.1 The fixed-path holding area problem as a blocking job shop problem

The holding area movement problem involves determining the sequence in which aircraft should move through the holding area. The optimisation version of this problem involves finding the movement that will optimise the take-off sequence, in terms of total delay (equivalent to minimising the sum of the take-off times) or minimising the time of the last take off. The feasibility problem involves confirming the feasibility of achieving a target take-off sequence. This may involve identifying the sequence in which aircraft should move through the holding area in order to achieve the take-off sequence.

If paths are pre-determined for aircraft, then the holding area movement problem can be formulated as a blocking (no swap) job shop problem with precedence constraints and release dates. The nodes of the holding area graph are equivalent to the machines in the equivalent job shop problem and the aircraft which traverse the holding area are equivalent to the jobs processed by the machines. An operation then represents the movement of an aircraft through a node of the holding area graph. The sequencing of the nodes on the (pre-defined) path determines the sequencing of the operations in the job. Dual occupancy of nodes in the holding area is not permitted during the holding area movement. This is precisely the blocking characteristic of the job shop problem as it means a job/aircraft cannot enter a node/machine until the previous job/aircraft has vacated the node/machine. The processing time of each operation should be the time required for the aircraft to traverse the node along its specified path. A holding area arrival time for an aircraft can be represented by a release date on the associated job. Precedence relationships between the movement of aircraft can be modelled as precedence relationships between the operations of the job shop problem. Such a job shop problem can be represented by an alternative graph.

4.3.2 The alternative graph for the job shop problem

The job shop problem, introduced in section 3.4, can be formalised in terms of the operations which need to be performed and the sequence in which they should be performed. The alternative graph is a directed graph, $G = (V, C \cup D)$, where V is a set of vertices representing operations, C is a set of conjunctive precedence arcs and D is a set of disjunctive precedence arcs, described below. The vertices of the alternative graph represent the operations to be performed, the arcs represent the precedence relationships between operations and the arc lengths specify the time gap required between the start times of the two related operations.

Three types of arcs can be used to represent different precedence relationships:

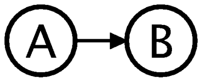


FIGURE 4.3: Conjunctive precedence arcs

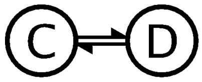


FIGURE 4.4: Disjunctive precedence arcs

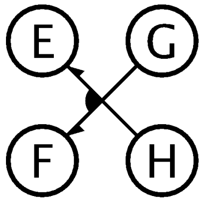


FIGURE 4.5: Generalised disjunctive precedence arcs

(1) Conjunctive arcs represent mandatory permanent precedence relationships between operations such that one will always occur before the other. A conjunctive arc is shown in figure 4.3 for two operations A and B, showing that A must occur before B. A label can be applied to the arc to specify the minimum time which must elapse between the start of the first and second operations. This is often the processing time of the first operation.

(2) A pair of disjunctive arcs can be used to indicate the situation when two operations can occur in either order, but not simultaneously. In figure 4.4, operation C can occur before or after operation D. Again, each arc can be labelled with the processing time of the operation at the source of the arc, indicating that the second operation cannot commence until this length of time has passed after the start of the first operation. Disjunctive arcs can be added between the operations that are processed by the same machine, to represent the fact that only one job can use the machine at once.

(3) The concept of disjunctive arcs in the disjunctive graph can be generalised to the situation where the vertices joined by the two disjunctive arcs are not the same, as shown by Mascis and Pacciarelli in [128]. Figure 4.5 shows an example of alternative arcs for the situation where event H must take place before event E, or event G must take place before event F. Again arcs can be labelled with the minimum time which must elapse between the start of the operations. The generalised disjunctive arcs or alternative arcs can be used to represent the blocking attribute of a job shop problem, indicating that a machine cannot start processing a second job until the next machine has started processing the first job (i.e. the first job has vacated the first machine). Further explanation can be found in [128], or with regard to the holding area problem below.

4.3.3 The alternative graph for the holding area problem

In order to model the take-off problem as a job shop problem, the runway and airspace need to be included in the model. The path assigned to the aircraft should therefore include all of the nodes traversed in the holding area, plus a node representing the runway, referred to as the runway node, and an airspace node relating to the fact that the aircraft has taken off. These are necessary in order to model the separation requirements at take-off, as will be seen below. The airspace effectively represents a single final machine with sequence dependent setup times, limiting the frequency of take-offs.

The processing time for the operations should be the expected occupancy time of the aircraft in the node represented by the operation. For the runway operations this should be the expected time the aircraft will take to line up and take-off, or the expected runway occupancy time, excluding delays waiting for other aircraft. For the other nodes, it should relate to the expected time that the aircraft will take to traverse the physical holding area location relating to the node. (Some problems with obtaining this value are discussed in section 4.3.6.)

An alternative graph can be created for the holding area problem using a similar method to that used for creating such a graph for the normal blocking job shop problem, [128]. The method for creating an alternative graph for this problem is described by the following steps:

- Add a node to the alternative graph for every operation of the equivalent job shop problem. This means one operation for each (aircraft,holding area node) pair, including a node for the runway and one for the airspace.
- Add a single 'initial' node and a single 'final' node.
- Add uni-directional conjunctive arcs between each pair of operations for the same aircraft, enforcing the traversal sequence for the nodes in the holding area. The length of the arc should be the processing time of the operation at the source of the arc, described above.
- Add disjunctive arcs between every pair of operations which represent traversal of the same node on the physical holding area graph, including the runway nodes but not the airspace nodes. The length of the arcs should be the processing times of the nodes at the source of each arc, described above. (Please note from section 4.3.5 that these arcs are strictly unnecessary but aid in understanding the movement sequence.)
- Add disjunctive arcs between every pair of operations which represent the airspace nodes. The length of the arcs should be the required take-off separation between the two aircraft joined by the arc, when the aircraft at the source of the arc takes off first.
- For each pair of operations *(a,n)* and *(b,n)* where aircraft a and b traverse the same node n, identify the subsequent operations, *(a,p)* and *(b,q)*, for the aircraft a and b respectively. (So aircraft a enters node p after node n and aircraft b enters node q after node n.) Add a pair of generalised disjunctive arcs from operations *(b,q)* to *(a,n)* and from *(a,p)* to *(b,n)*.

  These arcs represent the fact that either, (a) aircraft a cannot enter node n until aircraft b has moved to node q, or (b) aircraft b cannot enter node n until aircraft a has moved to node p. The sequence of entering node n will determine which applies.

These arcs should be given an extremely small but strictly positive weight, in accordance to Mascis and Pacciarelli's suggestion (in [128]) that the arcs to represent blocking operations should be given a small positive cost when swaps are not permitted, so that cycles of such arcs will have a positive length.

- Add an arc from the initial node to the node for the first operation for each aircraft. The length of this arc should equal the arrival time of the aircraft at the holding area.
- Add a zero length arc from the airspace operation for each aircraft to the final node.
- Identify the first operations for each job. Add a conjunctive arc between each pair of such operations to enforce the arrival sequence at the holding area. The arc should go from the operation for the earlier arriving aircraft to the operation for the later arriving aircraft. The length should equal the difference between the arrival times of the aircraft.
- Where aircraft have an earliest take-off time, for instance due to the presence of a CTOT, add an arc from the initial node to the airspace node for that aircraft, with a length equal to the earliest take-off time for the aircraft.

An example alternative graph for the take-off problem

An example alternative graph for a two-aircraft take-off problem is shown in figure 4.6. A CTOT (take-off time-slot) has been assumed to apply to aircraft *2* but not to aircraft *1*. Nodes in the graph are denoted by aircraft number and holding area node identifier (letter). The initial and final nodes are denoted by a '-' for the aircraft number and 'I' or 'F' for initial or final node respectively.

Precedence arcs exist between pairs of nodes. The arc lengths are labelled according to the following rules: $T(n, a)$ refers to the time aircraft a requires to traverse node n. $R(a)$ refers to the runway occupancy time for aircraft a. $S(a_1, a_2)$ refers to the minimum separation time required when aircraft a_1 takes off before aircraft a_2. $A(a)$ refers to the holding area arrival time of aircraft a. $C(a)$ refers to the earliest take-off time for aircraft a, for example due to the presence of a CTOT time-slot. In some cases a tiny non-zero separation time is required. These separations are denoted by the value d.

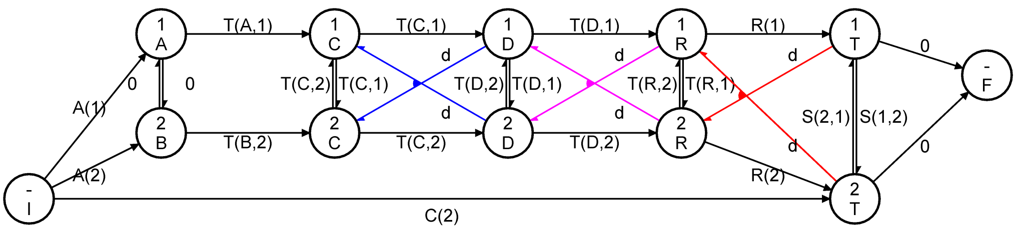


FIGURE 4.6: Example alternative graph for a two aircraft take-off problem

4.3.4 Using the alternative graph to solve the holding area problem

The production of the alternative graph is dependent upon the paths through the holding area having already been allocated to aircraft. A schedule is produced by making a selection of a single arc from each disjunctive or alternative pair of arcs. A schedule is feasible if, and only if, there are no positive length cycles in the graph. A zero-length cycle means that all operations on the cycle must commence simultaneously. A positive length cycle would mean that an operation has to commence later than itself, so obviously does not represent a feasible schedule.

The selected arcs define the order in which the associated operations are performed. The selected arcs between the airspace nodes for each aircraft will define the take-off sequence. Different take-off schedules can be created by changing the selection of the arcs from the disjunctive pairs such that positive length cycles are avoided.

The earliest start time of any operation is given by the length of the longest path from the initial node to the node for that operation. The earliest take-off time for an aircraft is given by the length of the path from the initial node to the airspace node for the aircraft. The makespan of the schedule is given by the completion time of the last job to complete, so is the length of the longest path from the initial node to the final node in the graph.

The makespan minimisation problem can be solved by finding the set of arcs such that there are no positive length cycles in the graph and the length of the longest path from the initial to final node is minimised. A total delay minimisation problem can be solved by finding the set of arcs such that there are no positive length cycles in the graph and the sum of the longest paths from the initial to airspace nodes for each aircraft is minimised.

4.3.5 Unnecessary arcs

Arc lengths for the disjunctive arcs between the operations representing traversal of the nodes of the holding area graph represent the minimum time which must pass between one aircraft entering the node and the next entering the node. The presence of the generalised disjunctive arcs makes the presence of these arcs unnecessary in the graph as the selection of the arc will always be enforced by the selection of an arc from an associated pair of disjunctive arcs and vice versa. (i.e. aircraft must exit a node of the physical holding area graph in the same sequence in which they enter it.) Moreover, as all arc lengths are non-negative, the lengths of the disjunctive arcs will always be no longer than the path between the connected nodes via the associated disjunctive arc, so the arc length is never constraining upon the longest path between nodes.

For example, assume the arc (1,C)→(2,C) is selected in figure 4.6. The selection of the arc (1,D)→(2,C) will be enforced as (2,D)→(1,C) would form a cycle in the graph, (2,D)→(1,C)→(2,C)→(2,D). The converse is also true, so the selection of the (2,D)→(1,C) arc would enforce the selection of the (2,C)→(1,C) arc, for example.

The total length of the arcs via node (1,D) (length T(C,1)+d) will always be at least as long as the length of the (1,C)→(2,C) arc (length T(C,1)). In addition, if any hold is required for aircraft *1* in node (1,C), for example due to another restriction upon how early (1,D) can take place, then the path length via (1,D) will be even longer.

4.3.6 The traversal time of a node in the holding area

In a job shop problem with set-up times, the minimum time gap between two operations should be the processing time of the first operation plus any setup time for the second operation. These set-up times could be sequence dependent. As an example, the take-off separation rules were modelled as set-up times and applied between the airspace nodes in figure 4.6.

It is actually possible for an aircraft to be able to enter a node of the physical graph earlier from one direction than another. For example, considering the 27R holding area graph in figure 2.3, consider the situation where an aircraft moves from node V to node X. It is sensible to assume that an aircraft could start entering node V from node U earlier than from node N, as the aircraft departing node V will be blocking entry from node N for longer. The implicit assumption with the usual timings is that an aircraft will have started taxiing into the node early enough that it arrives at the same time as the vacating one arrives at its new node. In this example, an extra delay may be required for aircraft entering from N. This sort of additional sequence-dependent delay can again be implemented by modelling it as a sequence dependent setup time and adding it to the length of the disjunctive and generalised disjunctive arcs between the operations for the associated aircraft. However, this adds further complexity to the model.

The biggest single problem for this kind of detailed model of the movement timing within the holding area is the inaccuracy of predicting how long the aircraft will be expected to take to traverse the nodes of the holding area, or the time to line up and take off which determines the runway occupancy. High accuracy in the traversal time estimates at this low level should not be expected, but the take off times may be highly dependent upon these timings. These timings will depend upon the the path taken through the node, the type of aircraft and also upon the pilot. When the additional sequence-dependent delay is included in the model, even less accuracy may be expected.

4.3.7 A simplified graph for testing only feasibility

It is possible to use the alternative graph for the consideration of the feasibility problem rather than for a makespan or total take-off time minimisation problem. For example, to consider whether a given take-off sequence is feasible, enforce the selection of arcs between the airspace nodes and determine whether there is a possible selection of the other arcs such that there are no positive length cycles in the graph.

If all arc lengths are known to be positive, then the feasibility problem is reduced to finding a cycle-free graph rather than having to consider the cycle length. Re-sequencing is feasible if there is a selection of arcs such that there are no cycles in the graph.

Some elements of the graph are then superfluous. All arc lengths can be omitted from the graph. It is not necessary to include the airspace operations as they must occur in the same sequence as the aircraft line up for take-off in the indicated model, so the arc selection between the runway nodes will define the take-off sequence. The initial node and the final node can also be removed, as these are used only for timings. Obviously, the related arcs can also be removed.

An example graph for determining the feasibility of re-sequencing is shown in figure 4.7 for a problem with three aircraft, 1, 2 and 3. Aircraft 1 and 2 take the path ACE to the runway (R). Aircraft 3 takes the path BCD to the runway.

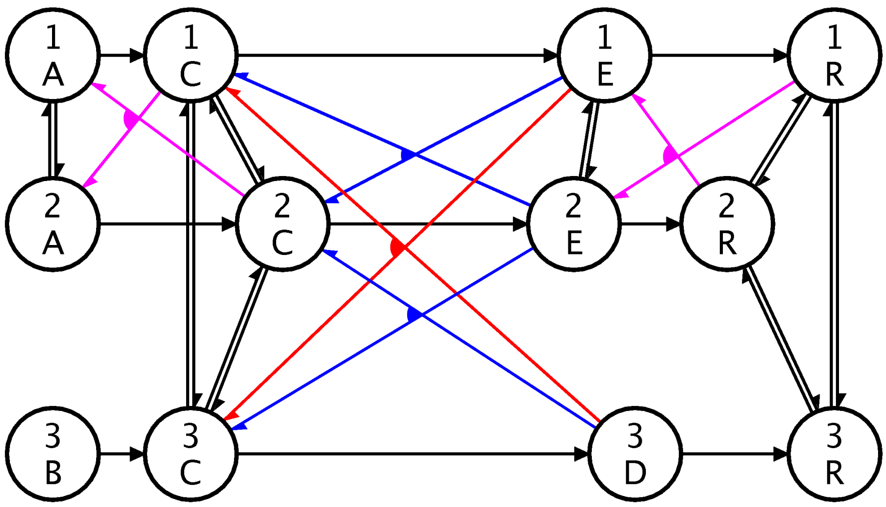


FIGURE 4.7: Example simplified alternative graph for a three aircraft holding area movement problem

A target take-off sequence can be enforced by appropriately selecting the disjunctive arcs between the nodes which represent the runway. The feasibility of a desired take-off sequence can be determined by verifying that there is a feasible selection of disjunctive arcs from the remaining disjunctive pairs such that there are no cycles in the graph. Given that the paths have to be allocated to aircraft before the alternative graph can be built, but the value of paths can only be determined from the target take-off sequence, testing the feasibility of the re-sequencing may actually be the most useful purpose for the alternative graph.

4.3.8 Solution difficulty

Klinkert and Gröflin called the alternative graph, presented in [128, 129], the generalised disjunctive graph, [121, 98]. They showed that the feasibility problem for the generalised disjunctive graph is NP-complete, using a reduction of the SAT-problem to the feasibility problem. Therefore, it is unlikely that a fast algorithm will be found for exactly solving the feasibility problem for the general alternative graph.

This result does not, of course, mean that the feasibility problem for a specific alternative graph is NP-hard, but it does indicate that any fast solution method is likely to be specific to the structures of the holding areas being used rather than a generic solution method for an arbitrary holding area structure.

Gröflin and Klinkert presented a tabu search algorithm for the problem in [99] and Pacciarelli presented a constructive heuristic algorithm for solving the alternative graph model in [139]. The constructive algorithm worked well when the blocking attribute was not present in the job shop problem, however the blocking attribute introduced deadlock problems which made it far more time consuming to solve. Pacciarelli concluded in [139] that further research was needed to develop efficient heuristic and exact solution algorithms for solving general blocking or no-wait job shop problems.

4.3.9 Reasons for not formulating the problem in this way

In summary, despite the fact that the take-off problem could theoretically be formulated and solved as an alternative graph, there are a number of reasons not to do so: Firstly, the formulation requires that paths have already been assigned to aircraft but the best path allocation to use depends upon the take-off sequence, which is an output of the solution of the problem. Secondly, the traversal times for nodes in the holding area are unlikely to be accurate. Thirdly, the problem is still difficult even when formulated as such a graph. And finally, the formulation only determines the earliest take-off times. The real objectives of the sequencing also include important elements related to CTOT take-off time-slot compliance and equity of delay.

4.4 Combinatorial Problems

The take-off problem is an example of a combinatorial problem, where a solution has to be selected from a finite (but arbitrarily large) set of possible solutions. The possible solutions for combinatorial problems can often be represented as a decision tree by formulating the problem as a sequence of decisions. Each level of the tree then represents a decision and the child nodes for any node represent the possible options for each decision.

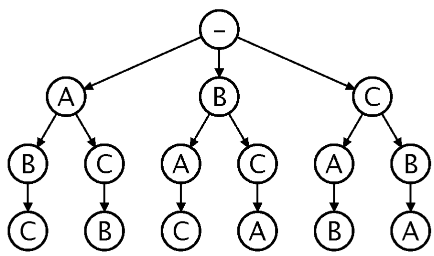


FIGURE 4.8: A small solution tree

For example, the aircraft sequencing problem can be represented as a solution tree by formulating the problem as a sequence of decisions about which aircraft should take off next. Figure 4.8 shows a solution tree for a three aircraft problem. Each path through the tree from the source to a leaf node represents a distinct take-off sequence. For example, the path down the left of the figure represents the sequence ABC and the path down the right represents the sequence CBA.

The first aircraft in a sequence of n aircraft can be any one of the n, the next can be any of the remaining $n-1$, the third any of the other $n-2$ that have not been chosen so far, and so on, giving $n!$ possible sequences in total. The number of solutions rises rapidly as n increases and can quickly make a complete enumeration of possibilities impossible.

This tree is useful for explaining the various ways in which different solution methods reduce the number of solutions which need to be considered. A simple depth first search algorithm, [21], could be used to exhaustively search this solution tree, however the time to do this can become prohibitive as the number of aircraft increases. With many problems there are ways to reduce the size of the search space which must be considered, for example by evaluating partial solutions to obtain a bound on the value of all solutions which contain them. In this way huge sections of the solution tree can be discarded and do not need to be evaluated.

4.4.1 Solution methods

There are a host of search methodologies available to the modern researcher. A review of many of these can be found in [47]. Each method has strengths and weaknesses, so the intention for any researcher is to find the method which best fits the problem.

Solution methods may be exact or heuristic. An exact method will guarantee that it will find the optimal solution but may take too long to execute to be of practical use. A well designed heuristic solution will often do well but will not guarantee to find the optimal solution. Indeed, a heuristic will usually give no guarantee at all about the quality of a solution. Obviously, an exact method is preferred when there is sufficient time to use such a method or the problem can be sufficiently simplified, but the take-off sequencing problem is not such a case. For example, Beasley et al. simplified the model of the arrivals problem by using a piece-wise linear cost function, [26, 27], allowing an exact solution method to be used, but explained in [28] that a linear objective function was not really appropriate and used a heuristic method for that reason. Exact methods could be used to solve very small versions of the take-off problem (containing very few aircraft), but the results in section 8.4 indicate that there are considerable advantages to be gained from considering more aircraft.

A second categorisation of solution methods may be made between constructive or iterative approaches. Constructive methods construct a complete solution piece by piece, so deal with partial solutions for much of the execution. A solution method for the control problem which adds one triplet at a time to the solution, attempting to determine the best triplet to add at each stage, is an example of a constructive solution method. Iterative solution approaches search by moving between complete potential solutions, seeking good ones. An example approach for the take-off problem may start at a known set of triplets and attempt to move or replace them such that a good feasible solution is eventually obtained.

The set of possible solutions for a problem defines the solution space. The solution space for some problems may be extremely large. Where a search method is used, the solutions that can be searched are collectively termed the search space.

4.4.2 Problem decomposition

A common approach to simplifying complex problems is to decompose them into easier sub-problems. The obvious candidate for decomposition in this problem is to separate the sequencing and control aspects of the problem. The sequencing problem could be further divided into the sequencing aspect and the take-off time prediction aspect. The path allocation and holding area movement problems could also be separated within the control problem. This results in four different sub-problems to be handed separately or simultaneously.

The inter-dependencies between the sub-problems cause difficulties for a problem decomposition. The preferred path allocation depends upon the take-off sequence, as discussed in section 2.8.3. The take-off sequence is determined by the sequence in which aircraft reach the runway, and hence by the solution to the holding area movement problem. A take-off sequence is useless if either it cannot be achieved, or the method to achieve it would be unacceptable to a controller. The holding area movement is dependent upon the paths assigned to aircraft. The take-off time prediction sub-problem is easy to solve once the take-off sequence and holding area movement have been determined, as detailed in section 5.5, but is dependent upon both of these other sub-problems. The value of a take-off sequence depends upon the take-off times that will be attained, so upon the solution to the take-off time prediction sub-problem. A starting position in this cycle of mutual dependencies has to be chosen.

It is possible to investigate the possible holding area movement and determine the consequent take-off sequence in each case. A decision then has to be made about whether to consider the entire control problem or to pre-define the paths to be taken. With pre-defined paths the alternative graph could be built and used to investigate the possible take-off sequences. The triplet model could instead be used if paths were not pre-defined, where the triplets for an aircraft would be determined by the path taken. The problem for this approach is the size of the search space. Some kind of search space reduction method would be needed. The problems with using common exact solution approaches for doing this kind of reduction are discussed in section 4.5.

An alternative approach is to start from the sequencing problem and attempt to find the best way to achieve the re-sequencing. Given a take-off sequence, a preferred path allocation can be determined. This ensures that the less preferred path allocations never need to be investigated, so reduces the complexity of the control problem considerably. However, the remaining control problem is still hard, as discussed in section 4.3.8, and has to be solved for each considered take-off sequence.

4.5 Exact Solution Methods

The application of three common exact solution methods for combinatorial problems is considered here.

4.5.1 A branch and bound implementation for the sequencing problem

Sometimes problems have a structure such that the value of a partial solution is a good indicator of the value of the full solution. When this is the case, an evaluation of the partial solution may allow it to be discarded, implicitly discarding all of the solutions for which it is a part. A branch and bound algorithm for this problem should at least be considered.

A full explanation of the branch and bound methodology can be found in [41], [71], [131] or [141]. Two requirements must be fulfilled to implement a branch and bound algorithm. Firstly, there must be a way to continually divide the feasible search space into two or more sub-regions. Secondly, there must be a way to obtain bounds for each of the sub-regions.

The algorithm starts by finding a (hopefully good) solution to the problem. Often a better initial solution will mean a faster search space reduction. The best known solution is maintained throughout the search, so when a better solution is found at any time it can be used to improve the search space reduction from then onwards.

The next stage is to divide the search space into two or more sub-regions and then determine bounds for the cost of each sub-region. If the bounds indicate that everything in the sub-region is worse than the best known solution then the entire sub-region can be rejected, as the optimal solution cannot be within it.

The application of a branch and bound algorithm to the sequencing problem is considered here. To illustrate the problems, the solution of the control problem will initially be ignored and the sequencing problem will still be seen to be difficult to solve exactly. The difficulties are due to a combination of the presence of CTOT take-off timeslots and the fact that equity of sequencing is very important.

The problem of partial solution evaluation

A natural way to divide the take-off sequencing problem is to divide the solution space based upon partial take-off sequences. In order to determine a cost for a sequence, it is necessary to predict take-off times for the aircraft in the sequence. The airport is usually busy enough that there is a queue for the runway, so the separation rules are usually a primary constraint upon how early an aircraft can take off. The take-off time for an aircraft is, therefore, very dependent upon the take-off times of the aircraft that take off before it. It is, consequently, only practical to predict a take-off time for an aircraft if the take-off times for earlier departures are already known. Therefore, useful evaluation of partial sequences is only possible when the partial sequence represents the start of the take-off sequence, and where the take-off time(s) of the first (few) aircraft in the sequence are already known.

In light of this, the sensible branching for a branch and bound algorithm is to branch on the aircraft that could be in each take-off position in turn. The first branch would be made based upon which aircraft is sequenced to take off first, the second on which aircraft is sequenced to take off next, and so on for each subsequent take-off position. In this way, the take-off times (and cost) can be predicted for each aircraft in the partial sequence at each stage, which will be a lower bound for the cost of any sequence starting with the partial sequence. The only thing that remains for the successful implementation of a branch and bound algorithm is to get a tight lower bound on the cost for the remaining part of the sequence. Adding the two costs will then form an improved lower bound for the cost of the sequences starting with this partial sequence, and will allow pruning if the lower bound is already higher than the cost of a known solution. In order to be successful the algorithm must be able to find a lower bound for the cost that is tight enough to allow pruning of the solution high enough in the tree to reduce the number of possibilities that need to be considered to a manageable level.

The difficulty of finding a tight lower bound for the cost

The cost of a sequence has a number of components. The three main components are related to CTOT compliance, total delay and equity of delay. A lower bound for the cost of CTOT compliance can be obtained by considering, in order of ascending CTOT time, only the aircraft with CTOTs. By applying a minimal separation time between them (for example assuming one minute separations) and predicting take-off times according to the earliest of the separation times and the start of the CTOT slot, a lower bound to the cost of CTOT compliance can be obtained. However, as CTOTs are applied to aircraft entering busy airspace, it is common for the aircraft to be on similar routes, so the lowest separations are unlikely to be obtained for all aircraft and the lower bound would not, consequently, be very tight.

An improved bound could be obtained by attempting to find an optimal sequence considering only the aircraft with CTOTs, thus determining where larger separations are actually required. This will give a tighter bound for this element of the cost, but may be very slow to perform. For example, at busy times a lot of the aircraft can have CTOTs, so the optimal sequencing is not always easy to achieve even when not considering aircraft without CTOTS. Additionally, such a sequencing ignores the fact that there may be advantages to be gained from slightly delaying certain aircraft in order to avoid excessive inequity of delay as it is not always desirable to prioritise aircraft with CTOTs.

Not only is it difficult to get a tight lower bound on the cost of the CTOT compliance for the remaining aircraft, but it is also difficult to evaluate the delay cost for the aircraft. Many simple methods turn out to be inaccurate. For example, take-off times could be predicted by sequencing the aircraft in the same order as they reach the holding area (so that the effect of any delayed arrival times upon the take-off times is minimised) and assuming one minute separations. A lower bound for the delay cost could then be calculated using these times as the delay based on these predicted take-off times is guaranteed to be no higher than the real delay. The problem is that the aircraft can hardly ever be sequenced in such an optimal way so the bound is not often very tight. There are three reasons for this: The first reason is that the runway can suffer from a lack of some departure routes while having a surplus on other routes. This can lead to enforced larger separations when there are no suitable aircraft available to fill the gaps between aircraft on similar routes. The second reason relates to the fact that wake vortex separations enforce occasional larger separations, which cannot be totally eliminated without an arbitrarily long delay for some aircraft. The third reason relates to the fact that the take-off sequence used for this lower bound is unlikely to be the same as that enforced by the need to comply with CTOTs, so it is unreasonable to expect a tight bound to be obtained from considering the two objectives independently.

Finally, the estimation of the cost of the equity of delay is interesting as there will always be a lower bound formed by allowing the aircraft to take off in the same order as they arrive at the holding area. However, as will be seen in section 8.3, the first-come-first-served sequence is usually extremely bad in terms of delay and CTOT compliance so it is unreasonable to assume that this bound will be particularly tight.

The similarity in cost between dissimilar sequences

In conjunction with the problem of finding a tight lower bound for the sequence cost, many sequences with quite dissimilar partial take-off sequences at the start of the take-off sequence can have very similar costs. To observe that this is the case, consider any two aircraft with very similar departure routes and the same weight class. Swapping the position of these aircraft in the take-off sequence will make little difference to anything other than equity of delay, as long as both can reach the holding area in time to take off at the same time. Similarly, there is a relatively low (but still existent) cost associated with delaying a single aircraft for a relatively long time as long as it is not moved out of CTOT compliance (if it has a CTOT). The relatively low cost associated with the equity of delay can mean that the sequence has a very similar cost to the more equitable sequences. Now consider a partial take-off sequence and that fact that many of these aircraft could be delayed to form another similar costing sequence with a different starting partial sequence. It is obvious that many different sequences have similar costs.

These problems with obtaining tight bounds on the cost of the remaining aircraft and the similarity of costs between sequences with different starting partial sequences mean that pruning of any sub-regions containing them can only occur very late in the solution tree, and so such pruning is unlikely to prune much of the tree. This prevents the branch and bound being as effective a technique as might be expected.

Including the control problem

The control problem cannot actually be ignored within the branch and bound approach. It is often the case that the re-sequencing required to create the better sequences (as measured by the sequencing objectives) cannot be performed within the holding area. For example, an aircraft with a tight take-off time-slot may not be able to overtake enough aircraft in the holding area to make the time-slot. Until the feasibility is considered the (infeasible) sequence with the aircraft overtaking and achieving the time-slot will always appear to be superior to the (feasible) sequences where it fails to do so. Even more commonly, an aircraft may only be able to make a time-slot if other aircraft are moved out of the way, but this in turn may prevent certain re-sequencing from occurring with these aircraft, resulting in the feasible sequences having a higher delay than the infeasible ones. This effect can be observed in the results presented in chapter 8.5, where the delay is always improved when time-slots are ignored and usually improved (slightly) when the constraints imposed by the holding area are removed.

As was seen in section 4.3, it is not a simple task to verify the feasibility of re-sequencing, and, moreover, it is not possible to accurately predict the feasibility of re-sequencing of full sequences from a partial sequence in take-off order. (The designed system presented in this thesis uses a heuristic rather than exact method to check the feasibility for this reason.) The feasibility must be verified for the known solution against which the bounds of sub-regions are compared, but it is common for the (apparently) lower cost solutions to be infeasible, so it may be hard to obtain a good known feasible solution. Furthermore, the presence of infeasible low cost sequences in the sub-regions contributes to the lack of effectiveness of the branch and bound approach as the solution space cannot be pruned unless these solutions can first be shown to be infeasible and hence excluded from consideration.

Heuristic pruning techniques

The difficulties of being unable to sufficiently prune the search space could perhaps be alleviated using heuristic pruning methods. For example, an estimated rather than exact lower bound could be obtained for the cost of the remaining aircraft. An alternative approach is to end the algorithm early, as in the truncated branch and bound algorithm applied to the asymmetric travelling salesman problem in [175, 176]. Neither approach was adopted here as the heuristic tabu-search method performs so well.

4.5.2 A dynamic programming solution considering the holding area movement

Dynamic programming is an exact solution technique for solving problems which can be formulated in a specific way. More information about dynamic programming can be found in [131],

[141] or [41] but a summary is given here in order to explain the problems of applying it to the Heathrow take-off sequencing problem.

To solve a problem using dynamic programming it must be possible to decompose the solution process into a finite number of stages such that the process can be in a finite number of states at the end of each stage. Furthermore, the states must be designed so that the optimality of the decisions that are made depends only upon the state not the method by which the state was reached. The problem is solved by determining the optimal decisions to make for the possible states at the final stage, using that information to determine the optimal decisions to make at the penultimate stage, and so on back to the states of the first stage. This approach often means that the number of decisions can be greatly decreased, by removing from consideration solutions which are obtained in a sub-optimal manner.

The state must be designed so that a cost can be determined and must contain all information which would affect the cost or feasibility of the final sequence. The most straightforward approach is for each stage in the algorithm to consider one more/less aircraft than the next. The aim would be to use dynamic programming to find an optimal solution to the final problem (with all aircraft sequenced) by considering the optimal solutions for the sub-problems where one less aircraft is considered at each stage along with the effect of adding the new aircraft to the sequence.

A dynamic programming approach was successfully applied to the simplified holding area problem in [124]. Despite this, the full take-off problem is not so amenable to a dynamic programming solution. Successful application in [124] required consideration of the positions of aircraft. In that case it was practical to do so as aircraft only had two paths that could be used so very few holding positions had to be considered. This is not the case for a real holding area structure, where there are many more paths available and where there may be advantages to be gained from holding further back in order to let other aircraft past, increasing the number of possible positions at which aircraft may hold.

The algorithm has to consider the fact that aircraft can be assigned to one of a number of paths and that the validity of the path allocation can only be determined once the final sequence has been determined. It does matter how the positions of aircraft were achieved, therefore the state needs to include information about the allocated paths as well as the positions of aircraft. This is especially true when the paths aircraft can take diverge then re-converge as, after the convergence there is no way to know from the position of the aircraft how it got there. This means that a vast number of solutions may be maintained at each stage, only to be discarded later once the path allocation is known to be unacceptable.

In addition to having more paths, there are also many more points at which aircraft may need to hold in order to make the full re-sequencing possibilities achievable, for example immediately before a node which has multiple incident arcs. In order to obey the condition that the state of the sub-problem includes everything which may affect the value of the full problem, it is important to consider all combinations of the positions at which aircraft may be held at each state.

The separation rules do not obey the triangle inequality. Separations of up to ten minutes or more may be required between aircraft along some departure routes at times of heavy congestion, although separations of more than four or six minutes are very unusual. No matter how unusual, it is not adequate to design a solution method which will not cope with this sort of situation. The take-off time of an aircraft can, therefore, be affected by the take-off time of any aircraft which has taken off in the previous ten minutes (or occasionally more) so the take-off times for all such aircraft must be included in the sub-problem state.

Any state must, therefore, include at least the paths allocated to aircraft, the current positions of the aircraft in the holding area and the sequencing of recent previous take-offs. The size of the state-space ensures that the dynamic programming approach is not appropriate for the take-off problem and it was consequently rejected.

4.5.3 Linear, Integer and Mixed Integer Programming

Many problems can be formulated as linear programs. These problems consist of a requirement to minimise (or maximise) a linear combination of continuous variables subject to a number of constraints which can be formalised as inequalities between constant values and linear combinations of the variables.

Although an explanation of the solution methods used for solving linear programming problems is beyond the scope of this thesis, it is worth noting that modern solvers can solve these problems extremely quickly. The interested reader is referred to [41], [57] or [141] for an introduction to the theory of linear programming and the simplex algorithm in particular.

If the variables in the solution for the linear program must take integer values, the problem is an integer programming problem. If both integer and linear variables are involved, this is a Mixed Integer Linear Program (MILP). Integer programs are often much harder to solve than linear programs as the solution space is no longer continuous. They can be solved by solving a linear relaxation of the problem then using a branch and bound type approach to enforce integrality upon the non-linear variables.

As has been observed in section 3.6, the arrival sequencing problem can be formulated as a MILP. The arrivals problem is usually simplified by assuming an objective to minimise the linear deviation of landing times from the optimal values. Time windows are usually applied to aircraft to prevent extreme inequity in the landing sequence. Such linear deviation costs are not practical for the departure sequencing problem as they have no factor to control the inequity of the sequence. (Consider the fact that summing the total delay means there is no difference between two six-minute delays compared with one two-minute and one ten-minute delay.)

The presence of the CTOTs and the variation in holding area delay mean that it is not practical to merely enforce a hard constraint of a tight take-off window. A convex piece-wise linear delay cost function could be used, but would need to have many pieces, slowing the solution of the problem. Alternatively, an additional factor could be added to account for the inequity of delay, but again this could be at best piece-wise linear. It should be further noted that the cost of compliance with CTOTs is (and must be) a further non-linear function, and again, although a piece-wise linear approximation would be perfectly adequate, such a function would have many pieces. Note that the function used in this thesis is not convex due to the discontinuities at the transitions between complying with a CTOT and requiring an extension and between requiring an extension and missing even an extension. These discontinuities are a reflection of the real rules and regulations, reflecting the fact that there genuinely is a greater problem with aircraft whose take-off times pass these boundaries.

Despite the problems with formulating the sequencing problem, the major problem for such an approach is not the sequencing problem but the presence of the constraints enforced upon the sequence by the holding area. The sequence in which aircraft pass each node in the holding area graph, and the paths that are taken, could be modelled within the MILP using this approach but the resulting program would be both extremely large and have an objective value that does not closely relate to the values assigned to the variables. The cost is not closely related to the sequence in which aircraft pass a node as many sequences have similar costs, as was discussed above for the dynamic programming and branch and bound approaches. Furthermore, some kind of evaluation would need to be performed about the appropriateness of the allocated paths and a cost would need to be applied in order to ensure a preference towards the use of simpler paths. This is not a situation conducive to swift solution by a MILP solver.

4.6 Selected Approach

Heuristic methods are used when the search space is too large to allow the problem to be solved exactly within the available time, as is the case here. The necessity for a decision support system to return decisions to the controller very quickly, combined with the complexities of the problem structure means that the problem cannot be solved exactly within the permitted search time for all but the most trivial of cases. This is not unusual for complex real-world problems. A heuristic solution approach is therefore proposed.

One reason for the problem difficulty is that the holding areas are usually flexible enough that re-sequencing that is feasible can be achieved in multiple different ways. The decision support system must ensure that it can be achieved in a way which is simple enough to be acceptable to the controller, rather than that it can merely be achieved.

The importance of the traversal paths allocated to aircraft cannot be over-emphasized, since it was seen in section 2.8 that they are related to the acceptability of the re-sequencing to a controller. Since the choice of path was observed in section 2.8 to be closely related to the target sequence, it is sensible to allocate paths only once the target sequence has been determined. The close relationship between the target sequence and value of the solution supports this approach. The proposed solution method attempts to solve the sequencing problem first and evaluate each solution for feasibility and ease of re-sequencing separately.

Once a target take-off sequence is known, the algorithm can ensure that optimal paths are allocated to aircraft, so that path allocation can be ignored within the actual feasibility test, thus vastly simplifying the search space for the problem. A heuristic rather than exact approach to the feasibility test ensures that a relatively generic solution method can be used (based around the timing of the movement within the holding area) and that the problem can be solved quickly.

4.6.1 Components of the selected solution approach

The selected solution approach has the following components. At the highest level, a meta-heuristic search is applied to the sequencing problem to determine a take-off sequence. Each potential solution is separately evaluated to determine the feasibility and cost of the re-sequencing. This evaluation involves a deterministic path-allocation method, guaranteeing that aircraft will not be given unnecessarily long paths to traverse and that the more difficult paths will never be allocated. Once paths have been allocated, a heuristic method is applied to verify the feasibility of re-sequencing and to determine how it will be achieved. Take-off times can then be predicted and a cost can be determined from the total delay and CTOT compliance of the predicted take-off times.

The decomposition includes some of the objectives directly in the objective function used by the search but simplifies the search by including others in the handling of the control problem. To be precise, the acceptability objectives are included in the path allocation and feasibility check rather than in the objective function, although an additional penalty could be included in the objective function for bad or unnecessarily long path allocations if required. The results presented in chapter 8 have demonstrated the efficacy of the approach.

4.6.2 Overview of the solution algorithm

An overview of the decision making algorithm used by the decision support system is given by algorithm 1. The algorithm consists primarily of a meta-heuristic tabu search algorithm (detailed in section 5.3) forming the outer loop (steps *1* to *4* and *15* to *25*) with a multiple-stage evaluation method for determining the cost of potential take-off solutions (steps *5* to *14*) as described in section 5.3.2.

Some of the steps in the algorithm involve complex and time consuming operations. The individual elements of the solution algorithm are explained in detail in chapters 5 and 6. In the experiments described in this thesis, the stopping criterion in step *2* involved repeating the loop from steps *2* to *24* one hundred times before stopping. The follow-on searches described in steps *26* to *28* were performed and involved testing a series of swaps and checking for local improvements to the take-off sequence, as described in section 5.7.3.

Algorithm 1 Overview of solution algorithm

```
1: generate initial feasible take-off sequence, s, as described in section 5.3.3
2: while tabu search stopping criteria are not met do
3:   generate fifty candidate neighbourhood solutions from the current solution, as described
     in section 5.4
4:   for each candidate solution do
5:     divide aircraft according to the holding area entrance at which they arrived, as described
       in section 6.4.3
6:     for each holding area entrance do
7:       allocate holding area traversal paths to the aircraft which arrived at that entrance,
         according to the overtaking which is required, as described in section 6.4.2
8:     end for each
9:     perform a feasibility test to determine whether the re-sequencing is achievable, as de-
       scribed in section 6.6
10:    if the re-sequencing is possible (i.e. the take-off sequence is feasible) then
11:      for each aircraft, a, in take-off order do
12:        predict the take-off time for aircraft a, as described in section 5.5
13:      end for each
14:      determine the cost of the take-off sequence, given the predicted take-off times, as
         described in section 5.6
15:      if the cost of the solution is better than that of the best solution found so far then
         record it as the best solution found so far
16:    else
17:      reject the take-off sequence
18:    end if
19:    if the take-off sequence matches an entry on the tabu list (see section 5.3.4) then
20:      reject the take-off sequence
21:    end if
22:  end for each
23:  identify the lowest cost candidate take-off sequence that was not rejected (if there is one)
     and adopt it as the new current take-off sequence
24: end while
25: identify the lowest cost take-off sequence that was found during the tabu search
26: if follow-on searches are being used then
27:   apply a heuristic improvement algorithm and/or a limited size exhaustive search within
      a rolling window to attempt to improve the take-off sequence. (Each take-off sequence is
      evaluated using the method described in steps 4 to 18)
28: end if
29: report the best take-off sequence found
30: end algorithm
```

4.6.3 Development history

An initial model was created for the problem to evaluate the feasibility of using a meta-heuristic solution method in combination with the heuristic holding area movement check. The first version of this problem decomposition was presented in [15] where a rolling window was moved across the dataset with repeated improvement of the aircraft sequence in the window at the time. The results from this research showed that it was feasible to solve the decomposed problem even with a very tight time limit and justified the decomposition. Although some of the constraints upon the take-off schedule were not included, most notably the take-off time-slots, the results showed promise that a decision support system could enable the controllers to improve upon the current delay.

The simulation that was developed was the forerunner of that which is described in chapter 7 of this thesis and which was presented in [14]. Due to the importance of taking taxiing aircraft into consideration, the effect of uncertainty upon the system was considered, [10]. The system was refined and used for experiments to evaluate the effects of holding area entrance allocation [11], planning horizon and freezing time [12] and the relative effects of the constraints upon the departure system, [13]. The results obtained against the simulation further support the decision to decompose and heuristically solve the problem.

Although not presented in [14], the simulated annealing, first descent and steeper descent algorithms were also tested on that problem. A limited size exhaustive search was also implemented, to exhaustively re-sequence the first seven aircraft at any stage. The results of these experiments are presented in appendix C for completeness.

The experimentation for [15] and [14] showed that the tabu search approach could solve the take-off problem quickly enough to be practical for a real-time online decision support system. The tabu search algorithm was more reliable than the simulated annealing algorithm and required much less tuning. Indeed, its performance was robust over quite a range of different tabu list sizes. In both cases, the first descent algorithm performed the worst, meaning that there must be local optima in the search space even with the neighbourhood designed to reduce the number of local optima. A meta-heuristic search was therefore useful for escaping the local optima.

4.7 Summary

The problem addressed by the decision support system was examined in this chapter. The characteristics of the problem were considered and the problems for an exact solution method were observed. An overview of the chosen heuristic solution method was presented, motivated by the structure of the problem. This method is detailed further in chapters 5 and 6.

Chapter 5

The Sequencing And Scheduling Problem

5.1 Introduction

An overview of the decision support system described in this thesis was given in section 4.6. The problem was seen to have both sequencing and control elements. The solution method for the sequencing sub-problem is considered in this chapter and the path allocation method and feasibility test are then explained in chapter 6.

This chapter starts with an overview of the theory of local search and an introduction to meta-heuristic search. The chosen tabu search methodology is then explained in detail, followed by a description of the multi-stage method that is used to determine the value of solutions.

The neighbourhood design is explained in section 5.4 and its importance for ensuring a fast and effective search is discussed. Once the feasibility of re-sequencing has been verified, it is necessary to predict take-off times for aircraft before a cost can be evaluated for the take-off sequence. The take-off time prediction method and evaluation function are both explained in this chapter.

The chapter ends with a consideration of the methods that have been applied to enhance the performance of the search, in terms of both speed and giving some guarantee of performance.

5.2 Local Search

Many search spaces for real problems are structured such that the value of a solution is strongly related to specific characteristics of the solutions. For example, in the travelling salesman problem (introduced in section 3.5), the good solutions to the problem often use a lot of shorter paths rather than a lot of long paths between nodes. These solutions with these shorter paths in common can be considered to be similar to each other. When there is not time to search the entire search space, it is sensible to put more emphasis on investigating the part of the search space which has more shorter paths in it than the parts which have more longer paths. Similarly, it may be sensible to investigate solutions which are similar to known good solutions.

Local search methods use the fact that good solutions are often similar to each other in order to concentrate the search on promising areas of the solution space. These methods start at a known solution and perform a number of iterations to move between solutions, seeking the better ones. Within each iteration of the search, one or more changes (moves) are made to a solution, generating one or more different but similar solutions. Each new solution is then evaluated to determine its worth and a decision is made about whether to move to any of the new solutions. The process then repeats and more new solutions are produced from the (possibly new) current solution. At some point, the process terminates and the best solution found by the search is usually returned as the result.

5.2.1 Definitions

A *'search space'* is "the space of all possible solutions that can be considered (visited) during the search", [82]. The set of solutions that can be reached via a single move (or 'local transformation') from a given solution is called the *'neighbourhood'* of the solution, [82]. Neighbourhood design (the choice of moves to allow) is a very important aspect of local search design and often has a major influence on the performance of a search, [132, 142, 154].

A *'local optimum'* (for a given neighbourhood) is a solution which is at least as good as every other solution in its neighbourhood (see definition 1.4 of [141]). A *'global optimum'* is a solution which is at least as good as every other solution in the entire search space, so that "all other points in the search space are worse than (or equal to) the current one", [46]. A local optimum will not necessarily be a global optimum, in which case any search which will only move to better solutions in the neighbourhood of the current solution may not be able to reach a global optimum. A good neighbourhood design would allow a search to move more easily towards a globally optimal solution, for instance by reducing the number of local optima.

There will be a number of constraints upon the solutions to a problem. Some of these may never be violated and are termed *'hard constraints'*, while some represent preferences that may be violated when necessary and are called *'soft constraints'*, [46]. The minimum separation times between aircraft, explained in section 2.3, are examples of hard constraints as they are mandatory. The CTOT take-off time-slot, described in section 2.4 is a soft constraint. Ideally the CTOTs would be treated as hard constraints, but the ability to use extensions when necessary means that these time-slot constraints can be (and need to be) broken at times.

An *'evaluation function'* or *'objective function'* is required in order to evaluate the quality of a solution with regard to the objectives of the optimisation. The evaluation function gives a "measure of the quality of the solution" and "is sometimes known as the objective, fitness or penalty function", [46]. The cost of a solution will usually include an evaluation of the degree to which the soft constraints are violated, for example the number of CTOT slots a candidate take-off sequence is predicted to miss. This function performs a mapping from the solution space to a numerical cost or value for each solution, [132]. The objective of the solution method is to find solutions which either maximise or minimise this value. A good evaluation function will be structured such that the value of a solution improves the closer the solution is to a global optimum, so that the search can be guided towards the optimal solution. If this ideal structure can be achieved then all local optima will also be global optima, so a local search will find a global optimum very quickly. This ideal is not usually practical, however, as will be seen for the problem considered in this thesis.

5.2.2 Simple local search algorithms

Probably the simplest local search algorithm is the *'simple hill climbing'/'first ascent'* (for maximisation) or *'first descent'* (for minimisation) algorithm. This search chooses a random neighbouring solution at each iteration and moves to that solution if doing so improves the value of the objective function. A new solution is then chosen from the neighbourhood of the current solution and the process repeats until some stopping criterion is met. An example first descent algorithm can be seen in section C.2 of the appendices.

The *'steepest descent'* or *'steepest ascent'* algorithm evaluates all solutions in the immediate neighbourhood and adopts the best as the next move. The *'steeper descent'* or *'steeper ascent'* algorithm samples multiple (but not necessarily all) solutions from the neighbourhood and moves to the best of the evaluated solutions. This is especially useful when the neighbourhood is large, perhaps too large to fully investigate. An example steeper descent algorithm can be seen in section C.3 of the appendices.

The major problem with each of the aforementioned algorithms is their inability to escape from even shallow local optima in the search space. When there are local optima in the search space, a search that includes the ability to escape from local optima will usually perform better than one which does not. Meta-heuristics were invented for this purpose.

If a steeper/steepest descent/ascent search accepts the best of the neighbouring solutions, even when it is worse than the current solution, then it has the ability to escape from the most shallow local optima. This is especially applicable to the steeper descent/ascent search, as these are less likely to move straight back to the local optimum on the next move since the previous solution may not be in the freshly sampled part of the neighbourhood.

5.2.3 Meta-heuristics

The term meta-heuristic was introduced by Glover in [90] as a name for a higher level algorithm to guide a heuristic search. Glover and Laguna defined a meta-heuristic in [88]:

> A meta-heuristic refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality. The heuristics guided by such a meta-strategy may be high level procedures or may embody nothing more than a description of available moves for transforming one solution into another, together with an associated evaluation rule.

Alternative definitions of meta-heuristics can be found in [35]. Burke and Kendall included the above definition, with elucidation in [46]. A meta-heuristic includes some heuristic basis for the method, such as a hill climbing local search, and some higher level guiding strategy, designed to improve the performance of the search.

An overview of meta-heuristics can be found in [87], along with details of many meta-heuristics. A bibliography of the various meta-heuristic approaches was produced by Osman and Laporte in [138], and more recent overviews of the use of meta-heuristics in combinatorial optimisation can be found in [35] and [83]. The algorithm for a tabu search meta-heuristic for the take-off problem is presented in section 5.3 and a simulated annealing algorithm for the same problem is provided in appendix C.

The methods for escaping local optima can be summarised as: (1) changing the neighbourhood of a solution (hopefully bringing better solutions into the neighbourhood, or removing some of the neighbouring solutions in order to guide the search), (2) changing the acceptance criteria for a solution (for example changing the evaluation function or (occasionally) accepting moves to worse solutions in order to escape a local optimum) or (3) restarting the search from another point. In each case, either the solution ceases to be a local optimum or the search will allow a move to a worse solution in order to escape a local optimum.

5.3 Tabu Search

In [90], Glover first introduced the term 'Tabu Search' when proposing an algorithm which remembers the path it has previously taken, to avoid retracing its steps. Glover stated that the method was originally designed for strategically changing the selection of the variables which were replaced by surrogate constraints in [89]. This idea of temporarily restricting further change to something that has recently changed is the key to tabu search.

Glover presented an extensive summary of the more basic versions of the Tabu Search algorithm in [91] and extensions to this in [92]. Glover and Laguna expanded on the tabu search algorithm in [88], extending the term to encompass algorithms which have (and use) memory in order to improve the success of the search. A recent explanation of the standard tabu search methodology can be found in [84].

The tabu search algorithm has been successfully applied to job shop scheduling problems in the past as well as to a wide variety of other problems, for example [30, 44, 67, 119, 133, 134, 156]. Tabu search is also an effective hyper-heuristic, where it is used to guide the selection of heuristics rather than of moves, [48, 152].

Despite the variety of uses for memory presented in [88, 91, 92], the most common variant of tabu search that is used is the variant Glover and Laguna refer to as short term memory tabu search, [88], originally proposed in [90]. Longer term memory may be of use when there is more search time, [170], however the short search time for the Heathrow take-off problem means it is sensible to restrict the tabu search to short term memory.

Many of the most successful meta-heuristics include an element to intensify the search, to concentrate upon a productive area of the search space, and an element to increase diversity, enabling the search to investigate new areas of the search space. In some ways, these elements can be considered to be in competition, in that the intensification element pushes the search to more thoroughly investigate the current part of the search space, while the diversification element seeks to leave to explore new regions. The time available for take-off sequencing makes the use of a strong diversification method problematic, however the clustering of the good solutions relatively close to the first-come-first-served sequence has made this unnecessary.

5.3.1 The Tabu Search Algorithm

The implemented tabu search algorithm is similar to a steeper descent algorithm in that it samples candidate solutions from the neighbourhood and adopts the best that is investigated. It also maintains details of the recent moves that have been made and prevents the search returning to previously explored solutions by eliminating from consideration those moves which reverse the effects of recent moves. This gives it a good ability to escape from local optima.

The tabu search is responsible for investigating the possible take-off sequences. A potential solution for the tabu search is merely a take-off sequence. Each sequence is separately evaluated for feasibility before a cost is determined for it. The implemented tabu search algorithm for the take-off sequencing problem moves through the feasible solutions in the search space by performing the following steps:

1. Obtain an initial solution. Generation of the initial solution is considered in section 5.3.3.
2. Evaluate the initial solution and determine a cost for it. A multi-stage method is used for this and is described in section 5.3.2.
3. Generate fifty new candidate solutions by selecting random solutions from the neighbourhood of the current solution. The moves used by the search determine the neighbourhood and are described in section 5.4.
4. Evaluate each new solution using the multi-stage process described in section 5.3.2.
5. Check each candidate solution against the current tabu list, described in section 5.3.4. If the solution matches a tabu entry, then reject it as a potential new solution.
6. If there is at least one feasible candidate solution that is not tabu, then select the feasible non-tabu candidate solution of lowest cost and adopt it as the new current solution.

   When the solution is adopted, add to the tabu list the details of the move made to obtain the selected solution.
7. If the given number of iterations have been completed, then stop the algorithm and report the best result so far, otherwise return to step 3.

In the experiments described in chapter 8, the algorithm was executed for one hundred iterations.

Ideally, the search would check all neighbours for the current solution at each iteration before accepting a move to a worse solution but a sampling system had to be adopted instead since the selected neighbourhood design means that there are far too many neighbouring solutions to exhaustively evaluate them all. The selected candidate list size of fifty aircraft is a compromise between the number of iterations that can be performed and keeping the candidate list large enough to be able to find better solutions in the neighbourhood.

5.3.2 Solution evaluation

The solution evaluation system has a number of stages:

1. Assign holding area traversal paths to aircraft for which the path has not already been fixed. Where paths have been fixed the previous path is kept. The path allocation process is discussed in sections 6.4 and 6.5.

2. Check the feasibility of the re-sequencing given the current holding area structure and allocated paths. This requires a model of the holding area and predictions for the current positions of aircraft, as explained in sections 6.6 to 6.9.

3. Predict take-off times for the aircraft. This is explained in section 5.5.

4. Determine the cost of the solution. The objective function used for this is explained in section 5.6.

   If the cost of the solution is better than the best solution found so far, then record it as the new best solution.

5.3.3 The initial solution

An initial feasible solution is required by the tabu search algorithm. If no other good sequence is known, then the first-come-first-served sequence is an ideal candidate for the initial solution as it is guaranteed to be feasible since it requires no overtaking. The decision support system will actually be running in a dynamic situation, being presented with a series of problems, each of which is an updated version of the previous situation as it changes over time. In the dynamic case a better initial solution may be created by using the solution from the previous problem as a baseline, as such a solution will have already been determined to be both feasible and of low cost.

An initial feasible solution can be created from the solution to the previous problem as follows: First remove aircraft that have already taken off and no longer need to be remembered for the purpose of meeting separation rules. Next add to the end of the take-off sequence any new aircraft which enter the system, in holding area arrival order. As long as these new aircraft arrive at the holding area after the aircraft which were already in the system, the new sequence will be guaranteed to be feasible since no overtaking is required that was not present in the previous (feasible) sequence. This approach to obtaining an initial feasible solution was adopted for the experiments presented in this thesis. Some additions had to be made, however, as described below.

In some cases the system may not become aware of aircraft in the order they are expected to arrive at the holding area. For example, when aircraft are only added only once they have pushed back from the stand, aircraft can push back from stands close to the runway and arrive before those which have been taxiing for a while. This means new aircraft may need to be added to an existing solution but should not necessarily be added to the end of the sequence. A simple way to guarantee recovery of a feasible sequence in this situation is to remove from the current sequence all aircraft that arrive after this new aircraft and add them back in at the end of the sequence, along with the new aircraft, in holding area arrival order. The removal of these aircraft could only reduce the overtaking required not increase it so this new sequence is again guaranteed to be feasible.

In the case of uncertain taxi times (as in live situations), the predicted arrival sequence should be expected to change over time, although this will usually be only minor perturbations in the sequence and can only happen for those aircraft still taxiing. A repair function can be applied to the sequence that was created to account for this. If the initial sequence is not feasible then, starting at the last holding area arrival and working earlier, aircraft are forced one at a time into the holding area arrival sequence until a feasible sequence has been attained. So, the first pass ensures that the last arrival is at the end of the sequence, and tests the feasibility, the second puts the last two arrivals at the end of the sequence and tests the feasibility and so on, until feasibility has to be regained eventually.

The aim of these recovery methods is to keep as much of the previous solution as possible (as it was found to be of low cost by the search in the previous step) while regaining feasibility to account for changes in taxi times. In particular, it ensures that the search (which is only heuristic) has the opportunity to consider this good sequencing for the aircraft at the start of the take-off sequence. These mechanisms for recovering a feasible initial sequence were used in the experiments reported in this thesis.

5.3.4 Structure of the tabu list

The structure of the tabu list is important. As Youssef et al. explain in [173], the tabu list structure and neighbourhood structure together embed the problem domain knowledge. Both therefore contribute to the success of the search. Salhi discussed the effect of the structure of the tabu list in [154].

The tabu list used for the decision support system described here stores the details of the last ten moves. Whenever a move is made, details of the aircraft that were moved are stored in the tabu list. The information stored for each move is a set of (aircraft,position) pairs, documenting the old take-off positions from which the aircraft were moved. The tabu list has a tenure of ten moves (not ten (aircraft,position) pairs as each move may have multiple pairs), so the eleventh move replaces the first move on the tabu list.

For example, assume that eight aircraft (labelled A to H) are scheduled to take off in the sequence $ABCDEFGH$, A is in the first take-off position and H is in the last position. Assume that a move is accepted which moves EFG to positions 3, 4 and 5 in the schedule, between B and C. The new take-off sequence is then $ABEFGCDH$. The following move is added to the tabu list: ((E,5),(F,6),(G,7)). Now assume that the next accepted move swaps the positions of G and E. The new take-off sequence is $ABGFECDH$. The following move is added to the tabu list: ((G,5),(E,3)).

A move is tabu if all of the (aircraft,position) pairs are matched in the state after the move. So, with the above example, if aircraft G is moved to position 5 and aircraft E is moved to position 3 the move will be declared tabu. As long as not all of the aircraft are moved back to the positions on the tabu list the move will be permitted. In other words, moves are only tabu if they reverse all of the effects of a previous move (ignoring aircraft which the previous move did not affect). To test for a move being tabu the algorithm should test against each stored move in turn. If a match is found for all (aircraft,position) pairs for any stored move then the move is declared to be tabu.

5.3.5 Alternative tabu list structures

A number of alternative tabu list structures were investigated. It is important that the tabu list prevents back-tracking and cycling without restricting the search too much. As the search cannot move through tabu solutions, it is possible for an over-restrictive tabu list structure to effectively split the search space into disjoint sets of solutions, so that a bad selection of initial moves could make it impossible to reach the good solutions. Alternatively, an under-restrictive one could easily fail to prevent cycling around local optima.

The first alternative was just declaring the previously visited sequence as tabu, so a search could adopt any move that did not return to a solution that has been evaluated over the last few moves. This is significantly less restrictive as very few solutions will be restricted and ineffective in this case due to the limited number of iterations for which the tabu search is executed. A second alternative tabu list structure is to mark as tabu any solution which matches any of the (aircraft,position) pairs. Given the example above where the tabu list stored the move ((E,5),(F,6),(G,7)), any move which places E in position 5, F in position 6 or (as opposed to 'and' in the selected algorithm) G in position 7 would be declared as tabu. This constraint is much more restrictive and a decision would have to be made about whether the tenure will consist of a number of moves or a number of pairs.

A number of different values for the tabu tenures were also investigated. The chosen structure and tabu list tenure were adopted as experimental results on static problems showed that the system performed at least as well with the chosen structure as with the others that were evaluated, although the tenure was found to have little effect as long as it did not deviate too far from the selected value. The validity of the selected tabu tenure for the application of the algorithm to the dynamic problem is considered in section 8.9.2.

5.3.6 Additional tabu search features

There may be a benefit to be gained from clearing the tabu list at times. Two obvious candidate times are when an improving move is made and when the best solution found so far is improved upon. The justification for this is that the search is more likely to need help to investigate new areas of the search space when it starts to fail to improve the solution value than when it is finding improvements.

In the first case, the tabu list would be cleared whenever a solution was found which improves upon the current solution. In the second case, the tabu list would be cleared whenever a solution was found which improves upon the best solution found so far. Experiments were performed with each of these algorithm changes. Both changes were found to make no improvement and were therefore not included.

A final feature that was experimented with was the aspiration criterion. An aspiration criterion is a condition which, if met, will cause the tabu restriction to be ignored. Common aspiration criteria relate to the cost of the solution, for example if the new solution is better than the current solution, no more than a given amount worse than the current solution or better than the best solution found so far.

The tabu list is useful for avoiding backtracking. However, as the tabu list does not store the entire solution, it is possible that good solutions which match an entry on the tabu list will be ignored. Indeed, as the search will be moving between good solutions and the tabu list stores some features of old solutions, it is quite possible that the solutions that match the tabu criteria represent some of the best in the solution space.

Different aspiration criteria were experimented with, relating the value of the new solution to the value of the current and best solution found so far. In experimental tests the various tested aspiration criteria did not improve the performance of the algorithm. In fact the algorithm that performed the best is the one selected and described here. The search is restricted from adopting any tabu solution no matter how good it is, however tabu solutions are still evaluated. If a solution is identified as the best solution found so far then it is recorded as such even if it is declared as tabu and so not adopted as the new current solution. The search therefore returns the best solution evaluated not the best solution adopted.

5.4 Neighbourhood Design

The neighbourhood design specifies the valid moves that can be applied to a current solution to generate new candidate solutions. A good neighbourhood design should allow the search to move easily between good solutions. This may not always be possible as the solution space may be very complex, but any local optima the neighbourhood design can eliminate are less that the search algorithm has to escape.

A decision has to be made about the size of the neighbourhood. There is a trade-off between making the neighbourhood large enough to reduce the number of local optima but small enough to take advantage of any solution space structure which clusters solutions. For example, a neighbourhood could be designed so as to include the entire search space, in which case, it is always possible to get to the global optimum in a single move. However, investigating the neighbourhood is equivalent to searching the entire solution space.

The first problem facing the designer of the local search neighbourhood is to identify the characteristics of a good solution to the problem. Ideally, all of the better solutions, including the optimal solution, would share some of the same characteristics. Solutions that did not have these characteristics could then be discarded. This is not always possible for real problems and where it is easy to characterise the good regions in this way, then the application of an exact solution method to take advantage of this characterisation should be considered.

In order to try to cluster good solutions for the take-off sequencing problem, it is important to consider the elements of the objective function, described in section 5.6. The sequencing problem for take-offs is ideally suited to this sort of technique due to the characteristics of the best solutions. The value of a take-off sequence is primarily determined by the CTOT time-slot compliance and the delay of aircraft.

If an aircraft has a tight take-off time-slot, then it is important to have a move which will allow it to be moved earlier in the take-off sequence, preferably without too much disruption for other aircraft. A shift move to shift the position of a single aircraft is an obvious candidate for an appropriate move.

Good take-off sequences will usually have an alternation of departure routes, as aircraft with similar routes have to be kept apart from each other. For example, a good take-off sequence may have an alternating north-bound and south-bound structure. Given this structure it is fairly obvious that a shift of a single aircraft is unlikely to form a good sequence from an existing good sequence. For example, if a north-bound aircraft is shifted forward then it will be taking off either directly before or directly after another north-bound aircraft, which will often introduce a larger separation into the schedule. Similar problems occur if a south-bound aircraft is moved. (The separation rules are actually far more complicated than this and there are other departure directions available but this example illustrates the problem.)

A move to shift multiple aircraft forwards or backwards was implemented due to the alternating structure of good solutions. Moving two aircraft in the alternating sequence has the possibility of maintaining an alternating sequence. A restriction upon shift moves was considered, whereby the shift could only move aircraft into positions where they were not adjacent to other aircraft on the same routes, but good sequences do sometimes have aircraft with the same departure route in adjacent positions in the sequence so this could not be implemented. For example, weight class separation rules or earliest take-off times may sometimes necessitate a larger separation, in which case there is no disadvantage from such sequencing.

An important aspect of the search is the very limited search time. The swap move was added to increase the flexibility of movement of aircraft through a sequence. This is particularly useful for swapping aircraft of similar characteristics, in order to move an aircraft into its take-off time-slot or to increase the equity of a take-off sequence while not affecting the rest of the sequence. As a high penalty is associated with missing the take-off time-slot, it is vital that moves allow other aircraft to move around in the take-off sequence without moving aircraft back out of the time-slot.

Results of experiments using the shift and swap moves showed that some good solutions were missed even though they were very similar to examined solutions. In particular, if a sequence of three or more aircraft had to be cycled then neither the swap nor shift would be able to do this in a single move. The randomise move was added to give the search a better chance to quickly escape this kind of local optimum.

All three move types were combined together to form a compound or composite neighbourhood. The use of a composite neighbourhood and only a simple short term memory is not unique to this problem, [134]. Meta-heuristic methods such as variable neighbourhood search would instead combine the neighbourhoods for the three types of moves in a different way, using one move until the search is stuck in a local optimum then changing to a different move. However, that sort of approach was impractical here due to the very short search time.

The moves that the search is permitted are listed below:

Swap single aircraft

There is a 30% chance that the *swap single aircraft* move will be used. This move takes two aircraft within five positions of each other and swaps the positions of the aircraft in the take-off sequence. The first aircraft is chosen randomly. The second aircraft is selected randomly from those within five positions of the first.

Shift aircraft

There is a 50% chance that the *shift multiple aircraft* move will be selected. This move selects a consecutive group of one to five aircraft and moves them to a new random position in the take-off sequence, either forwards or backwards.

Randomise a sequence of aircraft

There is a 20% chance that the *randomise a sequence of aircraft* move will be selected. This move selects a consecutive sequence of up to five aircraft as the target of the move. Each of these aircraft is then moved to a random position in the sequence. This move may, in some cases, emulate a shift or swap move but some of the sequences attainable through this move are not attainable by a single swap or shift move. In experimental results, this move has shown a valuable contribution in finding good schedules, when not overused.

Other moves, such as the reversal move (which reverses the take-off order of a sequence of aircraft) were experimented with but later rejected as they did not improve the performance when applied to test cases. The randomise move subsumes the reversal move anyway.

5.4.1 How the parameters were selected

The neighbourhood was tested and developed using the test system described in [15] and a steeper descent search, described in appendix C. This simple test system used a rolling window through the aircraft, sequencing twenty aircraft at a time rather than attempting to simulate the departure system. This simpler test system avoided the evaluation being affected by the complexity of the simulation. Similarly, the steeper descent search was used in preference to the tabu search precisely because of its reduced ability to escape local optima, since the aim of the neighbourhood design was to reduce the number and depth of these.

In addition to identifying the usefulness of moves, the system was used to determine the probabilities that each move should be selected. The success of the neighbourhood design was shown by the success of even the first descent algorithm in [15], however the need for the meta-heuristic approach was shown by the fact that the first descent search was outperformed by both the tabu search and simulated annealing approaches in [15]. The validity of the selected probabilities of adopting each type of move for the application of the algorithm to the dynamic problem is considered in section 8.9.1.

5.4.2 The search space structure

One important aspect of the search space structure, given these moves, is that the feasible solutions are connected. It is possible to move from one feasible solution to any other feasible solution by moving through only feasible solutions. To see that this is the case, consider that any feasible take-off sequence consists of a series of overtaking operations by aircraft. The shift and swap moves are sufficient to move from a first-come-first-served sequence to any target sequence by adding one or more overtaking operations at a time. Whether any two good solutions are connected by a path of good solutions must remain an open question as the answer will depend upon the holding area structure. The positive results from the system imply that, even if this is not the case, the tabu search algorithm is sufficient to cope, for the holding areas studied.

It is, therefore, possible to ignore infeasible solutions during the search. This is useful because the vast majority of possible solutions are usually infeasible at busy times of the day, as the overtaking opportunities start to decrease when the holding area gets more congested. Furthermore, the solutions of lowest cost are often infeasible at such times, due to the inability of some aircraft to overtake enough of the other aircraft to achieve a CTOT take-off time-slot. If infeasible sequences had to be considered, the algorithm may spend a lot of time striving for feasibility at the expense of increasing the cost.

5.5 Take-off Time Prediction

The decision support system must predict a take-off time for each aircraft in a take-off sequence. A cost cannot be determined until this has been done. In order to be able to predict take-off times for aircraft, an assumption is made that aircraft will take-off as early as possible, given all of the constraints upon how early the take-off can take place. The validity of this assumption will be shown when the results in chapter 8 are examined.

Aircraft cannot usually take off earlier than the calculated earliest take-off time (unless reduced separations are being used) and there is no advantage to be gained from taking off any later than the earliest take-off time. In contrast to this, in the arrivals problem aircraft can often fly faster or take a different path in order to land earlier than the ideal landing time, at the expense of increased fuel burn or workload.

The factors that constrain how early an aircraft can take-off are listed below:

1. Taxi time to the runway.

   An aircraft cannot take off before it can reach the runway. The earliest runway arrival time can be calculated by adding a holding area traversal time (which can be based upon the path taken through the holding area) to the arrival time of the aircraft at the holding area. In general this means that the traversal paths need to be determined prior to take-off time prediction.

2. Manoeuvring time.

   There may be times when an aircraft must wait for some other movement to take place before it can move to the runway. In this case it is important to allow for this requirement when predicting take-off times. The earliest take-off time is modified during the feasibility test, to allow for any delay waiting for other aircraft to pass and is a reason why the feasibility test has to be performed prior to predicting take-off times.

3. Wake vortex separation rules must be obeyed.

   An aircraft cannot take off before the wake vortices from the previous take-off have had time to dissipate.

4. Departure route and speed separation rules.

   In order to control the workload for downstream controllers and attain in-flight separation distances, a separation time is enforced at take-off between aircraft flying along similar routes.

5. Preparation time.

   Some of the pre-flight checks may be made after push-back from the stand, while an aircraft is taxiing to the runway. If taxi times are short it may be important not to schedule an aircraft to take off soon after it reaches the holding area. For example, if a large aircraft pushes back from a stand close to the runway it can easily reach the runway before it has had time to perform the pre-flight checks.

6. Take-off time-slot.

   Any aircraft with a take-off time-slot may not take off before the start time of the time-slot.

5.5.1 Taxi time limitations

The first and second take-off time restrictions in the list above are handled in this formulation of the problem by the feasibility test, described in chapter 6. In addition to determining the feasibility of the re-sequencing, the feasibility test also outputs an earliest take-off time, f_i, for each aircraft, i, which takes into consideration the time at which the aircraft arrives at the holding area, the amount of time the aircraft will need to traverse the holding area and any delay caused by having to wait for other aircraft to manoeuvre before the aircraft can take off. The value of f_i is given by equation 6.1, which is explained in section 6.9.3.

5.5.2 Separation rules

The separation rules for the third and fourth take-off time restrictions are regulated so a required separation can be calculated based upon the weight classes, departure routes and speed groups of each pair of aircraft, as described in section 2.3.

For any aircraft, i, let v_i be the weight class of the aircraft, r_i be the departure route allocated to the aircraft and s_i be the speed group of the aircraft. Let the position of aircraft i in the take-off sequence be given by c_i, so $c_i = 1$ for the first aircraft to take off.

For any pair of aircraft, i and j, where aircraft i takes off before aircraft j, values can be determined for the required separations between the aircraft, in accordance with current regulations. Let V_{ij} be defined as the required wake vortex separation between leading aircraft i and following aircraft j, determined from the weight categories v_i and v_j of the aircraft. Let R_{ij} be defined as the required separation based upon the SID routes, r_i and r_j, and the speed groups, s_i and s_j, of the aircraft i and j. A minimum required separation, M_{ij}, can then be calculated for each ordered pair of aircraft i then j, as shown in equation 5.1.

$$M_{ij} = \max(V_{ij}, R_{ij}) \tag{5.1}$$

Wake vortex separations obey the triangle inequality so that $V_{ij} + V_{jk} \geq V_{ik}$ for aircraft which take off in the order i,j,k, however route and speed separations do not. As the route separations do not obey the triangle inequality, it is not possible to ensure that all separations are maintained by merely ensuring sufficient separations between adjacent take-offs.

As long as take-off times are predicted for aircraft in take-off order then the predicted take-off time, d_j for all earlier aircraft, j will be known. The earliest time, e_i, at which an aircraft i can take off while obeying all separation rules can be determined from equation 5.2, where n is the number of aircraft in the take-off sequence. Although the take-off times for all earlier take-offs are considered in equation 5.2, in practice the separation rules are such that only the last two or three take-offs usually need to be considered. The number of previous departures that actually need to be considered is equal to the maximum inter-aircraft separation time (usually no more than ten minutes) divided by the minimum inter-aircraft separation time (usually one minute). The experiments performed for the results presented in this thesis assumed a consideration of the last ten aircraft. Although unnecessary, it was felt useful to ensure that the search would still be able to return results quickly when larger separations had been enforced.

$$e_i = \begin{cases} 0 & \text{if } c_i = 1 \\ \max\limits_{j \in \{1,\ldots,n\} | c_j < c_i} (d_j + M_{ji})) & \text{if } c_i \geq 2 \end{cases} \tag{5.2}$$

5.5.3 Ready time

The effects of the fifth take-off time constraint are approximated by ensuring that there is a minimum time between push-back and take-off for all aircraft. This minimum time is determined based upon the weight class of the aircraft, as larger aircraft need longer to prepare. The function P_i is defined for aircraft i to give the earliest time the aircraft will be ready for take-off. This is defined as the time at which the aircraft left its stand plus a minimum time depending upon the weight class of the aircraft.

In the simulation results presented in this thesis, P_i was treated as a constant for each aircraft, regardless of the desired take-off sequence. A live system would instead initially approximate this value but permit a controller to override this with a new time according to information from the pilot. For example, increasing the value if a pilot declares there are some difficulties and the aircraft will not be ready for take-off at a specified time.

The simulation presented in this thesis continues to use the estimated value throughout the experiments as no updated value will ever be available to the search. The function used in the experiments presented in this thesis is a simple one which returns a value of one minute, three minutes or five minutes after push-back for light, medium and heavy aircraft respectively. With a two-minute holding area traversal time, this is only usually constraining upon heavy aircraft which push back from stands close to the runway.

5.5.4 Take-off time-slot

Many aircraft have a Calculated Time Of Take-off (CTOT), described in section 2.4. If an aircraft, i, has a CTOT assigned, then let b_i be the start time of the consequent take-off window and let l_i be the end of the take-off time window. The fifteen minute CTOT time-slot is then denoted by the time interval $[b_i, l_i]$.

If aircraft i does not have a CTOT then b_i is set to a very low value (earlier than the start time of the dataset) and l_i is set to a large value, large enough that it is beyond the last take-off time of the last aircraft in the dataset. This effectively gives any aircraft that does not have a CTOT, a take-off window which spans the entire range of take-off times. In effect, no time window constraint is applied to these aircraft and a special case to deal with the absence of CTOTs is unnecessary.

5.5.5 Predicted take-off time

Given the above definitions of b_i, f_i, P_i and e_i for aircraft i, the equation for the predicted take-off time, d_i, can be determined for each aircraft, i, in take-off order, using equation 5.3.

$$d_i = \max(b_i, f_i, P_i, e_i) \tag{5.3}$$

5.6 Objective Function

The aim of this model is to enable the decision support system to find sequences with low delays and high CTOT take-off time-slot compliance while controlling the inequity of delay in the sequence and ensuring that the re-sequencing would be acceptable to a controller. Each of the objectives are considered below, followed by the description of the overall objective function.

5.6.1 Definitions

Let h_i denote the time at which aircraft i arrived, or will arrive at the holding area. Let d_i denote the predicted take-off time of aircraft i. Let l_i and b_i denote the limits of the take-off time-slot, as defined earlier. By expressing all times in seconds, in normal operation $l_i = b_i + 900$ for aircraft which have CTOTs. Where aircraft do not have CTOTs, l_i and b_i are set to very small and very large values respectively, so that d_i always lies between them.

Let a_i denote the position of aircraft i in the arrival sequence at the holding area and c_i denote the position of aircraft i in the intended take-off sequence. Where there is a previously known planned take-off sequence, let o_i denote the position of aircraft i in this previously take-off sequence. If there is no previous sequence containing aircraft i, then let o_i be set to the same value as a_i, so that the first-come-first-served take-off sequence is assumed.

5.6.2 Compliance with CTOT time-slots

Aircraft must adhere to their CTOT slots. It is always possible to delay aircraft to move them into the CTOT slot, but it is not always possible for aircraft to take-off early enough to meet the CTOT restrictions. As described in section 2.4, missing a CTOT slot by over five minutes is much worse than missing it by up to five minutes. All misses should be avoided but if a miss is inevitable then it is better to miss by less than five minutes.

The function $C(d_i, b_i, l_i, h_i)$, defined by equation 5.4, is responsible for determining a cost for any aircraft based upon the compliance of its predicted take-off time with its allocated CTOT. The value F_H is a constant.

$$C(d_i, b_i, l_i, h_i) = \begin{cases} 0 & \text{if } l_i \geq d_i \geq b_i & \text{(i)} \\ \omega_1(d_i - l_i) + \omega_2 & \text{if } (h_i + F_H) \geq d_i > l_i & \text{(ii)} \\ \omega_3((d_i - l_i)^{1.1}) + \omega_2 & \text{if } (l_i + 300) > d_i > max((h_i + F_H), l_i) & \text{(iii)} \\ \omega_4((d_i - l_i)^{1.1}) + \omega_5 & \text{if } d_i \geq (l_i + 300) & \text{(iv)} \end{cases} \tag{5.4}$$

Case (i) in equation 5.4 applies to any aircraft which takes off within the take-off time-slot for its allocated CTOT and ensures that no cost is associated with such an aircraft. If an aircraft has no CTOT then the values assigned to b_i and l_i will ensure that case (i) will apply to such an aircraft.

Case (ii) applies to aircraft which take off outside of the allocated CTOT time-slot, but within F_H seconds of arrival at the holding area. This case can only apply to aircraft which arrive at the holding area close to the end of the allocated CTOT slot, i.e. when $(h_i + F_H) > l_i$. When aircraft are delayed at the stands or in taxiing around the runway, their take-off slot may be so tight that meeting it becomes unrealistic. In these cases, severely penalising the number of seconds by which the time-slot was missed will result in a sequence where the aircraft is sequenced as early as possible, regardless of the effect that this has on the rest of the sequence. In order to avoid this, a period of flexibility is introduced whereby, for a given number of seconds, F_H, after arrival at the holding area, any CTOT miss by the aircraft will be less severely penalised. ω_1 should be set to a relatively low value and ω_2 should be relatively large compared with ω_1 to ensure that as few CTOTs as possible are missed.

Case (iii) applies to aircraft which take off later than the allocated CTOT time-slot, but within five minutes of the end of the slot, so that one of the permitted five minute CTOT extensions can be used. The cost has two elements, a fixed cost and a penalty which rises with the number of seconds by which the time-slot was missed. The relationship between the values of ω_3 and ω_2 determines the relative importance of the number of CTOT extensions used versus the total seconds by which CTOTs are missed. The non-linear form of the penalty helps to avoid the larger extensions, biasing towards equity of extensions where they are required. For example, the selected power gives a slight preference to a schedule with two two-minute extensions over a schedule with a one-minute and a three-minute extension. As the separation rules are in multiples of sixty seconds, moving the position of an aircraft in a sequence will usually alter the take-off times by a multiple of sixty seconds.

Case (iv) applies to aircraft which are scheduled to take-off even too late for a CTOT extension. If aircraft are more than five minutes late for their CTOT slot, then the CTOT slot will usually need to be rearranged. Term (iv) of equation 5.4 ensures that any sequence for which the departure time of an aircraft is more than 300 seconds after the end of the CTOT slot will be heavily penalised. Due to the importance of avoiding missing CTOT extensions, ω_4 and ω_5 should be given large values compared with ω_3 and ω_2 respectively.

The values of ω_1 to ω_5 determine the relative importance of the number of CTOT misses, the number of extensions missed and the number of seconds by which each was missed. Given the above comments about the appropriate values for these weights, the values $\omega_1 = 1$, $\omega_2 = 300000$, $\omega_3 = 2000$, $\omega_4 = 10000$, $\omega_5 = 10000000$ and the value $F_H = 240$ were used for the experiments presented in this thesis.

5.6.3 The holding area delay

The importance of keeping delay low has already been established and relates to fuel burn, pollution and passenger dissatisfaction as well as maintaining a high runway throughput. The time spent in the holding area is here used as a measure of the delay for each aircraft. Given the above definitions of d_i and h_i, the time spent by aircraft i in the holding area can be seen to equal $d_i - h_i$.

5.6.4 Schedule Equity

It is important to ensure a degree of equity in the take-off sequence. In general, it is better to avoid unnecessary overtaking or re-sequencing of aircraft. Aircraft can be considered to be positionally delayed when their position in the take-off sequence is later than their position in the arrival sequence at the holding area. When aircraft are positionally delayed in the take-off sequence, the affected pilots can feel dissatisfaction with the scheduling due to the perceived unfairness. Large positional delays can also restrict the re-sequencing opportunities for other aircraft, since a delayed aircraft must be held somewhere, thus blocking a part of the holding area structure and reducing the re-sequencing flexibility. Small positional delays are not a problem but larger delays become an increasing problem, both in terms of unfairness and constraining later re-sequencing. A negative positional delay (or positional advancement) is a problem for neither the fairness nor the flexibility of the holding area. The positional delay for aircraft i is given by $c_i - a_i$ if positive, or zero otherwise. A cost has been included in the objective function that is proportional to the sum of the squares of the positional delays for each aircraft.

In some cases it is useful to favour the first-come-first-served sequence even when aircraft are not positionally delayed. For example, if two aircraft with similar characteristics arrive at the holding area and overtake a number of other aircraft (thus neither has a positional delay), it is more equitable to allow them to take off in the same order in which they arrived. When neither aircraft has a positional delay then $(\max(0, c_i - a_i))^2$ will evaluate to zero regardless of the take-off sequence, so the third term will not ensure that the take-off sequencing favours the arrival sequence. An additional factor $(\texttt{abs}(c_i - a_i)(\texttt{abs}(c_i - a_i) + 1)/2)$ has been included in the objective function to ensure that the searches favour this preference even when no positional delay is involved. The fact that this term does indeed aid in keeping equity of delay can be observed in section 8.9.3. This sequencing should also help to improve the robustness of the take-off sequence as a more equitable sequencing should provide a more equitable holding area delay and this holding area delay acts as a buffer to allow for uncertainty in either traversal or arrival times.

5.6.5 Schedule Stability

There are often cases where there are multiple sequences with very similar costs. In this case, it is better to favour a previously used sequence rather than allow sequence changes that will have little benefit. This is especially important if there is some uncertainty in the data used to make the decisions as small perceived benefits may be purely down to data errors. To provide for this, a factor of $\texttt{abs}(c_i - o_i)$ is included in the objective function. (Due to the definition of o_i above, this factor will favour the first-come-first-served sequence if no previous take-off sequence is known.) The effects of this term are shown in section 7.8 where they are graphically illustrated using a comparison between figures 7.3 and 7.4.

The idea of penalising deviation from previous sequences is not a new one. As described in section 3.6.2, a penalty has been applied in previous arrival sequencing research into the dynamic problem, [27, 49], in order to restrict or penalise deviation from a previous arrival sequence. For the take-off sequencing problem, given that a controller would not be able to enact a take-off sequence until the aircraft reach the holding area (given the assumption that the runway controller is doing the sequencing), and that, providing sufficient information is available about taxiing aircraft, the sequencing is usually performed while aircraft are still on the taxiways (as discussed in section 7.8.3), it is less important to severely penalise deviations from previous schedules than it was in the arrival sequencing research. However, this is still an important element of the objective function, as discussed in section 7.8.

5.6.6 Avoiding penalising specific aircraft types

The function $Z(c_i, a_i, v_i, s_i)$ was added to avoid unnecessary penalisation of certain aircraft. As discussed in section 2.13, the objective to reduce delay will move problematic separations to a later place in the sequence. In some cases, delaying aircraft does not help and merely constrains the holding area re-sequencing by congesting the holding area unnecessarily. For example, there are so few light aircraft flying from Heathrow that it is highly unlikely that delaying one light aircraft will make it possible to group it together with another light aircraft to avoid the high wake vortex separation that will precede a light aircraft.

If an aircraft which will always have a large separation associated with it (such as a light or slow aircraft) is delayed to be sequenced near to the end of the take-off sequence, then this function will apply a penalty to penalise the sequence. Without this factor, there is a tendency for the system to sequence the aircraft later and later, to avoid the separation. The presence of this function prevents this.

In the experiments performed for this thesis, this function was implemented to apply a penalty to any sequence which positionally delays light or slow aircraft. The cost was parameterised but the parameters were selected so that any positional delay of more than one position in the sequence applied a cost equal to an increased total delay of two minutes in the sequence. Observation of the results showed that this small factor, in combination with the penalty for excessive positional delay that is applied to any positionally delayed aircraft, was sufficient to prevent the positional delay of such aircraft.

5.6.7 The objective function

The path allocation system and feasibility check ensure that only acceptable sequences are presented to a runway controller. The remaining objectives, described above, are considered in the objective function.

$$\begin{aligned}\sum_{i=1}^{n}(W_1C(d_i,b_i,l_i,h_i) + W_2(d_i-h_i) + W_3(\max(0,c_i-a_i))^2 \\ +W_4(\texttt{abs}(c_i-a_i)(\texttt{abs}(c_i-a_i)+1)/2) \\ +W_5(\texttt{abs}(c_i-o_i)) + W_6Z(c_i,a_i,v_i,s_i))\end{aligned} \tag{5.5}$$

The first term in the objective function refers to the compliance with take-off time-slots, as described in section 5.6.2. The second term relates to the holding area delay for aircraft. The third and fourth terms relate to schedule equity. The fifth term is included to aid schedule stability. The final term is related to avoiding penalising certain aircraft types.

The weights W_1 to W_6 are used to ensure that CTOT compliance is of primary concern, minimising delay is of secondary concern and maintaining equity and sequence stability are of only tertiary concern. In the experiments presented in this thesis, the weights $W_1 = 0.125$, $W_2 = 0.125$, $W_3 = 0.375$, $W_4 = 0.125$, $W_5 = 0.125$ and $W_6 = 0.125$ were used, so that the sum of the weights is one. When considering the values of the relative weights, consideration needs to be taken of the usual scale of the values for each element, since these differ. For example, the maximum positional delay for an aircraft is related to the number of aircraft, so is likely to be twenty or less, whereas the holding area delay is in seconds so will often be in the hundreds (or thousands) and the cost of a CTOT extension can be in the tens of thousands, or more.

5.7 Search Enhancements

A number of improvements were made over the research period. Rather than complicate the main explanation above, these improvements are described here.

5.7.1 Use of an exhaustive search when possible

If the number of aircraft that are free for re-sequencing at any time drops to seven or less, there is time to exhaustively sequence the aircraft. An aircraft is considered to be free for re-sequencing from the time it is added to the system until the time at which its position in the take-off sequence has been fixed. This search will guarantee that the optimal solution will be found for this sub-problem, and will complete almost as fast as the five thousand iteration tabu search when there are only a few aircraft as, even with seven aircraft there are only 5040 (= *7!*) different sequences to consider. In this case, the follow-on searches discussed below are not performed.

5.7.2 Use of a solution cache

All of the elements of the solution evaluation system (path allocation, feasibility check, take-off time prediction and objective function) are deterministic, so re-evaluating the same solution later will result in the same solution value. As solution evaluation can be time consuming, solution values are cached to avoid expensive re-evaluation.

Each solution that is evaluated is stored in a solution tree. The levels of the tree represent the positions in the take-off sequence. The nodes of the tree represent which aircraft is in that position in the sequence. A single root node has up to n child nodes (where n is the number of aircraft in the take-off sequence), representing the different aircraft that can be in the first position in the sequence. Each child node has $n-1$ child nodes representing which aircraft is in the second position in the take-off sequence. Each of these has $n-2$ child nodes representing the third aircraft in the sequence, and so on, to the end of the sequence.

The size of a fully populated solution tree would require a prohibitive amount of space as a fully populated tree for n aircraft would have $n!$ leaf nodes along with a large number of intermediate nodes. However, as the tabu search only investigates up to 5000 different solutions it will store a maximum of 5000 leaf nodes, with a similar reduction in the number of intermediate nodes.

Experiments showed that the tree usually had around 2500 solutions in it at the end of the tabu search, providing that sufficient aircraft were in the system. This meant that around half of the evaluations in the tabu search found a cached solution. Use of the cache allowed the search length to be increased from 5000 to 10000 solutions. So, for the tabu search this meant increasing the number of iterations from 100 to 200. The implementation of the follow-on searches described below meant that the number of iterations was later reduced back to 100.

5.7.3 Implementation of a follow-on search

Although the tabu search performed well in past experiments, it gives no guarantees about the quality of the sequences returned. Examination of the sequences produced, as discussed in section 8.8, showed that most aircraft were moved no more than two or three places away from the first-come-first-served sequence. However, some aircraft needed to be moved much further in the sequence. Whenever this happened it was because of CTOT compliance.

As a very high penalty is applied to any sequence where aircraft miss CTOTs, as was seen in section 5.6, the tabu search very quickly moves towards sequences where aircraft are in CTOT, if possible. The majority of the time is then spent improving the delay and fine-tuning the sequence to improve the equity. Experimental observations showed 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 and the improvement in the cost was nominal.

Swapping the positions in the sequence of two aircraft with the same departure route, weight class and speed group can often give a very similar costing sequence. As the tabu search gives no guarantee of finding 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, as discussed in section 5.3.3, helps to prevent this alternating between sequences but presents no guarantee.

These observations led to the development of a follow-on search. By reducing the number of iterations of the tabu search from two hundred back to one hundred, time was gained to perform a second search. The follow-on search starts with the best sequence found by the tabu search and has two stages.

The first stage of the follow-on search checks all possible swaps of aircraft in the sequence, ignoring aircraft for which the sequence 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 previously mentioned problem where there are two almost identical aircraft by ensuring that the sequence where the aircraft are in the reverse order is always considered. A bias in the objective function (detailed in section 5.6) towards first-come-first-served sequences and a further bias towards the previous sequence found ensure that the swapping will occur at most once.

The second stage of the follow-on search was also performed to make obvious local improvements to a sequence. The search starts by considering the first five aircraft in the sequence that are free to be re-sequenced. It exhaustively checks all possible take-off orders for these aircraft, with the rest of the sequence fixed. Any improving sequence is adopted as the new best sequence. The search then moves forward one place in the sequence and investigates the re-sequencing of the next five aircraft, again attempting to improve the take-off sequence.

Together, these searches ensure that a measure of confidence of at least local optimality can be given for the sequences produced. In particular, they ensure that a controller will not look at a sequence and see an obvious improvement that could be made; a situation which would greatly undermine any confidence in a decision support system.

Each of the second stage searches must evaluate 120 different solutions. With a maximum of twenty aircraft free for sequencing, 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.

The total number of solutions that needed to be evaluated has been reduced from 10000 when 200 iterations of the tabu search were used, to a worst case of 6940 (= 5000 (search) + 380 (swaps) + 120 + (15 x 96)) evaluations. Simultaneously, the performance in experiments was slightly improved and the occasional alternation between similar sequences, where the positions of two aircraft were swapped, was totally eliminated as the similarly costed sequence was guaranteed to be considered. The solution cache was made available to the follow-on searches so many of the sequences considered did not need to be re-evaluated, and there were further time savings.

The tabu search alone performed significantly better in experiments than a rolling window search alone; even a search with a window size of seven aircraft, as shown in appendix C. The tabu search has the advantage of being holistic, whereas the limited window exhaustive search is limited by its myopic view. Given the benefits of considering more aircraft in the sequencing that are seen in section 8.4, this is hardly surprising.

5.8 Summary

The sequencing and scheduling problem was considered in this chapter. A tabu search metaheuristic was proposed for the search through sequences and the design of the neighbourhood used was explained. Formulae were presented for predicting take-off times and determining a cost for a sequence. The necessary method for verifying the feasibility of re-sequencing and identifying any re-sequencing delays in the holding areas has been left for the next chapter. This chapter ended with an explanation of some of the additional features which were added to improve the performance of the selected solution method.

Chapter 6

The Control Problem And The Holding Area

6.1 Introduction

The holding area structures are vital elements of the Heathrow take-off problem and cannot be ignored by any practical decision support system. As any re-sequencing necessary to attain a target take-off sequence has to be performed within the holding area, it is important to verify that it is actually achievable. Determining whether the desired take-off sequence can be achieved, or how it will be achieved, is referred to as the control problem. The control problem involves both the allocation of paths through the holding area to aircraft and the verification of the feasibility of re-sequencing.

An overview of the proposed decision support system was given at the end of chapter 4. Most of the components of the decision support system were detailed in chapter 5. The descriptions of the components which deal with the control problem were reserved for this chapter.

This chapter starts with a consideration of the holding area models and a description of how to create such a model for a holding area. The chapter continues by providing important definitions of vital elements of the control problem.

The importance of the paths that are used to traverse the holding area was seen in section 2.8. The observations made there motivated the design of the path allocation system. An overview of the system is given in section 6.4. The algorithm is described in detail in section 6.5.

In section 6.6, the basic method used to determine the feasibility of re-sequencing is detailed. A heuristic method for restricting movement (described in section 6.8) is used so that temporal constraints can be implicitly included in the evaluation. In order to build more time-based constraints into the holding area movement, a mechanism for dealing with the current time is introduced. The chapter ends with the presentation of the improved feasibility check algorithm which uses the current time to limit the earliest take-off times for aircraft according to the expected manoeuvring in the holding area.

6.2 Holding Area Model, Paths And Path Suffixes

The directed graph model of the holding area was introduced in section 2.7.1. As discussed in section 2.7, the decision support system presented in this thesis uses a directed graph model to represent each holding area. The creation of these models from a holding area plan, such as that in figure 6.1, is described in this section.

These holding area models are similar to those used in the research into airport ground movement, reviewed in section 3.8. Although most of the previous research, for example [143, 96, 94, 81, 169], used the nodes to represent the locations at which aircraft could be located, the arcs have instead been used to represent locations in the past, [56]. Nodes rather than arcs were chosen to represent the locations of aircraft for this research as this makes the holding area graph more intuitive and easier to produce.

Holding area plans were available for each of the holding areas. The holding area plans show the taxiways and holding areas as a series of numbered blocks. It is common to be able to hold aircraft at the edge of blocks, so the block positions are important.

6.2.1 Creating the holding area models

The first step in creating the holding area graphs is to plot the paths through the holding area. These paths will later be divided into arcs and nodes. When building these models, controller preferences should be of prime consideration. Where the holding area is already in use, the initial paths added should be those the controllers currently use as these are guaranteed to be acceptable to controllers. If the holding area is not already being used, problem domain knowledge will be needed to identify the acceptable paths for aircraft to follow.

6.2.2 Unusual circumstances or modes of operation

Some paths may be acceptable to controllers, but only used in extreme circumstances. A decision can be made about whether to make these paths available to the decision support system under normal circumstances or only when the controller explicitly permits them. Multiple different models can be produced for the same holding area to support different operating modes. In some models these less desirable paths will exist, in others they will not. Some models may, for example, be used if there is a blockage at a specific point in the holding area, so that paths through the blocked position are not available but paths which would not normally be used become available.

It is possible to use the path allocation system to handle special cases within the same model rather than using different models. Additional flexibility could be included in the models, beyond that which the controllers would normally utilise, but it is important that every included path is achievable. The path allocation system (described in sections 6.3 and 6.4) has to make decisions about which paths to use, and can apply a preference for specific (simpler or shorter) paths rather than other longer/more complicated paths, or can ensure that less desirable paths are only ever allocated when specific circumstances dictate that it is sensible.

Using different models for different circumstances, rather than using a single combined model and handling special cases within the path allocation system hands control of the decision of when to change the mode of operation to the controller. The controller is deciding whether the situation justifies a change in operation rather than the system making the decision based on its perception of the situation. This is usually a good thing as long as this is done infrequently so that controller workload is not adversely affected. This also has the advantage that each holding area model will be simpler than the combined model, so the task faced by the decision support system will consequently be simpler.

6.2.3 Determining the nodes in the graph

Once paths have been determined, nodes on the paths need to be generated. Nodes should be added to paths at any point at which paths converge or diverge. If paths converge/diverge such that the nodes are very close together and there is insufficient room for aircraft to simultaneously occupy both nodes, then these can be combined into a single node. A node should also be added for each holding point, as these are the positions at which aircraft will usually hold. A node should be added for each exit from the holding area onto the runway (labelled R, T, V and Y in figure 6.2) and a node should be added for each entrance into the holding area (labelled D, G and K in figure 6.2).

Nodes should then be added between the existing nodes to represent the queueing positions between them. The number of such nodes added should be equal to the number of aircraft which can be queued between the nodes when there is an aircraft at each of the existing nodes. Finally, if desired, nodes can be added beyond the entrance nodes to represent aircraft queueing on the taxiways. The nodes labelled A to K in figure 6.2 are of this type.

If nodes are very close together, such that they cannot be simultaneously occupied by aircraft, then the two nodes should be replaced by a single node. This is important because the feasibility check will assume that each node can be simultaneously occupied and will use the number of nodes to determine the occupancy restrictions.

Once all of the nodes have been identified, arcs are added to join the nodes along the identified paths. The majority of arcs will be uni-directional, as aircraft usually head for the runway from the entrances. Some arcs may be bi-directional and will need special handling within the feasibility check described later. If the arcs are used in different directions only under different circumstances or modes of operation then the building of multiple models, one for each circumstance, should be considered. It is important to try to keep the holding area models as simple as possible, in order to reduce the complexity of the feasibility check.

As the graphs are used for determining whether re-sequencing is possible, it is important that the graphs represent the re-sequencing that can be done rather than strictly denoting the positions at which aircraft can be held within the holding area. At times this may mean that nodes denote regions rather than specific positions, to model the fact that an aircraft in that area will block other paths. It is important to understand how the controllers use (or would use) the holding areas to re-sequence the aircraft. With this understanding, it is possible to ensure that the models for the holding areas allow them to be used in the ways that the controllers desire.

Once the initial model has been created, a number of simple tests can be applied. It should be possible to simultaneously have an aircraft at each position represented by a node in the graph. It should also be possible for an aircraft to perform the movement related to the traversal of each arc in the graph even when all other nodes of the graph that are not linked by the arc are occupied by aircraft.

Plans of the four holding areas are given below. Directed graphs for the three commonly used holding areas (09R, 27R, 27L) are provided for comparison between the physical layout and the directed graphs used in this research. The fourth holding area (09L) can be considered to be a cut-down version of the 09R holding area if a graph is required so will be ignored here.

6.2.4 The 27R holding area

Figure 6.1 shows a plan of the 27R holding area. There are three main entrances for aircraft, at blocks 32(O), 39(O) and 49 on the plan. There are two exits from the holding area onto the runway, entering the runway at blocks 18 or 19. Block 19 is preferred as it allows the entire runway to be used and permits an aircraft to be lined up more swiftly.

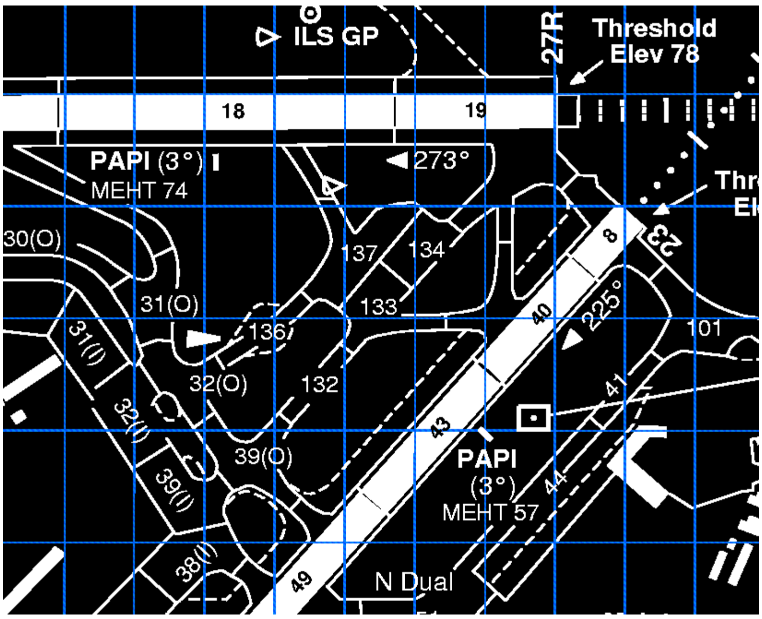


FIGURE 6.1: Plan of the 27R holding area

The 27R holding area can be thought of as having three take-off queues, with some limited opportunity for aircraft to move between queues, hold to be overtaken or overtake other

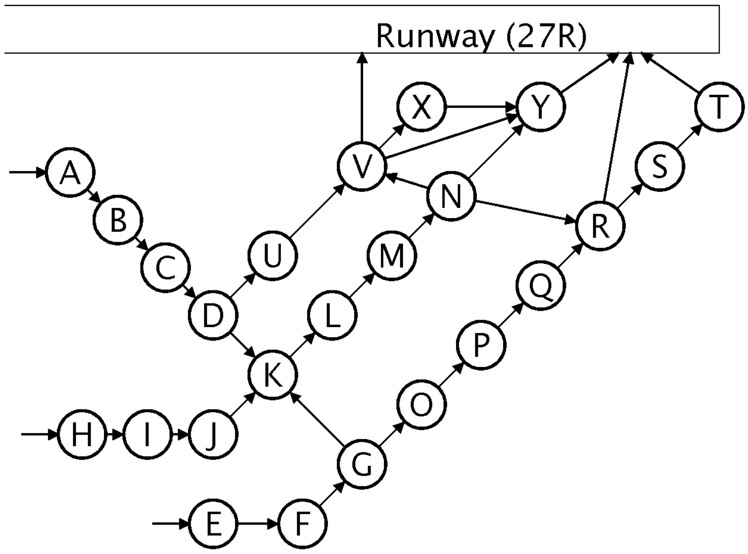


Figure 6.2: Directed graph model of the 27R holding area

aircraft in the queue.

The first take-off queue enters at block 136 and passes through blocks 137 and 134 before entering the runway. There is a possibility to overtake an aircraft that is holding at the front of 137 by turning right in 137 and passing down the right hand side of the holding aircraft, through the front of 133 and 134. Finally, an aircraft could turn left from 137 and enter the runway in block 18. This is not ideal, however, as the end of the runway is preferred.

The second queue runs from 132 to 133 then 134 and out to the runway. Aircraft may take a longer route if necessary, allowing them to be overtaken. For example, an aircraft which has to wait a long time for a CTOT take-off time-slot could be instructed to turn right at 133 into block 40. It could then taxi along to block 8 and wait. If necessary, the aircraft could be instructed to turn left at block 133, into the back of block 137 and enter the runway at block 18. Again this is not ideal.

Theoretically, an aircraft could turn left at block 133, into 137 then turn right and hold at the front of 137. It could then follow the normal route from 137 back into 134 and line up for the runway. Although possible (and playback of historic data shows it is used occasionally) a controller stated that the amount of manoeuvring this requires is prohibitive, so the path was eliminated from consideration.

The final runway queue uses the end of the third runway, labelled as blocks 49, 43, 40 and 8, before turning left onto the runway. If an aircraft using this queue has to overtake another aircraft in the queue it can turn left at block 40, allowing it to overtake anything queued at the front of block 40 or in block 8.

The way in which this behaviour has been modelled, as a directed graph, is shown in figure 6.2. The implementation of some nodes directly prohibits certain undesirable behaviour.

For example, node R on the graph denotes both the holding point between block 40 and block 19 and the south-west position of block 40 itself. Doing this prevents aircraft from taxiing from blocks 43 to 40 if there is an aircraft waiting to enter the runway from the holding point at block 40. If this behaviour is to be permitted then an additional node should be added to the graph between node R and the runway to represent this holding position.

6.2.5 The 27L holding area

The 27L holding area is more flexible and much bigger than the 27R holding area. A plan of the holding area is shown in figure 6.3 and a directed graph model in figure 6.4.

The 27L holding area consists of two disjoint holding areas, one north and one south of the runway. In normal operation, aircraft from terminal four will use the south side and aircraft from terminals one, two and three the north side. This reduces the number of runway crossings required.

South of the runway there is a single runway queue which splits into two queues (at block 95) which then converge again (in block 88). Aircraft can take the shorter route, or the longer route allowing holding at block 94. The entire queue can be bypassed by entering the runway from block 118 but as these aircraft do not enter the runway at the end this is not ideal.

North of the runway, the situation is much more flexible. Aircraft can enter the holding area via blocks 73/74, 48/54, 48/50 or 49/51, depending upon where they are taxiing from and the route they take to get to the holding area. The main route through the holding area is via block 131, after which aircraft can queue on the left (block 138) or right (block 75) before entering the end of the runway. An alternative route allows entrance to the runway via block 130. This appears to be used more for aircraft from 73/74 than the other entrances. As it does not enter the runway at the end, it is not as useful, but it allows aircraft to overtake all those waiting at 75 or 138.

Any aircraft entering via blocks 50 or 51 can be directed into the main route via 54 and 131, or can take the path 141, 56 and 71 to the end of the runway. Taking the latter path is quite restrictive, however, as there is then no way to further re-sequence the aircraft. A common way to use this path is to use blocks 140 and 139 as holding positions and route the rest of the aircraft straight down 141, 56 and 76. This assumes that any aircraft which need a delay will be removed from the queue and the rest of the aircraft will then be able to leave in the order they are queued, by appropriate interleaving them with aircraft waiting at 75, 138, 87 and 130.

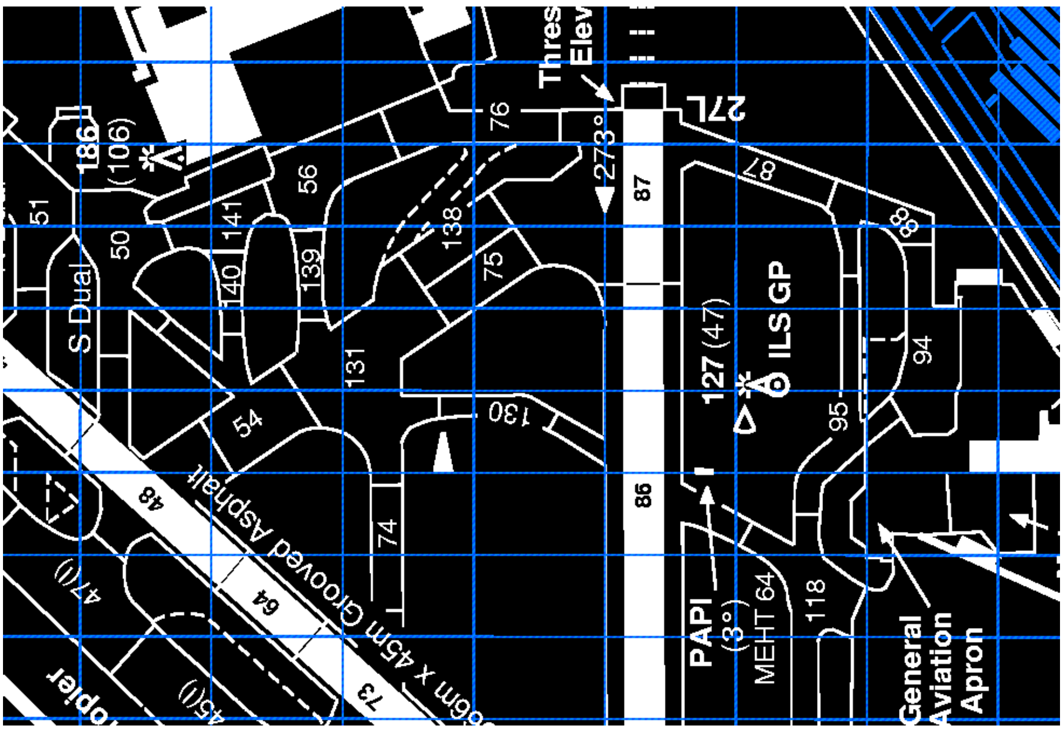


FIGURE 6.3: Plan of the 27L holding area

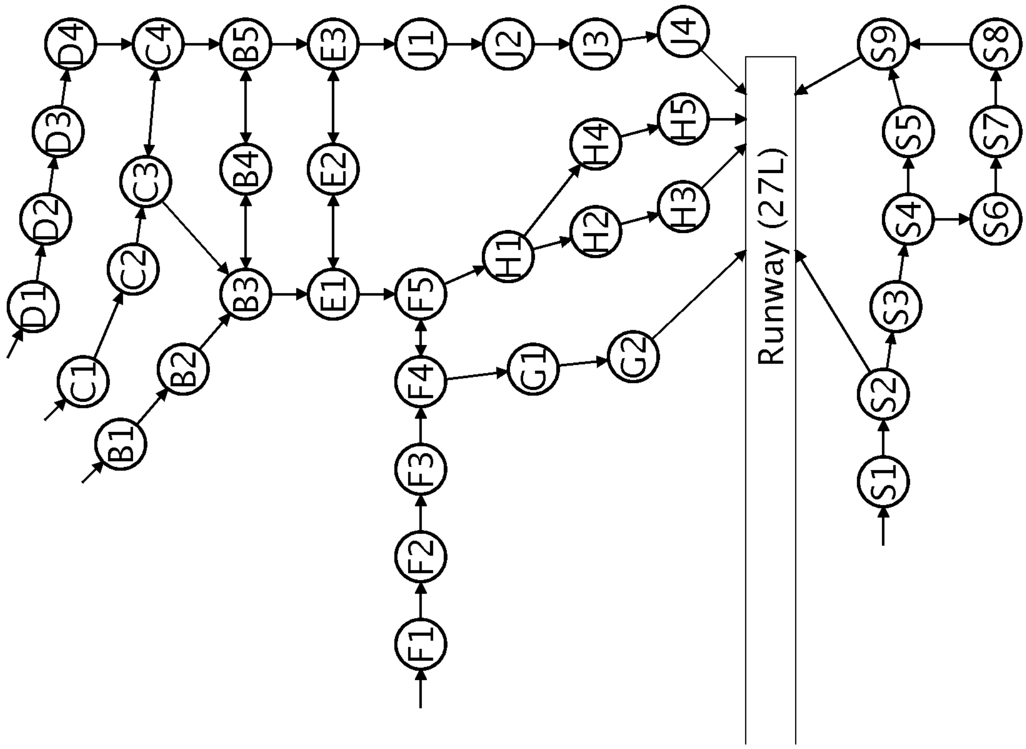


FIGURE 6.4: Directed graph model of the 27L holding area

6.2.6 The 09R holding area

Figure 6.5 shows a plan of the 09R holding area. There is one long queue north and one long queue south of the runway. This holding area has undergone much change recently, with the building of terminal five. The new plan is more flexible and has a second entrance queue north of and parallel to the northern queue. The old plan was kept for this research as it provides an interesting example of how the decision support system algorithm performs in a more constrained situation.

From both of the existing queues, there is the possibility to enter the runway early, rather than at the end. Controllers sometimes use these entrances at quiet times if aircraft do not need the full runway for take-off. They can also be used to overtake aircraft waiting near the end of the runway and are valuable for increasing the re-sequencing opportunities.

One controller mentioned that, prior to the changes, the throughput was often kept high by performing partial sequencing of aircraft prior to entry at the holding area. The results in chapter 8 will show that the decision support system can cope well even without this pre-sequencing.

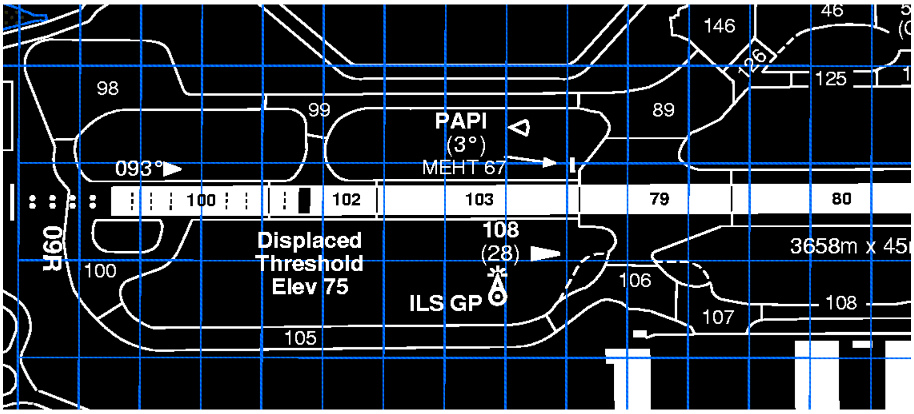


FIGURE 6.5: Plan of the 09R holding area

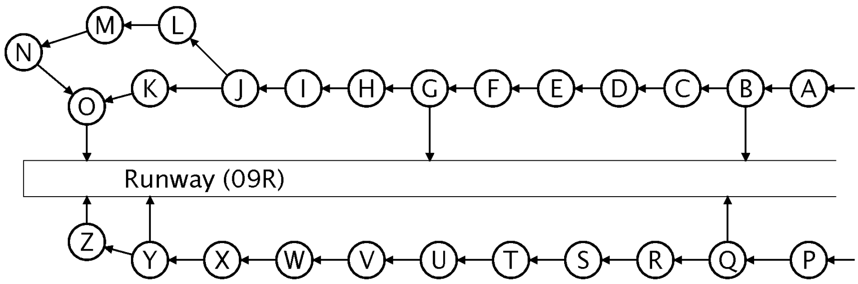


FIGURE 6.6: Directed graph model of the 09R holding area

A directed graph model is given for the holding area in figure 6.6. Even though a number of exits to the runway are shown there is no reason for a decision support system to support them all. Indeed, the decision support system described here would not use all of them in normal operation.

Near the end of the runway, the northern queue opens out into an overtaking area in block 98. Effectively, this overtaking area allows one queue of aircraft to be held around the outside of the area while other aircraft overtake along the inner side, nearer to the runway. This behaviour could be implemented either as two queues (a long one around the outside of the turn and a shorter one around the inside) or a parking place for a number of aircraft to be overtaken and an overtaking path. The graph in figure 6.6 makes no assumptions about how block 98 will actually be used.

The southern path also has a limited ability for overtaking. It is possible to move an aircraft along to the end of the path and line up right at the end of the runway while another aircraft takes the other path and lines up in front of it on the runway. This is not ideal though, due to the manoeuvring involved. In this case, in the holding area graph, node Z should be considered to represent the end of the runway.

6.2.7 The 09L holding area

The 09L holding area is hardly ever used, in order to control noise for nearby residents. The plan is shown in figure 6.7. It can be thought of as a reduced version of the 09R holding area. The overtaking that can be performed is very limited compared with any of the other three holding areas. This holding area is ignored in this research, because it is very simple and because it is very rarely used in practice. The solution method for the 09R holding area will also work for the 09L holding area, which has a reduced subset of the 09R functionality.

Leese et al. showed in [124] that a dynamic programming approach is appropriate for this kind of holding area as the number of re-sequencing options is so small, although that research did ignore some of the other constraints and objectives of the re-sequencing and it ignored the runway entrance from block 116 to 113.

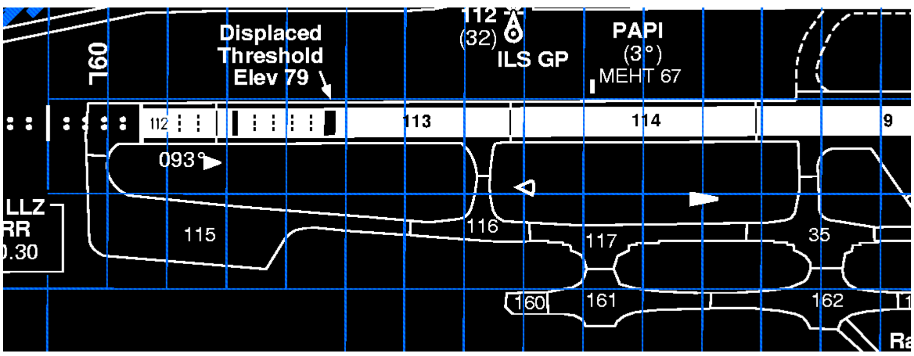


FIGURE 6.7: Plan of the 09L holding area

6.3 Important Definitions

A number of definitions are introduced here and will be used in the remainder of this chapter.

6.3.1 Traversal Paths

The concept of a traversal path and the importance of the path allocation were introduced in section 2.8. When the holding area is defined as a directed graph, such as in figures 6.2, 6.4 and 6.6, the traversal path followed by an aircraft can be uniquely defined in terms of the ordered sequence of nodes passed.

Figure 6.8 shows a graph for a simplified holding area. Assuming the entrance nodes are A and B, there are four possible paths through this graph: $ACER$, $ADER$, $BCER$ and $BDER$.

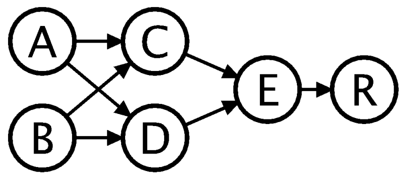


FIGURE 6.8: Example simplified holding area graph

6.3.2 Path Suffix

The concept of a path suffix is important for the algorithm to perform the feasibility test. A path suffix is an ordered set of nodes beyond the current node which one or more paths traverse to reach the runway. A path suffix defines a section of a path from a given node to the runway. Any nodes on the path before the given node are ignored for the purposes of defining a path suffix. Multiple paths often, therefore, share the same path suffix beyond a given node.

For example, in figure 6.8 there are two suffixes for A, which are CER and DER. The relationships between paths, suffixes and nodes are illustrated in figure 6.8. Suffixes effectively relate nodes and paths together. For example, there are two suffixes at nodes A and B and only one at nodes C, D and E. There is a one-to-many relationship between nodes and suffixes. There is a many-to-many relationship between paths and suffixes.

6.3.3 Aircraft Priority

The concept of a priority for an aircraft is introduced here, to simplify explanations later. The priority of an aircraft for the feasibility check is defined by the take-off order. If aircraft A must leave before aircraft B then its priority is said to be greater than that of aircraft B. By assigning a numerical priority to all aircraft (from the position in the take-off sequence), the number of higher priority aircraft on any suffix can be quickly counted.

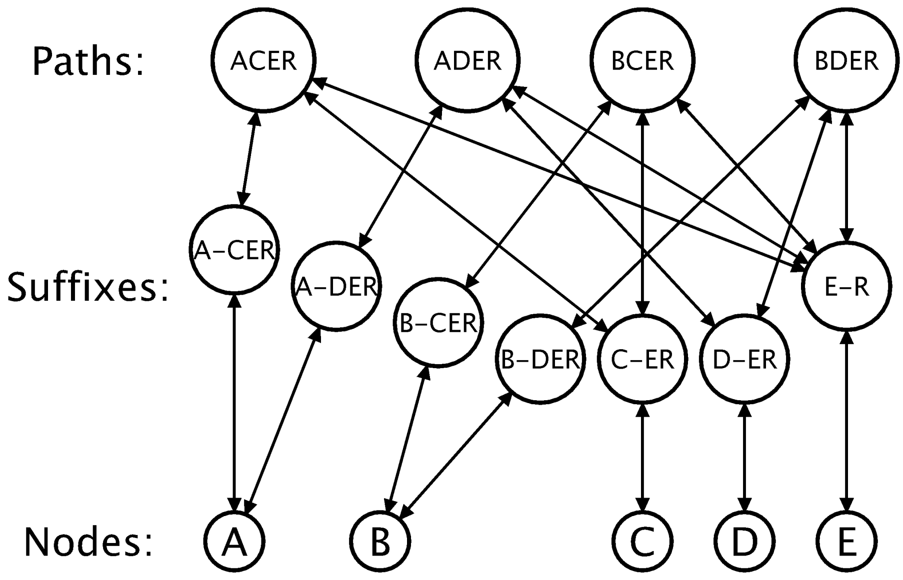


FIGURE 6.9: The relationship between paths, nodes and suffixes

6.3.4 Nodes of convergence

A *'node of convergence'* is a node which can be entered from more than one other node, so it is a node in the holding area graph where multiple paths converge. An example node of convergence is node E in figure 6.10. Here the paths $ABEFG$ and $CDEFG$ converge at node E. Nodes of convergence can be quickly identified as they are the only nodes with multiple incoming arcs. The possible nodes of convergence in figure 6.2 are K, N, R, V and Y.

Nodes of convergence are vitally important for the feasibility check as it is at these points that a decision must be made by a runway controller about which aircraft should pass first. The sequence in which aircraft pass a node of convergence will restrict the possible take-off sequences. The sequence in which aircraft pass any other node is uniquely determined by the sequencing at the nodes of convergence.

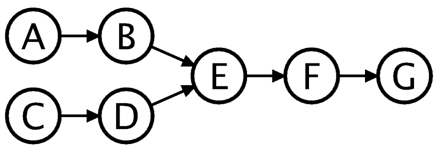


FIGURE 6.10: Path convergence

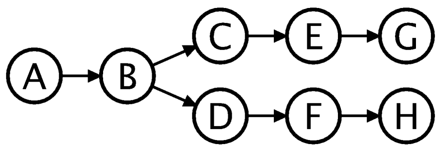


FIGURE 6.11: Path divergence

6.3.5 Nodes of divergence

'Nodes of divergence' are nodes at which two paths diverge. The diverging paths both use this node but the next node after the node of divergence differs between the paths. In figure 6.11, node B is a node of divergence.

Two paths will have different path suffixes at any node, N, if they diverge at node N itself or at any node between N and the runway. Given a holding area graph that has arcs only for paths that will be used, the nodes of divergence are the nodes with multiple outgoing arcs.

The possible nodes of divergence in figure 6.2 are D, G, N, R and V. The nodes that are actually nodes of convergence or divergence in practice will depend upon which paths are actually used at the time.

6.4 The General Path Allocation Algorithm

As was explained in section 2.8, the allocation to aircraft of paths through the holding area is a very important aspect of the problem considered in this thesis. The method presented here for allocating these paths is derived from the observations about sensible path allocations that were made in section 2.8.

The designed solution method requires that a path allocation heuristic is produced for each holding area. The production of the path allocation heuristics is discussed in this section. The detailed path allocation heuristics, which were used for the Heathrow holding areas considered in this research are detailed in section 6.5 and were produced by applying the principles which are explained in this section.

6.4.1 Identification of the types of paths

Before the path allocation process can take place, it is important to evaluate the different paths that are available for each holding area entrance and the circumstances under which they are used. A number of path categories have become apparent through discussing this problem with controllers and examining the ways in which holding areas are used. The ways in which these paths are used (for example when a slow path is used instead of a fast path) can be observed in the general path allocation algorithm, given by algorithm 2 and the specific algorithms for each holding area entrance, given by algorithms 3 to 10.

The fast paths

These are the most direct paths through the holding area, and are often preferred for that reason. The fastest paths in the 27R holding area graph, given in figure 6.2, are the paths $DUVYZ$, $KLMNYZ$ and $GOPQRZ$ for the holding area entrances D, K and G respectively.

The slower but good paths

These paths are slower but are fast enough and easy enough for an aircraft to traverse that they are worth allocating to aircraft. These paths are commonly used, but usually only when there is a reason why the fast paths cannot be used. The slower paths in the 27R holding area are $DUVXYZ$, $KLMNYZ$ and $GOPQRSTZ$.

The default path

It is important to identify the default path for each holding area entrance. This is the path that controllers will use if there is no restriction upon the re-sequencing of the aircraft. This will often be the fast path, but may be a slower path in cases where the controllers view this to be almost as fast and easy to navigate as the fast path, but more useful for maintaining flexibility for later re-sequencing. The default paths in the 27R holding area are the $DUVXYZ$, $KLMNYZ$ and $GOPQRSTZ$ paths. The $DUVXYZ$ path is a slower path but is almost as fast to use as the $DUVYZ$ path. The $GOPQRSTZ$ path is longer than the $GOPQRZ$ path but is useful for allowing later aircraft to overtake those at S or T and is very easy for a pilot to use.

Overtaking and short-cut paths

There may be one or more paths which are useful for overtaking aircraft already within the holding area, but which would not otherwise be used. For example, in the 27R holding area, it is useful to use the $DUVZ$ path to overtake an aircraft that has already been assigned to the $DUVYZ$ path, or to allow more than one aircraft from entrance D to be overtaken. The overtaking paths in the 27R holding area are the paths $DUVZ$ and $KLMNVZ$, for entrances D and K respectively.

Sometimes the overtaking path may enter the runway away from the end. Any path which enters the runway from node V in the 27R holding area in figure 6.2 is an example of such a path. These can be considered to take a short-cut to the runway, allowing additional overtaking, however the fact that the full runway cannot be used for the take-off means that they are not ideal. Since larger aircraft may need the full runway, the path allocation algorithms used for this research prevented the decision support system from assigning a heavy weight class aircraft to a path which would not allow it to use the full runway.

6.4.2 Overview of the general path allocation heuristic

Prior to applying the path allocation heuristic, the aircraft must first be divided according to the holding area entrance at which they arrive. It is then necessary to determine the overtaking that is required between the aircraft which arrive at the same holding area entrance. Relative overtaking between aircraft which arrive at different entrances is ignored. The paths that are allocated to some aircraft may have been fixed, as discussed in section 2.12.2. For these aircraft, the fixed path is retained and is allocated prior to using the path allocation heuristic. Finally, variants of algorithm 2 are applied to allocate the paths to the aircraft which arrive at each of the holding area entrances, in turn.

Algorithm 2 uses the concepts of a fast, slow, default and shortcut path which were introduced above. Steps *1* to *5* allocate the default path to any aircraft for which it is appropriate. Steps *6* to *10* allocate the slow path to aircraft which need to be overtaken, and do not need

Algorithm 2 General path allocation heuristic algorithm

```
1: for each aircraft a arriving at the entrance, which has not had a path allocated do
2:     if a is not overtaken and does not overtake then
3:         assign the default path to aircraft a.
4:     end if
5: end for each
6: for each aircraft a arriving at the entrance, which has not had a path allocated do
7:     if a is overtaken but does not overtake then
8:         assign the slow path to aircraft a.
9:     end if
10: end for each
11: for each aircraft a arriving at the entrance, which has not had a path allocated do
12:     if a only overtakes aircraft which have been allocated to the slow path, and these are few
        enough that it can overtake using the fast path then
13:         assign the fast path to aircraft a.
14:     end if
15: end for each
16: for each aircraft a arriving at the entrance, which has not had a path allocated do
17:     if paths have been allocated to all of the aircraft that a overtakes and it is valid for a to
        use the overtaking or shortcut path then
18:         assign the shortcut path to aircraft a.
19:     end if
20: end for each
21: if each there exists an aircraft a which has not had a path allocated then
22:     declare the re-sequencing to be infeasible
23: end if
```

to overtake, in order to move them out of the way of the overtaking aircraft. Steps *11* to *15* then allocate the fast path to aircraft which only need to overtake those aircraft which have been allocated to the slow path. Note that, in order to overtake using the fast path it is important to know the number of aircraft overtaken and the number of positions at which aircraft can be parked on the slow path to be overtaken by aircraft which are on the fast path. It may be necessary to use the shortcut path to overtake if there is insufficient space to park aircraft in order to overtake using the fast path. Finally, steps *16* to *20* allocate the overtaking or shortcut path to the remaining aircraft where possible, as long as an overtaking path exists for that holding area entrance and the aircraft is permitted to use it. For example, heavy aircraft are constrained by the implemented algorithms to only be allocated paths which will allow them to use the full runway for take-off.

6.4.3 Reasons for a per-entrance allocation of paths

As discussed in section 1.3, although the intention is for the decision support system to present a controller with a suggested take-off sequence, it is important that the method to achieve the desired re-sequencing is obvious to controllers. This means that a controller must be able to easily identify the paths to allocate to aircraft and the sequence in which aircraft should pass different points within the holding area. As the size of the control problem increases, this identification becomes more difficult and it becomes necessary to decompose the problem into smaller sub-problems. The obvious and intuitive way to do this is to consider the paths allocated to aircraft on a per-entrance basis. The designed decision support system uses this approach.

The intuitiveness of the per-entrance decomposition of the problem is a major advantage of the approach, enabling the controllers to consider a much smaller set of aircraft and paths in isolation. The controllers can then use a simple method to determine the paths to allocate to aircraft. Moreover, the controllers can be sure that such a simplistic consideration of the paths to allocate to aircraft will be sufficient to achieve any suggested take-off sequence as the designed decision support system provides a guarantee that the suggested sequences are achievable using a per-entrance path allocation method.

A per-entrance path allocation has been adopted, despite the fact that it is possible to construct scenarios where a per-entrance path allocation will not allow the aircraft to be successfully re-sequenced but where there is an alternative path allocation that could be assigned to aircraft which would allow the re-sequencing to be achieved, since it works well for the Heathrow holding areas and has a number of important advantages. The advantages can be summarised as follows:

1. The per-entrance path allocation ensures that the default path is allocated whenever possible. Furthermore it ensures that paths will be allocated in a manner which will seem sensible to controllers and will be acceptable.

2. If the per-entrance path allocation fails but a different path allocation would work then, when considering the aircraft arriving at each entrance in isolation, the path allocation will not be as sensible as a per-entrance path allocation would be for at least one of these entrances (otherwise it would be the per-entrance path allocation). It would be necessary in that case to verify that the path allocation would still be acceptable to controllers.

3. If the per-entrance path allocation is not used then there is no guarantee that the controller will be able to identify how the re-sequencing is to be achieved. As explained above, the decomposition of the problem by holding area entrance makes it easier for controllers to determine how a take-off sequence can be attained. Without this, the controller is less likely to see the sequence as sensible, so less likely to accept the sequencing advice. This could undermine long term controller confidence in the system as well as meaning that the controller ignores the immediate advice.

4. If the system ever suggests any take-off sequences which are not achievable using a per-entrance allocation method then the controller can no longer use the fact that the system only uses such per-entrance allocated paths in order to simplify the determination of the paths necessary to achieve any suggested sequence. Consistently using such a system allows the controller to rely upon this fact in determining how the sequence will be attained and ensures that the method will be obvious to the controller.

The per-entrance path allocation method has been observed to be effective for the

Heathrow holding areas, since the number of points at which aircraft can conflict with each other is usually quite small and, in most practical cases, where a per-entrance allocation fails to achieve the re-sequencing, either the re-sequencing would not have been possible with any other path allocation, or the path allocation that would have been necessary would not have been acceptable. The observations below give some insight into the reasons for this.

Reasons why the per-entrance path allocation is effective

Firstly, in some cases, such as the 09R holding area, the independence of path allocation between aircraft arriving at different entrances is assured as the holding area structures beyond the entrances are disjoint. In this case, consideration on a per-entrance rather than global basis can make no difference.

Secondly, problems can only occur when aircraft are moved from the queue from one entrance into the queue for another entrance, potentially interfering with the re-sequencing that is occurring with the aircraft from that second entrance. The only time an aircraft cannot be allocated to the preferred path (and thus may need to be moved into the queue from another entrance) is when it overtakes or is overtaken by another aircraft which arrives at the same entrance. When this is not the case, re-sequencing merely requires the interleaving of the queues from the different entrances at the node at which the queues converge. There is an objective to control the inequity in the take-off sequence and this helps to control the total amount of re-sequencing that is required. Since so much re-sequencing can be performed by the re-sequencing of aircraft which arrive at different entrances relative to each other, the amount of re-sequencing that is usually required between aircraft in the same queue is even lower.

The effect of moving an aircraft between queues often depends upon how long the moved aircraft will be present for, since it will block a path from the other entrance for the duration for which it is present. The bias towards more equitable sequencing helps to ensure that an aircraft is not usually delayed for long in the holding area, so reduces the length of time for which any path will be blocked. As will be seen in chapter 8, aircraft only have long holds when they are waiting for the start of a CTOT take-off slot. In practice, the controllers would know about any excessive required delay for a late CTOT in a live situation and (if an earlier CTOT could not be negotiated) would release the aircraft from the stand later (when possible) or park the aircraft at a remote holding location, in order to reduce the duration of the delay in the runway holding area. (As will be seen in chapter 8, where long delays are observed with the test data, the controllers had actually re-negotiated the CTOTs to allow aircraft to take off earlier.) This is important since it helps to limit the duration for which parked aircraft block paths through the holding area.

Note that it is possible to estimate the take-off time of an aircraft prior to performing the feasibility check, as discussed in section 6.9.3. If large delays are expected, then the length of time for which the path would be blocked could be predicted and used in the path allocation system for the affected entrances to avoid allocating to the blocked paths aircraft which need to take off before the blocking aircraft does.

6.4.4 Controllers are not obliged to use the predicted paths

It should be noted that controllers are not obliged to achieve the take-off sequence in the same way as the system predicts they will, although the focus upon using the preferred paths should mean that they usually will. It is useful to be aware of this, since it allows the path allocation system to maintain some flexibility without forcing the controllers to do so. This applies particularly to the allocation of a default path, where there may be an option to choose between the shorter less flexible path or a longer path which allows for greater flexibility for later aircraft to overtake.

For example, the path allocation system could always allocate the $DUVXYZ$ path in the 27R holding area shown in figure 6.2 in preference to the $DUVYZ$ path whenever the situation is such that either path allocation could achieve the desired re-sequencing. In this case the preference is to use the longer path as the default, on the basis that the controller is not told the path that has been allocated and so would use the $DUVYZ$ path if it was more appropriate anyway. This actually mirrors what controllers really do in that they will direct aircraft which enter at the entrance related to the D node towards the position related to the X node in order to provide the facility to hold at X to be overtaken. If the controller decides that the aircraft does not need to be overtaken then they can direct the pilot to turn right into Y prior to reaching X, thus changing the path taken.

Conversely, there may be times when there are advantages for the system to assume a less flexible path allocation than a controller may use. For example, the decision support system could allocate the path $GOPQRZ$ in preference to $GOPQRSTZ$ as the default path for aircraft entering at node G. Obviously, the default path only applies to any aircraft which does not need to be overtaken by another aircraft from the same entrance, as can be observed from algorithm 2, in which case either path allocation would permit identical re-sequencing. In particular, the controller would still have the flexibility to send the aircraft via nodes S and T if desired, without decreasing the amount of re-sequencing that can be achieved. However, this allocation allows the path allocation system for entrance G to ignore whether an aircraft from entrance K has been parked at S or T in order to be overtaken by another aircraft from entrance K. Without this, aircraft from entrance K would not be able to enter node R from node N until all earlier take-offs from entrance G had done so. This altered preference reduces the potential delay for aircraft at node N.

6.4.5 When the feasibility check becomes unnecessary

The path allocation heuristic may be able to assign paths to aircraft and determine that these paths will not interfere with each other. In this case, the feasibility check which follows can be skipped as the sequence will be known to be feasible. Unfortunately, checking the feasibility at the holding area can be a complex issue and the holding area path assignment cannot always catch all infeasible solutions due to the complex interactions between holding area paths so the feasibility check detailed below is usually necessary.

Obviously, if the path allocation heuristic cannot assign paths then there is no need (or ability) to test the feasibility of re-sequencing. Many of the more complex re-sequencing can be declared impossible at the path allocation stage. One example (in most cases) is trying to achieve a reversal of the sequence of four aircraft which arrive at the same holding area entrance.

6.5 The Path Allocation Heuristics Used For This Research

The algorithms for the path allocation heuristics used for the research in this thesis are given in this section. The general algorithm was described by algorithm 2. In each case, aircraft are first separated into groups according to the holding area entrance at which they arrived or are predicted to arrive. Once this has been performed, the aircraft that are in the same group and which overtake or are overtaken by each aircraft are identified. At the end of this pre-processing, the system has a number of disjoint groups of aircraft, and a record for each aircraft of which other aircraft it overtakes and which aircraft overtake it.

In the following description, paths are defined by the ordered list of identifiers for the nodes which are passed. All paths are assumed to end at the runway so the runway node is not included.

6.5.1 The 27R holding area

The 27R holding area has three holding area entrances, and is shown in figure 6.2. Aircraft from terminals 1 to 3 are assumed to arrive at entrance D or K depending upon which is the most convenient for the stand from which the aircraft departed. Aircraft from terminal 4 (south of the runway) are assumed to arrive at entrance G. The path allocation algorithms for entrances D, K and G can be expressed by algorithms 3, 4 and 5 respectively.

Each of these algorithms follows the form of algorithm 2. Each entrance has a default, fast and slow path. The default paths are $DUVXY$, $GOPQR$ and $KLMNY$ The fast paths are $DUVY$, $GOPQR$ and $KLMNY$. The slow paths are $DUVXY$, $GOPQRST$ and $KLMNY$. Entrances D and K have an overtaking path available which does not use the entire runway, but entance G does not. The shortcut paths are DUV and $KLMNV$.

6.5.2 The 27L holding area

The 27L holding area has many holding area entrances, and is shown in figure 6.4. Aircraft from terminals 1 to 3 were assigned to entrances $B1$ or $F1$ depending upon which was more appropriate for the stand they were taxiing from. Aircraft from terminal 4 were assigned to entrance $S1$. Entrances $C1$ and $D1$ are used by ground controllers according to pre-sequencing

Algorithm 3 Path allocation heuristic for the 27R holding area, entrance D

for each aircraft a in the current group which has not had a path allocated **do**
  if a is not overtaken and does not overtake **then**
    assign (default) path $DUVXY$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a is overtaken but does not overtake **then**
    assign (slow) path $DUVXY$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a overtakes a single aircraft, b, and b is on the $DUVXY$ path **then**
    assign (fast) path $DUVY$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a path has been allocated to every aircraft which a overtakes, and a is not a heavy aircraft **then**
    assign (shortcut) path DUV to aircraft a.
  end if
end for each
if each there exists an aircraft a which has not had a path allocated **then**
  declare the re-sequencing to be infeasible
end if

Algorithm 4 Path allocation heuristic for the 27R holding area, entrance K

for each aircraft a in the current group which has not had a path allocated **do**
  if a is not overtaken and does not overtake **then**
    assign (default) path $KLMNY$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a is overtaken but does not overtake **then**
    assign (slow) path $KLMNRST$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a overtakes three or less aircraft and all of these are on the $KLMNRST$ path **then**
    assign (fast) path $KLMNY$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a path has been allocated to every aircraft which a overtakes, and a is not a heavy aircraft **then**
    assign (shortcut) path $KLMNV$ to aircraft a.
  end if
end for each
if each there exists an aircraft a which has not had a path allocated **then**
  declare the re-sequencing to be infeasible
end if

heuristics, and the usage will differ depending upon the controller. For example, a controller may use them for aircraft with long holds (awaiting a CTOT time-slot), or those which need to be prioritised (to meet a CTOT time-slot). For this research we assume no ground controller

Algorithm 5 Path allocation heuristic for the 27R holding area, entrance G

for each aircraft a in the current group which has not had a path allocated **do**
  if a is not overtaken and does not overtake **then**
    assign path $GOPQR$ to aircraft a (see section 6.4.4 for why the default path used by the controllers is not used in this case).
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a is overtaken but does not overtake **then**
    assign (slow) path $GOPQRST$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a overtakes two or less aircraft and all of these are on the $GOPQRST$ path **then**
    assign (fast) path $GOPQR$ to aircraft a.
  end if
end for each
if each there exists an aircraft a which has not had a path allocated **then**
  declare the re-sequencing to be infeasible
end if

Algorithm 6 Path allocation heuristic for the 27L holding area, entrance $B1$

for each aircraft a in the current group which has not had a path allocated **do**
  if a is not overtaken and does not overtake **then**
    assign (default) path $B1B2B3E1F5H1H4H5$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a is overtaken but does not overtake **then**
    assign (slow) path $B1B2B3E1F5H1H4H5$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a overtakes two or less aircraft and all of these are on the $B1B2B3E1F5H1H4H5$ path **then**
    assign (fast) path $B1B2B3E1F5H1H2H3$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a path has been allocated to every aircraft which a overtakes **then**
    assign path $B1B2B3B4B5E3J1J2J3J4$ to aircraft a.
  end if
end for each
if each there exists an aircraft a which has not had a path allocated **then**
  declare the re-sequencing to be infeasible
end if

aid with the re-sequencing so these entrances were not used. The path allocation algorithms for entrances $B1$, $F1$ and $S1$ can be expressed by algorithms 6, 7 and 8 respectively.

Each of these algorithms follows the form of algorithm 2. Each entrance has a default, fast and slow path, as indicated. The default paths are $B1B2B3E1F5H1H4H5$, $F1F2F3F4F5H1H4H5$ and $S1S2S3S4S5S9$. The fast paths are $B1B2B3E1F5H1H4H5$, $F1F2F3F4F5H1H4H5$ and $S1S2S3S4S6S7S8S9$ The slow paths are $B1B2B3E1F5H1H4H5$, $F1F2F3F4F5H1H4H5$ and $S1S2S3S4S6S7S8S9$. Entrances $F1$ and $S1$ have an overtaking path available which does not

Algorithm 7 Path allocation heuristic for the 27L holding area, entrance $F1$

```
for each aircraft a in the current group which has not had a path allocated do
  if a is not overtaken and does not overtake then
    assign (default) path F1F2F3F4F5H1H4H5 to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated do
  if a is overtaken but does not overtake then
    assign (slow) path F1F2F3F4F5H1H4H5 to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated do
  if a overtakes two or less aircraft and all of these are on the F1F2F3F4F5H1H4H5 path
  then
    assign (fast) path F1F2F3F4F5H1H2H3 to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated do
  if a path has been allocated to every aircraft which a overtakes, and a is not a heavy
  aircraft then
    assign (shortcut) path F1F2F3F4G1G2 to aircraft a.
  end if
end for each
if each there exists an aircraft a which has not had a path allocated then
  declare the re-sequencing to be infeasible
end if
```

Algorithm 8 Path allocation heuristic for the 27L holding area, entrance $S1$

```
for each aircraft a in the current group which has not had a path allocated do
  if a is not overtaken and does not overtake then
    assign (default) path S1S2S3S4S5S9 to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated do
  if a is overtaken but does not overtake then
    assign (slow) path S1S2S3S4S6S7S8S9 to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated do
  if a overtakes three or less aircraft and all of these are on the S1S2S3S4S6S7S8S9 path
  then
    assign (fast) path S1S2S3S4S5S9 to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated do
  if a path has been allocated to every aircraft which a overtakes, and a is not a heavy
  aircraft then
    assign (shortcut) path S1S2 to aircraft a.
  end if
end for each
if each there exists an aircraft a which has not had a path allocated then
  declare the re-sequencing to be infeasible
end if
```

use the entire runway. The overtaking path from entrance $B1$ does allow the entire runway to be used for take-offs. The shortcut paths are $B1B2B3B4B5E3J1J2J3J4$, $F1F2F3F4G1G2$ and

$S1S2$.

6.5.3 The 09R holding area

The 09R holding area has two holding area entrances, and is shown in figure 6.6. Aircraft from terminals 1 to 3 are assumed to arrive at entrance A and aircraft from terminal 4 (south of the runway) at entrance P. This is realistic as it ensures that no runway crossings are required. The path allocation algorithm for entrance A can be expressed by algorithm 9 and for entrance P by algorithm 10.

Algorithm 9 Path allocation heuristic for the 09R holding area, entrance A

for each aircraft a in the current group which has not had a path allocated **do**
  if a is not overtaken and does not overtake **then**
    assign (default) path $ABCDEFGHIJKO$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a is overtaken but does not overtake **then**
    assign (slow) path $ABCDEFGHIJLMNO$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a overtakes three or less aircraft and all of these are on the $ABCDEFGHIJLMNO$ path **then**
    assign (fast) path $ABCDEFGHIJKO$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a path has been allocated to every aircraft which a overtakes, and a is not a heavy aircraft **then**
    assign (shortcut) path $ABCDEFG$ to aircraft a.
  end if
end for each
if each there exists an aircraft a which has not had a path allocated **then**
  declare the re-sequencing to be infeasible
end if

Each of these algorithms follows the form of algorithm 2. Each entrance has a default, fast and slow path. The default paths are $ABCDEFGHIJKO$ and $PQRSTUVWXYZ$. The fast paths are $ABCDEFGHIJKO$ and $PQRSTUVWXY$. The slow paths are $ABCDEFGHIJLMNO$ and $PQRSTUVWXYZ$. Entrance A has an overtaking path ($ABCDEFG$), but an overtaking path has not been included for entrance P. The overtaking path does not use the entire runway.

6.6 The Basic Feasibility Test

This section aims to answer the question of whether an intended take-off sequence can be attained, and if so then how it would be attained. The developed algorithm simplifies the problem by noting that there are only a limited number of decision points in real holding area structures. The generic feasibility check algorithm is described below. Holding area specific modifications

Algorithm 10 Path allocation heuristic for the 09R holding area, entrance P

for each aircraft a in the current group which has not had a path allocated **do**
  if a is not overtaken and does not overtake **then**
    assign (default) path $PQRSTUVWXYZ$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a is overtaken but does not overtake **then**
    assign (slow) path $PQRSTUVWXYZ$ to aircraft a.
  end if
end for each
for each aircraft a in the current group which has not had a path allocated **do**
  if a overtakes only a single aircraft, b, and b is on the $PQRSTUVWXYZ$ path **then**
    assign (fast) path to aircraft a.
  end if
end for each
if each there exists an aircraft a which has not had a path allocated **then**
  declare the re-sequencing to be infeasible
end if

are made to the generic solution algorithm where required to ensure that the solution method is fast even when holding areas are slightly more complex than those the algorithm can cope with normally. An enhanced version of this algorithm is presented in section 6.9, but the basic algorithm is presented first for clarity.

6.6.1 Overview of the feasibility check

The feasibility check involves testing whether the currently assigned paths allow the desired take-off order to be achieved. Once every aircraft has been assigned a path through the holding area, the next stage is to feed aircraft through the holding area graph for the holding area that is currently in use. As described in section 2.7, each node in the network can only hold one aircraft at a time and the arcs specify the valid movement for aircraft through the holding area.

Initially, if aircraft positions within the holding area are known, each aircraft that is already in the holding area is placed in the node related to its current position. If the aircraft is between nodes, it is placed at the node it will enter next. If an aircraft is currently on the taxiways, or if aircraft positions are unknown, then aircraft are placed in queues, in predicted arrival order, at the holding area entrance at which they are predicted to arrive.

Whenever an entrance node is empty, the first aircraft from that entrance queue is placed in the node and removed from the queue. Aircraft enter the entrance nodes in arrival order and exit onto the runway in the intended take-off order. The key to the feasibility check is to determine the sequence in which the aircraft should move in order to ensure aircraft do not block each other from reaching the runway in the correct take-off sequence.

6.6.2 An exact back-tracking algorithm

The basic principle of the feasibility check can be demonstrated using a simple recursive back-tracking algorithm, as illustrated by algorithm 11. This algorithm performs a depth-first search of the possible movement within the holding area, seeking a set of moves which will allow the take-off sequence to be attained. The operation of algorithm 11 can be seen to be analogous to an exhaustive evaluation of sequences of triplets in the triplet representation that was introduced in section 4.2.

Algorithm 11 Recursive back-tracking feasibility test

```
1: for each aircraft, a, in the holding area do
2:    if the next node that aircraft a should enter is empty then
3:       move a to next node
4:       if a vacated an entrance node and the entrance queue is not empty then
5:          identify the first aircraft, f, in the queue at the entrance
6:          remove f from the queue at the entrance
7:          place f within the holding area at the vacated entrance node
8:       end if
9:       if a entered the runway node then
10:         remove a from the graph
11:         if all aircraft have entered the runway then
12:            declare the re-sequencing feasible and end this recursion of the algorithm
13:         end if
14:      end if
15:      recursively call algorithm 11 to move the next aircraft
16:      if re-sequencing has been declared to be feasible then
17:         end this recursion of the algorithm
18:      end if
19:      undo any changes that were made in this iteration:
20:      if a entered the runway node then
21:         replace a in the graph
22:      end if
23:      if a vacated an entrance node and the entrance queue is not empty then
24:         if an aircraft f was moved into the entrance node from the queue then
25:            remove f from the entrance node of the holding area
26:            place f back into the front of the queue at the entrance
27:         end if
28:      end if
29:      move a back to its previous node
30:   end if
31: end for each
32: end this recursion of the algorithm
```

Steps *1* and *2* of algorithm 11 test each aircraft that could potentially move and evaluate the effects of moving it. The aircraft is moved in step *3*. Each iteration of the loop in step *1* considers the movement of a single aircraft forward a single node. (The movement of further aircraft is considered using the recursion in step *15*.) Steps *4* to *8* ensure that any entrance node that was vacated is filled with the next aircraft. Steps *9* to *14* remove the moving aircraft if it reaches the runway. Step *15* uses a recursive call to the same algorithm to evaluate the effects of further moves. Steps *16* to *18* ensure that the algorithm terminates as soon as a feasible solution is found. Steps *19* and onwards undo the changes that were made by moving this aircraft, so that the effects of moving a different aircraft instead can be evaluated.

Problems with algorithm 11

There are a number of problems with using algorithm 11 for the feasibility test. Firstly, this recursive version of the algorithm may have to search an extremely large number of possible sequences of movement within the holding area in order to find a sequence which achieves the desired take-off sequence. This can make it extremely slow. The greatest problem with this occurs in cases where the sequencing is infeasible. In this case the algorithm has to consider every possible sequence of movement and eliminate each one in turn before it will determine that the re-sequencing is not possible. A faster method for performing the feasibility check was required.

Additional problems with algorithm 11 compared with the implemented deterministic algorithm, discussed in section 6.6.3, are that it has no preference for solutions which are practical in terms of how long aircraft are delayed for, and has no facility for determining an earliest take-off time based upon the holding area movement. The ways in which these important elements are handled by the designed solution algorithm are explained in the following sections. Although these elements could be added to algorithm 11, doing so would greatly increase the size of the search space (for example by requiring a decision about whether to increase the current time or not in each recursion) which would make the algorithm even slower.

6.6.3 A deterministic version of the feasibility check

Algorithm 12 Basic feasibility test

```
while aircraft remain in the graph do
  clear movement flag, m
  for each aircraft, a, in the holding area do
    if a can move to the next node without preventing the re-sequencing then
      move a to next node
      set movement flag, m
      if a vacated an entrance node and the entrance queue is not empty then
        identify the first aircraft, f, in the queue at the entrance
        remove f from the queue at the entrance
        place f within the holding area at the vacated entrance node
      end if
    end if
  end for each
  if flag m has not been set then
    no aircraft moved so declare re-sequencing to be infeasible
    end algorithm
  end if
end while
declare re-sequencing as feasible as it has been achieved
end algorithm
```

Due to the problems with the simple recursive algorithm, a deterministic algorithm was developed. Rather than try every possibility of moving an aircraft, this algorithm instead attempts to predict whether later problems will be caused by moving the aircraft, and to prevent the move when this is the case.

The basic feasibility check algorithm that was used in this research can be expressed by algorithm 12. A more in-depth study of the algorithm is presented in section 6.9.1, where additional time-based features of the algorithm are added and explained.

Algorithm 12 is deterministic, with no backtracking to undo bad moves, in order to aid the execution speed. At any point in time, if no aircraft can move then the take-off sequence is declared infeasible and is rejected. This no backtracking method is only possible when there is a method to determine whether an aircraft can move at each point, so that moves do not need to be reversed. Statement 4 in algorithm 12 represents the complex test which has to be performed in order to determine whether this is the case. This test is detailed in section 6.8 and is summarised by the following rules:

1. Aircraft must enter the runway node in the desired take-off sequence.
2. Aircraft can only enter the next node if it is empty.
3. Aircraft cannot enter a node if doing so would prevent another aircraft from entering the runway in the required take-off sequence.

The third rule is the only one that is difficult to resolve. Determining whether moving an aircraft will block another aircraft is not always easy for a generic graph as this is equivalent to determining the validity of selecting an arc in the feasibility problem for the alternative graph, discussed in section 4.3.

6.6.4 A heuristic approach

Using a heuristic rather than exact approach, it is possible to perform this test quickly for the real holding areas. In fact, it will also be seen later that the implementation of the heuristic approach has an important role to perform in ensuring that the re-sequencing can be performed without excessive delay for the aircraft involved. This heuristic method does not guarantee to find a feasible schedule if there is one, but the feasible schedules that it misses are very unlikely to be useful in practice, as discussed in section 6.8.

The selected approach detects infeasible re-sequencing very quickly. As soon as no aircraft can move, the sequence is declared to be infeasible. This is very important and is an extremely valuable property of the method as, usually, many of the evaluated sequences are infeasible and this is where an exact approach often has the most difficulty. If the sequence is feasible, an exact feasibility test can stop as soon as the feasible sequencing is found but when the sequence is infeasible an exact approach has to rule out all possible sequencing before it can determine this fact.

No temporal aspects have been explicitly mentioned so far in the feasibility test. For example, no measurement of the amount of time aircraft have to spend waiting for other aircraft to move has been made. With the approach described so far, it is actually possible for re-sequencing to be theoretically achievable but for aircraft to be delayed so long to achieve the re-sequencing that target take-off times would not be achieved. The heuristic approach to movement prevents this from happening as it prevents lower priority aircraft from advancing in front of higher priority aircraft unless there is a way for them to immediately move out of the way. The sequences which an exact method would determine to be feasible (but this method evaluates as infeasible) are those which had the possibility of delaying aircraft such that the achievement of take-off times may have been put in jeopardy.

6.6.5 Results of the feasibility check

By the end of the feasibility check, not only are the traversal paths known but the sequence in which aircraft pass each node in the holding area will also have been determined. This sequencing is important as it allows a comparison to be made between what happens with controllers in control and what the decision support system assumes will happen. When running with the simulation described in chapter 7, this allows an evaluation of how realistic the re-sequencing really is, and allows the simulation to predict holding area positions for aircraft.

6.7 A Comparison With Previous Approaches

The designed decision support system can be compared with the previous ground movement research since the holding area movement problem considered in this thesis can be considered to be a limited size ground movement problem. The similarities and differences between the designed path allocation and feasibility check aspects are considered below.

6.7.1 Differences in path allocation methods

The decision support system design that is presented in this thesis utilises the fact that there are strong preferences for specific paths through the holding area. In contrast, previous ground movement research has often performed the path allocation implicitly within the optimisation, [56, 96, 94, 127, 143, 169].

The primary danger of assigning paths implicitly is that it is important to ensure that excessively long paths are avoided. For this reason, it is common to pre-determine a set of preferred paths for each aircraft, and limit the search to utilise only these paths in order to limit the unnecessary travel distance that is assigned to any specific aircraft [56, 96, 94, 127, 143, 169]. In some cases, [157], the path to allocate was uniquely determined by the start and end points of the movement, thus ensuring that all aircraft were assigned the preferred paths, but possibly limiting the flexibility of the movement.

The system described in this thesis can be considered to have similarities to both approaches. By knowing in advance the re-sequencing that is required, it is possible to ensure that the preferred path is used whenever possible, which is similar to the approach in [157], but the flexibility to use longer paths (within limits) when required gives similar benefits to those of the more flexible approaches such as [56, 96, 94, 127, 143, 169].

6.7.2 The sequence of solving the problems is reversed

In previous research such as [56, 94, 96, 127, 143, 169], the paths were allocated based upon the timing of movement rather than upon the requirement to achieve a specific take-off sequence. In contrast, since the holding area movement algorithm in this thesis is executed within the search for take-off sequences, there will always be a pre-determined target take-off sequence. The order of performing the path allocation and take-off sequencing operations can be considered to be reversed between the two approaches. In [56, 94, 96, 127, 143, 169], the allocated path determined the possible take-off sequences. In this research, the take-off sequences determine the paths to allocate.

6.7.3 Differences in objectives

The decision support system described here has an emphasis upon the take-off sequencing rather than the control problem. This is important since the sequencing problem is the vital element which will affect the delay. This reflects controller working practices. The allocation of longer paths to aircraft is acceptable as long as it can be justified from the point of view of the re-sequencing that is required. For example, it is acceptable in the problem considered in this research to assign longer paths to aircraft which must be overtaken, to hold them out of the way in order to achieve a better take-off sequence, but this increase in taxi time would be a problem for any system which had an objective to minimise the taxi time.

As long as the required re-sequencing can be performed, and aircraft can reach the runway early enough that they do not delay the take-offs, the method in which the control problem is handled will have no effect upon the take-off sequencing. However, the fact that many good take-off sequences cannot be attained (as shown by the effects of the constraints, in section 8.5), means that the control problem does have an effect upon the sequencing problem so cannot be handled separately. The closest objective in the previous research to the way in which the control problem is handled here is where the objective for any ground movement is for all aircraft to reach the runway by pre-determined take-off times, although the feedback from the control problem to the sequencing problem in the system described in this thesis (via the earliest take-off time constraint) means that even this is not an exact match. Since most of the previous ground movement research, for example [81, 127, 143, 157, 158, 169], considers the problem of minimising the time between push-back and reaching the runway, or considers only very simple wake vortex separations, the objectives can be seen to differ between this research and the previous ground movement research.

6.7.4 Differences in level of detail

Previous ground movement research often considered a far more detailed model of the ground movement. For example, the model in [94] considered turning angle and traversal difficulty when determining the value of an arc, while the level of detail used in TAAM models (see section 3.8.3) is often even higher. These models suffer from the problems of requiring detailed calibration [8], that were discussed in sections 3.8.3 and 3.8.5. This level of detail was unnecessary for this research.

In addition to the solution space for the holding area problem being smaller than for the ground movement around the entire airport, there is a consequently shorter taxi time available to traverse the holding area. With comparatively shorter times, the traversal time predictions become even more sensitive to any errors and any prediction is far less likely to be accurate. Consequently, it is not practical to build a system which relies upon accurate taxi times for movement over such short distances. Rather than attempting to determine the shortest or expected time to traverse the arcs or nodes on the allocated path, the designed system instead ensures that aircraft will have plenty of time to traverse it.

6.8 Limitations Upon When An Aircraft May Move

When performing the feasibility check, it is important to be able to determine whether an aircraft can safely enter the next node on its path without blocking another aircraft from reaching the runway in time for take-off. A fast method for making this determination is explained in this section.

6.8.1 Movement restrictions

The key to the movement restrictions is determining whether moving a given aircraft, A, will prevent another aircraft, B, from entering the runway in the required position in the take-off sequence. The relationship between aircraft A and B determines how this can be ensured.

Any aircraft can freely enter any node which has only one incoming arc as no other aircraft can possibly enter the node before this one can. The sequencing decisions have to be made at nodes of convergence, where more than one traversal path converges on a single node. Using this fact, the number of times that movement blocking rules have to be applied can be substantially reduced, as they only need to be checked for aircraft which are about to enter a node of convergence.

Furthermore, given the earlier definition of a path suffix, it is obvious that, where two or more paths do not diverge beyond a node, aircraft on these paths have the same path suffix at that node and there is no possibility for them to overtake each other beyond that point. The aircraft on the same path suffix at a node must pass the node in that order relative to each other. The sequencing problem at a node is therefore a case of interleaving the partial take-off sequences formed by the aircraft on each path suffix through that node.

At any specific node of convergence, the procedure for determining whether an aircraft will block another aircraft can be divided into the consideration of a number of cases based upon the path suffixes that the aircraft have at that node. The rules for avoiding the blocking of other aircraft can then be categorised as follows, in increasing level of difficulty:

1. Do not block another aircraft on the same path suffix. An aircraft can only enter a node if it is the next aircraft on its path suffix, as described in section 6.8.3.

2. Do not block another aircraft passing this node which is on a different path suffix. This is a more complicated proposal and the method is explained in section 6.8.4. As will be seen in section 6.8.5, the application of these rules is why the feasibility check is heuristic rather than exact in nature.

3. Do not delay another lower priority aircraft from entering this node if this aircraft will in turn delay a higher priority aircraft that may not itself pass this node. This is far more complicated and requires the application of holding area specific rules. Some of these are discussed in section 6.8.7.

6.8.2 Holding area graph categorisation

A holding area problem can be characterised by considering the set of paths that are available and the sequencing of the nodes of convergence and divergence on each path. The most complex sequence found can be used to characterise the graph. This characterisation of the graph determines the difficulty that will be experienced in solving the holding area problem and is useful as this understanding may highlight ways to simplify the problem or to treat it as a simpler problem with special cases.

If multiple nodes of divergence follow each other, they can be considered in the same way as a single node of divergence. Similarly, nodes of divergence prior to the first node of convergence can be ignored. In other words, the graph can be categorised according to the number of nodes of convergence and whether there are nodes of divergence between them or after the last one.

Convergence graphs

As an example, the graph in figure 6.12 can be characterised as a convergence graph as that is the most complex path through it. It is actually a divergence-convergence graph but the initial divergence can be ignored. This graph is simple to handle as there is only one suffix at node G so aircraft can only enter node G in take-off order.

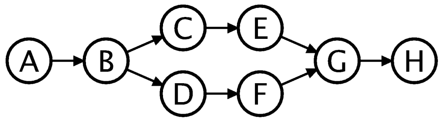


FIGURE 6.12: Sample convergence graph

Convergence-divergence graphs

Figure 6.13 shows a sample convergence-divergence graph. Due to the divergence at node D there are two path suffixes at the node of convergence C. The sequencing of aircraft entering node C must ensure that aircraft on the other suffix are not blocked from entering the runway in the correct sequence.

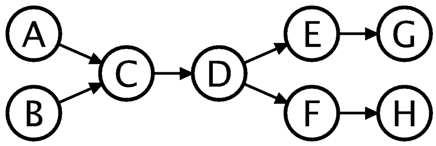


FIGURE 6.13: Sample convergence-divergence graph

More complicated graphs

Special holding area specific rules are required when the graph is more complicated than the convergence-divergence graph. In this case, as much of the graph as possible is handled using the general rules and the exceptions to the general rules are handled individually. Examples of this will be seen below.

For example, two extra rules are needed for the 27R holding area network. The first is to account for the final convergence at node Y and the second is to account for the initial convergence at node K. With these convergences present, the graph is a Convergence-Divergence-Convergence-Divergence-Convergence graph, as can be seen from the path HIJKLMNVYZ, for example. Without the convergences at K and Y it is a standard Convergence-Divergence graph so can be handled easily.

The solution method presented in this thesis works so quickly because the majority of holding areas are relatively simple in terms of the graph categorisation and can thus be solved relatively simply using path suffix information.

6.8.3 Blocking aircraft on the same suffix

The feasibility check for any graph which can be characterised as a convergence graph can be solved using only the partial sequences created for the path suffixes. In this case, the sequence in which aircraft should pass the node is pre-calculated according to the desired take-off sequence.

An ordered list of aircraft is maintained for each suffix for each node. An aircraft can only move into the next node if it is at the top of the list of expected aircraft for its suffix at that node. Once an aircraft is moved into the node, it is removed from the expected aircraft list so that the next aircraft can move into the node. The method by which the lists are created for the path suffixes is described in section 6.8.6.

The 09R and 09L holding area problems can be solved using only this approach. However, this approach is not sufficient for the more complex 27R and 27L holding areas.

6.8.4 Blocking aircraft on other suffixes

For the reasons that will be explained in section 6.8.5, using the definition of take-off priority given in section 6.3.3, an aircraft is only permitted to enter a node in front of a higher priority aircraft on another path suffix if there is, at that time, room for it to immediately exit the node to get out of the way of the higher priority aircraft.

Compliance to this condition can be verified by first pre-calculating, for each path suffix for each node, a count of later nodes that are not shared with each of the other path suffixes at that node, then calculating how many of these nodes are occupied. This count is here named the *'not-shared'* nodes count. For example, in the 27R graph shown in figure 6.2, let the runway node be denoted by the letter Z. Node V has three path suffixes: Z, YZ and XYZ. R has no nodes that are not shared by YZ and no nodes that are not shared by XYZ. YZ has one node that is not shared by Z and no nodes that are not shared by XYZ. XYZ has two nodes that are not shared by Z and one that is not shared by YZ. These counts can be easily and automatically calculated off-line for each holding area structure.

In the absence of later nodes of convergence, the number of nodes an aircraft can move into in order to move out of the way of a higher priority aircraft is the number of nodes which are on the path suffix for that aircraft but not the suffix of the higher priority aircraft. This is the not-shared node count described above. The number of such nodes that are currently occupied can easily be determined by tracking how many aircraft on this suffix have already passed the node but not yet taken off. By subtracting the number of such aircraft from the not-shared node count, the number of available nodes can be obtained. The lower priority aircraft is then only permitted to enter the node if there is at least one free node available to move into.

A list is maintained of the partial take-off sequence for each suffix at each node of convergence. The highest priority aircraft remaining on each suffix will always be the first aircraft remaining in the partial sequence. The algorithm counts the number of aircraft on this suffix that have passed this node and have not yet taken off, then adds one for the aircraft that currently wishes to move. If this number is greater than the number of not-shared nodes, then this aircraft cannot move and is considered blocked. If the node has more than two suffixes, then this comparison needs to be performed against each of the other suffixes in turn.

Where there is a later node of convergence, the check must be more complicated. Aircraft on the converging suffix may occupy some of the not-shared nodes, leading to inaccuracies in the count of occupied nodes. Special rules are necessary to determine which nodes are occupied by aircraft from which suffix in this case, as will be seen below.

6.8.5 Reasons for the stronger restriction on movement

A movement restriction is applied so that a lower priority aircraft can only enter a node ahead of a higher priority aircraft on a different path suffix if there is room for the lower priority aircraft to immediately move out of the way again. In fact, an exact feasibility test need only ensure that the lower priority aircraft can move out of the way before the higher priority aircraft needs to enter the node rather than that it can immediately vacate the node. The stronger restriction implemented here is useful for ensuring that aircraft are not delayed excessively, as well as for simplifying the feasibility check significantly. Moving out in front of a higher priority aircraft risks adding a delay to it, but ensuring an immediate exit is available ensures that any delay for the other aircraft is minimal. Without this restriction, there is no guarantee of how long it will be before the aircraft can move out of the way; only that it will be possible before the blocked higher priority aircraft is the first in the take-off queue.

6.8.6 Ensuring a short search time

Appropriate use of data structures ensures that the blocking checks are extremely fast and usually involve a comparison of very few values to verify whether each aircraft can move. If there is only one input arc to a node, then there is never any decision about which aircraft should enter it next as it will always be the same as the aircraft which is the next to enter the preceding node.

Further reduction in pre-calculation work and execution time can be obtained by only considering the path suffixes for the paths that are actually being allocated at the time, and removing any arcs from the holding area graph which are not currently in use. For example, if the path in the 27R holding area which is equivalent to the use of arc NV in figure 6.2 is not being used, then this arc can be eliminated from the graph. Node V is then no longer a node of convergence, so no check has to be made to enter it. In practice, arc NV is only used for the shortcut path with the suffix NVMZ, so is often not used in the problems being solved.

Off-line pre-processing for path suffixes

Two levels of pre-processing are used to ensure a swift feasibility check. The first is based purely upon the holding area structure and the possible paths that could be used. This stage is performed off-line for each holding area graph. The aim is to identify nodes where possible contention may exist, and to identify any convergence of paths through the graph.

For each node of convergence, the possible path suffixes are noted, and the paths which use each suffix are identified. Where paths diverge, a count is made of how many nodes are on each path suffix beyond that node that are not on each other path suffix through the node. This is the not-shared node count referred to earlier.

Different modes of operation are possible, to cope with different circumstances or different controller preferences. Each will be associated with a set of traversal paths and a traversal path allocation heuristic. For example, if node Y in the 27R holding area is blocked by some aircraft which is unable to move for a while for some reason, it may be necessary to route aircraft differently. Where there are different modes of operation, the pre-processing is performed for each possible mode of operation and the counts and information about suffixes and nodes of convergence should be cached for each.

As this information is calculated off-line, the calculation need not be fast, although in practice it is. This is a one-off calculation that needs to be performed once per combination of paths on the graph rather than once per take-off sequence. The pre-processing can be quickly performed using algorithm 13.

Algorithm 13 Determine suffixes and nodes of convergence

```
for each path p through the holding area do
  for each node n on path p do
    determine the path suffix s for the path p at node n
    if path suffix s has not been stored for node n then
      store the suffix s against node n
    end if
    record the fact that path p uses suffix s at node n
    determine the node m which path p passes immediately prior to node n
    if no previous node has been recorded for n then
      record node m as the previous node for node m
    else
      if the previous node for node n is not node m then
        record the fact that node n is a node of convergence
      end if
    end if
  end for each
end for each
```

At the end of this process each node has a set of one or more path suffixes associated with it and each path suffix has one or more paths associated with it. Furthermore, each node which is a node of convergence under the current selection of paths has been labelled as such. Once this has been performed a second stage, described by algorithm 14, is used to count the not-free nodes.

On-line pre-processing stage

The second pre-processing stage requires knowledge of the desired take-off sequence so must be performed separately for each feasibility check. The purpose of this stage is to calculate partial take-off sequences for the path suffixes at each node. These can be calculated using algorithm 15, using the results of the first pre-processing stage.

The lists so created will be in take-off order (as the aircraft were considered in take-off

Algorithm 14 Determine not-shared node counts

for each node n **do**
    for each ordered pair of suffixes s_1 and s_2 at node n **do**
        clear the count of not-shared nodes for s_1 with respect to s_2
        for each node m on suffix s_1 **do**
            if m is not on suffix s_2 **then**
                increment the the count of not-shared nodes for s_1 with respect to s_2
            end if
        end for each
    end for each
end for each

order in step 1) so will form a partial sequencing for the aircraft which pass that node. Step 3 is extremely fast due to the pre-calculation performed by algorithm 13.

Algorithm 15 Create partial take-off sequences

1: **for each** aircraft a, in take-off order **do**
2:     **for each** node n on the path p assigned to aircraft a **do**
3:         identify the suffix s for path p (pre-calculated from algorithm 13)
4:         add aircraft a to the aircraft list for suffix s at node n
5:     **end for each**
6: **end for each**

6.8.7 More complicated rules

The rules given in sections 6.8.3 and 6.8.4 form the basis for the movement restriction rules. The special rules described below provide methods to handle more complicated graphs, either by reducing the complexity to that of the convergence-divergence graph (for example by handling earlier nodes of convergence separately) or by providing special ways of handling additional later nodes of convergence.

The exceptions and special rules can be summarised as follows:

1. The sequence in which aircraft pass a node can be pre-determined. For example to enforce a first-come-first-served sequence.
2. The count of occupied nodes on a suffix can be altered, for instance to account for occupation by aircraft on other suffixes from a later path convergence.
3. Custom rules can be provided to prioritise the sequence in which aircraft pass a node, for example by implementing a movement level.
4. Custom handling is required for bi-directional arcs.

The holding area graph can sometimes be designed in a way which will aid in reducing the movement complexity. For example, the feasibility of movement is often easier to verify if there are fewer nodes of convergence so combining multiple nodes into one will simplify the problem.

6.8.8 Eliminating early nodes of convergence

If early nodes of convergence can be eliminated, it is often possible to significantly reduce the complexity of a graph. There may be simple ways to handle these early nodes.

In some cases, for example, where the early nodes actually represent positions on the taxiways (as node K does in the 27R graph), a first-come-first-served sequencing is advisable at the node, as anything else would mean aircraft waiting (blocking taxiways) for other aircraft to pass. Any fixed sequencing can be enforced within the existing system by appropriately modifying the partial take-off sequences; for example, by specifying a single partial sequence which all aircraft must obey. Modifying sequences in this way means that no extra complexity is required within the movement check itself.

In the 27L graph, nodes B4 and E2 are usually used as remote holds. If using them in a west to east direction (for example E1-E2-E3), it is simple to determine when to release aircraft as there is only one path suffix at both E3 and B5 so the sequencing can be uniquely determined. If using the holding nodes in an east to west direction, then there is a question about how to fit the held aircraft into the sequence of passing aircraft. This is usually quite simple; for example, the aircraft must enter in the correct position for its own path suffix and must not block another suffix, as usual. Furthermore, within these suffix list constraints, it is better to hold the aircraft until all higher priority aircraft have passed the node of convergence, if this is possible, so that this aircraft cannot possibly block these aircraft. The movement level system described in section 6.8.10 could be used for this if necessary.

6.8.9 Eliminating later nodes of convergence

As mentioned earlier, the rules for blocking other suffixes rely upon having no later nodes of convergence and no more than one later node of divergence. If these later nodes of convergence and divergence can be handled, they can be eliminated from consideration thus simplifying the problem. The number of nodes that are being used by aircraft on other suffixes must be added to the number used by aircraft on this path suffix before comparing this value to the not-shared node count.

This problem can occur in the 27R holding area when considering whether an aircraft can enter node V along a path ending UVXYZ without blocking an aircraft along a path ending with MNVZ. It is important to consider the fact that node Y may be blocked by an aircraft along a path ending MNYZ so may not be available to allow the aircraft to move out of the way again.

In this example there is a simple method that can be used, as there is only a single node, Y, that is shared by other converging paths. As there is only one path suffix at node Y, the next aircraft to pass the node will always be known. It is then simple to determine whether the next aircraft to use the node is on the same suffix as the aircraft waiting. If so, it will have already been included in the count of occupied nodes. If not, then the count must be increased, as there will be one less node available to move to. Handling the convergence at Y separately simplifies the problem considerably and means that the method in section 6.8.4 can be used to handle node V.

A generalisation of this case for n nodes is more complicated but could work backwards from the runway predicting the next occupant for each of the nodes in turn. This will work as long as the nodes have a single path suffix as the next aircraft can then be uniquely determined. The number of nodes which would be occupied by aircraft from other suffixes should be taken from the number of available nodes when determining the feasibility of an aircraft entering the node of convergence. However, this was not necessary for the Heathrow holding area graphs.

6.8.10 Prioritisation at nodes

Three-way contention problems can sometimes occur. Prioritisation rules can be used to resolve many of these problems. For example, it may be the case that there will never be a problem if aircraft from one path, p_1, go first, but there may be if aircraft from the other path, p_2, go first. In this case, prioritising the aircraft on path p_1 will ensure that the problem cannot occur.

This prioritisation can be ensured by using a movement level. At problem nodes, a minimum movement level can be applied to a path, such that aircraft cannot enter the node along that path until the movement level has been increased high enough. All of the aircraft on the safe paths will therefore be able to enter the node freely. As soon as movement is blocked, the movement level is increased so that aircraft on more risky paths can enter the node. The full implementation of this method is explained in algorithm 16. It should be noted that excessive use of the movement level system can reduce the speed of the feasibility check so it is better to avoid it where possible.

An example of the problem occurs at node V of the 27R holding area. Aircraft along paths from node N are prioritised over those from path U as all aircraft from node U pass through node V, so the blocking of any aircraft along these paths will be considered by aircraft entering from node N using the standard path suffix system. On the other hand, not all aircraft passing node N will pass node V, so would not be considered in the suffix blocking rules. In particular, a high priority aircraft could be queued behind the aircraft waiting at N, implying that the aircraft at N should move out of the way, but the suffix priority rules at V will not consider this.

Prioritisation of paths may delay aircraft on the lower priority paths, however the introduction of the current time variable, described in section 6.9 ensures that this delay is limited and that aircraft will only have to wait for other aircraft that are already in the holding area rather than waiting indefinitely.

6.8.11 Handling specific problem conditions

An alternative, less generic, approach to the movement level can be used if a specific condition is known to cause a problem for a specific holding area. For example, if a problem would only occur in moving aircraft a into node N when a higher priority aircraft on a different path p had not yet passed node N, then a custom check could be made against the path suffix for path p at node N to see whether the next aircraft on the suffix had a higher priority than aircraft a. This can be performed very fast as the pre-processing has already identified the path suffix for each path at each node, and created a list of the aircraft on the path suffix in take-off order (i.e. the highest priority remaining aircraft will be at the front of the list for the suffix). When this specific problem is known then it may be much faster to check the specific conditions rather than using a generic movement level.

6.8.12 Bi-directional arcs

Bi-directional arcs, such as those which exist in the 27L holding area, can cause a problem as aircraft could wait on both ends of the arc to traverse it in opposite directions and block each other. This would results in a deadlock situation and the movement would be declared infeasible. There are a number of ways in which bi-directional arcs can be handled.

Firstly, it may be the case that the arcs are not traversed in both directions at once, for example, the usual usage of the 27L graph is to use nodes B4 and E2 as remote holding areas. It may be possible to then define two or more different modes of operation and create separate holding area graphs for each mode, as explained in section 6.2.2. The arcs in each graph can then be treated as being uni-directional and the problem has been eliminated.

The second solution method is to design the holding area graph so that these arcs do not exist, for example by combining multiple adjacent nodes into a single combined node. This may be possible, but there are two problems which must be resolved. Firstly, the loss of capacity of the holding area graph, as combining nodes means that there are less available in which to hold aircraft. Secondly, this can not be done if it would apply constraints between paths which would not normally exist. For example, in the 27L holding area, combining nodes F4 and F5 would apply a relationship between aircraft following the E1-F5-H1 path and those on the F3-F4-G1 path, which are actually independent in practice.

A third solution method could consider the nodes on the other side of the bi-directional arcs before allowing movement into a node on one side of it. A check is performed upon attempting to enter the first node and entry is only allowed if a deadlock situation will be avoided. The condition that will prevent the aircraft entering the node can differ. The simplest possibility is to prevent entrance to one node if the other node is currently occupied. This effectively treats both nodes as a single node, as dual occupancy of these nodes is prevented, but has the same holding area reduction problem as combining the two nodes into a single node has. A better approach could check the node at the other end of the arc whenever an aircraft wishes to enter one end of the arc, to verify that both aircraft do not intend to traverse the arc and prevent the second from entering if they do. i.e. to explicitly avoid the deadlock condition. The problem with each case is that the path suffixes are not combined, so the check is less effective than the true combination of nodes as blocking of other suffixes will not be considered.

There is really no ideal way to handle this possible deadlock situation quickly. The best way is to remove the bi-directional arcs where possible, even if this means combining nodes, as long as the holding area size reduction is not sufficient to cause a problem. More complicated approaches could look ahead to avoid problems, but the more complicated look-ahead risks slowing down the system to the point where it may be better just to allow the system to backtrack if it finds problems.

In the experiments performed for the thesis, the path allocation method was chosen so that the arcs became uni-directional. If this had not been the case, the situation in the 27L holding area would have been handled by adding an arc from F5 to G1, and making the bi-directional arc between F4 and F5 unidirectional from F4 to F5. Node F3 would then have been eliminated and the situation where an aircraft went from F5 to G1 while another was at F4 awaiting a move to G1 would be assumed to represent the aircraft at F4 having held at the (removed) F3 node while the other aircraft passed in front of it. This would complicate the graph somewhat (and is not ideal as a node is lost on the path F1-F2-F3-F4-F5-G1-G2) but this allows the feasibility check to be performed much more quickly and easily than if the deadlock situation had to be considered. The method described in section 6.8.9 could then have been used to eliminate the new node of convergence that would be introduced at G1.

A generalisation of the bi-directional arc problem can be seen when there are larger cycles in the graph. Any cycle raises the possibility of deadlock and the bi-directional arc is merely an example of a two-node cycle. Larger cycles are unlikely in practice due to the fact that all aircraft will usually be trying to go in roughly the same direction, i.e. from the taxiways to the end of the runway. If other cycles are present in the graphs, they would need to be resolved in a manner similar to the bi-directional arcs.

6.8.13 Summary of movement restrictions

An aircraft can enter the next node if it is not a node of convergence, or all of the following conditions hold:

1. It is the next aircraft on the partial take-off sequence for its own path suffix.
2. Either no other path suffix has a higher priority aircraft waiting to pass the node, or the not-free node count and the count of occupied nodes show that there is room for this aircraft to immediately move out of the way of the other aircraft.
3. Path prioritisation rules (for example a movement level) are obeyed.

4. Any pre-conditions are met, for example preventing an aircraft moving until specific other aircraft have passed the node.

5. Any special movement restrictions such as for bi-directional arcs have been obeyed.

Rather than perform an exact feasibility check to determine whether the aircraft could be re-sequenced, this method uses a strengthened movement constraint to ensure that large delays do not occur for aircraft waiting for lower priority aircraft to move. The method will detect infeasible re-sequencing as being infeasible, and will do so very quickly. It will also sometimes declare feasible re-sequencing to be infeasible, but only where the re-sequencing required to attain the sequence is complex or time-consuming. The removal of the necessity to backtrack in the feasibility check helps to ensure that it can be performed extremely quickly. As the holding area complexity increases, it becomes more and more difficult to do this, but the real holding areas are simple enough to enable this approach.

6.9 The Enhanced Feasibility Test

The feasibility check for movement was presented in section 6.6 and the rules for allowing movement or not were presented in section 6.8. The described test implicitly includes most of the time based constraints but there are cases which are not covered. The enhanced feasibility check which is actually used is presented in this section.

A modification has been made to the feasibility check to include the current time variable, explained below. The current time is used to determine the earliest time, f_i, at which an aircraft, i, can reach the runway. This variable explicitly ensures that any delay needed to perform the re-sequencing will be taken account of in the take-off times.

A second facility performed by the feasibility check is to predict holding area positions of aircraft. This facility is used by the simulation described in the next chapter when it takes a snapshot of the positions of aircraft during the feasibility test and then uses this to predict the positions that aircraft will have in future.

6.9.1 Modified Feasibility Test

The enhanced feasibility check is performed to evaluate each suggested take-off sequence and is initialised in the same way as the basic check. Aircraft which are already within the holding area are placed at the node representing their current physical location. Aircraft on the taxiways are placed in virtual queues at the holding area entrances. The feasibility test performed is given by algorithm 16, which is an extension of algorithm 12.

Firstly, steps 4 to 23 are repeated to move as many aircraft as possible through the graph. Once no more movement is achievable, some of the movement restrictions are slackened and the steps are repeated. If movement is not achievable even after the restrictions have been lifted, the re-sequencing is declared to be infeasible.

Algorithm 16 Advanced feasibility test

```
1: set current time, c to start time of feasibility check
2: set movement level, l to 1
3: clear the movement level block flag, b
4: clear movement flag, m
5: for each aircraft, a, in the holding area do
6:   if a can move to the next node without preventing the re-sequencing then
7:     if movement level restrictions prevent movement then
8:       set the movement level block flag, b
9:     else
10:      move a to next node
11:      set movement flag, m
12:      clear movement level block flag, b
13:      set movement level, l to 1
14:      if a vacated an entrance node and the entrance queue is not empty then
15:        identify the first aircraft, f, in the queue
16:        if aircraft f is predicted to arrive at the holding area no later than time c then
17:          remove f from the queue at the entrance
18:          place f within the holding area at the vacated entrance node
19:        end if
20:      end if
21:    end if
22:  end if
23: end for each
24: if all aircraft have left the holding area and reached the runway then
25:   declare re-sequencing as feasible as it has been achieved
26:   end algorithm
27: end if
28: if the movement flag, m, was set then
29:   return to step 4
30: end if
31: if the movement level block flag, b, was set then
32:   increment the movement level, l = l + 1
33:   return to step 3
34: end if
35: if any holding area entrance node is empty and the queue has aircraft then
36:   identify the earliest arriving aircraft, f, that is queued for an empty entrance node
37:   increase the time c to move the feasibility check later in time (see explanation below)
38:   for each empty holding area entrance node do
39:     identify the first aircraft, f, in the queue at this entrance
40:     if aircraft f is predicted to arrive at the holding area no later than time c then
41:       remove f from the queue at the entrance
42:       place f within the holding area at the empty entrance node
43:     end if
44:   end for each
45:   return to step 2
46: else
47:   declare the re-sequencing infeasible
48:   End algorithm
49: end if
```

The arrival of new aircraft at the holding area is now limited by a current time value, detailed in section 6.9.2 which acts to introduce a time-based element into the feasibility check. The aim is to know which aircraft would actually be in the holding area at the time, in order to avoid any delay waiting for a later arrival to reach the holding area.

The concept of a movement level was introduced in section 6.8.10. Whenever a movement is blocked by a rule which depends upon the current movement level, this flag is set. If all movement is blocked, this flag will cause the movement level to be increased before another attempt is made to move the aircraft. Whenever an aircraft is actually moved within the holding area graph, this flag is reset and the movement level is reset to level 1.

Steps 28 to 30 ensure that the movement is repeated while aircraft can still move. Once aircraft stop moving, steps 31 to 34 increase the movement level if this was preventing the movement, making more aircraft available to move. When movement stops again steps 35 to 45 advance the time, allowing more aircraft to reach the holding area and enter the system. If even this fails to restore movement then step 47 declares the re-sequencing to be infeasible.

When the current time is advanced, in step 37, the amount by which it is advanced is optional. It could be advanced in one second or one minute increments, or to the time that the next aircraft, f, arrives, or to another value such as the time f arrives rounded up to the nearest minute. It is possible that the advancement of the time will allow multiple aircraft to enter the holding area. One minute intervals were used in the experiments performed for this thesis.

The conditions under which an aircraft is permitted to move to the next node on its path were initially the same as explained in section 6.8, so are not repeated here. However, a weaker version of the movement restrictions is sometimes used by the simulation when predicting the current positions of aircraft, as described in the next chapter.

6.9.2 The current time value

A current test time value is maintained during the feasibility check and is used to limit how early aircraft are released from the entrance queues into the feasibility check. This ensures that the system considers only aircraft that would actually be in the holding area at that time, and has two benefits. Firstly, it allows a better earliest take-off time to be predicted for aircraft, as explained below. Secondly, it reduces the complexity of the feasibility check by limiting the aircraft which need to be considered in each iteration.

As the limitation is only upon release into the feasibility check and not upon exiting onto the runway, aircraft will leave the holding area graph as soon as the entrance is clear. As the take-off times cannot be calculated prior to the feasibility check (the earliest take-off time depends upon the holding area movement) it is not practical to limit the time at which aircraft leave the holding area. If desired, an estimation of the take-off time could be used to limit the removal from the holding area, but there are desirable side-effects from removing aircraft as quickly as possible. One effect can be see in the later stages of the feasibility check. Each time the current time is advanced, the only aircraft remaining in the holding area will be those which are just being added, those which are overtaken by the aircraft that have just arrived or those which cannot perform their holding area movement until one of these arriving aircraft has performed some necessary movement. This provides a reduction in the aircraft under consideration for movement at any time so simplifies the feasibility check.

Note that the only time that the precise predicted holding area positions matter is when a simulator takes a snapshot of the positions. This is always done before any aircraft have actually entered the runway, as will be seen later, so allowing aircraft to leave the holding area early cannot affect these position predictions. Consideration of the fact that aircraft must leave the holding area in take-off order (so, if aircraft a is still in the holding area all aircraft which will take off after a will also still be present) means that the purpose of the current time value (allowing for delay imposed by the movement of aircraft which take off later, as described below) will not be compromised by the removal of earlier take-offs from the holding area.

6.9.3 The earliest take-off time for an aircraft

The concept of an earliest taxi time to the runway is re-introduced here to explain the second use of the current time mechanism. Rather than add complexity or require precise taxi time estimates, the designed system ensures 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 derived is easily achieved, a minimum of two minutes was allowed to traverse the holding area in the experiments presented in this thesis, thus adding slack to the taxi time. Therefore, the take-off time is limited 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. This means it has at least three minutes to traverse the holding area.

The only time there could be a problem is when an aircraft has to hold for some other aircraft that is not overtaking it and which takes off later than it. (If the blocking aircraft takes off earlier than the delayed one, then the delayed aircraft will have sufficient time to reach the runway as long as the blocking aircraft has enough time, so only blocking by aircraft which will take off later has to be considered.) The last limitation on take-off time is designed to handle this problem. For example, it is possible for an aircraft A to have an earlier take-off position than aircraft B and C, but for B to only be able to overtake C if it does so before A moves. For instance, moving A may block a necessary space in the holding area. In this case, A cannot take-off before B and C move so the arrival times of B and C and the time taken to perform the overtaking should limit how early A can take off. Even more complex scenarios are also possible, involving the interactions of more aircraft, all of which need to be taken into account.

This extra delay that must be imposed upon an aircraft which must wait for the movement of other aircraft is modelled using the current simulation time. The runway arrival time of the aircraft which must wait is limited by the arrival time of the aircraft for which it must wait. Whenever the current time has to be advanced, the only aircraft remaining in the holding area are those which are waiting for an aircraft which has not yet arrived to do something. Whenever the current time is advanced, a limitation is placed upon the take-off time of all aircraft still in the holding area such that the earliest take-off time is the current time plus the traversal time rather than the aircraft's own arrival time plus the traversal time.

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 5.5. The general equation for the earliest take-off time is given by equation 6.1, where h_i is the time the aircraft is predicted to arrive at the holding area. T_i is a safe holding area traversal time for aircraft i, given the holding area traversal path that aircraft i is using, and ct_i is the value of the current time variable the last time it was incremented prior to aircraft i leaving the holding area graph. This will be recorded for the aircraft during the feasibility test. If aircraft i leaves the holding area before the current time variable is incremented then ct_i is set to be zero, so that h_i always takes precedence.

$$f_i = T_i + \max(h_i, ct_i) \tag{6.1}$$

The value T_i should be defined in a way that it will be achievable by aircraft, so should be a safe large value rather than a minimum or mean expected traversal time for the holding area along that path, so that f_i is the earliest time at which aircraft i can be safely assumed to reach the runway. A value of $T_i = 120$ was used in the experiments performed for this thesis. This means that no aircraft can be scheduled to take off within two minutes of arriving at the holding area.

The value f_i is used in the take-off time prediction model detailed in section 5.5. If required, an approximation of f_i may be obtained without needing to perform the feasibility test by assuming that $ct_i = 0$. An approximation of the earliest take-off time can then be determined from the approximated value of f_i.

6.10 Summary

The solution method for the control problem was considered in this chapter. The control problem is solved heuristically, to allow a fast solution method to be developed, but doing so has been seen to have obvious advantages in ensuring that the paths that are used will be acceptable to controllers and that the re-sequencing that is suggested will be easy to achieve. The structure of the holding areas, and in particular the limited number of nodes of convergence and divergence are key to the successful application of this method.

The entire solution method for the take-off problem has now been described. The simulation that was developed for evaluating the performance of the system is described in the next chapter. The success of the approach will then be shown by the results in chapter 8.

Chapter 7

Testing and Simulation

7.1 Introduction

The method used to test the developed decision support system algorithms is described in this chapter. Testing an on-line decision support system is not simple, but an accurate testing system is vital if any confidence is to be gained in the validity of any predicted benefits from such a system. Obviously, testing the system in a live environment is far from ideal, even when it is possible, as a solution system for a complex problem is unlikely to perform well until it has been tuned to the problem. In this case, the safety considerations of increasing the controller workload during any live testing phase make it impractical.

One problem that any decision support system for a runway controller faces is the lack of accurate information about the future and particularly the lack of information about aircraft which may arrive at the holding area later. The effects of this problem can be partially alleviated by considering the aircraft currently upon the taxiways, however there is inevitable uncertainty intrinsic to any taxi time prediction, limiting the value of such information. The experimental results presented in chapter 8 show the benefits of having perfect knowledge of taxiing aircraft a specific number of minutes before the aircraft reach the holding area. The effects of errors in these predictions are also considered.

It is vital to test the system in the way in which it will actually be used. Even when including taxiing aircraft into the considered problem, it is inevitable that the system is still only solving a series of partial, instantaneous problems but aiming to find an overall combined solution which will be good for the entire day. It is important for any evaluation method to present the system with realistic sub-problems to solve, to enact the decisions and then to evaluate the worth of the overall final solution obtained for the entire day or half-day period. In this way, more confidence can be gained about how the decision support system could perform in a real situation.

7.2 Simulation Overview

The primary objective of the simulation is to provide enough information to test the decision support system and evaluate the effects of decisions made, without providing unnecessary detail about aspects that will not affect it.

The simulation has to provide a series of problems for the decision support system to solve. In some ways this is similar to the rolling time window [117] or rolling horizon approaches [105, 106], since all of these approaches attempt to obtain an overall good schedule by solving a sequence of sub-problems. The main difference is that the sub-problems are presented by the simulation because there is only limited information available at any time, rather than in order to reduce the complexity of the problem to be solved.

The features and responsibilities of the simulation are as follows:

1. Create an appropriate sub-problem for the decision support system.

   This means creating the input problem for the decision support system, thus specifying the current state of the airport departure system. The simulation uses real historic data, provided by NATS to generate the initial input data for the decision support system. The effects of decisions made by the decision support system are then used to modify the later input problems that are presented.

   The input data is detailed in section 2.14 and must include:

   - The positions of any aircraft within the holding area and the path planned for each one through the holding area.
   - Information about any aircraft which recently took off.
   - Predicted arrival times and arrival entrances for all aircraft currently taxiing around the airport towards the holding area.
   - Details of all aircraft in the system, for example, the weight class, speed group and assigned departure route.

2. Accept the output from the decision support system under test.

   At a minimum this means a planned take-off sequence. Predicted take-off times are also useful. The implemented simulation goes beyond this as it also receives details of how the output state was achieved, including the paths which were assigned to aircraft and even a snapshot of the expected positions of aircraft within the holding area at some point in the future.

3. Enact the decisions made in order to see the later effects of the decisions.

   This means at least assigning paths through the holding area to aircraft and predicting the movement of aircraft.

In the implemented departure system simulation this task is delegated to the decision support system and the paths it assigned are used by the simulation.

4. Define a set of rules to specify how the state changes over time. For example:
   - Aircraft lined up for take-off will take-off.
   - Aircraft within the holding area may move forwards through the holding area, along their assigned path, or may have to wait for other aircraft to overtake them first.
   - Aircraft on the taxiways will advance towards the holding area over time.

     This means reducing the time the aircraft is expected to remain on the taxiways, and possibly reducing any simulated uncertainty or prediction error in the value passed to the decision support system.

5. Optionally, provide some simulation of the uncertainty in information with which a real decision support system must contend.

   In the simulation described here, information about the exact times at which things will happen is maintained, but 'estimated' values can also be produced by adding prediction errors. It is, then, the inaccurate predicted values which are provided to the decision support system as input data at each stage. This simulates the fact that the decision support system would not necessarily have accurate information in advance.

7.3 Data Maintained By The Simulation

The problem presented by the simulation to the decision support system includes different levels of detail for aircraft, depending upon the current position of the aircraft in the departure system. The current position of an aircraft is determined by comparing the current simulation time with the historic times at which aircraft pushed back from the stand, the times the aircraft reached the holding area and the predicted take-off times provided by the decision support system. Aircraft will either be at their stands, on the taxiway, at the holding area, or in the air. Aircraft move forward through the states over time. The level of detail the simulation maintains for aircraft varies as the state changes.

7.3.1 Aircraft at the stands

No information is given to the decision support system about aircraft that are at their stands. Aircraft only enter the problem after they have pushed back from their stands. Until this point, the level of uncertainty currently involved in the time they will push back from their stand is such that there is little use in incorporating these aircraft into the system. As described in section 3.9, there is a possibility that this may change in future and such aircraft could then be taken into consideration.

7.3.2 Taxiing aircraft

Taxiing aircraft are dealt with at a relatively high level of abstraction and are only included in order to give some idea of what will be arriving at the holding area, when it will be arriving and where it will arrive. Experiments have shown that the decision support system performs better if it is given knowledge of the taxiing aircraft, thus enabling the search to have visibility of the effects upon these taxiing aircraft of decisions currently being made.

One common case that including taxiing aircraft can assist with is when there is a taxiing aircraft with a tight take-off time-slot (CTOT). This aircraft may require a straight run through the holding area in order to achieve take-off within the CTOT slot. Knowing about such aircraft in advance can ensure that other aircraft are not in the way when they arrive.

Another potential benefit is seen when 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.

Knowing about taxiing aircraft can reduce the amount of re-sequencing required of the aircraft already in the holding area. For example, knowing that a taxiing aircraft could be slipped into a gap in the take-off schedule means that it is not worthwhile eliminating the gap at the expense of other parts of the take-off sequence.

The following information is maintained for aircraft currently on the taxiways.

- The historic arrival time at the holding area. This is available in the historic input data. An expected arrival time is provided to the decision support system by optionally modifying the historic arrival time by a variable prediction error to simulate a level of uncertainty for the values. Prediction errors would be expected to decrease over time for any specific aircraft as it approaches the holding area. The simulation assumes that aircraft will leave their stand and arrive at the holding area at the times at which they did so historically, but the decision support system will not necessarily know these time accurately in advance due to the added prediction errors.

- For all aircraft on the taxiways, the holding area entrance at which the aircraft will arrive is predicted based upon the stand from which the aircraft left. For example, there are three entrances in the 27R holding area diagram given in figure 6.2 and they are labelled D, K and G. The simplest case assumes the Ground Movement Controller (GMC) directs aircraft to the closest entrance to the stand, as discussed further in section 8.2.7.

7.3.3 Aircraft within the holding area

More information is maintained for aircraft in the holding area as these aircraft are under the control of the runway controller or the decision support system in these experiments. The effects of decisions made for these aircraft have to be maintained.

The simulation uses the same directed graph representation of the holding area as is used by the decision support system. This allows the simulation to provide initial holding area positions of the aircraft to the decision support system and also allows the simulation to ask the decision support system to predict new positions for aircraft, as will be seen later.

The following information is available about these aircraft:

- The actual holding area arrival time and holding area entrance at which the aircraft arrived. This is important as the holding area delay is calculated based upon the arrival time, and the arrival entrance affects the traversal paths that can be taken and the possibilities for overtaking.
- Any holding area traversal path that was previously allocated. This is maintained as the decision support system is not permitted to change this for any aircraft that is already in the holding area, as described in section 7.4.6.
- The current position of the aircraft in the holding area. This is predicted by the simulation and depends upon the allocated path. It will change over time as the aircraft moves forward through the holding area. The simulation uses the results of the feasibility check to predict holding area positions, as described in section 7.7. An implemented decision support system could use ground radar data to determine the position of aircraft.

7.3.4 Aircraft which have taken off

Once aircraft have taken off, the only information that is needed is the take-off time and aircraft characteristics, so that the required separations can be calculated for future aircraft.

7.4 Simulation Loop

The simulation maintains a list of the aircraft that the decision support system should be aware of at the time, with the details described earlier for each aircraft.

For evaluation purposes, it is assumed that a controller will follow the advice of a decision support system. The simulation can then be used to evaluate the effects of various changes to the departure system, putting the decision support system in the place of the controller.

The simulation starts at the start of the dataset and performs the steps described in algorithm 17. In algorithm 6, T_S refers to the current simulation time and the other variables and constants are described in the explanation of the algorithm given below.

For each aircraft, a, the simulation needs to know the push-back time, P_a, of the aircraft, the holding area arrival time, H_a, of the aircraft and, once the decision support system has scheduled a take-off time for an aircraft, the predicted take-off time, D_a, and the position, O_a, of the aircraft in the take-off sequence.

Algorithm 17 The simulation loop

initialise simulation time, T_S, to start time of the dataset under test
empty the set of aircraft S_A to provide to the decision support system
while there are aircraft which have not taken off **do**
    for each aircraft a in the dataset **do**
        if $(a \notin S_A)$ and a has not already taken off and been removed from S_A **then**
            if the flag E is set or $T_S \geq P_a$ **then**
                if $T_S \geq (H_a - T_P)$ **then**
                    add a to the set S_A
                end if
            end if
        end if
    end for each
    pass the set of aircraft S_A to the decision support system and solve the take-off problem
    advance T_S by the current simulation step size, for example 60 seconds
    for each aircraft $a \in S_A$ **do**
        if $T_S \geq H_a$ **then**
            aircraft a is within the holding area
            retrieve the allocated traversal path for a from the decision support system and fix it for the aircraft, so it can no longer be changed in future
            retrieve the predicted holding area position for a from the decision support system. This will be passed to the decision support system in future iterations
        end if
        retrieve the position O_a of the aircraft in the desired take-off sequence from the decision support system
        retrieve the predicted take-off time D_a for the aircraft from the decision support system
        if $T_S \geq (D_a + T_M)$ **then**
            the aircraft has taken off long enough ago that it can be ignored so remove a from the set S_A
        else if $T_S \geq (D_a - T_F)$ **then**
            the aircraft is close to take-off. Fix the position, O_a, in the take-off sequence and the take-off time, D_a of aircraft a
        end if
    end for each
end while
end simulation and return the predicted take-off times for all aircraft

Configuration parameters are required in order to determine when the decision support system should be made aware of each aircraft. If aircraft are included before they push back, then the flag E should be set to true, otherwise it should be false to indicate that the system only considers aircraft after they push back. A planning horizon, T_P, must be specified, determining how early before holding area arrival the system will be given information about taxiing aircraft. The planning horizon is considered in more detail in chapter 8.

The take-off sequence is fixed for aircraft close to their time of take-off to allow for the fact that aircraft would need some time to line up and both controllers and pilots would need some firm knowledge of the future take-off sequence. The length of time during which the take-off time is assumed to be fixed is given by the variable T_F. As the positions of these aircraft are fixed, the searches have no flexibility to re-sequence them any more.

In order to maintain separations from aircraft which have taken off, an aircraft must remain in the system even after take-off until it can have no effect upon later take-offs. Let T_M

be the memory time, the time after take-off for which the system must remember aircraft. A safe value of T_M should equal or exceed the maximum duration of a separation rule. A value of ten minutes was used in the experiments, allowing for the application of a large MDI, increasing the required separation on a departure route.

7.4.1 The problem passed to the decision support system

When the current problem is passed to the decision support system in step 13, all of the required details such as weight class, speed group and departure route are passed for each of the aircraft in the set S_A. The decision support system is told the current state of the aircraft; whether it is taxiing, at the holding area or has taken off. Additionally, where predicted holding area positions are known they are also passed, as are the details of any traversal paths that have been fixed and details of whether the position of the aircraft in the take-off sequence has been fixed.

The previously determined take-off sequence (if there is one) is also passed to the decision support system. As the search moves through the time period of the dataset, later searches use the partial schedule built by previous searches, with knowledge of any extra aircraft that arrived, to re-sequence the remaining aircraft. Eventually the aircraft will all have a fixed take-off time and a final schedule will have been created and can be evaluated. The outputs of the simulation are discussed in section 7.5, along with a description of the method used to evaluate these outputs.

7.4.2 Updating the internal model

The internal model of the departure system is updated in a number of ways. The positions of the aircraft in the holding area will move forwards over time, as updated predictions of positions are retrieved from the decision support system.

Aircraft will move through the departure system towards take-off as the simulation time is advanced. Although the decision support system may only receive an inaccurate, predicted holding area arrival time, the (real) arrival time for each aircraft that is known by the simulation is fixed, so eventually each aircraft will arrive at the holding area. At this time, the last traversal path that was allocated to an aircraft is fixed and cannot be changed.

The decision support system predicts take-off times for aircraft and provides these predictions to the simulation. The simulation assumes that the decisions of the decision support system are enacted, so aircraft will take-off when they are predicted to do so. Eventually, aircraft will reach the predicted take-off time and be assumed to take off. At some point before this time the position of the aircraft in the take-off sequence should be fixed.

7.4.3 Extra outputs from the decision support system

The decision support system has an important role to play in the simulation as it performs much of the prediction work. The decision support system already calculates a lot of information in order to perform its task and rather than have the simulation re-calculate these things, the decision support system is responsible for storing the information it calculated for aircraft and providing it to the simulation when needed. This information includes a predicted position in the holding area, a predicted take-off time and the assigned holding area traversal path.

7.4.4 Predicting holding area positions

For each aircraft within the holding area, the current node the aircraft is occupying must be specified as one element of the input data for the decision support system. The positions have to be determined by the simulation and the implemented simulation does this by delegating the task to the decision support system.

The simulation tells the decision support system to re-play the feasibility test for the selected take-off sequence. At a relevant point in the re-play, the decision support system takes a snapshot of the positions of the aircraft and uses these positions as the predicted positions to supply to the simulation. This snapshot is taken at the point where the first aircraft in the holding area takes off or the first time the current time is advanced; whichever happens first. The implementation is actually more complex than this as it needs to ensure that the positions are realistic, so the system is considered in more detail in section 7.7.

7.4.5 Simulation step time

The simulation time is increased in pre-determined steps, as indicated by step *14* of algorithm 17. A sixty second increment was used for the experiments performed for this thesis. With a search time of a second, one second increments would be more realistic, simulating a constantly running decision support system, however, this would allow the decision support system multiple attempts to obtain a good schedule in response to changing situations and would therefore make evaluation of the performance of a single iteration much harder. The selected step size makes the problem much harder for a decision support system to solve.

7.4.6 Fixing the traversal path

The holding area traversal path is important. Once an aircraft has reached the holding area, changing the path it uses to traverse the holding area is more difficult than doing so before it enters the holding area. It is important for the simulation to recognise this difficulty.

The simulation is deliberately over-restrictive about what can be performed. If an action may be difficult for a controller or pilot, then it is not accepted. The system aims to only ever recommend good sequences that are easy to achieve. To this end, the holding area traversal path is fixed for an aircraft as soon as it enters the holding area, as indicated by step *18* of algorithm 17. The reasons for this are discussed in section 2.12.2. In a real decision support system, with a controller present, more flexibility could be allowed and the schedules obtained may be even better.

7.5 Simulation Outputs

The primary output of the simulation is an overall take-off sequence for the dataset, with predicted take-off times for aircraft. These take-off times can then be used to determine how many aircraft would miss take-off time-slots (CTOTs) and to calculate the length of time that each aircraft is predicted to be delayed in the holding area for. These values could be considered to be the most important aspects of the decision support system, in that it is important to ensure take-off is within CTOT where possible and the delay is a good measure of the potential throughput of the system.

7.5.1 Equity of delay

Although the delay and CTOT compliance are vital measures of the worth of a take-off sequence, it is not valid to ignore the other objectives described in section 2. For example, it is often easier to keep the delay low by delaying the aircraft with specific characteristics until the end of the take-off sequence. A slow, light aircraft will often need a larger separation before it (to allow for the wake vortices from the preceding aircraft to dissipate) and a larger separation after it for any other aircraft on a similar departure route, to allow for the fact that following aircraft will take off faster and catch up. Delaying such aircraft for as long as possible will limit the effects of these separations by reducing the number of aircraft affected.

An extra measure that is checked to verify the performance of the system is the positional delay of aircraft. The aircraft with the highest positional delay are examined to identify why the positional delay was applied. The positional delay can be determined directly from the take-off sequence. If required, the equity of the seconds of delay per aircraft can also be determined from the predicted take-off times in the primary output.

This primary output is the only information which can be usefully evaluated by someone other than a problem domain expert. As such, the results presented in this thesis show the delay, CTOT compliance and positional delay of the aircraft in the generated take-off sequences.

7.5.2 Other outputs

A number of other outputs can be generated by the simulation. The other outputs are used for tuning the decision support system and the simulation. The aim is to calibrate the simulation to make it act in a similar way to the real world and to tune the decision support system so that it produces sequences which would be acceptable to controllers.

Key outputs for these purposes include the detailed holding area movement, the sequences which were adopted at each step of the simulation and details of the sequences that the decision support system considered.

7.6 Simulation Tuning And Calibration

A secondary output of the simulation is a set of holding area moves which can be performed to achieve the take-off sequence. These moves give the same information as the triplets in the triplet model, described in section 4.2.

The numerical delay and CTOT compliance results of the analysis of the final sequences obtained will indicate situations where the path allocation system or feasibility check are not flexible enough to obtain schedules with low delay and a low number of CTOTs missed. The numbers will not show whether the allocated paths would be acceptable to a runway controller, so methods were needed to verify this.

It is important to calibrate and validate any simulation and this can be complex [8]. Validation of the decision support system and simulation described in this thesis was performed by providing a visual display of the operation. The visualisation is not intended to be any kind of final user interface for the system, but instead was designed for obtaining information from the experts, such as air traffic controllers, and for illustrating to them the acceptability of the solutions from the final tuned system. It is not uncommon to provide a visualisation for such a system, for example the visualisation of the ATOS system [143, 96, 94] was discussed in section 3.8.2.

The path allocation algorithm suggested in this thesis is deterministic. Once tuned to controller preferences, the algorithm will continue to give acceptable results, but the difficult task is tuning it to controller preferences in the first place. It is not possible to automate this sort of tuning so holding area movement replay software was developed for this purpose.

The holding area movement described by the triplets can be replayed using the holding area movement viewer, described in section 7.6.2. The holding area viewer is ideal for obtaining feedback from problem domain experts and for comparing a replay of the sequencing the system achieves against the real sequencing that was performed. Even a count of how many times each path had been allocated would not show whether the allocation of a slower path was valid at the time the allocation was made. The viewer described in section 7.6.2 provides a fast time playback of the sequencing, so that the flow of aircraft through the holding area can be viewed and the path allocation can be considered and tuned.

The simulation works by presenting a series of problems for the decision support system to solve. The realism of these problems needs to be verified if the results from a simulation are to be accepted as viable. An optional user interface was developed for the simulation in order to simplify the examination of the problems created at each step, in addition to the playback facilities already discussed. The user interface allows step by step movement forwards or backwards through the simulation, so that the state at any stage in the simulation can be examined.

The replay program was mainly used to verify the acceptability of the path allocation rules. The user interface on the simulation was mainly used for verifying the realism of the problems created by the simulation.

7.6.1 Running With A User Interface

A user interface is helpful to illustrate what the search is doing in real time. The empirical results in chapter 8 are presented for experiments which were executed in the absence of a user interface (although sample experiments were re-executed with a user interface just to ensure that performance was as expected). This ensured that experiments could be performed as batch processes and the results analysed later.

Components of the user interface

The user interface has three components. The first is a simple control window, allowing a user to step forward or backwards through the stored sequences, or to display a real-time update showing each new sequence as it is received. The sequencing information includes both the positional information for aircraft and the chosen take-off sequences. This information is displayed in the other two windows. Sequencing information can also be saved to a file and loaded later for examination and playback. This allows a detailed examination of the sequencing that the system performed and of the holding area positions the simulation predicted for aircraft.

The second window, shown in figure 7.1, displays information about the aircraft in the take-off sequence. The suggested take-off sequence and predicted take-off times can be seen. All times are shown relative to the start time of the dataset.

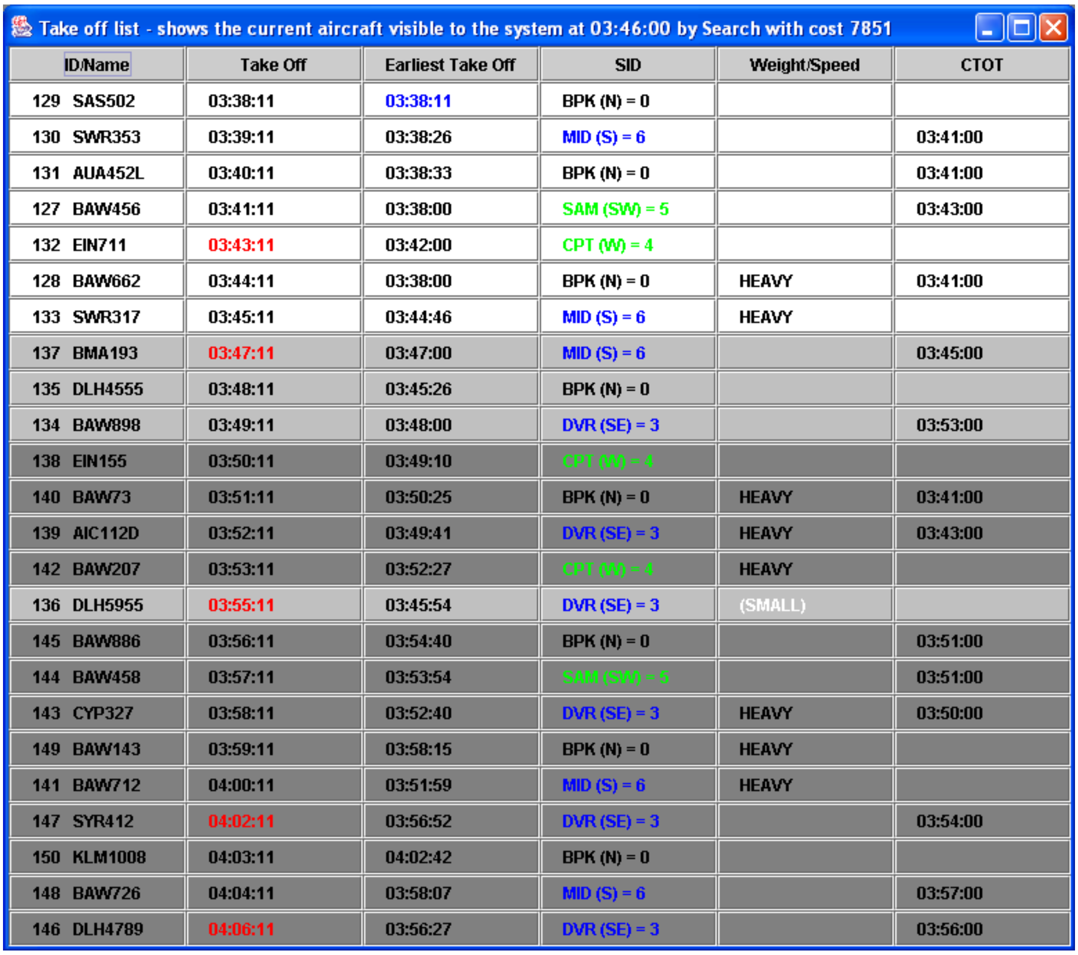
The first column shows an identifier for the aircraft and the position of the aircraft in the first-come-first-served sequence. The second column shows the predicted take-off time for the aircraft. Any separation which is more than the minimum one-minute is highlighted in red.

An earliest take-off time can be estimated for an aircraft based upon the holding area arrival time, traversal time and the results of the feasibility check. This time is shown in the third column and is important in understanding why aircraft were not sequenced to take-off earlier than they did.

The departure route, weight class and speed group determine the separation rules. The allocated departure route (SID) is shown in the fourth column. The rough direction of departure (for example, North, South or West) is shown in parentheses to aid in understanding the sequencing.

Most aircraft are medium weight class and are speed group three. Where the characteristics of an aircraft differs from these norms, the information is noted in the fifth column. The final column in the usual display shows the allocated CTOT time for the aircraft. Aircraft must take off no more than five minutes before the CTOT time and should take off no more than ten minutes after the CTOT time. A number of extensions are available, allowing aircraft a five minute extension to the end of the CTOT window. Any take-off after that time is likely to require a new CTOT to be determined for the aircraft.

Optional extra columns provide more detailed information including the allocated traversal paths through the holding area and can show both the actual timings and the times which include the prediction errors to account for any simulated uncertainty.



Take off list - shows the current aircraft visible to the system at 03:46:00 by Search with cost 7851

ID/Name	Take Off	Earliest Take Off	SID	Weight/Speed	CTOT
129 SAS502	03:38:11	03:38:11	BPK (N) = 0		
130 SWR353	03:39:11	03:38:26	MID (S) = 6		03:41:00
131 AUA452L	03:40:11	03:38:33	BPK (N) = 0		03:41:00
127 BAW456	03:41:11	03:38:00	SAM (SW) = 5		03:43:00
132 EIN711	03:43:11	03:42:00	CPT (W) = 4		
128 BAW662	03:44:11	03:38:00	BPK (N) = 0	HEAVY	03:41:00
133 SWR317	03:45:11	03:44:46	MID (S) = 6	HEAVY	
137 BMA193	03:47:11	03:47:00	MID (S) = 6		03:45:00
135 DLH4555	03:48:11	03:45:26	BPK (N) = 0		
134 BAW898	03:49:11	03:48:00	DVR (SE) = 3		03:53:00
138 EIN155	03:50:11	03:49:10	CPT (W) = 4		
140 BAW73	03:51:11	03:50:25	BPK (N) = 0	HEAVY	03:41:00
139 AIC112D	03:52:11	03:49:41	DVR (SE) = 3	HEAVY	03:43:00
142 BAW207	03:53:11	03:52:27	CPT (W) = 4	HEAVY	
136 DLH5955	03:55:11	03:45:54	DVR (SE) = 3	(SMALL)	
145 BAW886	03:56:11	03:54:40	BPK (N) = 0		03:51:00
144 BAW458	03:57:11	03:53:54	SAM (SW) = 5		03:51:00
143 CYP327	03:58:11	03:52:40	DVR (SE) = 3	HEAVY	03:50:00
149 BAW143	03:59:11	03:58:15	BPK (N) = 0	HEAVY	
141 BAW712	04:00:11	03:51:59	MID (S) = 6	HEAVY	
147 SYR412	04:02:11	03:56:52	DVR (SE) = 3		03:54:00
150 KLM1008	04:03:11	04:02:42	BPK (N) = 0		
148 BAW726	04:04:11	03:58:07	MID (S) = 6		03:57:00
146 DLH4789	04:06:11	03:56:27	DVR (SE) = 3		03:56:00

FIGURE 7.1: An example take-off sequence

The third window, shown in figure 7.2, is actually the same as used by the holding area playback tool described in section 7.6.2 and is used to show the positions of aircraft within the holding area. This can aid in understanding why the decision support system makes the decisions that it makes.

This real-time feedback is especially useful for investigating the validity of the simulation system as well as the decision support system itself. For example, the display will show whether the predicted positions of aircraft are reasonable. It was precisely this feedback, and the replay of the holding area movement, that revealed the initial problems with positional predictions that are discussed in section 7.7. These problems were subsequently rectified.

Once the path allocation system had been tuned in this way and appropriate measures implemented to ensure the sequencing, movement and predicted positions were sensible, these areas were not assessed any further, beyond sampling to ensure that the measures continued to work as expected and that bugs had not been inadvertently introduced.

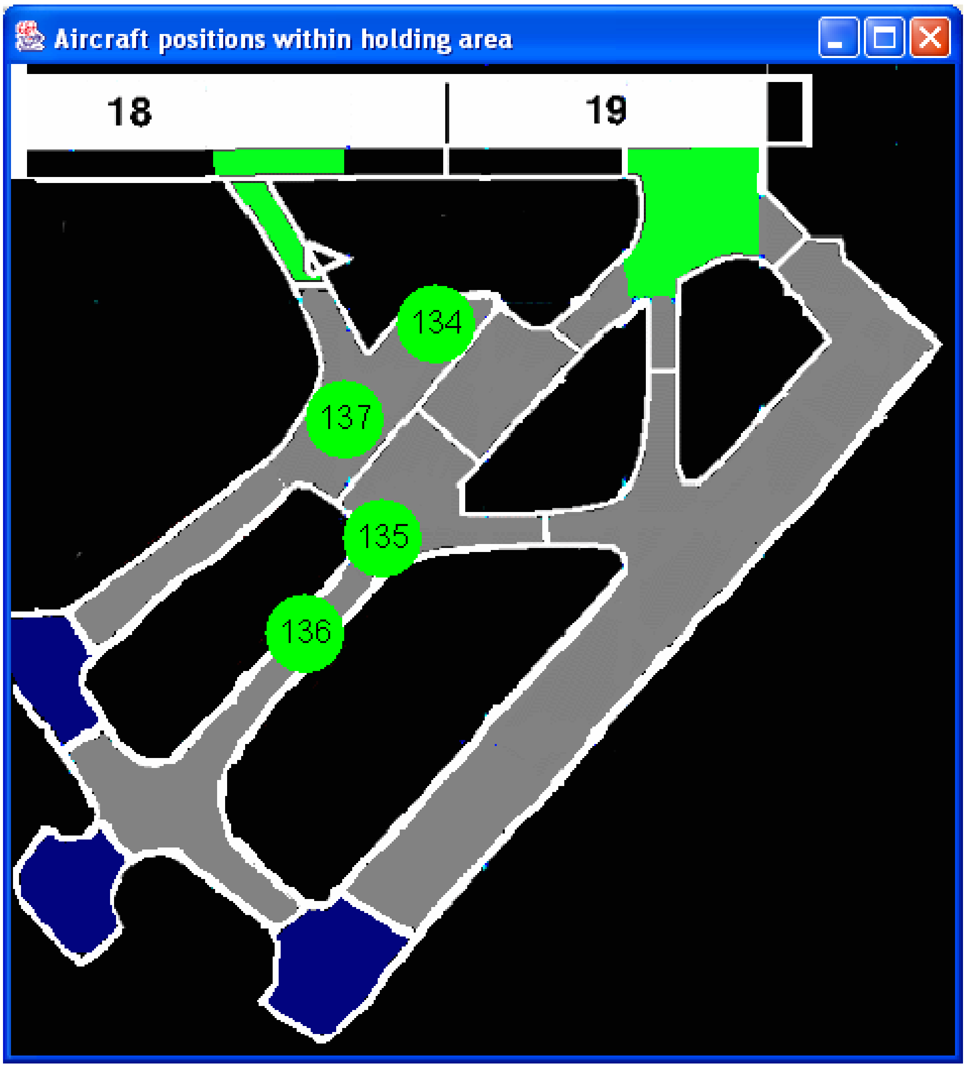


FIGURE 7.2: The holding area contents during a simulation

7.6.2 The Holding Area Viewer

The path allocation system was tested using the holding area viewer. This consisted of a graphical display similar to that 7.2 and a control window with controls to load new files, pause, play, step forwards or backwards through the playback and alter the displayed details or playback speed.

Due to the way in which the decision support system works, guarantees can be given that unacceptable traversal paths will never be used and less desirable paths will only be used when doing so is justified by the sequencing requirements. This can only be achieved once the controller preferences have actually been identified. The main purpose of the holding area viewer was to obtain feedback from the problem domain experts about whether the path allocation was acceptable. This is important since a consideration of the historic data is not always sufficient. One path in particular was observed to be used in the playback of the historic data, but playback to a controller of a sequence generated using this path allocation prompted the feedback that he would not usually use it as it was too difficult. The secondary purpose of the holding area viewer was to demonstrate and verify that the path allocation actually performed as expected and that the holding area movement was simple enough to be acceptable.

The holding area movement viewer is given three pieces of information. The first is the set of triplets (similar to those described in section 4.2) which define the movement which took place in the holding area during a simulation run of the decision support system. The next is a background image to use, which shows the structure of the holding area. The third consists of the coordinates on the holding area picture of each of the nodes of the holding area graph.

With this information the holding area viewer can replay the movement of aircraft from the triplets specifying the old and new positions of each aircraft. Each aircraft is represented by a labelled circle on a background image for the holding area structure. Processing of a triplet involves moving the representation of the aircraft from the coordinates for the source node to the end node. By interpolating the coordinates for multiple points between the two nodes, the illusion can be given of a relatively smooth movement for the aircraft through the holding area structure. The number of interpolation points can be modified to increase or decrease the speed of replay and the tool can step backwards, forwards or run in a continuous play mode. An additional coordinate is available, if necessary, for any pair of nodes, specifying the mid-point on the path between the nodes. Using this, the path can appear to bend, so that aircraft will appear to more accurately follow the layout of the holding area.

Another feature of the playback tool is the ability to re-sequence triplets, or provide simultaneous movement. Where the sequence in which triplets are processed does not affect the feasibility of re-sequencing, the tool can swap the order in which the triplets are played back, if the reversed order appears more realistic. Similarly, some triplets can actually take place simultaneously, where the nodes and aircraft involved are disjoint, and the playback tool is capable of processing these at the same time, giving the more realistic appearance of multiple aircraft moving through the holding area at the same time.

In summary, the playback tool provides an easy facility to view the holding area movement, for comparison against real movement and for demonstration to runway controllers. It was invaluable in collecting information about controller preferences.

7.7 Holding Area Position Prediction

The simulation predicts positions for aircraft in the holding area by requesting the decision support system to perform a modified feasibility check and taking a snapshot of the positions during this. The snapshot is taken at the point at which the first aircraft enters the runway node or when the simulation time is advanced, whichever is earlier. These positions are then used as the predicted initial positions for the next problem presented to the decision support system. When the user interface was used to examine the positions that were generated, two related problems were seen.

The first problem was due to the point at which the snapshot is taken. At this point, it is possible that some aircraft have not had the opportunity to be moved forward through the holding area. A real controller would usually only advise aircraft to hold further back in the holding area if it was necessary in order for overtaking to take place, as doing so increases congestion at the holding area entrances. In practice, aircraft would probably be moved as far forward as possible but this was not always being done by the simulation. It would be useful to move aircraft forward before the snapshot is taken if they can do so without causing any sequencing problems.

The second problem is linked to taxiway congestion. Holding aircraft further back in the holding area, in order to provide possibilities for other aircraft to overtake them, can result in a queue of aircraft behind them, possibly blocking the taxiways and causing problems for the controllers. Unnecessary holding area or taxiway congestion may then occur behind these aircraft. This situation would not be permitted to occur in practice so any simulation should also not permit it. Controllers would not leave aircraft congesting the taxiways if it is at all possible to fit them into the holding area.

7.7.1 Additional modifications to enhance realism

To simulate the behaviour of a real system, some further checks are performed immediately prior to taking the snapshot of positions. Firstly, 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 sequence. This is performed by repeatedly checking each aircraft in turn and attempting to move it to the next node in the holding area, using the same rules about blocking that are normally used so that the feasibility of re-sequencing will not be affected. The only extra restriction is that aircraft cannot enter the runway. 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.

If the taxiways are not clear after this has been performed, the process is repeated with relaxed movement restrictions. 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, a slightly modified rule is used: *'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.'* This modified rule is more time consuming to apply but, of course, is only being applied to the final sequence rather than to every solution that is evaluated during the search, so the extra time is not a problem.

The normal movement rules require empty nodes further in the holding area to move a potentially blocking aircraft into. The modified check considers the priorities of aircraft in these nodes and treats the node as empty if the priority is such that the aircraft will have taken off by the time the potentially blocking aircraft must move. This is not desirable from the point of view of providing schedule flexibility as it possibly leads to aircraft temporarily blocking other aircraft which take off before them and so is normally avoided. However, it does mean that aircraft can move further into the holding area, possibly freeing nodes so other aircraft can move off the taxiways.

In the experiments performed for this thesis, the first method (giving aircraft the opportunity to move forward before taking the snapshot) usually cleared taxiway congestion. In the few cases where this did not clear it, the second method, relaxing the movement restrictions, always did so. The selected recovery method if these fail was to accept the taxiway congestion but this was never needed in experimentation.

With these enhancements, the predicted positions for aircraft within the holding areas were observed to be much more realistic. If the system did actually leave taxiway congestion at any point, the fact was flagged up in the output files. It was not observed in the experiments. These changes are only used by the simulation and this kind of holding area prediction would not be needed in a live situation.

7.8 Sequence Stability

Ideally, a decision support system should give advice and never have to change it. That ideal is not practical, however, as the changing situation and increasing knowledge over time may mean that there are advantages to be gained from re-visiting earlier suggestions. Assuming that the simulation steps are such that only a few aircraft are added to the system at each step, any sequence other than the first-come-first-served sequence would require new aircraft to be fitted into an existing take-off sequence. This would alter the positions of aircraft already in the take-off sequence, so some positional deviation over time is usually expected.

7.8.1 Observations of the strict insertion sequence

The sequence created by inserting each aircraft, in first-come-first-served order, into the existing take-off sequence will here be called the strict insertion sequence. In fact, the strict insertion sequence is also highly unlikely to be a good take-off sequence. To see why, consider the fact that the departure route separations mean that sequences that represent alternate departure routes are often good, thus partial sequences that are found are likely to have this form. As described in section 5.4, the insertion of a new aircraft into such a sequence is likely to introduce a larger separation into the take-off sequence.

There are reasons, such as a tight CTOT or specific aircraft characteristics, that may mean an aircraft must overtake a number of aircraft in the current take-off sequence. With a strict insertion sequence, this would introduce a large separation. Only the insertion of further new aircraft into the sequence would rectify the problem. Even if a later aircraft is available to be inserted into the gap, advancing such an aircraft unnecessarily is unlikely to be the most equitable take-off sequence.

In summary, therefore, some re-sequencing is almost certain to be required when new aircraft enter the system in order to find good take-off sequences. Due to this requirement, it is not advisable to merely measure the amount of re-sequencing that is performed and use this as a measure of the stability of the sequencing. What is needed is a method to determine whether unnecessary re-sequencing is being performed; in other words, to identify whether the re-sequencing that is performed is actually necessary or not.

Not all change has the same cost. For example, the early part of the sequence is important as these are usually the aircraft that are already within the holding area and under the control of the runway controller. These are the only aircraft which would have already been given instructions. It is important that at least the early part of the sequence is stable over time, rather than constantly fluctuating between wildly different sequences of similar cost.

7.8.2 The sequence viewer

The problem is to determine what is a necessary or worthwhile change and what is unnecessary. Over the duration of each experiment, the simulation generates a number of static problems for the decision support system to solve. In order to visualise what the search is doing, the sequences that were created by the decision support system at each simulation step were output and a sequence viewer was created. This allowed a visualisation of the points at which the sequencing of aircraft changes and an obvious visualisation of the reason for doing so.

The sequences that were generated during a simulation can be loaded into the sequence viewer which will illustrate the ways in which the positions of aircraft change in response to new aircraft entering the system. The position of each aircraft at each step is displayed graphically. The horizontal axis of the display shows time, or the simulation step, and the vertical axis shows the position of the aircraft in the final take-off sequence.

An example of the display of the sequence viewer is given in figure 7.3. Each vertical column on the display represents a take-off sequence. The simulation iteration number is displayed below the circles representing the aircraft. Each aircraft is shown with a specific colour and the circles for each aircraft are connected with a line.

The point at which an aircraft is added to the sequence is represented by a larger circle. In general, horizontal lines are good (as they represent a stable position for an aircraft in the take-off sequence) and diagonal lines are bad, as they represent changes in the take-off sequence. However, a change in sequence is not a bad thing when there is a reason for it. Consideration of the sequencing shown in figure 7.3 shows that the sequence only changed in response to the addition of new aircraft. The sequencing is extremely stable here.

Some example problems can be easily visualised using this display. For example, when the lines for two aircraft cross then the aircraft are reversing their positions in the take-off sequence. This can be a bad thing and reveals problems such as when two aircraft have very

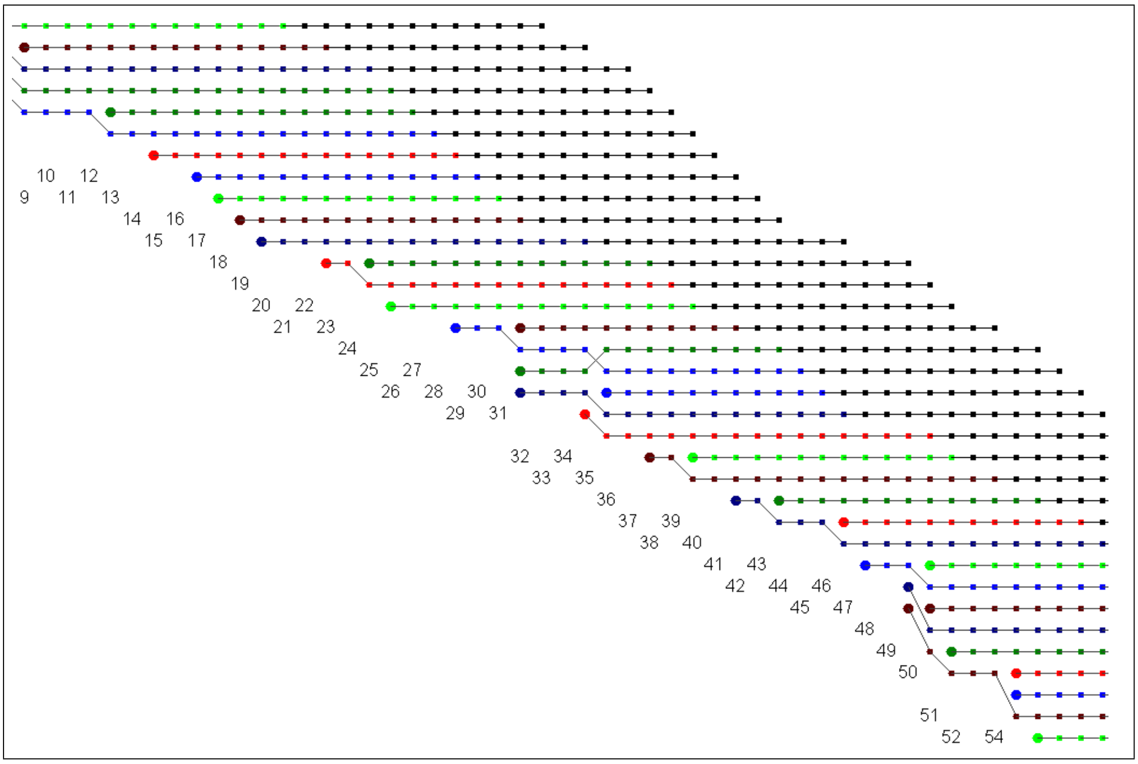


Figure 7.3: Viewing normal sequencing

similar properties, so that the take-off sequences have equal or very similar costs regardless of the relative take-off sequence for these aircraft. In this case, a decision support system may alternate between sequences which swap the positions of these aircraft. This situation can be easily identified from the graphic displays.

This display was especially valuable initially as the decision support system was found to (very occasionally) alternate between identically costing sequences. A first-come-first-served deviation cost was added, along with an additional cost for moving aircraft from the positions they took in the previous sequencing. Although very low, these costs acted as 'tie-breakers' and (together with the follow-on searches described in section 5.7 which ensured the alternative sequences were actually both considered) eliminated the alternation between sequences.

To illustrate the visibility of this kind of problem when the sequence is viewed, figure 7.4 shows the effects of a deliberate change to the objective function in order to favour the sequence least similar to the current sequence when considering similarly costing sequences. Here, the presence of so many diagonal lines illustrates the instability in the sequencing.

7.8.3 Resulting sequence stability

The sequences were found to be remarkably stable once the additions to slightly penalise changes had been made. Indeed, the sequence was not usually observed to change unless a new aircraft was added. Moreover, with adequate taxi knowledge of aircraft, the sequencing stabilised long before the aircraft reached the holding area and sequencing changes within the holding area were very rare. This is important as the controller will only ever be enacting the sequencing once the

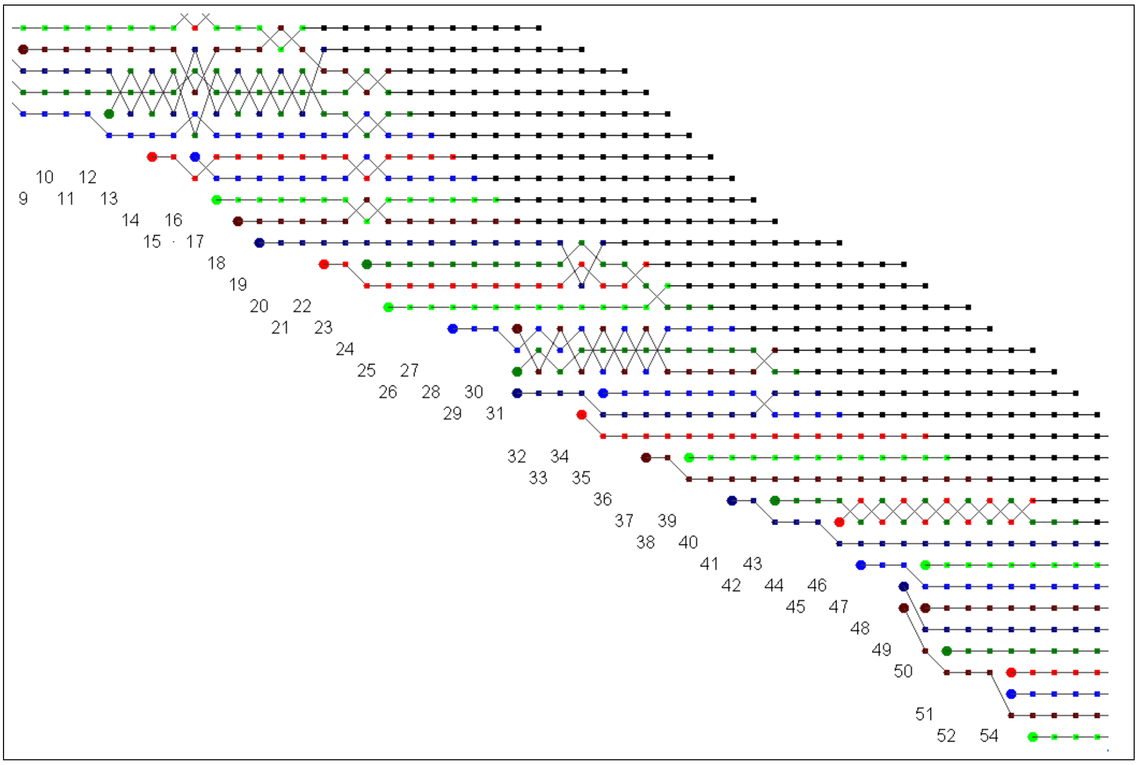


Figure 7.4: Picture of bad sequencing

aircraft reach the holding area, so changes prior to that time are virtually irrelevant.

The relative stability of the early part of the schedule is not unexpected as the delay-based objective function penalises larger separations much more if they are nearer the start of the schedule. The objective function, therefore, puts much more value on getting the first part of the schedule right. This means that the search is much more likely to find schedules which schedule the first few aircraft well than those which schedule the last few aircraft well. Additionally, the fact that each subsequent search is seeded with the resulting schedule from the previous iteration helps to ensure that the best schedule found in the previous search will be rediscovered in this search, so if there is a good schedule that is similar to the previous iteration, it is more likely to be found than a good schedule which is very different. In the majority of cases, new aircraft leaving their stands do not overtake more than a couple of aircraft to reach a good take-off position, and these overtaken aircraft are usually those still on the taxiways so, most of the time, only the last few aircraft in the schedule are affected.

The most excessive change to the sequence is observed when aircraft with tight CTOT time-slots leave their stands. At this point, the system will try to schedule the aircraft within CTOT if possible, often requiring overtaking a number of other aircraft to do so. However, even in the case of a tight CTOT, the constraints that are imposed upon the re-scheduling within the holding area and upon late re-scheduling act to limit changes. For instance, the first two minutes of the sequence is fixed, so no new aircraft can overtake these aircraft. Similarly, aircraft in the holding area that are not already being overtaken will have been assigned a direct path through the holding area, so a later arrival at the same holding area entrance can only overtake if a short-cut path is available and permissible.

Another observation from this display was the confirmation that the decision support system was finding the good solutions straight away rather than using multiple iterations to solve a single problem.

7.8.4 Sequence differences

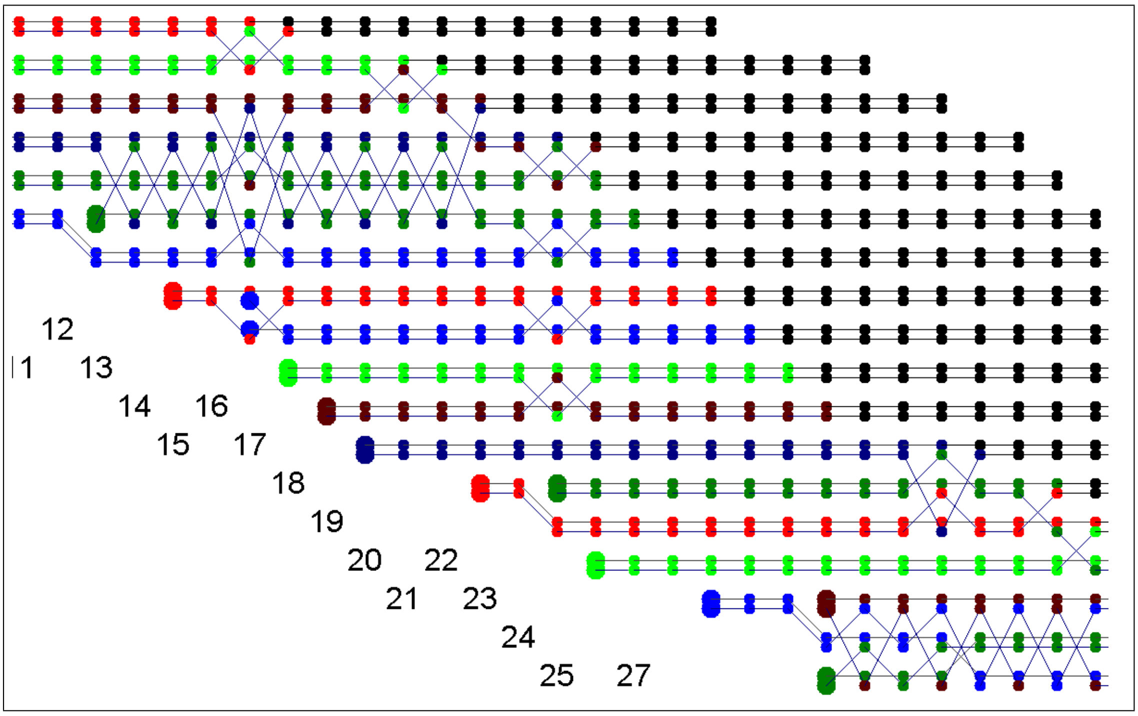


Figure 7.5: Comparison of sequencing

The viewer has facilities to load the results of multiple experiments and combine them side-by-side, in order to better understand the differences between the way final sequences were created. This can sometimes shed light on to the reasons for differences between the results obtained in different executions of an experiment. The user can zoom in or zoom out of the display, or move through the displayed sequence. An example of such a comparison between the earlier two sequences is shown in figure 7.5, where the viewer has been zoomed in.

This sort of comparison was particularly useful when used in conjunction with an output of the schedule costs for the partial schedules that the decision support system selects at each stage. One thing that this revealed was that the best final sequences were sometimes obtained by selecting slightly worse solutions at intermediate stages. This implies that the benefits to be gained by implementing an exact solution for the sequencing of the partial-solutions that the system is aware of at any time may be limited.

7.9 Summary

A simulation was built to allow the designed decision support system to be tested. This simulation was described in detail in this chapter. It uses real historic data to create realistic problems for the decision support system and enact the decisions made, so that the consequences of earlier decisions are faced later.

A user interface and playback tool were developed to allow simple verification of the behaviour of the simulation and these were also described in this chapter. The method in which the simulation predicts positions for aircraft in the holding area is important and was also described along with the description of some improvements that were added to the system to cope with problems.

Finally, the sequence stability was considered. Another tool was developed to aid in visualising the circumstances under which sequencing changes occurred. When the sequencing performed under simulation was examined, the sequencing changes that did occur were seen to be justified.

The experiments used to evaluate the performance of the system are described in the next chapter, along with the experimental results.

Chapter 8

Results

8.1 Introduction

The decision support system described in this thesis was tested using the simulation detailed in chapter 7. Four main sets of experiments are described in this chapter, with details of the parameters used for each. Together, they illustrate the performance of the proposed decision support system under different operating conditions and allow an estimation of the possible benefits of implementing the system.

This chapter starts with a consideration of the input data that was tested, the parameters that were utilised and the expected outputs of the system. Experiments showed a distinct benefit from including taxiing aircraft in the sequencing decisions. The first experiment considers the effect of changing the time at which such aircraft are introduced into the system.

The effects of the constraints upon the system are important as an analysis could aid in understanding how the system performs and may help in tuning and improving the system. In particular, if some constraints have negligible effects, or effects that can be ignored in the presence of other constraints, then it is useful to know this. The second experiment was performed to evaluate the relative effects of the different constraints upon the sequencing. This experiment also provides an example of how the system could be used to evaluate the effects of structural modifications to holding areas, an extension which may be of use in future.

A live system is unlikely to have accurate taxi times for aircraft. When taxiing aircraft are to be included in the system it is important to understand how much effect taxi time uncertainty is likely to have upon the results. In particular, it is useful to understand, when developing an estimation method for taxi times, how accurate such a system would need to be. The third experiment investigates the effects of uncertainty.

The fourth experiment was performed to better understand the short-term effects of the sequencing decisions. Experiments were performed using partial datasets and the results showed that the performance can vary across the time covered by a dataset. Example or summary results are given for each of the four main experiments and an examination of the results is performed. Full results for these experiments can be found in appendices D to G.

Following the fourth experiment, results are presented to illustrate how the performance of the system varies as different parameters are changed. The chapter ends with some examples of the holding area movement and sequencing performed by the algorithm.

8.2 Experimental Configuration

The simulation described in chapter 7 presents a series of problems to the decision support system to be solved, enacts the decisions made, increments the current simulation time and presents the new problem to the decision support system. There are many parameters which can be changed in the simulation and decision support system. The aim in this section is to detail the actual configuration used for the experiments.

8.2.1 Supplied datasets

The input data for the experiments consisted of real historic recorded data which was provided by NATS. Ten datasets were produced from the provided data and were used by the departure system simulation to build a set of realistic problems to be solved by the decision support system. The test datasets include the weight class, departure route and speed group for each aircraft. Various historical timings were also provided for each aircraft, including the time at which each aircraft left its stand, reached the holding area and took off. One required item, the holding area entrance that was used, was not included in the supplied data, so had to be estimated, as explained in section 8.2.7.

Details of the holding area configuration in use at the time, the number of aircraft in the dataset which had take-off time-slots (CTOTs) and the sizes of the aircraft are shown in table 8.1.

TABLE 8.1: Details of the datasets

Dataset	Holding Area	Aircraft	CTOTs	Light	Medium	Heavy
1	27R	329	172	0	235	94
2	27L	305	90	0	205	100
3	09R	331	74	1	227	103
4	27L	330	100	1	239	90
5	27R	326	77	1	211	114
6	27R	345	98	1	244	100
7	27L	259	42	2	193	64
8	09R	318	149	0	229	89
9	27R	341	158	0	245	96
10	27L	331	89	1	213	117

8.2.2 Number of evaluations

There is a stochastic element to the decision support system. For this reason it is not adequate to perform each experiment once for each configuration. Instead, each configuration under evaluation in an experiment was tested one hundred times using a batch process on the University of Nottingham high performance cluster[1]. The mean, minimum and maximum results are shown in the results tables presented later.

8.2.3 Simulation Step Size

A step size of one minute was used in these experiments. This ensured that the problem would usually change between iterations of the simulation so that the decision support system only had one attempt to solve each sub-problem. A smaller step size would be more realistic, but it would then be impossible to judge the performance of the decision support system from the final results alone as the system could have taken multiple attempts to solve the same problem.

8.2.4 Allocated Runway

The allocated runway is important as it affects both the holding area configuration that is in use and also some of the separation rules. The holding area configuration is important as it can greatly change the constraints upon the re-sequencing. The runway is usually changed near the middle of the day. The datasets were created by considering the runway at use at the time so that each dataset covered a single runway and spanned around half a day. The test datasets span all three of the usual runway configurations.

By ensuring that each dataset was for a single runway configuration, the issues related to runway allocation could be ignored. Runway allocation would be better performed prior to aircraft leaving the stands, as otherwise aircraft may taxi to the holding area only to have to taxi back again to another holding area, thus wasting time and fuel. Runway allocation is beyond the scope of the research programme presented by NATS and addressed in this thesis.

In order to have more experimental data and to be able to compare the flexibility of the sequencing for the different runways, experiments were performed using each dataset on each of the three common runway configurations. Where useful, the results are shown for all three configurations and the results for the actual runway configuration in use at the time are highlighted. Care must be taken when comparing datasets that were tested against a runway configuration other than the real one used. For example, it is not feasible to compare real results from one runway configuration against automated results from another, even when the same aircraft were involved.

8.2.5 Holding area traversal times

The traversal time of the holding area is important as it limits how early an aircraft can be scheduled for take-off, as shown by equation 6.1. One problem with simulating the movement of the aircraft through the holding area is that there can be a great variability in holding area

[1]A central cluster of over 500 dual 2.2GHz Opteron processor nodes with 2GB RAM each, running an operating system based on Suse Linux.

traversal times even along the same path. There can be even more variability between paths. Even if precise estimates of minimum traversal times could be made, it is questionable whether it is advisable to use minimum times. It is far safer, in terms of seeking take-off sequences which are easily achievable, to use unnecessarily long prediction times to ensure that the planned take-off sequence is actually achievable.

Rather than have to predict the exact traversal time that will be needed for each path, large, safe values are used for the traversal times. Aircraft should be able to get to the runway much faster than these values assume, but using these values makes the schedule easy to achieve. For this thesis, all required holding area traversal times were assumed to be equal and independent of the path taken, as only good paths were used. Aircraft assigned to slower paths, which may take longer to traverse, will be those which are overtaken and will therefore have extra time available to traverse the holding area. Additionally, as this is a *minimum* traversal time, most aircraft will actually have even longer to traverse the holding area.

The holding area traversal time actually serves three purposes in an automatically generated schedule. Firstly, it limits the re-sequencing to ensure that there is plenty of time for an aircraft to traverse the holding area. This is so that the schedule is less likely to get disrupted by delays. Secondly, it allows for some unpredictability in the arrival time at the holding area. Thirdly, it is used in the evaluation of the final schedule to calculate the actual take-off times.

Only the third of these purposes is relevant when assessing a manually produced schedule. When assessing a manual schedule, the large leeway in the traversal time is unnecessary so a smaller traversal time could be used. Obviously the traversal time is irrelevant when assessing the real take-off times.

The high minimum traversal time can be seen to be overly constraining upon the automated sequencing algorithms, however, if the searches can find good sequences under these constraints then they will be able to find schedules at least as good when there is a human controller present to loosen these constraints when it is desirable to do so.

8.2.6 Separation rules

Standard separation rules were used throughout these experiments, as provided by NATS. In these tests there were no extra MDI constraints added, beyond those which are permanently present on some routes and are already included in the separation rules given in section 2.3.

8.2.7 Holding area entrance allocation

In order to allocate a path through the holding area, the decision support system needs to know the holding area entrance at which each aircraft arrives. This information was not available in the supplied data so was estimated based upon data that was available.

The allocated stand was known for each aircraft so each aircraft was allocated to the holding area entrance that was most convenient for its allocated stand. This is a pessimistic approach as it assumes the Ground Movement Controller (GMC) will not perform any sequencing prior to the aircraft reaching the holding area. In practice the ground movement controller can make the sequencing a lot easier for the runway controller by intelligent allocation of aircraft to entrances according to expected sequencing requirements.

The implemented approach has been chosen because it ensures the minimum workload for GMCs. More complex approaches could assume some marshalling is being performed by the GMC and could, therefore, assign aircraft to entrances in a way which will increase the general flexibility for overtaking. For instance, approaches which allocated all north-bound aircraft to a specific entrance often improved the solutions the system could obtain. Even better performance was obtained with a load-balancing approach, for instance, alternating the entrance to which aircraft were allocated. Another approach could be to attempt to identify useful overtaking and make it possible, for example, to allow an aircraft with a tight CTOT to overtake by assigning it to a specific entrance, or to assign aircraft which have to wait for the start of their CTOT to an entrance where they can easily be overtaken. There are many different methods that could be used by the GMC but the nearest-to-stand allocation was adopted as it is the solution of lowest work for the GMC and pilots.

Despite the fact that the selected approach may be pessimistic, it is safer for an evaluation of the system performance. A live system should perform at least as well as the system performs under these conditions. Importantly, no expectations are made of the GMC so there is no reliance upon the GMC performing exactly as predicted.

8.2.8 Holding area arrival times

A holding area arrival time for each aircraft is a required input for the decision support system, although this time may not be entirely accurate. Taxi times are intrinsically uncertain but most of the uncertainty is thought to occur near to the stand. Many of the stands at Heathrow are on long cul-de-sacs, and contention can occur between users of the cul-de-sac or aircraft at nearby stands at push-back time. This contention can cause unexpected delays and contribute greatly to the uncertainty of the time from push-back to holding area arrival. Once aircraft are taxiing, the uncertainty is thought to be low, unless a runway crossing is required (for example, to travel between terminal four and the northern runway).

Real taxi times were provided in the input datasets. To ensure that the final sequences that are evaluated are always comparable, the real holding area arrival time is always used for evaluation of final sequences. Where the effects of taxi time uncertainty are to be simulated, this was achieved by modifying the predicted values that were supplied to the decision support system algorithms by adding or subtracting prediction errors based upon the remaining taxi time before providing the times to the decision support system.

As a further expected benefit of any future decision support for take-off sequencing, it is reasonable to expect that predicted take-off times (and possibly also predicted holding area

TABLE 8.2: Real controller results and predicted results for real and first-come-first-served sequences

		Controller		Predicted		First-Come-First-Served	
Dataset	Runway	CTOT	Delay	CTOT	Delay	CTOT	Delay
1	27R	10	99805	22	136252	99	438130
2	27L	9	127891	19	180378	68	753106
3	09R	13	128763	17	150052	61	814325
4	27L	5	120893	12	151869	51	581071
5	27R	14	140075	16	162834	56	547461
6	27R	5	107786	7	152713	55	718855
7	27L	4	96235	17	214445	33	667507
8	09R	10	99507	20	129234	57	267753
9	27R	6	117894	17	140460	100	462251
10	27L	11	120329	20	167114	46	551127

delay) could eventually be transmitted back to GMCs to specify how tight the take-off timing is for that aircraft. A GMC could then strategically prioritise aircraft at taxi-way intersections based upon the expected delay in the holding area. In this way, aircraft would be more likely to reach intended take-off times and the slack built into the traversal times to allow for taxi time unpredictability could be decreased.

8.2.9 Other assumptions

Not all suggested re-sequencing would necessarily be accepted by the real runway controller. The controller may have a number of other objectives to keep in mind, for instance expressed pilot preferences or a declaration by a pilot of a delay due to some difficulty. In these experiments, however, there is no controller to make the decisions so it has been assumed that all advice will be accepted.

8.3 The Manually Produced Sequences

It is useful to begin by considering the performance of the controllers for each of the supplied datasets. Table 8.2 presents details of the controller performance for each of the ten datasets supplied. The first two columns specify the dataset number and the runway configuration in use at the time. The CTOT compliance and total delay (in seconds) are then given for the real schedule the controller achieved, in the columns labelled 'Controller'.

One of the elements of the decision support system is a method for predicting take-off times for aircraft given a take-off sequence. The take-off time prediction system was applied to the sequence the controller used in order to predict take-off times for the aircraft. The consequent number of CTOTs missed and the predicted total delay are shown in the columns labelled 'Predicted'. Finally, the take-off time prediction system was applied to the first-come-first-served sequence, where aircraft take off in the sequence they arrived at the holding area. The results of doing this are shown in the columns labelled 'First-Come-First-Served'.

When evaluating the manual and first-come-first-served sequences, the start of the CTOT take-off slot was ignored in calculating the earliest take-off time. In practice, controllers or airlines can sometimes re-negotiate the CTOT on some aircraft, meaning that they can occasionally release aircraft earlier than the CTOT seems to allow. If this time was not ignored, the predicted take-off times would be wildly pessimistic in these cases. Ignoring these can, at worst, mean that predicted take-off times are earlier than may seem to be allowed by the CTOT, so a lower predicted delay would be expected than the real value. In fact, the accumulated delays over a sequence, discussed in section 8.7.3, mean that this rarely affects many aircraft anyway, but does affect a few, as will become obvious from the discussion in section 8.8.1. Ignoring the start of CTOT is only ever done when evaluating the manual sequences, never for the automated sequences, as it should not be assumed that a controller will be able to re-negotiate CTOTs.

Two conclusions can be drawn from the results in table 8.2:

8.3.1 Comparisons with predicted take-off times are 'unfair'

The first conclusion is that the predicted total delays can be observed to be much higher than the the real total delays for aircraft. Explanations for this are presented in section 8.7.3, where the effects of the take-off time prediction system are considered in more detail and over shorter time periods. It is important to realise that any comparison where the take-off time prediction system is used for the evaluation of one element and real times are used for the evaluation of the other element will be unfair upon the element which uses the take-off time prediction system. It will be observed later, that the decision support system compares admirably with the real take-off times, even with the unfair comparison method.

8.3.2 Re-sequencing take-offs is vital

The second conclusion is that the first-come-first-served sequence is an extremely bad take-off sequence. The importance of the sequencing performed by controllers cannot be underestimated given the total delay that would be expected for the first-come-first-served take-off sequence. This importance is illustrated by the large disparity between these delays and even the delays for the predicted times for the real take-off sequence. It is obvious that the controllers obtain a far better average delay for aircraft than would occur if aircraft were not re-sequenced.

8.4 The Effects Of Considering Taxiing Aircraft

A runway controller will usually only consider the aircraft that are already within the holding area when determining the take-off sequence. If time permits, the controller may also look at the aircraft that are on the taxiway close to the holding area and use their experience of which departure routes are usually allocated to which flights. However, the times when this will be most useful are when the controller is likely to be most busy and so is less likely to have the time to be able to do this. The experimental results presented in this section show the effects upon the performance of the system of knowing about aircraft on the taxiways.

8.4.1 The planning horizon

A number of experiments were performed with different amounts of knowledge of the taxiing aircraft. The length of time before holding area arrival at which the aircraft are added to the system is here called the *'planning horizon'*. At the extremes, this means giving the system knowledge at the time of push-back from the stand or only giving it knowledge of aircraft that are already within the holding area.

It is useful to understand the effects of the planning horizon as it is obviously easier to obtain accurate short term predictions of taxi times than longer term predictions. Consequently, the task of any taxi time prediction system to provide these to a decision support system would be simplified if only short term predictions were necessary. Additionally, larger planning horizons mean incorporating more aircraft in the search, so the search space is much larger, making the job of the decision support system harder. Understanding the effects of the planning horizon can avoid the need for an unnecessarily large planning horizon.

For these experiments, perfect knowledge of the taxiing aircraft was assumed. However, no aircraft were added to the system before the actual push-back time of the aircraft, as a great deal of uncertainty is associated with the push-back times. Given the known holding area arrival times and push-back times from the stands, the system was given knowledge of the aircraft a specified number of minutes before the aircraft arrived at the holding area, or at the time of leaving the stand, whichever was later. Twenty different experiments were performed for each dataset, varying the planning horizon from zero to nineteen minutes. Each experiment was executed one hundred times and the mean, best and worst results were calculated for both delay and CTOT compliance.

The mean delay and CTOT compliance results are presented in figures 8.1 to 8.8. The experiments were executed for each of the three holding area structures (shown in red, green and blue for the 27L, 27R and 09R holding areas respectively in figures 8.1 to 8.8), thus allowing a comparison of the effect of the holding area upon the sequencing as well as a comparison of the effects of the planning horizon and the expected system performance compared with the real controller performance. The vertical axis shows the mean delay in the generated schedules or the mean number of CTOTs that were missed as a percentage of the delay in the real schedule or number of CTOTs that were missed by the real controllers. For example, executing the decision support system using dataset 1, using the 27R holding area, with a seven minute planning horizon, the system found schedules with an average delay 87% (87095 seconds rather than 99805) of that of the manual schedules and only missed 40% of the CTOTs that were missed by real controllers (missing 4 CTOTs rather than 10). The actual delay and CTOT compliance figures are presented in tabular form in tables D.1 to D.5 in appendix D.

8.4.2 The effects of not considering taxiing aircraft

The system was actually designed with the intention of taking taxiing aircraft into consideration. A decision was made to freeze the holding area traversal paths for aircraft within the holding area. This decision had an adverse effect upon the performance of the system in the absence of knowledge about the taxiing aircraft as it was over-restrictive on the re-sequencing. Consideration of the results in figures 8.1 and 8.2 shows that the overall performance of the system appears to be similar to that of the controllers with the real take-off times. The system improves on the controller sequences in some cases but does worse in others. With no look-ahead to what is currently taxiing, the traversal path allocation system can work extremely badly. This effect can be observed in figures 8.1 and 8.2 where both the delay and CTOT compliance values for the automated results are worse for some datasets than were the schedules the controllers achieved.

To understand the effect on the path allocation system, consider three aircraft arriving at the same holding area entrance, one at a time. The first aircraft will be assigned the default path through the holding area, which is often the shortest path. It is unwise to assign it a longer path than necessary and the system is unaware of any other aircraft that may need to overtake it. Now, consider the situation where the second aircraft should overtake the first. The only way to overtake an aircraft on the shortest path is to use a short-cut path, most of which enter the runway away from the end and so may not be suitable for the largest aircraft. This is far from an optimal path allocation for the two aircraft and also restricts later overtaking. If a third aircraft then needs to overtake the first two, there is probably no way for it to do so. Moreover, even if this aircraft must take off more urgently than the second aircraft (so if a sequence of 312 is better than 213, for example) there is no way to 'undo' the path that was assigned to the second aircraft.

Conversely, when aircraft are sequenced while still on the taxiways, as was envisaged when the system was designed, paths can be re-allocated to aircraft as long as the system is aware of the requirement for overtaking prior to aircraft reaching the holding area. In this case, the first aircraft would be re-allocated to a slower path providing the system became aware of the second aircraft before the first reached the runway. The desired 321 or 312 sequences described above could then be attained.

There are only two solutions for this problem; either to allow path-reallocation to aircraft within the holding area, or to take the taxiing aircraft into consideration. The former could easily be performed as long as the system had knowledge of the position of the aircraft within the holding area. The path allocation system would then assume the aircraft could be re-allocated to a new path providing it had not passed the point in the holding area where its current and new paths diverged. This is the way the controllers currently work, attempting to keep flexibility for as long as possible.

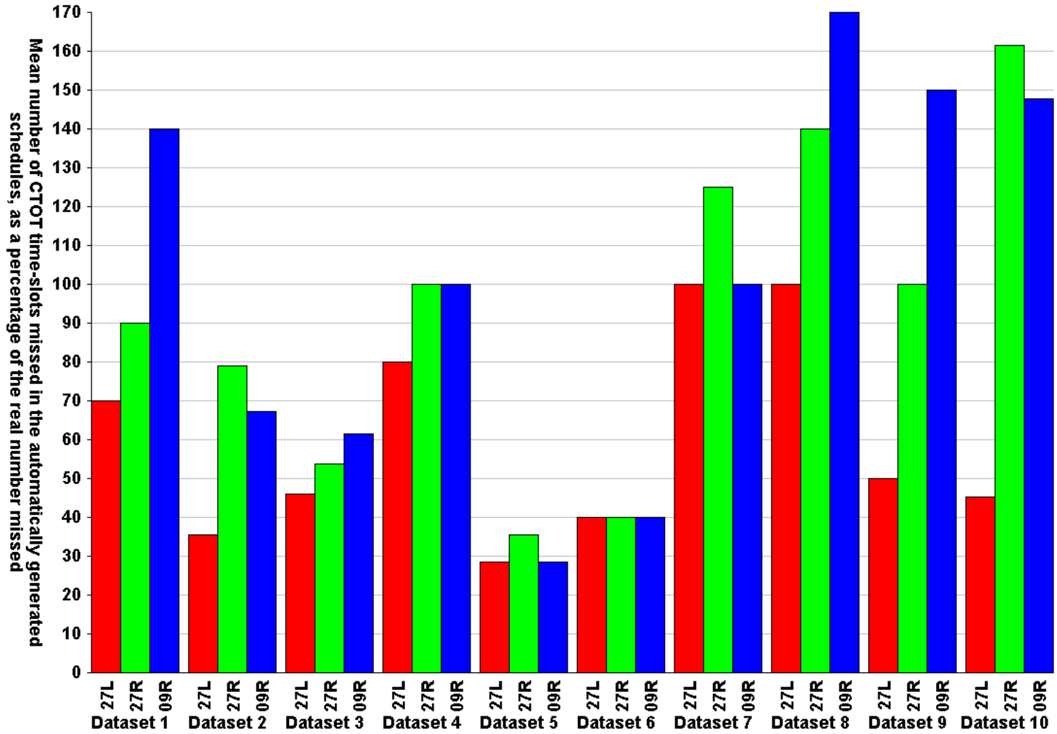


FIGURE 8.1: Number of CTOTs predicted to be missed by the decision support system given no notice of aircraft before arrival at the holding area, as a percentage of the number missed in the real take-off schedule

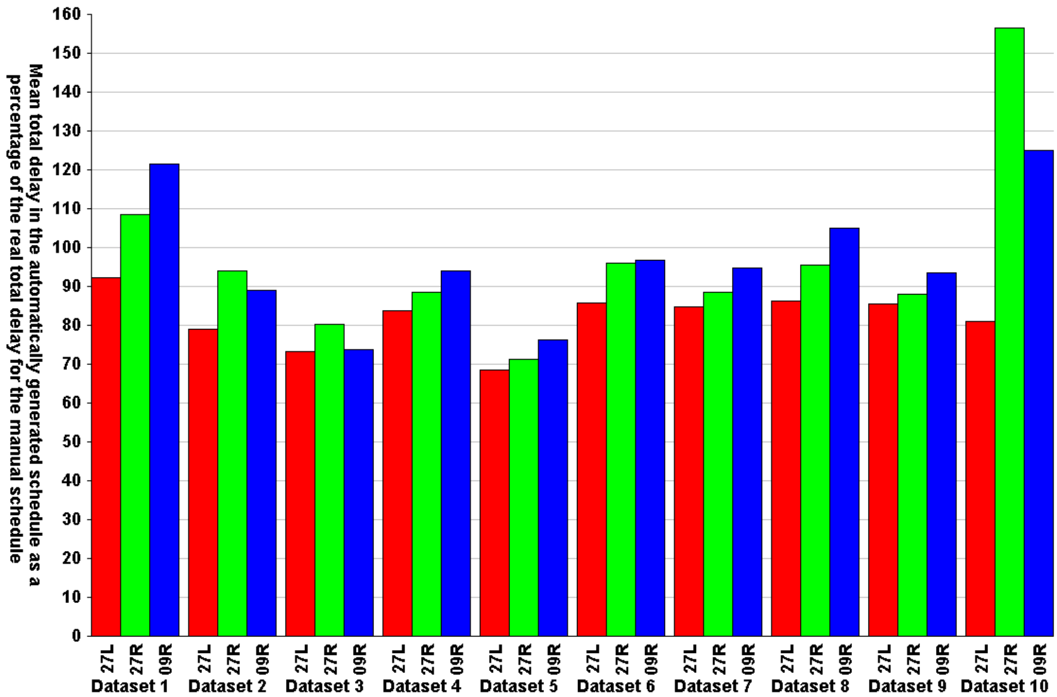


FIGURE 8.2: Total holding area delay predicted for aircraft for schedules generated by the decision support system given no notice of aircraft before arrival at the holding area, as a percentage of the total holding area delay in the real take-off schedule

8.4.3 CTOT compliance vs delay with reduced re-sequencing possibilities

Allocating paths without consideration of taxiing aircraft, then fixing the path allocation as soon as the aircraft enters the holding area, leads to an extremely restricted path allocation system. The effects of the restricted path allocation can be seen in both the CTOT compliance and delay. For example, if two aircraft are both assigned to the same traversal path then they cannot overtake each other. If the second aircraft has a tight CTOT then it may not be able to perform the overtaking required to achieve it. Alternatively, if the first aircraft has a CTOT which will require it to delay until the start of the take-off time-slot then all aircraft behind it will be delayed, regardless of whether they have a CTOT or not. If both of these cases apply then there is no way for both aircraft to achieve their CTOTs. The CTOT compliance of the generated sequence is therefore expected to decrease if insufficient notice of taxiing aircraft is provided.

As far as the delay is concerned, not only does the restrictive path allocation reduce the flexibility of the re-sequencing, so the lower delay sequences may no longer be achievable, but there is often also a trade-off between CTOT compliance and delay. CTOT misses are penalised so severely that aircraft with CTOTs will be prioritised to allow take-off within CTOT, even if this adversely affects the throughput of the sequence or considerably increases the delay for other aircraft. The effects of this will be seen in section 8.5 where the delay is considerably reduced when the CTOT restrictions are lifted. It is sensible then, to expect that the delay will be worse for the schedules for which the CTOT compliance is better, but higher delays can mean that later aircraft are then less likely to be able to meet CTOTs and the effects can escalate.

8.4.4 The benefits of considering the taxiing aircraft

The results in figures 8.3 and 8.4 show that the performance of the system improves considerably even when only notified of aircraft one minute before holding area arrival. With a planning horizon even as small as this, the system is already finding take-off sequences which have better CTOT compliance and delay for most datasets.

It was mentioned in section 3.6.1, that the major gains in implementing a decision support system to help with solving the arrival scheduling problem were considered to be from giving controllers a greater visibility of arriving aircraft [151]. The results obtained using the decision support system described in this thesis support this view for the departure process as well.

8.4.5 Comparing automatically and manually produced sequences

The results in figures 8.1 to 8.8 compare the performance of the take-off schedules produced by the decision support system against those produced by the real controllers. Figures 8.5 and 8.6 imply that, with a planning horizon only five minutes before the holding area arrival time, the

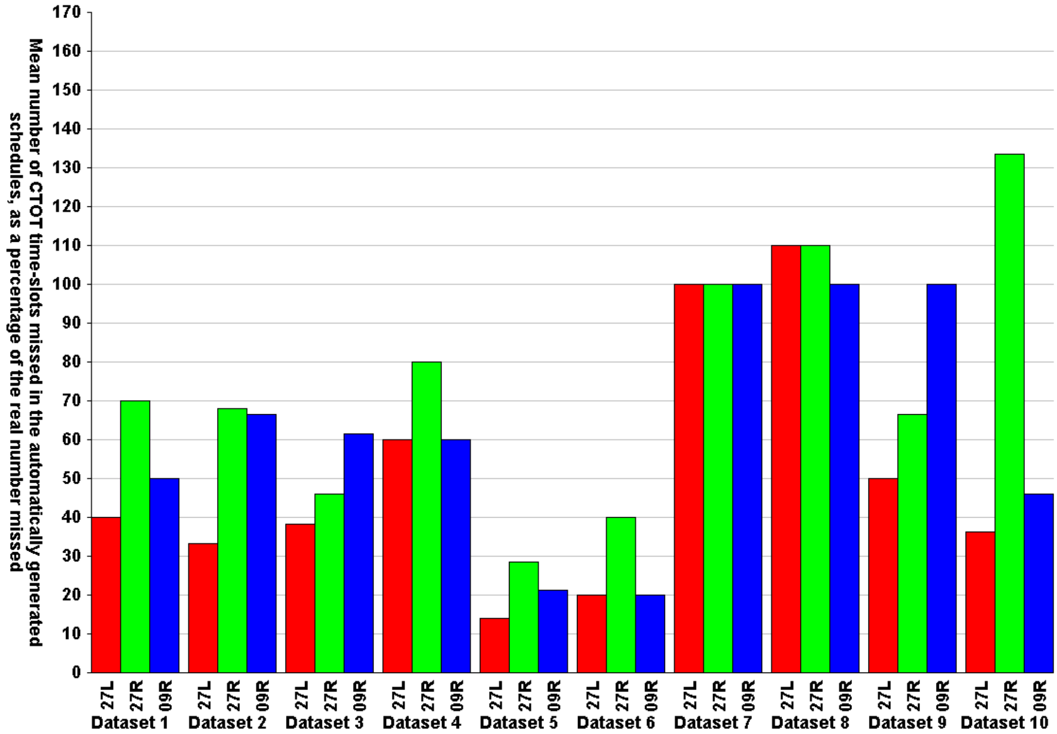


FIGURE 8.3: Number of CTOTs predicted to be missed by the decision support system given 1 minute notice of aircraft before arrival at the holding area, as a percentage of the number missed in the real take-off schedule

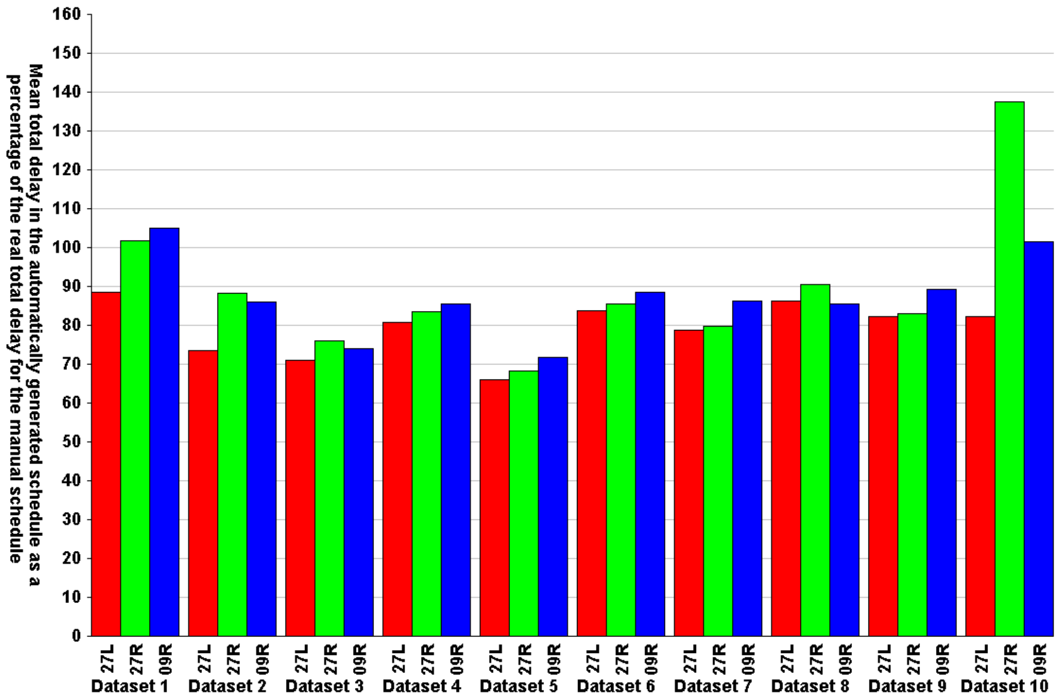


FIGURE 8.4: Total holding area delay predicted for aircraft for schedules generated by the decision support system given 1 minute notice of aircraft before arrival at the holding area, as a percentage of the total holding area delay in the real take-off schedule

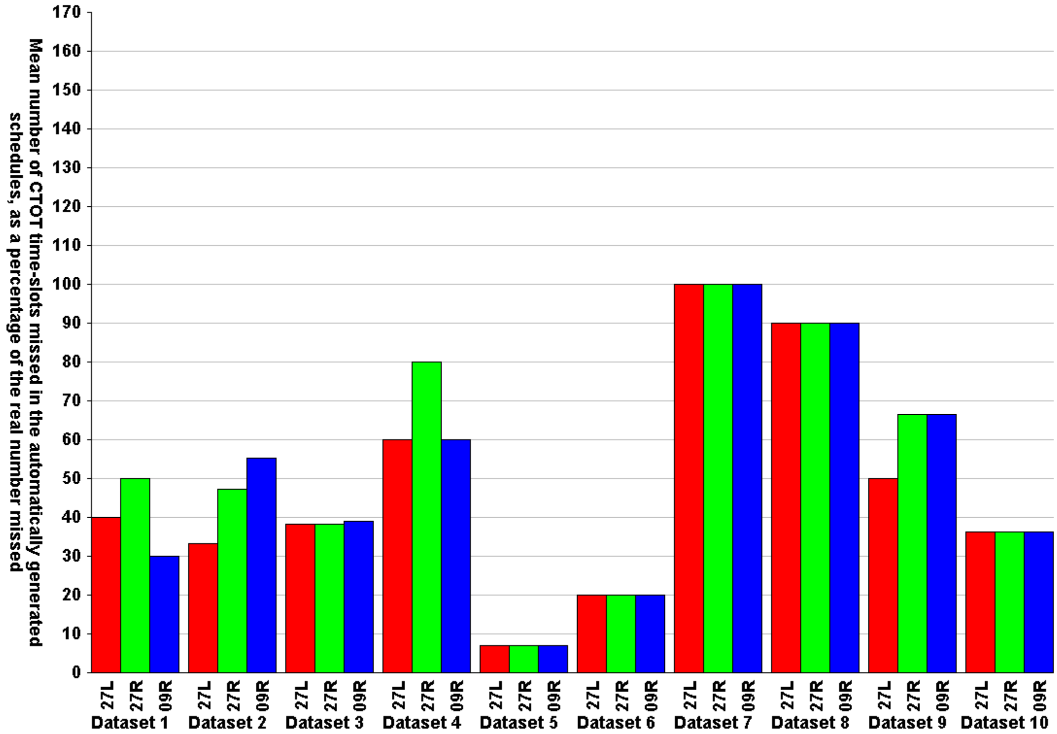


FIGURE 8.5: Number of CTOTs predicted to be missed by the decision support system given 5 minutes notice of aircraft before arrival at the holding area, as a percentage of the number missed in the real take-off schedule

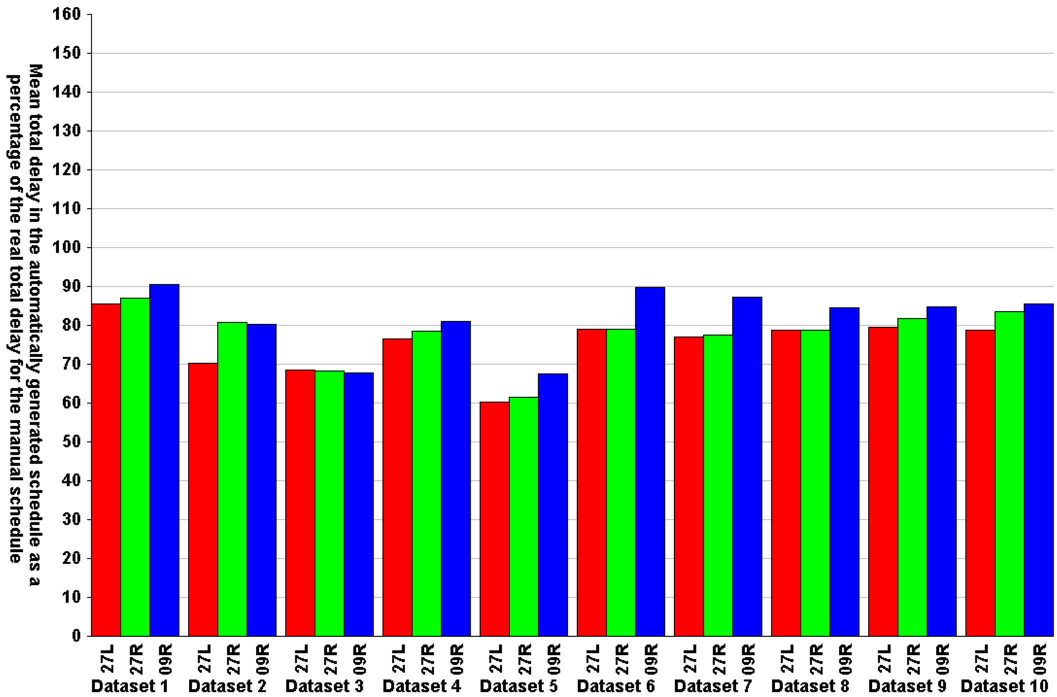


FIGURE 8.6: Total holding area delay predicted for aircraft for schedules generated by the decision support system given 5 minutes notice of aircraft before arrival at the holding area, as a percentage of the total holding area delay in the real take-off schedule

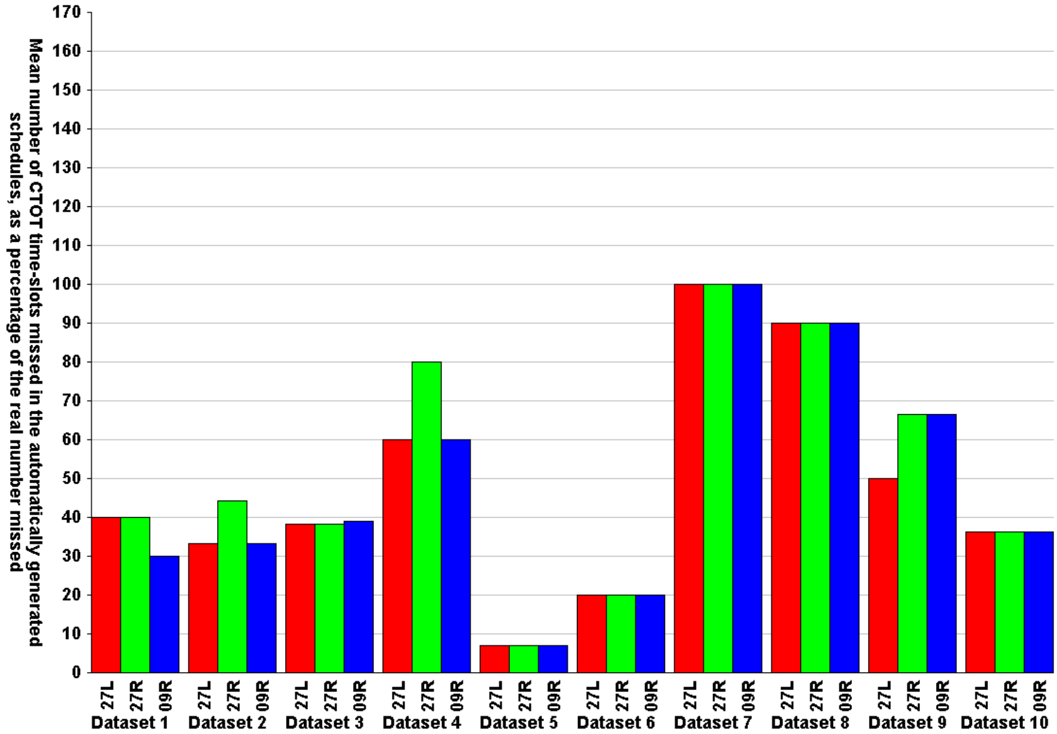


FIGURE 8.7: Number of CTOTs predicted to be missed by the decision support system given 15 minutes notice of aircraft before arrival at the holding area, as a percentage of the number missed in the real take-off schedule

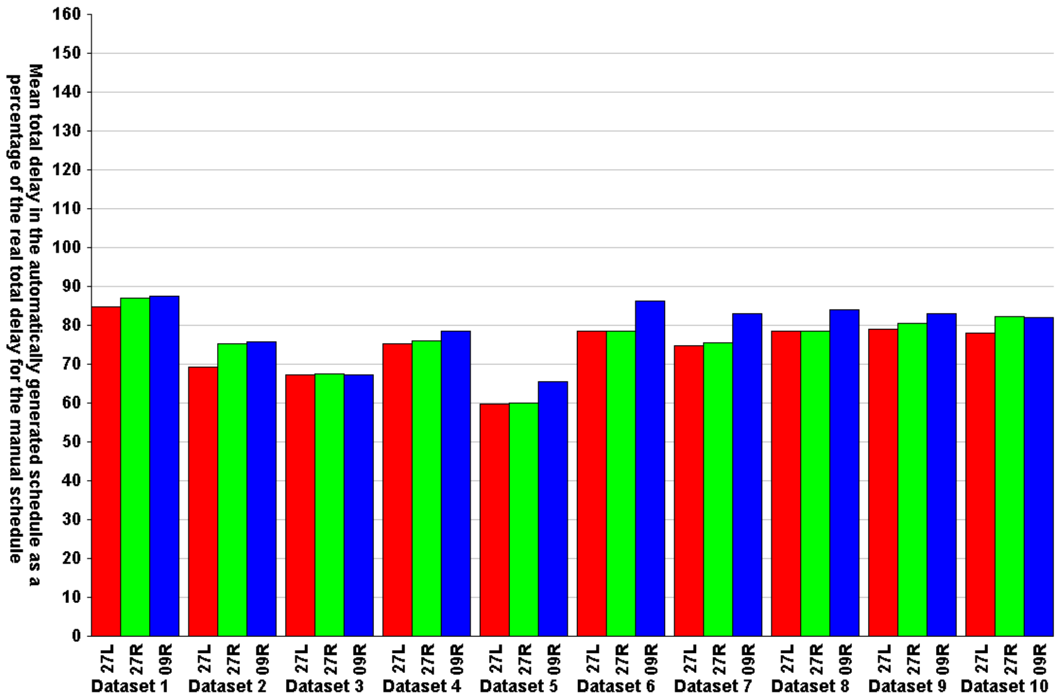


FIGURE 8.8: Total holding area delay predicted for aircraft for schedules generated by the decision support system given 15 minutes notice of aircraft before arrival at the holding area, as a percentage of the total holding area delay in the real take-off schedule

system already considerably outperforming the controllers in terms of both CTOT compliance and delay for aircraft under the simulation. The results for fifteen minutes taxi time knowledge that are presented in figures 8.7 and 8.8 indicate that the system performs even better with a longer planning horizon.

It will be observed in section 8.7.3, that the take-off time prediction system which was used to evaluate the automatically produced schedules is pessimistic in the results it gives and so is unfair upon the automated solutions. Even with this unfair comparison, the automated solutions can be observed to provide much lower total delays than the manual solution and this provides significant evidence for the high value of this approach. In fact, the manual solutions are very good, with very few separations above the minimum, but there are a number of reasons why the automated solutions are so superior in terms of delay and CTOT compliance:

Firstly, maximising throughput is not the same as minimising delay, as discussed in section 2.13. The controller is trying to maximise throughput rather than directly to minimise total delay. Minimising the delay will maximise the throughput over the long term but the converse is not true.

Secondly, the searches had more knowledge about the future than the runway controller did. Good sequencing by the search was often the result of knowing which aircraft were going to be arriving later and sequencing the current aircraft with this in mind. Furthermore, without the knowledge about what was going to happen in future, controllers have to allow extra flexibility in the sequencing, which may not be needed by a system that knows what which aircraft are likely to arrive soon. For example, if an automated system can identify that there is no possibility of a north-bound aircraft arriving to fill a gap in the schedule then there is no need for the system to provide the flexibility for such an aircraft to overtake to achieve the take-off position. Without such knowledge, it is worth ensuring that there is sufficient flexibility to allow an aircraft to do so if necessary, even if this may result in a slightly higher overall delay.

Thirdly, the searches also considered more aircraft at once than the controllers would usually do, thus enabling them to find solutions which are not achievable when considering less aircraft.

Finally, the real controllers were working amidst the uncertainty of the real world rather than with precise, historic, data. Real take-off times can vary from predictions and the controllers had to allow for this fact. Real schedules had to be designed to be able to take account of real world change, although this does mean that a controller can capitalise on advantageous occurrences in the real world as well as having to allow for disadvantageous ones.

8.4.6 Over-restrictive holding area movement

Some of the rules on holding area movement may need greater flexibility in a live system, but this is something that should only be done with controller guidance at the time. The decision support system is more constrained in what it can do in a holding area and only ever uses good paths through the holding area. Some of the re-sequencing that is currently rejected as being complicated may be acceptable to some controllers in some circumstances. For example, a real controller may at times want to use longer or more difficult paths in order to achieve the desired sequencing, but this should not be assumed of the controller. These rules will differ at different times and between controllers but will be more flexible than the current (deliberately constraining) path assignment heuristics. The current system is deliberately over restricted so that the results obtained will be achievable in practice. Better results should be obtainable when combining the system with a real controller, as in the desired decision support system.

8.4.7 Reliability of the system

In order to understand the reliability of the system when taxi times are known in advance, tables 8.3 and 8.4 are presented as examples of the performance of the system. Full results for all of the datasets can be found in appendix D. These tables show the best (minimum) and worst (maximum) performing results for the various planning horizons as well as the mean performance over the one hundred executions. The rows of the table relate to the planning horizon the system was working with for the taxiing aircraft. These examples were selected as they illustrate how the delay changes as the CTOT compliance changes. In each case the system is remarkably consistent in the sequences it selects despite the stochastic element to the tabu search.

8.4.8 The effects of the holding area configuration

A difference in performance between the different holding area configurations can be observed in figures 8.1 to 8.8. Part of this difference is due to the difference in separation rules between the holding areas and part is due to the restrictions imposed by the holding area configuration.

Discussion with controllers revealed that the 09R holding area that was used for these experiments is a lot less flexible than the 27R and 27L holding areas. One controller explained that they avoided the problems by making increased use of intersection entry points to the runway and by the GMC performing intelligent partial sequencing of aircraft prior to delivering them to the holding areas. The 09R holding area has since been redesigned as a part of the current work on the new terminal five, however the original restrictive layout was kept for this work in order to see the performance of the system in a less flexible environment.

The effects of the constraints imposed by the physical layout can be seen in the next section and are surprisingly small. To understand this, it is important to remember that the system performs sequencing differently to, and often earlier than, the runway controllers. With sufficient knowledge of the aircraft on the taxiways, sequencing is performed while the aircraft are still taxiing towards the runway. The conclusion drawn from this relatively low effect is that the relatively poor performance with the 09R holding area is due to the separation rules in place.

Table 8.3: Dataset 1, 27R results

Planning Horizon	CTOTs missed Mean	Min	Max	Total delay (s) Mean	Min	Max
No notice	9.00	9	9	108461	108461	108461
1 minute	7.00	7	7	101789	101789	101789
2 minute	6.00	6	6	90858	90858	90858
3 minute	6.00	6	6	87942	87942	87942
4 minute	5.00	5	5	86256	86256	86256
5 minute	5.00	5	5	86833	86745	86842
6 minute	4.00	4	4	87599	87442	87682
7 minute	4.00	4	4	87095	87024	87204
8 minute	4.00	4	4	87095	87024	87204
9 minute	4.00	4	4	87095	87024	87204
10 minute	4.00	4	4	87030	87024	87714
11 minute	4.00	4	4	86879	86150	87079
12 minute	4.00	4	4	86867	86150	87104
13 minute	4.00	4	4	86884	86150	87104
14 minute	4.00	4	4	86888	86150	87104
15 minute	4.00	4	4	86905	86150	87104
16 minute	4.00	4	4	86907	86150	87104
17 minute	4.00	4	4	86906	86150	87079
18 minute	4.00	4	4	86903	86150	87079
19 minute	4.00	4	4	86903	86150	87079

Table 8.4: Dataset 8, 09R results

Planning Horizon	CTOTs missed Mean	Min	Max	Total delay (s) Mean	Min	Max
No notice	17.00	17	17	104517	104517	104517
1 minute	10.00	10	10	85146	85146	85146
2 minute	10.00	10	10	85554	85554	85554
3 minute	9.00	9	9	85140	85140	85140
4 minute	9.00	9	9	84801	84801	84801
5 minute	9.00	9	9	84223	84223	84223
6 minute	9.00	9	9	84175	84175	84175
7 minute	9.00	9	9	83887	83887	83887
8 minute	9.00	9	9	83893	83887	84043
9 minute	9.00	9	9	83795	83779	83935
10 minute	9.00	9	9	83843	83779	83995
11 minute	9.00	9	9	83927	83887	84043
12 minute	9.00	9	9	83690	82712	84055
13 minute	9.00	9	9	83802	82712	84055
14 minute	9.00	9	9	83654	82712	84067
15 minute	9.00	9	9	83602	83356	84055
16 minute	9.00	9	9	83537	82712	83947
17 minute	9.00	9	9	83540	82712	83965
18 minute	9.00	9	9	83554	82712	83965
19 minute	9.00	9	9	83554	82712	83965

8.4.9 Continuing to increase the planning horizon

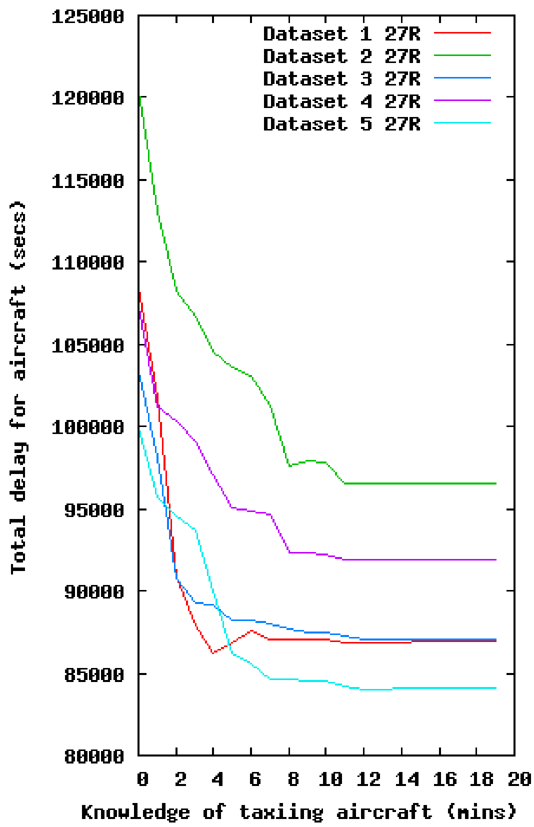


FIGURE 8.9: Delay vs planning horizon for 27R, datasets 1 to 5

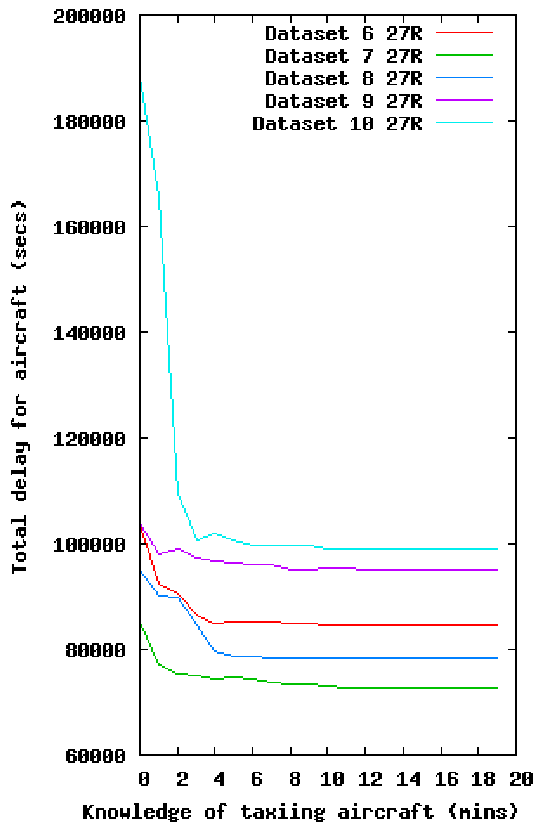


FIGURE 8.10: Delay vs planning horizon for 27R, datasets 6 to 10

Figures 8.9 to 8.14 show how the delay in the generated sequences varies as the planning horizon is changed. The vertical axis shows the mean total delay over one hundred executions of the algorithm. The horizontal axis shows the number of minutes prior to arrival of an aircraft at the holding area that the system is given information about the aircraft. It should be noted that the scale of the vertical axis differs between the graphs, so care should be taken in using these for any comparison of how the performance of the system varies across datasets. Figures 8.1 to 8.8 are better for that purpose.

The delay benefits of the automatic scheduling can be seen from figures 8.9 to 8.14 to further increase as the system is given increasing knowledge of taxing aircraft, however the benefits often plateau with around seven to ten minutes knowledge of taxiing aircraft. The plateauing effect was also seen in the experiments performed for and described in [12].

In theory, the best sequences could be found more consistently by further increasing the planning horizon (and including aircraft before they leave the stand) to the point where the system can predict the later effects of early sequencing decisions. However, there are problems with doing this. Firstly, the intrinsic uncertainty in a live environment would make accurate long-term predictions improbable. Secondly, the size of the search space would increase enormously.

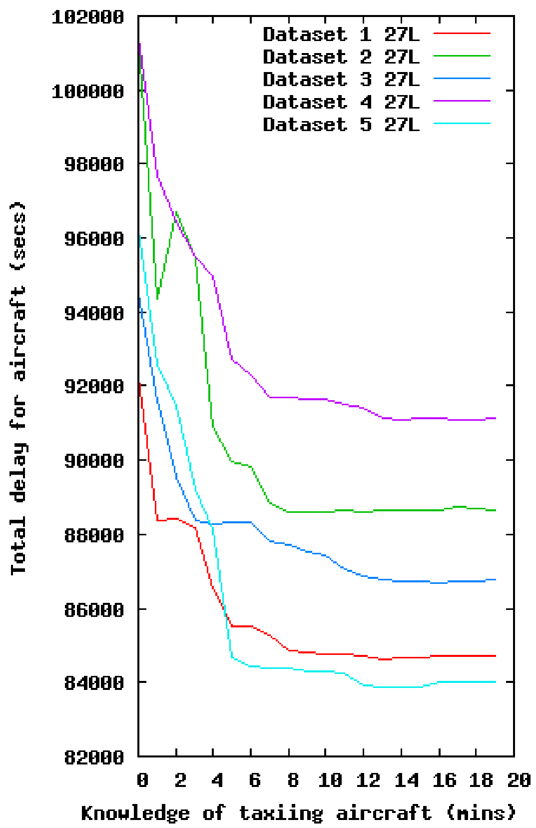


FIGURE 8.11: Delay vs planning horizon for 27L, datasets 1 to 5

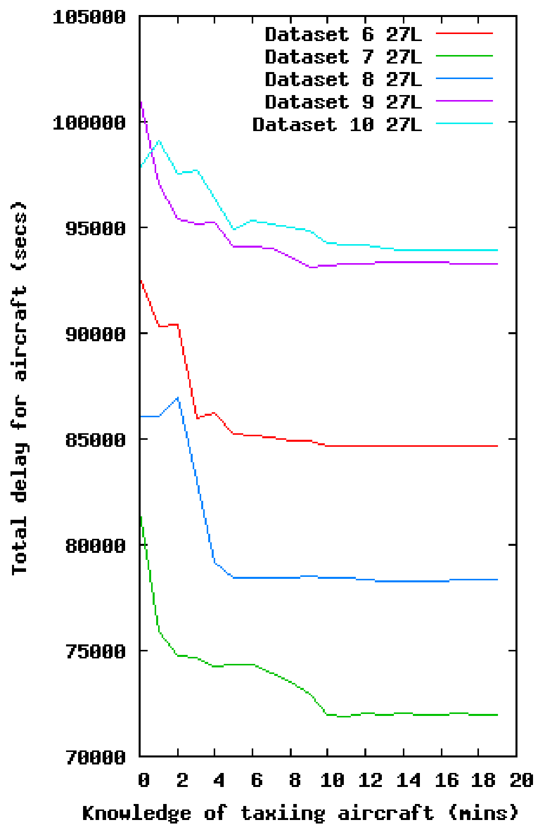


FIGURE 8.12: Delay vs planning horizon for 27L, datasets 6 to 10

The meta-heuristic approach would soon find it very hard to investigate the good solutions within the desired run-time. It is even possible that the number of possible sequences would mean the performance degraded, rather than improved, as relatively less time may be spent considering the sequencing of the first few aircraft and these are more important.

8.4.10 Sub-problems and global problem optimality

One observation that was made was that the best solution to the current problem, considering only aircraft on the taxiways and in the holding area, was not always the best solution for the global problem. It was observed that better overall sequences could sometimes be obtained by selecting a slightly worse (for example, slightly less equitable) solution for some of the earlier sub-problems, as this sometimes left increased flexibility for late re-sequencing. In general, however, a better local solution gave a better global solution. Without knowing in advance the aircraft which will enter the system later, it is impossible to predict whether it is worth adopting a worse solution to the local problem in order to increase flexibility later.

8.4.11 The freezing time

The concept of a *'freezing time'* was introduced in section 2.12.3 to refer to the length of time before take-off at which the position of an aircraft in the take-off sequence is frozen. In the

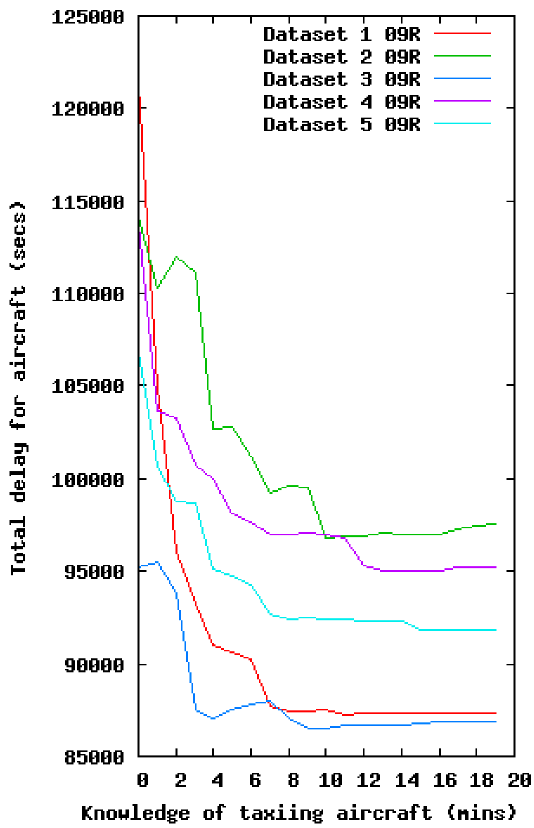


FIGURE 8.13: Delay vs planning horizon for 09R, datasets 1 to 5

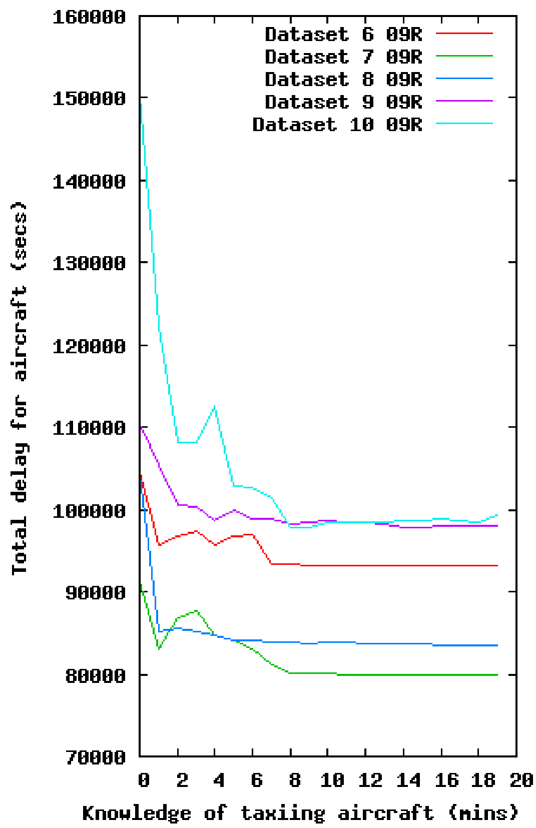


FIGURE 8.14: Delay vs planning horizon for 09R, datasets 6 to 10

experiments performed for this thesis, the freezing time was assumed to be two minutes before the predicted take-off time for all aircraft. An investigation of the effect of the freezing time was performed and the conclusions were presented in [12] along with a discussion of the practical meaning of the freezing time. The conclusions are summarised below.

A close correlation was seen between the effects of increasing the freezing time and decreasing the planning horizon. Consideration of what these times mean to the aircraft that are free for sequencing means that this is expected behaviour. If the freezing time is increased, then the planning horizon should be increased by a similar amount in order to obtain a similar performance from the system. Doing this can be seen to be moving the sequencing window earlier in time. The relationship breaks down after a while however, as aircraft are only added when they leave the stand (so pushing the planning horizon further back in time will not introduce these aircraft any earlier) and there are more constraints upon the sequencing once aircraft reach the holding area (as discussed above) so in some cases moving both the planning horizon and freezing time earlier can reduce the sequencing constraints upon the same set of aircraft and lead to better results.

8.5 The Effects Of Constraints

It was important to understand the effects of the various constraints and combinations of constraints upon the sequencing for three main reasons. Firstly, to understand the reasons for the performance of the system, secondly, to motivate the structure of the neighbourhood for the meta-heuristic search and thirdly to understand how the departure system could be improved.

8.5.1 Experimental details

The effects of the constraints were investigated by relaxing different combinations of the primary constraints and then examining the consequent performance of the system. A planning horizon of twenty minutes before holding area arrival was used, unless the aircraft pushed back later than that, in which case the aircraft was added at its push-back time. A minimum holding area traversal time of two minutes and a freezing time of two minutes before take-off were used. The experiment was repeated one hundred times for each dataset in order to alleviate the effect of the stochastic element of the tabu search.

Example results for one dataset are presented in table 8.5. Similar tables for the remaining results can be found in appendix G. The tables show the mean results obtained for each dataset and holding area configuration. The top row of results shows the performance of the system with all of the usual constraints in place. The rows below show the improvement in the number of CTOTs missed and the percentage decrease in the delay (compared to the delay when all constraints were present) when each of the four types of constraints were removed. When CTOT constraints have been ignored, results will always show that no CTOTs were missed. Since the CTOT improvement value is not relevant in these cases, a hypen ('-') is shown for the associated CTOT entries in the table to indicate this fact.

The four constraints which are considered for removal are the departure route separations (labelled S), the wake vortex/weight class separations (labelled W), the effects of the holding area (labelled H) and the CTOT take-off time-slots (labelled C). For example, the row labelled *'W H'* shows the improved performance of the system when both weight class and holding area effects have been removed.

Departure (SID) route separations were removed by assuming all SID/speed separation rules required one minute separations regardless of the route allocated to the aircraft or the speed group of the aircraft. Wake vortex separations were eliminated by assuming that all wake vortex separations were one minute regardless of the weight classes of the aircraft involved. The effects of the holding area structure were eliminated by assigning a unique imaginary path with a traversal time of two minutes to each aircraft rather than performing the usual path allocation and feasibility check. The CTOT time-slot constraints were eliminated by merely removing the CTOTs allocated to aircraft.

TABLE 8.5: Effects of constraints, dataset 1

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				**4.00**	**86903**	4.00	84698	3.00	87382
S	-	-	-	**1**	**20%**	1	19%	0	20%
-	W	-	-	**0**	**10%**	0	11%	0	12%
-	-	H	-	**0**	**3%**	0	0%	0	2%
-	-	-	C	**-**	**14%**	-	12%	-	13%
S	W	-	-	**1**	**29%**	1	27%	0	29%
S	-	H	-	**1**	**21%**	1	19%	0	21%
S	-	-	C	**-**	**33%**	-	31%	-	34%
-	W	H	-	**0**	**13%**	0	11%	0	12%
-	W	-	C	**-**	**23%**	-	21%	-	21%
-	-	H	C	**-**	**14%**	-	12%	-	14%
S	W	H	-	**1**	**29%**	1	27%	0	29%
S	W	-	C	**-**	**41%**	-	39%	-	41%
S	-	H	C	**-**	**33%**	-	31%	-	34%
-	W	H	C	**-**	**23%**	-	21%	-	21%
All				**-**	**41%**	-	39%	-	41%

8.5.2 Interpreting the results

In all cases, except dataset 2, the holding area structure had a surprisingly low effect upon system performance, although in some cases the 09R holding area had a significant effect. In dataset 2, there is an aircraft which must be held for a long time (for reasons explained in section 8.8.1) and this can block or restrict the less flexible holding areas, so the holding area structures have a larger effect in this dataset. Comparison for dataset 2 of the results with CTOTs removed and those with both CTOTs and holding area constraints removed shows that the holding areas have very little effect once the CTOT constraints have been removed. This provides support for the effects of the holding area being due to the aircraft with CTOTs needing to overtake or be overtaken.

The relatively low effects of the 09R holding area structure were the most surprising as this was thought to be quite restrictive. These results show that the path allocation and feasibility test system are working well and that the restriction to only using good paths (to ensure acceptability to a controller) is not restricting the sequencing too much. This does not mean that the holding area is having little effect upon the sequencing. There are often many diverse sequences with similar throughput and delay so it is quite possible for the holding area to eliminate many different good sequences and to force the adoption of very different sequences than would otherwise be adopted, but the new sequences are obviously of very similar cost to the rejected good ones.

The constraints which have the most effect upon the performance are the downstream constraints, primarily the route and speed based separations, but the CTOT take-off time-slots also have a significant effect. The fact that the route and speed separations are the most constraining is not really surprising due to the proximity of Heathrow to several other airports, the congestion in the local airspace and the consequent flow control measures applied to reduce the take-off rates along busy routes. Unless something is done to improve the situation or change the departure routes, there is little that can be done about this at the local airport.

The fact that the downstream constraints are so important for the sequencing problem is not surprising given how busy the airspace around Heathrow is. It does imply that a the two-stage approach such as that described in [4, 5], where the downstream constraints are ignored in the first stage, is unlikely to be as effective at Heathrow as at airports where the surrounding airspace is less congested.

Since CTOTs are usually applied to control flow through the busy European airspace, it is interesting to note the heavy penalty paid by aircraft on the ground for these airspace congestion control measures, as the effects of these may not always be obvious. It is surprising that the CTOT constraints have such a great effect upon the delay. Examination of what is happening reveals that the objective function ensures that aircraft which arrive late for their CTOT overtake aircraft that are already in the holding area. From the point of view of CTOT compliance this is a good thing, however, it is not only extremely unfair but it may also force the rest of the aircraft to adopt less desirable traversal paths. For example, aircraft may need to be moved into positions where they can be overtaken. This can also occasionally force overtaken aircraft to take off in a less efficient sequence, as the overtaking needed to obtain the more efficient sequence may no longer be possible. Removing the CTOT time-slot constraint ensures that better take-off sequences can be attained and so can reduce the delay for aircraft. The presence of CTOTs can be observed to have a large effect upon the positions of aircraft in the take-off sequences generated by the system and these results indicate the detrimental effect that this is having upon the overall delay.

The SID-route constraints are the most restricting in all cases. Each dataset tends to have a surplus of either north-bound or south-bound aircraft, depending upon the current flight paths over the Atlantic but, even in datasets with an overall surplus for one direction, there are still times when there is a deficit at the holding area of aircraft going in that direction and consequently wasted throughput. The generated schedules show that there is actually significant spare capacity for aircraft to take-off from Heathrow, provided that they take off at the right times and along the right departure routes. Of course, these are not the times that passenger or airline preferences would favour, as the capacity would have been absorbed already otherwise.

The wake vortex constraints also have a significant effect upon the delay for the aircraft. They are roughly comparable to the effects of the CTOTs but not to the route/speed separations. In some cases, they have more effect than the presence of CTOT constraints and in some cases less. Given that wake vortex separation rules are often used to estimate airport capacity, this is perhaps surprising.

In all cases, a considerable further reduction in delay can be obtained by removing any two of the SID, wake vortex and CTOT constraints rather than just one of them, and even more benefit can be seen from removing all three. It is apparent that both wake vortex separations and SID separations matter and indeed, from the scale of the extra delay benefits gained by removing both, they appear to be relatively independent. Furthermore, the CTOT time-slots are constraining the solutions in such a way that they still have an effect when one or both of the SID-route and wake vortex constraints have been removed. The effects of these three major constraints seem to be relatively independent.

This indicates that all of the constraints should be considered in the sequencing and it is not possible to assume that the effects of one will be subsumed by the effects of others. For example, it is not possible to assume that the sequences can be built by considering only the downstream constraints (CTOT and SID/route constraints) and that increased wake vortex separations can be subsumed into the natural increased SID separations, even though the SID separations are the most constraining.

The holding area structure is, perhaps, an exception. As more of the other constraints are removed, the effects of the holding area structures reduce rapidly and soon disappear altogether. This result is sensible as the holding areas only restrict sequencing rather than increasing separations so, once enough constraints have been removed that the sequencing is less important, the structure of the holding area will have no effect.

Finally, it should be noted that, with the future introduction of larger aircraft (for example, the Airbus A380) and the consequent expected increase in wake vortex separations, the wake vortex separations are likely to start to have more effect than they currently do. However, the full effects of these cannot be evaluated until the separation rules have been finalised and the rough times of departure of such aircraft can be determined.

8.5.3 The meaning for capacity increases at Heathrow

These results imply that the biggest delay benefits for passengers would be gained by a reconsideration of the downstream constraints and that wake vortex constraints are not actually the biggest limitation on the runway capacity.

Unless the simulations show that the holding area structures are restricting the resequencing, then any additional cost for laying down more concrete would be hard to justify. Conversely, an examination could be performed into the effect of closing some holding area paths or runway entrances or into building on part of the holding area using the same techniques that were used for the evaluation here.

The adoption of mixed mode for the runways at different times has been discussed in the 2003 White Paper on aviation, [66]. The best that can be expected from mixed mode is to eliminate the wake vortex separations between aircraft. A similar constraint modification approach could be used to evaluate the effects of doing this, but the results presented here imply that a re-consideration of the downstream flow constraints would benefit the system more.

8.6 The Effects Of Uncertainty

The experiments presented so far have assumed perfect knowledge of the taxi times for the aircraft in the system. In practice, this perfect knowledge is unlikely in a live situation, although taxi times are thought to be fairly predictable once the aircraft have started to taxi towards the holding area.

Most aircraft at Heathrow are parked on stands on long cul-de-sacs. Leaving the stand often means being pushed back into the cul-de-sac, then starting the engines and taxiing around to the holding area. Due to the necessity to push back either into a cul-de-sac or directly onto the taxiways, there is a high possibility for contention between the push-backs of multiple departing aircraft, arriving aircraft which need to use the cul-de-sac and aircraft on the taxiways. As this push-back contention and the varying engine start-up times are considered to be the two major elements of any taxi time uncertainty, aircraft are not added into the system until they have pushed back.

Once aircraft have pushed back and started to taxi, the taxi time should be fairly predictable, especially if the taxi speeds of the other aircraft on the taxiways are also considered. However, a taxi time prediction system would require considerable additional research and is beyond the scope of this thesis.

The decision support system described here assumes that it is provided with predicted arrival times and arrival entrances for aircraft, in the same way that the data is provided by the simulator that was built for evaluating the system. The system performance would be expected to change as the accuracy of taxi time information changes. Evaluating this will give an estimate of the likely performance of the system in a live situation and will also help in determining how accurate a taxi time prediction system will need to be in order to gain benefits from a system.

Ten different experiments were repeated for each dataset. In the first experiment, the system was provided with perfect taxi time information for aircraft from the point at which they pushed back from the stand. A planning horizon of twenty minutes was used. Using a long planning horizon means that the system should perform well in the deterministic case. It also means that taxi times for aircraft in the system will be longer so variations in taxi time will also be greater and should be more obvious.

A further nine experiments were then performed where a random taxi time error percentage was determined in advance for each aircraft. Each time a taxi time had to be passed to the decision support system, the remaining taxi time was modified according to this value before being used to predict a holding area arrival time. The first of these nine experiments selected a random error from -10% to +10% for each aircraft, the second selected from -20% to +20%, the third from -30% to +30% and so on, to the ninth which was from -90% to +90%.

As there is a random element to both the search itself and the random taxi time error determination, each experiment was again repeated 100 times. Each execution used a different random seed so the taxi time errors were different for each one.

8.6.1 Experimental results

The results of these experiments are shown in appendix E and sample results are shown in tables 8.6, 8.7 and 8.8. Each table shows the results for a single dataset. The first column indicates the experiment to which the results in that row refer, the next three show the mean, minimum and maximum number of CTOT slots missed and the final three the mean, minimum and maximum delay. The first row shows the results for the manual sequence with predicted take-off times, the next the real results the controller attained and the third the results for the first-come-first-served sequence with predicted take-off times. The fourth row shows the performance of the system with known/certain taxi times. The remaining rows show the performance as the taxi time uncertainty is increased.

TABLE 8.6: Dataset 1, 27R results

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual	22	-	-	136252	-	-
Real	10	-	-	99805	-	-
FCFS	99	-	-	438130	-	-
Certain taxi times	4.00	4	4	86903	86150	87079
+/- 10% error	4.40	4	6	86449	84053	88349
+/- 20% error	4.86	4	6	86421	84426	89451
+/- 30% error	5.13	4	6	86829	84993	90475
+/- 40% error	5.20	4	7	87535	85023	91935
+/- 50% error	5.25	3	8	88726	84898	97292
+/- 60% error	5.33	3	7	90266	85330	98525
+/- 70% error	5.51	3	8	92029	86628	102383
+/- 80% error	5.98	4	10	97335	89996	109501
+/- 90% error	7.07	3	14	106423	93910	131590

In each case, the performance of the system decreased as the level of uncertainty increases, although in some cases the introduction of a small amount of taxi time uncertainty can actually improve the delay results. This may seem surprising but there are two reasons for this:

First reason: CTOT versus delay trade-off

The first reason for this is illustrated in table 8.7, where there is a trade-off between CTOT compliance and total delay. In most cases, the schedules that were found to have a lower total delay when prediction errors were introduced were also those where more CTOTs were missed. According to the objective function used, these were actually worse sequences (since CTOT misses are so highly penalised), so the system was actually performing worse in the presence of uncertainty rather than better.

Table 8.7: Dataset 10, 27L results

Experiment	CTOTs missed Mean	CTOTs missed Min	CTOTs missed Max	Total delay (s) Mean	Total delay (s) Min	Total delay (s) Max
Manual	20	-	-	167114	-	-
Real	11	-	-	120329	-	-
FCFS	46	-	-	551127	-	-
Certain taxi times	4.00	4	4	93933	93726	94276
+/- 10% error	4.00	4	4	94155	93509	94910
+/- 20% error	4.00	4	4	94451	93427	95620
+/- 30% error	4.00	4	4	94962	93523	98140
+/- 40% error	4.01	4	5	95802	94015	99124
+/- 50% error	4.04	4	5	96741	94662	101933
+/- 60% error	4.09	4	5	98265	95606	104209
+/- 70% error	4.24	4	6	101042	95796	114023
+/- 80% error	4.43	4	7	105832	98354	118079
+/- 90% error	5.00	4	10	116840	103228	133628

One important effect of the taxi time uncertainty can be seen in relation to aircraft 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.

Second reason: Possible benefits of uncertainty

The experiments were actually performed using each of the three holding area models, rather than just the one used in the recorded data. An interesting result occurred with dataset 10 and the 27R holding area, for which results are shown in table 8.8. In this case, with a 20% error, the average delay was actually lower than in the case where there is no uncertainty. However, this is at the cost of a much reduced worst case performance. One conclusion that could be drawn from consideration of only these results could be that it may be worth injecting some errors into taxi times in order to find better sequences.

One suggestion for an improvement to the system could be to inject these errors anyway, or to solve both the uncertain and exact problem at each stage and take the best result. The main problem with doing this is that in most cases the mean system performance is not actually improved by injecting taxi time errors and in all cases the worst case performance is reduced so the reliability of the system could actually be reduced. A secondary problem is concerned with determining whether the added uncertainty is helping or not. When examining the values

TABLE 8.8: Dataset 10, 27R results

Experiment	CTOTs missed Mean	Min	Max	Total delay (s) Mean	Min	Max
Certain taxi times	4.00	4	4	99054	98974	99079
+/- 10% error	4.03	4	5	98572	94882	103968
+/- 20% error	4.00	4	4	98727	94915	103497
+/- 30% error	4.03	4	6	99278	95476	110907
+/- 40% error	4.04	4	5	100180	95238	109459
+/- 50% error	4.08	4	5	101986	96047	121838
+/- 60% error	4.08	4	5	102234	96707	110643
+/- 70% error	4.16	4	6	104089	97833	121199
+/- 80% error	4.35	4	7	108118	100328	121271
+/- 90% error	4.84	4	8	115829	104211	136087

of the objective functions for sub-problems, it was found that, occasionally, the best overall sequences were actually obtained when the system failed to find the better solutions for certain sub-problems. However, the benefits to the global solution were nominal and were due to the specific details of the aircraft which arrived later. In general, a better solution to the sub-problem will usually yield a better overall solution. It is possible that, at times, the addition of the taxi time error was preventing the system from finding the best solution to the current problem, and hence accidentally finding the improved global solution.

Comparisons with controller-produced sequences

In general, the system still out-performs the controllers under the simulation conditions, even with moderate levels of uncertainty added to the taxi times. However, as the uncertainty increases, not only does the mean performance decrease, but the worst case performance decreases extremely rapidly. Indeed, with sufficient error in the taxi time predictions the system actually performs worse than if taxiing aircraft were not considered at all, as a comparison with the results in appendix D will show.

It matters where the errors are

The random prediction errors for each aircraft differed between executions of the experiments. The variation in performance across the different executions of each experiments 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. The experiments were inconclusive in terms of identifying where the system is most sensitive to errors in the taxi times. As the sequencing problem is a highly complex one, it was not clear whether it was the uncertainty associated with aircraft with tight CTOTs, aircraft of specific weights or specific departure routes or something else which most affects the results.

It can be conjectured, however, that the important aircraft will differ depending upon the other aircraft currently in the holding area, on the taxiways and (probably) on the aircraft which will arrive later. This is a possible area of further research but is beyond the scope of this thesis. If these effects could be determined, then it may be possible to prioritise these aircraft during the taxiing, thus improving the predictability of their taxi times and the performance of the system as a whole. Of course, the complexity may be too high to be able to do this, or it may rely upon information not available at the time of making the sequencing decisions, such as the types of the later aircraft.

8.7 Effectiveness Over Shorter Periods Of Time

The previous experiments were performed using the entire half-day datasets. However, during a half-day period there will be some times which are quieter than others. It can be difficult to understand how the system works over a shorter time period when analysing a dataset covering a half-day period. There may be benefits to using smaller datasets as it is then much easier to see how the accuracy of the take-off time prediction model differs according to how busy the holding area is at the time.

Differences between the real and predicted times can accumulate over time so a fairer assessment of the performance of the decision support system can be made if it is assessed over shorter periods of time. Over short periods of time, especially at busy times, the take-off time prediction model is very accurate but over longer periods of time the cumulative differences between real and predicted times can increase.

The supplied datasets were each divided into around ten to twelve smaller datasets, by taking samples of sixty consecutive aircraft starting from indices 0, 25, 50, 75, and so on, in 25 aircraft increments. The take-off positions and take-off times for the first ten aircraft were fixed to be the same as they were historically, giving the current problem a history of recent take-offs, which will affect the sequencing decisions. Overlapping datasets were used in order to reduce any bias introduced by imposing artificial boundaries between them.

Fifty aircraft is a small enough number to be able to easily see the local effects of any re-sequencing, but large enough that the effects of earlier decisions will be seen in later decisions so that the system is still not just solving a static problem.

8.7.1 Results with the smaller datasets

Experimental results showing the performance of the automatically generated sequences and a comparison with the real and predicted take-off times for the real take-off sequence are presented in the tables in appendix F. The results which are presented in appendix F show for each of the partial datasets, the real and predicted delay and CTOT compliance for the manually produced take-off sequences and the predicted delay and CTOT compliance for the automatically produced take-off sequences. The percentage changes in the delays are also given for comparisons between the different delay values. These results are summarised and illustrated by figures 8.15 and 8.16.

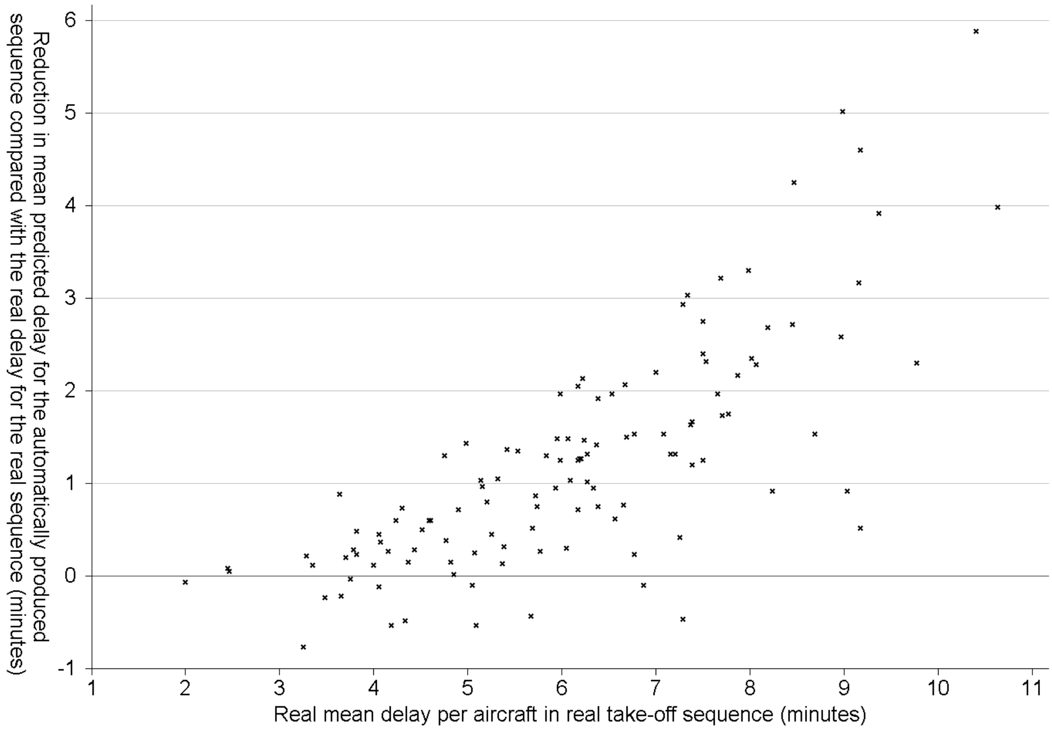


FIGURE 8.15: Graph of the decrease in delay per aircraft in the automated schedules versus the real delay for each partial dataset

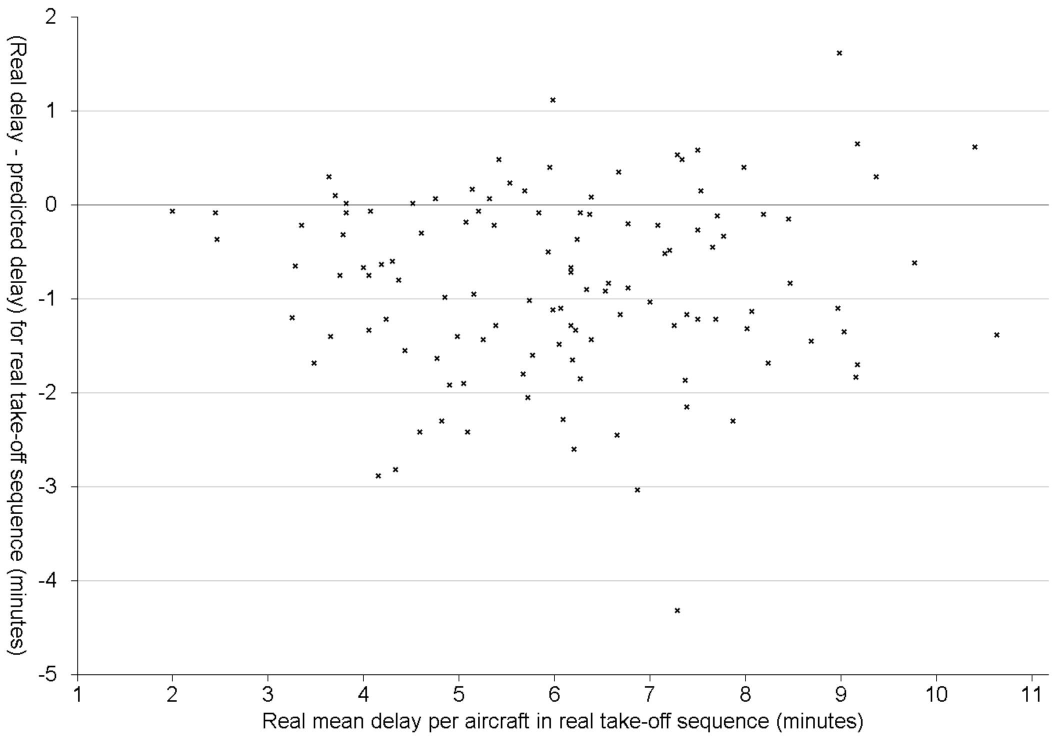


FIGURE 8.16: Graph of the difference between the predicted and real delay per aircraft versus the real delay for each partial dataset

Figure 8.15 illustrates the performance of the automated decision support system compared with the performance shown in the real schedule. Each point in the graph represents one of the partial datasets which was used for the experiments. The x-axis shows the real mean delay for the aircraft in the partial dataset, using the real take-off times. The mean delay was also calculated for the aircraft in the automatically produced schedules, and the reduction in the mean delay compared with the real schedule is shown on the y-axis. If the decision support system produced solutions which were of equal delay to the manually produced schedules then all points would be on the line $y = 0$. If the system consistently produced better solutions then the delay reduction would be positive and the points would be above the line $y = 0$.

8.7.2 Estimating how busy the departure system was

As the holding area gets busier and the number of aircraft queueing for take-off increases, it is intuitive that the holding area delay for aircraft will be longer. This intuition is supported both by anecdotal evidence and by the findings of researchers at other airports, [108]. Indeed, Idris et al. used this fact as the basis for a queueing model to predict taxi out time in [108], which they defined as the time from leaving the stand to take-off. The holding area delay in the real schedule can therefore be considered to represent an approximation of the runway queue size, and the consequent load upon the controller and departure system.

8.7.3 The accuracy of the take-off time prediction system

It is important to understand whether the take-off time prediction system is accurate or not, since it is used to predict the take-off times for the evaluation of an automatically generated take-off sequence. In particular, it is important to ensure that any perceived benefits from the automated sequencing are not merely an effect of an optimistic prediction of take-off times.

Figure 8.16 illustrates the effect of using the take-off time prediction system. As in figure 8.15, each point in the graph represents one of the partial datasets which was used for the experiments. The take-off time prediction system was used to predict the take-off times for the aircraft in each of the real take-off sequences, and to calculate the delay for aircraft. The x-axis shows the real total delay for the aircraft in the partial dataset, in minutes, using the real take-off times. The y-axis shows the difference between the real delay and the predicted delay for each partial take-off sequence, in minutes. A positive value on the y-axis indicates that the predicted mean delay was lower than the real mean delay for that take-off sequence, indicating that the take-off sequence was optimistic.

The results shown in figure 8.16, where the majority of points are below the $y = 0$ line, indicate that the take-off time prediction system is usually pessimistic, in that it usually predicts a higher delay for aircraft than is actually experienced in practice. The difference between the real and predicted values is lower for the quieter times (real mean delay less than three or four minutes) but the spread of the values is relatively consistent for delays between about four and nine minutes.

Figures 8.15 and 8.16 can be considered to be comparable in that the position on the x-axis of any given partial dataset will be the same in the two figures, since this represents the delay in the real schedule. The y-axis in figure 8.15 shows the predicted delay for the automated schedule compared with the real delay, and the y-axis in figure 8.15 shows the predicted delay for the real schedule compared with the real delay, so a comparison can give some indication of the improvement in the predicted delay between the real and automatically produced sequences.

8.7.4 Reasons for the take-off time prediction errors

It should be remembered that the separation rules can mean that delaying or advancing a single aircraft can affect the take-off times of later aircraft as well. Slight discrepancies can accumulate, making a substantial difference to the take-off times of aircraft later in a take-off sequence. With this in mind, some of the reasons for the prediction errors are considered below.

The durations of the quarter schedules, which are provided in appendix F, show that the take-off time prediction system usually deviates by less than two minutes per fifteen aircraft and that it is usually pessimistic (so predicted take-off times are later than real times), with occasional times at which the predicted values are earlier than the real values. Even a two minute change to the holding area delay for a number of aircraft can make a significant difference to the total holding area delay for the aircraft. The scale of the difference will depend not only upon the size of the reduction but also upon the position in the take-off sequence at which it was achieved. A reduced separation early in the sequence will affect the take-off times for many aircraft, so will have a larger effect upon the delay than if the reduced separation happens later in the sequence. This explains the fact that the effects were shown to vary widely in figure 8.16.

Controllers can reduce some separations

Controllers can sometimes reduce certain separations where they are not related to safety, as discussed in section 2.3.3. As these reductions are at the discretion of the controllers, the prediction system does not assume that this will be done, although it would be a simple matter to provide controllers with an interface to inform the system that they are permitting reduced separations and in what circumstance, so that the system could take advantage of these when scheduling. An examination of the actual flight times showed that the controllers often did manage to negotiate such separation reductions. Since the predicted take-off times assume a theoretical earliest take-off without allowing for reduced separations, there is no way the controller could attain these reduced levels of delay without lower taxi times or altering the separation rules.

Once a reduced separation has been negotiated, in the absence of a slack period, many later take-off times may also be advanced. Furthermore, the effects of subsequent separation reductions may be cumulative. For this reason, large discrepancies can accumulate even over the time period covered by a partial dataset, and enormous discrepancies can accumulate over the half-day time period of a fill dataset. Reduced separations were not assumed by the take-off time prediction system used in the experiments presented here and this has led to a large difference between the actual and predicted take-off times of the aircraft in some cases. This is especially apparent in the predicted delays for the overall datasets, presented in table 8.2.

Take-off times are unlikely to be accurate to the second

In the real-world take-offs rarely happen to the exact second that they could. Slight prediction errors can accumulate over a schedule. An implemented system would have constant feedback from the real world situation to correct for slight deviations between the predicted and actual take-off times, whatever the reason. This feedback is absent from the automated tests so cumulative discrepancies accrue through the busy times, leading to pessimistic predictions.

An excessive predicted holding area traversal time

A two-minute holding area traversal time was used for these take-off time predictions. This is actually excessive for take-off time prediction and is only used for the automated sequencing in order to provide flexibility to allow for uncertainty. When an aircraft actually took only one minute, in the absence of other constraints upon the earliest take-off time, the real take-off time would lead the predicted take-off time for that aircraft by a minute. Separation rules for later aircraft can then mean that this prediction error can be propagated through many of the predictions for later take-off times. This has similar effects to reducing the separations by a minute, but applies to any aircraft for which the holding area traversal time was the limiting factor on the earliest take-off time (i.e. whether e_i or f_i is the greater in equation 5.3 in section 5.5) rather than the common case where the separation rules are the limiting factor. Table 8.9 shows the results that are obtained when one-minute traversal times are used instead of the usual two minute times. Take-off times (and the consequent total delay for aircraft) are then much closer to the real times (and delay), but still lag behind them due to the reduced separations which controllers can use. Nevertheless, two-minute times were used for all experiments performed for this thesis, to allow leeway for slow taxiing through the holding area or uncertain taxi time

TABLE 8.9: Real and manual results for sixty second traversal times

		Controller		Predicted		First-Come-First-Served	
Dataset	Runway	CTOT	Delay	CTOT	Delay	CTOT	Delay
1	27R	10	99805	16	116632	92	418510
2	27L	9	127891	15	162258	68	753106
3	09R	13	128763	13	130972	59	795305
4	27L	5	120893	9	132129	49	561369
5	27R	14	140075	13	143574	56	547401
6	27R	5	107786	6	132253	55	698395
7	27L	4	96235	13	198965	33	652027
8	09R	10	99507	16	113078	52	248793
9	27R	6	117894	15	120136	96	441927
10	27L	11	120329	17	147614	46	532767

predictions.

8.7.5 Real vs Automated take-off sequences

Examination of the results for the percentage improvements, given in appendix F and shown in figures 8.15 and 8.16, shows that almost 90% of the time the automated sequences, even though evaluated with pessimistic predicted take-off times, are at least as good as the real sequences with real take-off times and over 80% of the time the improvement in the delay is at least 5%.

In addition to the delay improvement, the CTOT compliance for the automatically produced schedules is almost always at least as good as, and often better than, the compliance of the controller produced schedules, with real take-off times. Where the automated sequencing fails to improve upon the controller produced sequencing, this is due to issues with the take-off time predictions. This is shown by the fact that these are the times when there is a great disparity between the real and predicted take-off times for the real sequence. In these cases, the facilities available to the controller to improve upon the normally required separations allow the controller to outperform the automated sequencing.

The results in figure 8.15 imply that the potential benefits of the system are greater as the load on the holding area increases, and relatively low at quiet times, when there are very few aircraft waiting.

Importantly, the points in figure 8.16 and the consideration of the fact that the take-off time prediction system is actually pessimistic indicate that the benefits suggested by the points in figure 8.15 are not due to an optimistic take-off time prediction system but are indeed due to a significant improvement in the sequencing. In fact, the pessimistic nature of the take-off time prediction system reasonably implies that a more accurate system may allow even greater delay reductions to be obtained.

A distinctive trend can be seen in figure 8.15 such that not only are the solutions produced by the decision support system usually better than the manual schedules, in terms of having a lower mean delay per aircraft, but also the number of seconds of improvement tends to increase as the mean delay in the manual schedule increases. Indeed, the reduction in mean delay can be very large for the real schedules which have a large mean delay for aircraft. Conversely, when there is a very low delay associated with the manual schedule, there is little benefit from the decision support system and in some cases the decision support system had a negative mean delay improvement, implying that it failed to produce schedules which would be expected to perform as well as the controllers really did perform.

8.7.6 The performance at busy periods

There are three important observations that can be made about busy times. The first observation is that the sequencing is more important at busy times than at quiet times. Unnecessary separations apply a delay to the following aircraft until there is a gap in the take-off sequence. If there is a queue of aircraft then an extra one minute of separation can add a minute to the take-off time for every later aircraft. At quiet times more larger separations are inevitable, merely due to the lack of aircraft of the correct types, and these larger separations form natural gaps in the sequence. Furthermore, the fact that more aircraft will be affected by delays inevitably means that reduced separations are more likely to be used by controllers.

The second observation is that there are more aircraft available for any position in the take-off sequence when the holding area is busier. Often it is possible to eliminate an additional large separation by considering more aircraft at once and allowing a slightly higher positional shift for aircraft in a sequence than might be the case at quieter times. Two obvious examples of this are by grouping more heavy aircraft together or by simultaneously combining a larger route-based separation with a weight class separation. The decision support system usually sequences the aircraft before they reach the holding area, so more sequences are often available. If the sequencing is performed within the holding area, the current positions of the aircraft may limit the re-sequencing and prevent the best sequences from being achievable.

The third observation is that the controllers are busiest when there are more aircraft to control. The times when there are more aircraft available for sequencing, so the sequences should be better, are precisely the times at which the controller is busiest and tends to use a heuristic sequencing method which considers only a few aircraft at once.

8.7.7 Performance at quiet periods

The decision support system was designed to help controllers to cope with busy periods of the day. At times the departure system is quiet enough that there is no real need for complex sequencing. In the extreme case, if only one aircraft can reach the runway at a specified time then that aircraft should be sequenced next. At these times the designed decision support system does not perform so well due to the various constraints placed upon the sequencing. The main cause is the excessive minimum holding area traversal time that the system applies.

The system could be designed to work in the way that the controllers do; namely, 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, since these are the times the controller has time available to do that. In practice, if a controller has no need for a decision support system, they are unlikely to refer to it anyway.

On the other hand, at quiet times there is less pressure to take off as early as possible. This means that reduced separations are less likely to be used than at busy times, so the take-off time prediction system is likely to be more accurate, as indicated by figure 8.16. This means that, at busy times an implemented system could reasonably be expected to perform better than these results indicate (since the take-off time prediction system is likely to be more pessimistic at such times) but at quiet times it is unlikely to perform significantly better than indicated by these results.

8.8 Equity Of Delay

The question of equity of delay has to be addressed, as it would be easy for a decision support system to reduce total delay by strategically delaying specific aircraft. In order to verify the equity of the sequencing, the maximum, second and third highest sequence position delays were recorded for each sequence, along with details of the number of aircraft positionally delayed, the total positional delay and the total squared positional delay. The positional delay is here defined as the index of the aircraft in the take-off sequence minus the index of the aircraft in the arrival sequence at the holding area, with a minimum value of zero.

The results are taken from the experiments detailed in section 8.6 for each of the automated sequences, with a twenty minute perfect knowledge of the taxiing aircraft and are summarised in table 8.10. One row is dedicated to each of the manually produced sequences (labelled 'manual/real') and the holding area it was produced under is noted in the second column. Each row for the automated results shows the mean of one hundred executions for each dataset and holding area.

The first column specifies the dataset and holding area configuration used for the experiment. The second column specifies whether the row refers to manual or automated sequencing. The next three columns show the highest, second and third highest positional delay suffered by aircraft in the take-off sequences. The sixth column shows the number of aircraft which had a positive positional delay. The seventh column shows the mean positional delay for those aircraft with a positive positional delay, excluding the three with the highest delay, as the positional delay for these aircraft has already been shown. The last column shows the sum of the squares of the positional delay for all aircraft delayed, including the first three. The sum of the squares of the positional delays is useful as it gives an estimate of equity of this positional delay. A more equitable allocation of the same total positional delay should have a lower value in this column. The value for the second dataset illustrates this as the 25 or 27 position delays greatly increase the sum of the squares of the delay.

The results show that the system tends to positionally delay the occasional aircraft more than the controllers do and in this way can appear less equitable. However, the automated system usually delays less aircraft than the controllers do and, once the first three aircraft are excluded from consideration, the average positional delay for the other aircraft is similar to that in the controller sequences.

The decision support system does appear to be strategically delaying aircraft so as to obtain a better throughput and this needs to be investigated. Doing this is intrinsically unfair for those aircraft that are delayed, so some consideration should be put into whether the delay and

CTOT compliance benefits are worthwhile. The main problem with the sequencing seems to be the excessive positional delay applied to specific aircraft. In particular, the reason for the high positional delays in datasets 2 and 10 needs to be explained. However, the usually lower values for the total squared positional delay (especially once the squares of the most delayed aircraft have been subtracted) shows that this delay is at least as equitably spread amongst the delayed aircraft as the delay the controllers applied was.

8.8.1 Excessive positional delay

One of the aircraft in dataset 2 had an arrival time at the holding area of 14:34 but a CTOT of 15:29. As is usual in this sort of case, the CTOT was obviously re-negotiated as the real take-off time was 14:40. However, whatever modification of CTOT was attained, it was not recorded in the input data used in these experiments. Rather than attempt to guess a CTOT and possibly give the decision support an easier task than the controller actually had, the original CTOT was kept. This meant that the aircraft could not take off before 15:24 if it was to meet the CTOT, so would spend almost an hour in the holding area, during which time it would be expected to be overtaken by many other aircraft. In fact it is possibly more surprising that only 25 aircraft had to overtake it in that time, as the number expected to overtake is the number that will take off in that time, less the number already in the holding area waiting when it arrived. In the experimental results, the aircraft took off at the start of the take-off slot, which is expected as the cost associated with additional positional delay will be high by that point.

A similar aircraft exists in dataset 10, with an arrival time at the holding area of 15:49, a CTOT of 16:19 and a real take-off time of 15:57, seventeen minutes prior to the start of the CTOT take-off slot. In both cases, the automated system did not assume a re-negotiation of the CTOT, thus many aircraft had to overtake the aircraft which were awaiting the start of CTOT. In both cases, these aircraft had an excessive individual delay, which contributed to the total delay for aircraft, however the total positional delay for all of the aircraft was still not excessive. With a CTOT re-negotiation, the total delay would be expected to be further reduced.

A further fact to consider is that these aircraft were parked in the holding area for this delay time. Realistically, if the CTOT could not be re-negotiated, the aircraft would probably be parked at a remote holding position rather than congesting the holding area. In this case, however, their positions would be expected to restrict the sequencing that could be done, yet the system still performed well.

There are higher positional delays on some of the other aircraft in the automated sequences as well. In each case this was caused directly or indirectly by the presence of CTOTs. In most cases the delay was for an aircraft awaiting the start of a CTOT. In some cases, a small positional delay had been applied to aircraft without CTOTs but which were overtaken by similar aircraft which had tight CTOTs so were prioritised by the system. To understand the reasons, consider an example of three north-bound aircraft queueing for taking off, so they have

Table 8.10: Summary of positional delay in manual and automated sequences

Dataset	Sequencing	Highest delay 1st	2nd	3rd	Number delayed	Total delay	Mean other delay	Squared delay
1, 27R	Manual / Real	8	7	6	93	204	2.03	642
1, 27R	Automated	8	8	8	85	201	2.16	717
1, 27L	Automated	8	8	8	80	191	2.17	698
1, 09R	Automated	8	7	6	87	185	1.95	574
2, 27L	Manual / Real	7	7	6	114	205	1.67	551
2, 27R	Automated	25	8	8	85	223	2.21	1373
2, 27L	Automated	27	8	7	83	210	2.10	1335
2, 09R	Automated	27	6	6	80	203	2.12	1328
3, 09R	Manual / Real	6	6	5	107	224	1.99	664
3, 27R	Automated	7	5	5	90	173	1.79	488
3, 27L	Automated	7	7	5	89	178	1.85	522
3, 09R	Automated	5	5	4	82	167	1.92	465
4, 27L	Manual / Real	7	6	5	117	223	1.80	613
4, 27R	Automated	8	5	5	94	189	1.87	550
4, 27L	Automated	8	5	5	92	186	1.88	551
4, 09R	Automated	7	5	5	104	206	1.87	563
5, 27R	Manual / Real	8	7	6	118	241	1.91	683
5, 27R	Automated	7	7	7	87	167	1.74	510
5, 27L	Automated	7	7	7	91	168	1.67	501
5, 09R	Automated	7	6	6	85	176	1.91	555
6, 27R	Manual / Real	5	5	5	103	173	1.58	393
6, 27R	Automated	8	7	6	90	161	1.69	450
6, 27L	Automated	8	7	6	90	161	1.61	454
6, 09R	Automated	7	5	5	96	169	1.62	429
7, 27L	Manual / Real	7	6	4	90	158	1.62	386
7, 27R	Automated	9	5	5	77	153	1.81	445
7, 27L	Automated	9	7	5	73	152	1.87	483
7, 09R	Automated	6	5	5	77	152	1.83	431
8, 09R	Manual / Real	6	6	5	88	173	1.84	469
8, 27R	Automated	7	5	5	77	155	1.85	445
8, 27L	Automated	6	5	5	80	155	1.80	416
8, 09R	Automated	11	6	5	86	163	1.69	507
9, 27R	Manual / Real	7	6	6	112	209	1.74	575
9, 27R	Automated	11	9	8	94	190	1.77	675
9, 27L	Automated	8	7	5	94	189	1.85	580
9, 09R	Automated	8	7	6	100	202	1.86	667
10, 27L	Manual / Real	7	6	6	118	209	1.65	557
10, 27R	Automated	16	9	9	87	218	2.19	998
10, 27L	Automated	18	9	8	88	227	2.25	1116
10, 09R	Automated	15	9	9	88	225	2.26	1091

a minimum two-minute separation between them. The first has no CTOT and the other two have to take off within the next four minutes to achieve their CTOTs. As the system applies a high penalty to missing a CTOT, the first will be placed after the other two aircraft in the take-off sequence. Due to the separation rules, it is often possible for a south-bound aircraft to take-off between them without affecting their take-off times. In this case, the delayed aircraft could witness four aircraft overtaking it, giving a positional delay of four, which is already much higher than average.

In summary, the perceived inequity in the automated sequencing is due to the presence of CTOTs which the controller obviously re-negotiated. Disregarding these, the positional inequity is at worst only as bad as that in the controller sequences so the level of inequity is either acceptable or justified by the CTOTs.

8.8.2 Take-off times within a CTOT take-off slot

The comparative weights for CTOT compliance of aircraft, total schedule delay and positional delay mean that even aircraft delayed until the start of a CTOT may not be sequenced as early as possible within the CTOT time-slot if doing so would be detrimental to the delay or mean that another aircraft misses a CTOT or CTOT extension.

Controllers tend to put a high priority upon getting these delayed aircraft airborne as soon as possible but the decision support system does not always do so, although the penalty associated with positionally delaying aircraft means that the system will do so unless there is a total delay or CTOT compliance benefit from not doing so. Given that these aircraft have already been considerably delayed, it is sensible to delay them no longer than necessary, however, in light of the fact that the predicted take-off times are likely to lag the real take-off times (as discussed in section 8.7.3), it is actually at least questionable whether it is wise to attempt to sequence take-offs as early in the CTOT slot as possible. Once the issue of schedule slippage has been resolved (a number of resolution methods are discussed in section 9.3.4), then it may be worth considering doing so, but not before then. An evaluation could then be performed to determine the exact cost of sequencing these aircraft as early as possible rather than allowing the sequencing some flexibility to delay these aircraft slightly more. An informed decision could then be made about which is the better way of sequencing the take-offs.

8.8.3 Reasons for using a measure of positional delay

Even given these issues with CTOT delayed aircraft, the positional delay is a good measure of inequity of sequencing. An alternative approach could consider equity of delay through a schedule, however delay naturally increases and decreases over time according to the number of aircraft queueing, so some measure of 'local' mean delay would be needed in order to identify inequity

Positional delay has an added benefit in that it is also a good measure of perceived inequity. In particular, it approximates the number of aircraft that the pilot will have seen arrive after him/her but take off before him/her.

8.9 The Effects Of Algorithm Parameters

Three questions are considered in this section: Firstly, given that the percentages of different types of moves (discussed in section 5.4) were selected based upon the performance of the steeper descent algorithm, it is important to consider whether these are appropriate values for the final problem. Secondly, since the tabu tenure of the tabu search was tuned using static problems (as discussed in section 5.3.5), in order to avoid tuning the system to the specific problems upon which it was evaluated, it is important to determine whether the selected tenure of ten moves is appropriate for the application of the algorithm to the dynamic problem. Thirdly, it is important to consider whether the weights in the objective function (equation 5.5 in section 5.6.7) correctly affect the results that are obtained. In particular, do these weights correctly influence the relative importance of CTOT compliance, delay and equity.

8.9.1 Changing the parameters

Experiments were performed using the tabu search algorithm and varying the percentage chance for each of the three moves to be used. The final system was used for these experiments, but without the follow-on searches, so that the performance of the search is purely due to the tabu search algorithm. Each experiment was executed ten times and the mean delay over the ten experiments was determined. Table 8.11 provides an example of how the mean total delay in the schedule can vary as the percentage chance of each move type being used is varied. The columns specify the percentage chance for a swap move to be used and the rows the percentage chance for a shift move to be used. The remaining moves were randomise moves. For clarity, the value in the body of the table is the increase in total delay, in minutes (rounded off to the nearest minute) compared with the parameter selection which had the best results.

For example, the value in the cell in the 50% row, 30% column of table 8.11 is zero. This indicates the fact that the results for the percentages 50, 30 and 20 respectively for shift, swap and randomise moves had a deviation of zero minutes (i.e. 30 or less seconds) from the best delay obtained. (In fact these percentage selections performed the best for dataset 6 and match the percentages that were selected for the experiments presented in this thesis.) If the percentage of shift moves is decreased to 40%, and the percentage of randomise moves is increased to 30% then the results are shown in the cell in the 40% row and 30% column, indicating a value of 3 minutes. In this case the total delay was between 2.5 and 3.5 minutes greater than the best delay obtained.

The example results in table 8.11, for dataset 6, illustrate the importance of including

TABLE 8.11: Example number of minutes variation in total delay, for dataset 6, as the percentage of each of the three move types is varied.

Percentage of shift moves:	Percentage of swap moves: 0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
0%	18	19	10	12	20	15	16	18	12	17	16
10%	8	10	7	10	9	12	10	10	9	13	
20%	6	6	7	9	6	8	9	7	6		
30%	6	4	4	6	3	0	3	3			
40%	4	4	7	3	0	4	3				
50%	5	7	3	0	3	7					
60%	4	3	3	5	5						
70%	3	3	4	5							
80%	7	5	3								
90%	5	2									
100%	4										

all three types of move in the neighbourhood. The overall delay was increased by 18 minutes when only the randomise move was used (i.e. when the percentages of the other moves was 0) and by 16 minutes when only the swap move was used. In these results, the delay was only increased by four minutes when the shift move alone was used, but this is still a poorer performance than when all of the moves were used. The shift move is very flexible (since multiple aircraft can be shifted at once) so it is unsurprising that solutions with a high percentage of shift moves perform better than those with a high percentage of randomise or swap moves.

The best solutions for this dataset were all obtained when there was a 20% chance of selecting the randomise move and a 30%, 40% or 50% chance of selecting each of the shift or swap moves. The results with this dataset validate the selection of movement percentages that was made.

8.9.2 Changing the tabu tenure

Experiments were performed using tabu search algorithms with different values for the tabu tenure, again without using any of the follow-on searches. Example results are shown in figures 8.17 and 8.18, where the increase in the total minutes delay compared with the best result found for any of the tested tenures is shown for each value of the tabu list tenure. Each experiment was executed ten times, and the mean result for each tenure is shown in the graph. The best results were obtained with a tenure of 11 moves for dataset 5 and 10 moves for dataset 6. For comparison of the scale of the values, note from tables E.5 and E.6 in appendix E, that the tabu search with a tenure of ten moves reduced the delay by 55902 seconds (or almost 932 minutes) for dataset 5 and 23117 seconds (or 385 minutes) for dataset 6, compared with the real delay for the manually produced schedule.

In each case, the steeper descent algorithm did extremely well, but the addition of a tabu list improved the performance. These results are similar to those from the earlier research, discussed in section 5.3.5 and appendix C. The optimal value for the tabu tenure differs in these

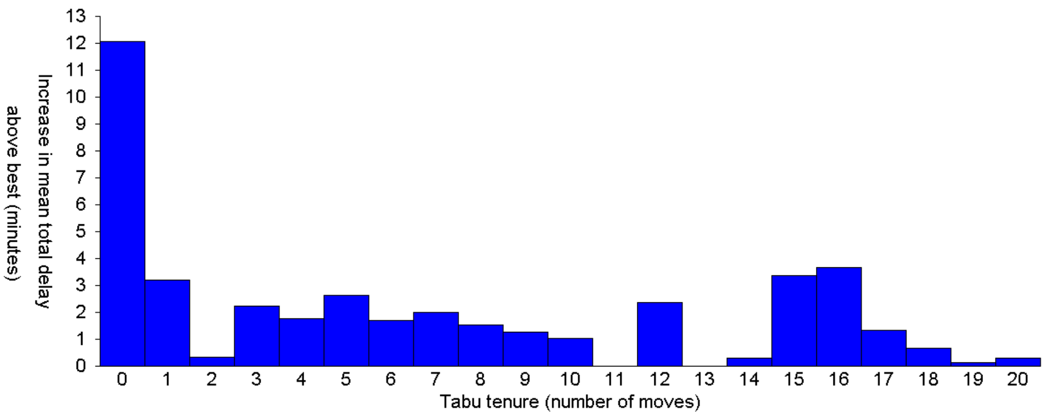


FIGURE 8.17: Graph of the variation in total delay as the tabu tenure is varied, dataset 5

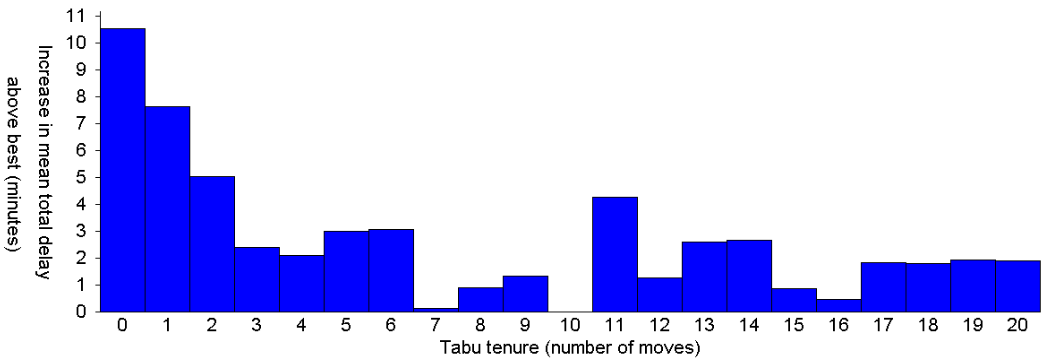


FIGURE 8.18: Graph of the variation in total delay as the tabu tenure is varied, dataset 6

results, depending upon the dataset, but the selected value of ten moves gave reasonable results in each case. This implies that the tabu tenure which was selected based upon its performance on a static problem is still an appropriate choice for use with the final system on a dynamic problem. Given that the system is actually solving a series of static problems in order to solve the dynamic problem, this should not actually be surprising. Furthermore, the earlier findings that the performance is not particularly sensitive to the exact value of the tabu tenure are further supported by these results.

8.9.3 Changing the weights in the objective function

A number of experiments were performed where the values of the different weights W_1 to W_6, used in objective function (function 5.5, detailed in section 5.6), were varied and the effects were evaluated in terms of the delay, CTOT time-slot compliance and equity of the resulting schedules. The values of the weights used in each experiment are given in table 8.12. Each row refers to a different experiment. The first column specifies a name for the experiment and the following columns show the values of the weights that were used for the experiment.

The three weights W_1, W_2 and W_3 are the primary means for determining the correct

TABLE 8.12: Weights used in experiments, dataset 6.

Experiment	Values of weights: W_1	W_2	W_3	W_4	W_5	W_6
A	0.125	0.125	0.375	0.125	0.125	0.125
B	1	0	0	0	0	0
C	0	1	0	0	0	0
D	0	0	1	0	0	0
E	0	0	0	1	0	0
F	0	0	0	0	1	0
G	0	0	0	0	0	1
H	0.5	0.5	0	0	0	0
I	0.5	0	0.5	0	0	0
J	0.5	0	0	0.5	0	0
K	0	0.5	0.5	0	0	0
L	0	0.25	0.75	0	0	0

TABLE 8.13: Example results of the effects of varying the weights, dataset 6.

Experiment	CTOTs missed	Total delay	Positional delay: Number	Total	Squared	Max	Number advanced
A	1.0	84,669	90.0	161.2	448.8	8.0	111.7
B	1.3	308,265	84.2	285.0	1739.2	16.4	102.3
C	2.1	83,560	90.9	178.3	554.5	8.8	123.3
D	61.0	811,215	0	0	0	0	0
E	61.0	811,215	0	0	0	0	0
F	62.8	855,885	146.1	360.5	1222.3	7.7	130.2
G	61.0	811,215	0	0	0	0	0
H	1.0	84,489	90.5	182.8	578.4	8.2	126.8
I	1.1	280,565	89.5	109.6	155.4	3.1	42.6
J	1.3	266,361	79.4	101.9	156.5	3.8	50.6
K	2.3	83,649	91.3	159.8	418.6	8.0	109.7
L	2.6	83,687	90.6	157.6	413.2	8.0	108.1

prioritisation of CTOT time-slot compliance, delay reduction and equity of delay for aircraft in the take-off sequence. Weight W_4 usually has a small value and is used to ensure equity of positional advancement of aircraft, as discussed in section 5.6.4. Weight W_5 is used to ensure schedule stability, so its primary effect is to avoid continuous changing between schedules. Weight W_6 is used to avoid penalising specific types of aircraft.

The results of the experiments to see the effects of varying the weights are shown in table 8.13. Each row of table 8.13 gives the results for a different experiment. Each experiment was performed ten times and the average results are presented in the table. The first column specifies which experiment was being performed, the second column specifies the average number of CTOT time-slots that were missed and the third column shows the average total delay for the aircraft (rounded off to an integer value). The fourth to seventh columns provide detail about the positional delays which were applied to aircraft. The fourth column specifies the number which had a positional delay, the fifth the total positional delay across the aircraft, the sixth the sum of the squares of the positional delay and the seventh the maximum positional delay that any aircraft had. The eighth column specifies the number of aircraft which had a positional advancement. The total number of positions advanced must always equal the total positional delay so does not need to be shown. A higher number of aircraft advanced implies a lower average positional advancement for each of the advanced aircraft.

The results in table 8.13 show that the weights are working as expected. Experiment A is the baseline, with the same weights as used in the other experiments presented in this thesis. Experiment B shows the effect of only penalising CTOT misses. One interesting effect of removing the penalty for delay is that the system can actually miss more CTOTs when this is done than with the normal weights. This is unsurprising given that the normal weights already apply an extremely large penalty for missing a CTOT. If there is no penalty for increased delay there is no need for the system to attempt to keep the throughput high, as long as the aircraft with CTOTs take off early enough. This can have extremely detrimental effects at those times when there is no aircraft with a CTOT present, since the aircraft can be extremely badly sequenced, leading to a low throughput and accumulated delays. These delayed aircraft are then blocking the holding area when the system does become aware of aircraft with CTOTs, and these aircraft may then not be able to achieve CTOTs that they may otherwise have achieved.

Experiment C shows the effect of ignoring CTOTs and equity and considering only the total delay. Unsurprisingly, given the results presented in section 8.5, where the removal of CTOT constraints decreased the delay, the delay is improved in this case but the number of CTOTs missed is increased.

Experiments D and E show what happens when equity is the only consideration. Unsurprisingly, the first-come-first-served sequence is used in each case so that aircraft take off in the same sequence in which they enter the holding area. Note that these results are worse than those for the first-come-first-served sequence in table 8.2 despite being the same take-off sequence since the start of the CTOT slot was not enforced when evaluating the results for table 8.2, as discussed in section 8.3.

The take-off sequences generated by experiment F were more complicated. Since aircraft are only added to the system when they have pushed back and are within the planning horizon, the pushback sequence and planning horizon designate the order in which aircraft enter the system. The feasibility recovery method, described in section 5.3.3, forces the end of the initial sequence (beyond the point at which the new aircraft is inserted) into the first-come-first-served take-off sequence, however, the objective function aims to find a take-off sequence which places aircraft which were already in the previous take-off sequence back into the positions that they previously had. This means that the system aims for a take-off sequence which matches the sequence in which aircraft entered the system. As expected, the resulting take-off sequences were very similar to the sequence in which the aircraft entered the system, with some modifications due to issues of feasibility of re-sequencing. A large number of aircraft were positionally delayed, and some were delayed by a significant amount, reflecting the fact that the sequence in which aircraft enter the system can be very different from the sequence in which they reach the holding area.

Experiment G shows the effect of using an objective to purely avoid excessive positional delay for the small or light aircraft. As described in section 5.3.3, recovery of feasibility on adding aircraft to the system is ensured by forcing later aircraft into the first-come-first-served holding area arrival sequence. Since no aircraft has a large delay in this sequence, and there is no incentive for the system to ever deviate from this sequence, aircraft in experiment G took off in the first-come-first-served take-off sequence.

Experiment H shows the effects of ignoring equity. Compared with experiment A, the delay for aircraft is slightly decreased, but the total positional delay, total squared positional delay and maximum positional delay are increased.

Experiment I shows the effect of optimising for CTOT compliance and equity. Here the system uses the first-come-first-served sequence, except where deviations are desirable due to CTOT compliance. A large total delay for aircraft is accumulated as the schedule progresses, and aircraft with CTOTs have to be moved more and more places forward in the schedule, but the penalty upon positional delay means that each aircraft is positionally delayed very few places.

Experiment J is similar to experiment I, but aims for equity of positional advancement as well as delay. In this case the number of aircraft advanced is increased while the total positional delay (which equals the total positional advancement) is decreased, so the mean positional advancement is decreased, as expected for an experiment where positional advancement is penalised.

Experiments K and L penalised both delay and inequity, but ignored CTOT compliance. The only difference between experiments K and L was the relative weights of the two objectives. In experiment L, a higher weight was given to the objective of maintaining equity of positional delay than was given to it in experiment K, and the results show the expected decrease in positional delay and increase in total delay.

These experiments show that the test system behaves as expected when the weights for the different objectives are altered. Many of the objectives can be seen to cooperate in general, but conflict in the best solutions. For instance, a better overall CTOT compliance is obtained by also penalising the delay, to avoid accumulating delays for aircraft which would prevent later aircraft being able to achieve their CTOTs. However, once the relatively good solutions are found, the trade-off between the CTOT compliance, delay and equity of positional delay that was observed in section 8.5 will take effect.

8.10 Sequencing Examples

Example results from the decision support system are presented in this section. These show the compliance of the developed system with the path allocation algorithms and the validity of the sequencing of the movement within the holding area.

8.10.1 The take-off sequence

Table 8.14: Example automatically generated take-off sequence

Take-off Index	FCFS Index	Take-off Time	Earliest Take-off	Departure Route	CTOT	Path Assigned	Path Type
0	0	00:00	00:00	MID	05:00	HIJKLMNY	Default
1	1	02:00	01:55	MID	05:00	EFGOPQR	Default
2	3	04:00	03:00	MID	08:00	HIJKLMNY	Default
3	2	06:00	06:00	BPK	11:00	ABCDUVXY	Default
4	4	09:14	09:14	MID		HIJKLMNY	Default
5	5	10:23	10:23	BPK	14:00	EFGOPQR	Default
6	6	12:16	12:16	SAM		HIJKLMNY	Default
7	7	13:19	13:19	DVR	14:00	HIJKLMNY	Default
8	8	15:54	15:54	WOB		ABCDUVXY	Default
9	9	18:00	18:00	CPT		ABCDUVXY	Default
10	11	25:07	25:07	MID	25:00	ABCDUVXY	Default
11	12	26:37	26:37	BPK	29:00	HIJKLMNY	Fast
12	10	30:00	30:00	DVR	35:00	HIJKLMNRST	Slow
13	14	31:00	30:36	WOB		ABCDUVXY	Default
14	16	32:05	32:05	CPT		ABCDUVY	Fast
15	15	33:05	31:27	BPK	34:00	ABCDUVXY	Slow
16	17	34:05	34:00	MID	39:00	ABCDUVXY	Default
17	13	35:05	35:00	BPK	40:00	HIJKLMNY	Default

An example take-off sequence that was generated for a quiet period of the day can be observed in table 8.14. Each row shows the details for a single aircraft. All aircraft in this sequence were medium weight class and speed group three so these details have been omitted from the table. The first column shows the (zero-based) index of the aircraft in the generated take-off sequence. The second column shows the (zero-based) index of the aircraft in the arrival sequence at the holding area. So this would denote the take-off sequence for a first-come-first-served take-off sequence. The third column shows the predicted take-off time. The fourth column specifies the earliest take-off time that could be allocated to the aircraft, according to the taxi time, CTOT and ready time restrictions discussed in chapter 2.6. The fifth column specifies the (SID) departure route of the aircraft. This will determine the required separations between aircraft. The sixth column specifies any CTOT allocated to the aircraft. The take-off time-slot for a CTOT is from CTOT-5 minutes to CTOT+10 minutes. The penultimate column specifies the holding area traversal route that the system allocated to the aircraft. The final column specifies the type of the allocated path. A default path may be the same as the fast path (for example EFGHOPQR or HIJKLMNY) or the slow path (for example ABCDUVXY), as described in sections 6.4 and 6.5.

In the following descriptions, aircraft will be named by their position in the take-off sequence, so aircraft *0* is the first aircraft to take off. Although this example is from a quiet period of the day, chosen in order to be able to illustrate the holding area movement in a simple manner, most of the aircraft had CTOTs. The presence of CTOTs can be observed to require aircraft to delay their take-off, shown by the fact that the earliest take-off times for a number of aircraft is restricted by the allocated CTOT. For example, aircraft *3* arrived before *2* but took off later due to its allocated CTOT.

Most aircraft in the sequence can be observed to take off at their earliest take-off time. This means that the holding area delay is close to the minimal value it can take. Since so few take off later than the earliest take-off time in this example, it is easy to see that the system has, in this case, found the take-off sequence with the minimal delay. The delay could only be decreased by advancing aircraft which do not currently take off at the earliest take-off time. It can be observed that sequences which would advance these aircraft would delay other aircraft by an equal or larger amount, so the delay would not be decreased. For example, in order for aircraft *2* to take off any earlier it would have to take off before at least one of the aircraft *0* or *1*. Whichever was overtaken could then not take off before time 05:00:00, due to the two minute separation rule after the earliest take-off of aircraft *2*. The increased delay for the overtaken aircraft would exceed the decreased delay for aircraft *2*, so the total delay would increase.

In order to examine some of the holding area movement, the take-off sequence will be divided into independent sub-sequences. The triplet model, introduced in section 4.2, will be used to illustrate the holding area movement. The triplet representation of the holding area movement, that was output by the system, is shown pictorially in figures 8.19 to 8.22. The movement in a triplet model is defined by a single long sequence of triplets. For clarity, in figures 8.19 to 8.22, this sequence has been split across multiple lines, and arrows have been added to indicate the fact that the rows should actually continue from one another. The break has been made after the sixth triplet in the figure 8.19. In figures 8.20 to 8.22, a line break has been introduced whenever the aircraft being moved was changed. The horizontal alignment of triplets has then been adjusted to ensure that the triplets for each individual aircraft appear in sequential columns. This graphically illustrates the way in which the movement of some aircraft is interrupted by the movement of others.

8.10.2 An example of simple alternating movement

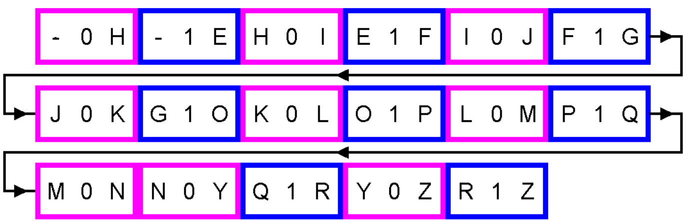


FIGURE 8.19: Triplets for simple holding area movement

Aircraft *0* and *1* can leave in the same order in which they arrive. Since they arrive at different entrances, each can be allocated to the default path for its holding area entrance and the results in table 8.14 show that the system has correctly done so. Since both aircraft are in the holding area at the same time, the holding area movement involves moving both forwards at the same time (which is shown by the alternating triplets), then entering the runway in the sequence dictated by the desired take-off order. Re-sequencing between these aircraft only has to be performed at the point where their paths converge. The triplet representation in figure 8.19 shows that the system performed as expected.

8.10.3 Simple re-sequencing

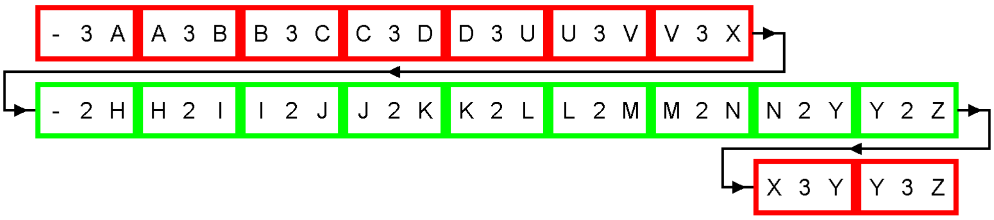


FIGURE 8.20: Triplets for simple re-sequencing within the holding area

Aircraft *2* and *3* need to be re-sequenced. Aircraft *2* arrives after *3* but takes off before it. This is due to the CTOT that aircraft *3* has been allocated. Figure 8.20 shows that aircraft *3* arrived first at the holding area and performed all possible movement, advancing as far as node *X* before holding. Aircraft *2* then enters the holding area, moving right through it. Once aircraft *2* has vacated node *Y* (entering the runway), aircraft *3* can enter node *Y* and then enter the runway and take off. The triplet representation in figure 8.20 shows that the system performed as expected.

8.10.4 Holding within the holding area

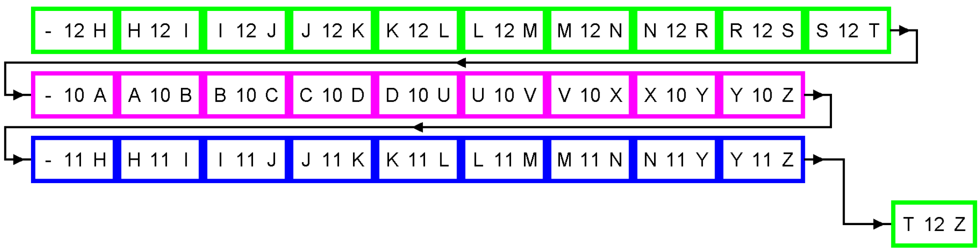


FIGURE 8.21: Triplets showing longer holding within the holding area

Aircraft *10* to *12* are a more interesting example. Aircraft *12* arrives first, but takes off after aircraft *10* and *11*. The triplet representation in figure 8.21 shows that the system moved aircraft *12* as far as node *T* then held it there until aircraft *10* and *11* had arrived and taken off. Once aircraft *10* and *11* had taken off, aircraft *12* entered the runway and took off. Again the system allocated paths and performed holding area movement as expected.

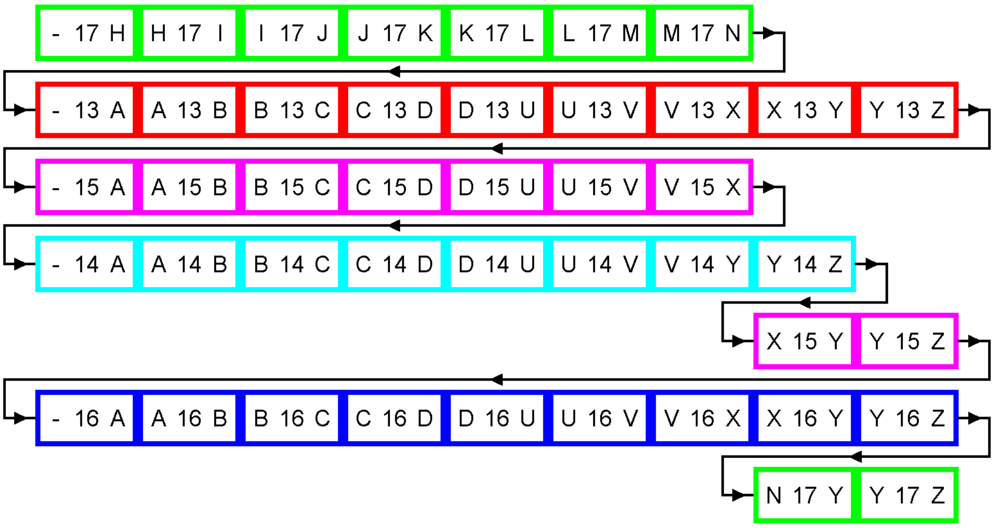


FIGURE 8.22: Triplets for more complex holding area movement

8.10.5 More complex re-sequencing

The final example of holding area movement is more complex. Aircraft enter the holding area in the sequence *17*,*13*,*15*,*14* then *16*. Aircraft *17* has to be held until the other aircraft have all passed through the holding area then taken off. Aircraft *14* and *15* also need to reverse their take-off sequence, while aircraft *17* is holding.

Only aircraft *15* was overtaken by an aircraft from the same holding area entrance. It was thus allocated to the slow path. Only aircraft *14* overtook an aircraft from the same holding area entrance. It was allocated to the fast path. All other aircraft were allocated to the default paths. This is correct behaviour according to algorithms 3 to 5.

The triplet representation in figure 8.22 shows that the system moved aircraft *17* as far as node N then held it there. Next aircraft *13* entered the holding area, passed through and took off. Aircraft *15* then entered the holding area and was moved as far as node X before it was held. Aircraft *14* entered the holding area, overtook aircraft *15* which was held at node X and aircraft *17* which was still held at node N, and took off. Aircraft *15* then left node X, travelled to node Y and then took off. Aircraft *16* then entered the holding area, overtook aircraft *17* which was still at node N, and took off. Finally, aircraft *17* was released from node N, travelled to the runway via node Y and took off.

The triplet representation of the movement the system performed for the aircraft shows that the system correctly allocated the appropriate holding area traversal paths and correctly sequenced the holding area movement to achieve the desired take-off sequence.

8.10.6 Summary

These small examples illustrate how the holding area movement is performed by the system and show that in each of these cases the system performed as expected.

8.11 Summary

The designed system performs very well at busy times of the day, but the restrictions imposed upon the sequencing mean that it will not perform so well at quiet times. At quiet times, the controllers have less pressure and easier problems to solve, so are unlikely to consult a decision support system anyway.

The take-off time prediction system that the decision support system uses is relatively accurate but tends to be pessimistic as it allows more time than necessary for aircraft to achieve the sequencing (to ensure that sequences are easy to achieve) and does not assume that controllers will use the reduced separations facilities that are available to them. This problem will need to be handled by any live system and some examples of how to do so are discussed in section 9.3.4.

The results in section 8.4 showed that the system performance increases tremendously when taxiing aircraft are considered in addition to those already within the holding areas. This is even the case when moderate levels of uncertainty are associated with the taxi time predictions for the aircraft, as shown by the results in section 8.6. The system performance decreases as the uncertainty in the taxi times increases, so the development of an accurate taxi time prediction system should be of great benefit.

The system puts a lot of emphasis upon CTOT slot compliance, possibly even more than the controllers do. This can be seen by comparing the number of CTOTs missed in the real and automatically generated schedules. It is obvious from the real take-off times that many CTOTs were re-negotiated, but the simulation is not able to do this as there is nobody to negotiate with. This emphasis on CTOT compliance can cause prioritising of these aircraft and consequent positional delays for other aircraft. Excessive delays waiting for CTOT can give excessive positional delays to aircraft, making the sequencing appear less equitable than it actually is. If the aircraft with large positional delays for reasons of CTOT compliance are excluded, the system appears to find sequences which are no less equitable than those the controllers find.

The simulation was used to evaluate the effects of some of the different take-off sequencing constraints. The holding areas were found to be very flexible in most cases, as the values of the sequences with and without the effects of the holding areas usually varied by less than 5%. The exceptions were when an aircraft had to be delayed within the holding area, thus blocking one or more paths through it and reducing the flexibility. The route and speed separation rules were found to have the most effect upon the system. The wake vortex and CTOT time-slot constraints were found to have a secondary, but considerable, effect. Each of the evaluated constraints seemed to be relatively independent so all should be considered when performing the sequencing.

The effects of the tabu search parameters were also evaluated. The selected tabu search tenure and percentage chance of selecting each type of move were all seen to be reasonable for the considered problems, and the weights in the objective function were observed to correctly affect the balance between the considered objectives. Finally, some examples of the paths assigned by the system and the holding area movement that was predicted were presented in section 8.10, showing that the system performs as expected.

Chapter 9

Conclusions

The job of the runway controller is a difficult one and there are many demands upon his or her time. The runway controller is responsible not only for the aircraft at the holding area but also for the aircraft that have taken off, until they leave the local airspace. Although the controller spends most of his or her time talking to pilots and observing the situation at the holding area out of the window of the control tower, he or she must also find the time to determine the sequence in which the available aircraft should take off.

The departure problem is complicated since there are many constraints upon the take-off schedule, including the sequence-dependent separations between aircraft. The route and speed based separations at Heathrow are not only asymmetric, but also do not obey the triangle inequality, so it is not sufficient merely to look at adjacent pairs of aircraft. The complexity of the re-sequencing task, physical constraints of the holding area and limited time which the controller has available mean that the runway controllers have an extraordinarily difficult task to perform.

Given the complexity of the sequencing problem, it is unsurprising that the controllers have to resort to mental heuristics for the take-off sequencing and so do not always obtain the optimal take-off sequence from the point of view of delay and time-slot compliance. It is for just this kind of problem that a decision support system is ideal.

The work presented in this thesis was motivated by the desire to determine whether it would be possible for a computerised decision support system to solve this problem fast enough to be of use in a real-time system. Once this question was answered satisfactorily, the question of the potential benefits from using a decision support system had to be addressed.

The developed simulation shows that the decision support tool could potentially aid the controllers to perform considerably better than they do at present. Significant decreases in both the total delay for aircraft and the number of CTOT time-slot extensions required are predicted from adopting the automatically produced sequences, however, importantly, these sequences can be attained without requiring unacceptable holding area movement.

9.1 Key Results

A number of key results were presented in this thesis. These results are summarised here, along with conclusions which can be drawn from them.

9.1.1 The controllers perform well

The results show that the controllers perform a valuable task in re-sequencing the aircraft and that they do very well. This is especially obvious when the sequences are compared with the first-come-first-served sequence, where no re-sequencing at all is performed.

The controllers do so well that a decision support system would not be expected to provide benefits if only given the limited view of the aircraft at the holding area that the controllers have to work with, as shown by the results for the system with no knowledge of taxiing aircraft in section 8.4.

9.1.2 The decision support system does even better

The simulation results predict that the decision support system is capable of finding good quality take-off sequences when given adequate knowledge about taxiing aircraft. The experimental results showed that the decision support system could outperform controllers over half-day periods, when taking taxiing aircraft into consideration. Both the CTOT compliance and the delay were considerably improved over the manual take-off sequences, while the system simultaneously ensured that the generated take-off sequences would be easy for the pilots and controllers to achieve. Importantly, the system found these good quality solutions in a time which would not be prohibitive to implementing an advisory system for the controllers.

9.1.3 Take-off time predictions are pessimistic

It was shown in section 8.7.3 that the take-off time prediction system which was used by the decision support system for evaluating the automatically produced take-off sequences was pessimistic. For the purposes of proving that the decision support system works well, this is better than being optimistic. Even when sequences were evaluated using the pessimistic take-off time prediction system, the decision support system found sequences which improved the delay and time-slot compliance considerably. However, even better results could reasonably be expected if the system was able to allow for the pessimistic take-off time prediction system. Ways of doing this are considered in section 9.3.4.

9.1.4 The performance over shorter time periods

Performing experiments upon smaller datasets allowed the evaluation of the performance of the system under different levels of system load. Experimental results 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 results predict a reduction in delay of up to 50% at busy times. The usual controller method of attempting to get as many one-minute separations as they can is obviously not always optimal in these circumstances.

At other, usually quieter, times the controllers can outperform 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 whilst allowing the controller to utilise extra flexibility at quieter times, when the workload permits.

Even better performance could be expected at busy times if the system were allowed to take advantage of some of the methods that controllers have for reducing separations or by predicting that controllers will do so. Even the occasional small improvement to the separations at these times can have a large effect upon the delay for following aircraft. If the decision support system is to be used at quieter times, there could be an option for a controller to specify which separation reductions or higher workload activities should be permitted or what the minimum holding area traversal times should be. However, the value of using it at quiet times is questionable, since the controllers already perform very well at quieter times and the pressure on the controller is less, so the controller would be far less likely to refer to a decision support system anyway.

9.1.5 The system is deliberately over-restrictive

The designed system is deliberately restrictive in the schedules it will produce. It will eliminate from consideration any take-off sequences which are not likely to be acceptable to controllers, even when it is theoretically possible to achieve them. The system will also allow far more time than is likely to be needed for aircraft to traverse the holding area. At busy times, when there are many take-off schedules to choose from, this ensures that schedules where the taxi time may be an issue are less likely to be adopted. However, at quieter times, when there may not be reasonable alternative schedules to adopt, the large traversal time can instead have a detrimental effect upon the take-off time prediction system. These restrictions mean that the system may be incapable of matching the controller results at quieter times.

9.1.6 The benefits of knowledge of taxiing aircraft

A decision support system can consider many more aircraft than a human controller can and thus can help avoid future problems from aircraft not currently under the control of the runway controller. The results presented in this thesis indicate that there is an advantage to be gained from including taxiing aircraft when determining a take-off sequence. They also show that the designed system is capable of using this information quickly enough to provide real-time decision support for controllers.

9.1.7 The behaviour with uncertain taxi times

It can be seen from the results that the searches can endure the situation whereby at any iteration they only have imperfect knowledge about the future and only a partial ability to reorganise the aircraft already in the holding areas. In order to achieve this, the evaluation of the worth of the current, partial schedule that is visible at any iteration includes elements to avoid causing unjustified problems to later iterations.

The concept of a taxi-out time that is usually referred to is the time from leaving the stand to reaching the runway. This time is highly influenced by the variable waiting time at the holding area (which depends upon the size of the runway queue). The delay in the holding area is after holding area arrival so can be ignored in the taxi time prediction system required by the decision support system described in this thesis. Additionally, the positions of the stands at Heathrow, where most are on long cul-de-sacs, mean that the contention at push-back from the stand and engine start-up is considered to be by far the greatest element of the uncertainty in taxi time to the holding area. By considering aircraft only after push-back has been completed, the taxi times to the holding area should be relatively predictable. Taxi time prediction in this context would be an interesting area for further research, and is discussed in section 9.3.3.

9.1.8 Guarantees of the quality of the suggested sequence

The follow-on searches that are utilised after the tabu search has completed provide some guarantee of the value of the presented take-off sequence. For example, they help to avoid the situation where the heuristic search misses obvious improvements to the schedule from swapping aircraft or from re-sequencing aircraft that are close to each other in the take-off sequence.

The possibility of swapping between two identical or similar cost schedules over time has been explicitly accounted for in the objective function and the follow-on searches ensure that the alternative sequences are evaluated. This increases the stability of the scheduling suggestions.

9.1.9 The downstream constraints are the most restrictive upon the sequencing at Heathrow

An interesting result from the experiments presented in this thesis was that the downstream constraints (the CTOTs and the departure route separation rules) are actually more restrictive for the take-off sequences than the wake vortex separation rules, in that removal of these constraints has the most benefit for the total delay for aircraft. As discussed in section 3.3, it is not uncommon to measure the theoretical capacity of an airport by using the wake vortex separations. This is obviously not appropriate for Heathrow.

9.2 Decision Support For The Runway Controller

A number of conclusions can be drawn here about the decision support system and departure system simulation design presented in this thesis.

9.2.1 Difficulty of the task

The take-off problem is an incredibly complex one, with similarities to, but also significant differences from, the asymmetric travelling salesman problem and the blocking job shop machine scheduling problem. The difficulty of the problem was considered in chapter 4. With holding areas as flexible as those at Heathrow this is a difficult problem to solve as, if the re-sequencing is possible, there are often many ways to perform the re-sequencing, but usually only one best way.

9.2.2 Advantage of using a tool

A decision support tool can consider more aircraft that a human controller can and, hence, can help to foresee and avoid future problems for aircraft not currently under the control of the runway controller. If a computerised system is to be of any use to the runway controller in sequencing the aircraft at the holding areas, then it must be able to find good take-off sequences in real time. All sequences that are presented must be achievable and should be easy to achieve rather than requiring complex manoeuvring or complicated instructions from controllers. As the situation changes, the system must be able to rapidly react to the changes and re-sequence aircraft accordingly. The simulation results presented in this thesis show that the decision support system described here is capable of achieving these benefits.

9.2.3 Complex objectives

As this is a real-world problem, the objective function is not easy to capture but is key to obtaining good results for the problem. The objective function is designed to help the search to find schedules with a low delay and high CTOT compliance. The path allocation system and the feasibility check ensure that sequences can be attained in a manner which would be acceptable to a controller.

9.2.4 Problems with applying previous research

The presentation of the take-off scheduling problem in the academic literature usually fails to take account of the physical constraints upon the scheduling problem at Heathrow. Much of the previous research has been concerned with the arrivals problem, sometimes with a statement that the approach should also work for departure sequencing. However, there are considerable differences between the arrival and departure problems.

Where previous research has looked specifically at the departure problem, it was usually assumed either that re-sequencing constraints could be ignored or that the constraints were so restrictive that they simplified the problem. For example, if runway scheduling can take place at the stands/gates or there are extremely flexible holding areas then the physical holding area structures are not important. This is not the case at Heathrow. Alternatively, if the holding areas are extremely constrained, for example only allowing the interleaving of two take-off queues, then the sequencing constraints will vastly simplify the problem. Again, this is not the case at Heathrow. At Heathrow, it is more practical to perform the sequencing at the holding areas, but the holding areas have significant flexibility, so these structures are vitally important.

Finally, constraints such as the true effect of CTOT time-slots and time-slot extensions were not considered in the previous research. They are explicitly considered in this thesis, and have been shown to have a significant effect upon the sequencing problem. In particular, there is sometimes a trade-off between CTOT compliance and total delay, and CTOTs can sometimes mean aircraft need to be held for a long time, or prioritised in the holding area.

9.2.5 The selected approach

A design for a decision support system to aid the runway controllers was presented in this thesis. It uses a problem decomposition to enable the production of good results very quickly. This decomposition was motivated by observations of controller preferences and vastly reduces the size of the search space which must be considered.

Real world constraints such as partially fixed schedules and fixing the paths through the holding areas were included, in addition to the usual constraints such as maintaining required separations and meeting CTOT take-off time-slots. The results showed that it is feasible to check the effects of the holding areas for each generated solution, using a heuristic rather than exact verification method. The automated searches perform very well even with the very limited time available to them.

9.2.6 Departure system simulation accuracy

A departure system simulation was developed to enable the performance of the system to be evaluated. This simulation provides all of the information that would be available in the live situation and also allows prediction errors to be simulated.

The accuracy of the take-off time prediction system was evaluated by predicting take-off times for the aircraft in the real take-off sequence. The predictions were seen to be pessimistic in most cases rather than optimistic and especially so at busy times of the day.

The accuracy of the simulation was tested by comparing the positions the aircraft took under the simulation against the positions they were expected to take, by observation of holding locations of aircraft and playback of full sequences. Taxiway congestion was 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. 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.

9.3 Extensions And Future Work

The design presented here could be used or extended in a number of ways in future.

9.3.1 Evaluate the effects of CTOT compliance

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 do. The controller view makes sense as they are permitted a number of extensions so they are happy to use them if it will help to keep the throughput high and delay low. By knowing about tight CTOTs in advance (while aircraft are still taxiing) the designed system can ensure that CTOTs are met and the delay is still kept low. 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.

There may be benefits to be gained from not achieving so many CTOTs, so the weights in the objective function could reasonably be expected to be modified if used live. A profitable course for future research may be to explicitly investigate the trade-off between CTOT compliance and delay, and to determine whether there are easy ways in which the effects could be reduced, for example, by highlighting and rectifying problematic CTOTs earlier.

9.3.2 Evaluate the effects of applying MDIs

The effect that the application of an MDI has upon a take-off sequence will vary depending upon the the length of the take-off queue and the number of aircraft which have been allocated to the routes with the MDIs. Short-term increases in MDIs can sometimes be absorbed by re-sequencing. Longer-term increases can have a long term effect upon the take-off times of aircraft. An interesting area of research would be to investigate the short and long term effects of MDIs of varying duration and size on different routes.

9.3.3 Develop a taxi time prediction system

Test results in section 8.6 indicate that a taxi time prediction system may be required to obtain the maximum benefit from the decision support system presented here. Such a system could consider taxiway congestion, standard aircraft taxi speeds and source and destinations for the taxiing to derive a predicted taxi time. Designing and calibrating such a system may form an interesting subject for future research.

9.3.4 Investigate and handle schedule slippage over time

As was seen in chapter 8, the predicted take-off times tend to be later than the real take-off times. Reasons for this were discussed in section 8.7.3. This means that the predicted take-off schedule slips later over time, as the prediction errors accumulate. Evaluating and handling the slippage in a live situation would form another interesting area of further research. A decision support system which was built to take advantage of the natural schedule movement and reduced separations could reasonably be expected to perform even better than the current results imply.

One key reason for the schedule slippage is the fact that controllers can reduce the required separations at times. The first suggestion for a method to handle the schedule slippage is to allow the controller to specify which reduced separations are being used at that time. This may be practical at times, especially for general reductions possible in good weather. The 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. This was not assumed for these tests, however, and may not be practical in the live situation.

The second suggestion for handling schedule slippage is to use systematically lower separation times, for example by a specific number of seconds per minute. The aim here would not be to simulate the real reductions in separations, but to simulate the average reductions. In this way the gap between the predicted and real times should not increase over time.

The third suggestion for handling the slippage is to build sequences which can be moved earlier in time without problems. Many restrictions upon take-off times are calculated relative to previous take-offs, so these should not pose a problem if the entire sequence is shifted earlier. Some times, such as the earliest time at which an aircraft can reach the holding area or the limits of a CTOT time-slot, are absolute rather than relative values. If the sequence is expected to systematically move earlier in time then some consideration needs to be taken as to how to cope with this. For the earliest take-off time for holding area arrival and traversal, a two minute traversal time is used and is already generous, so could be considered to allow for some leeway in practice. In other words, if the schedule is moved earlier, the aircraft should still be able to make the new take-off time. In the same way, the start and end of the CTOT time-slot could be manipulated, or the function $C(d_i, b_i, l_i, h_i)$ in section 5.6.2 could be modified to penalise schedules where the aircraft takes off within the CTOT slot but close to the start or end, in order to favour schedules where some movement in the take-off time will be possible without moving the aircraft out of its CTOT time-slot.

9.3.5 Simulation to evaluate potential changes

It was shown in sections 8.4 and 8.5 how the simulation could be used to evaluate the effects of varying the amount of knowledge and accuracy of the knowledge about the taxiing aircraft. It was shown in section 8.6 how the simulation could be used to evaluate the effects of the different sequencing constraints. Throughout the experiments, the simulation was performed with models of all three of the Heathrow holding areas.

The designed simulation could also be used to evaluate the effects of changes to holding areas, changes to traversal path allocations (for example, closing a runway entrance), changes to entrance allocation schemes that the ground movement controller may use, changes to separation rules, changes to other constraints upon the sequencing, the effects of the accuracy of the taxi times given to the system or the effects of varying the planning horizon and freezing time of the schedule.

Evaluating the effects of changes to the design of the holding areas would be especially useful due to the cost of making the physical changes. This sort of evaluation is already performed by NATS using existing simulations (for example TAAM) but the difference with the simulation presented in this thesis is that (with the appropriate path allocation rules) it will make decisions as a controller would and could possibly give a more accurate estimation of the performance of the system with a controller present.

9.3.6 Develop as a training aid

The designed system could potentially be developed into a training aid for runway controllers. By providing details of holding area structures and sequencing rules and then calibrating the system using experienced runway controllers, it could be used to produce training scenarios for trainee runway controllers. Although the calibration stage would involve controller input to ensure that the suggested path allocations and sequencing decisions are what the controllers desire to teach, a vast number of scenarios could then be created for trainees to learn from. Furthermore, the later effects of trainee decisions could be viewed and played back, providing a useful feedback and learning experience. Properly implemented, this could potentially save much valuable time for experienced controllers wishing to pass on their knowledge.

9.4 Publications From This Work

The development history of the decision support system design described in this thesis is given in section 4.6.3, including references to the various conference presentations that were made. The key publications that have resulted from this work are discussed as follows:

The initial proof of concept, showing that the designed algorithm could handle the holding area problem fast enough for real time decision support was presented at the **9th International Conference on Computer-Aided Scheduling of Public Transport, 2004** [9] and accepted for publication in the Springer post-conference volume of selected proceedings, [15]. These papers included a version of the tabu search which is very similar to that presented in chapter 5. The feasibility check was performed using a simpler version of the algorithm described in chapter 6 and a rolling window through the aircraft was used instead of a departure system simulation. Both a simulated annealing and a tabu search algorithm were applied to the problem for the 27R holding area and a comparison was made of the comparative effectiveness of these and first descent and steeper descent algorithms. The meta-heuristic searches were shown to perform significantly better than the simpler searches, showing that there are still local optima in the search space.

Given the success of the algorithm applied to the test problems, a simulation was produced to present it with more realistic problems such that it had knowledge only of the aircraft that a real system would know about at the time of making the decision. This simulation was a simplified version of that presented in chapter 7. This system was then described in a paper published in **Transportation Science** [14]. The same four algorithms were evaluated for this work as for [15], although only the results for the tabu search were presented in the paper. The results for the remaining algorithms can be found in appendix C.

The developed system was used to evaluate the effect of the planning horizon for a paper accepted for the **2nd International Conference on Research in Air Transportation, 2006**, [12]. The effects of varying the knowledge of taxiing aircraft and the point at which the take off sequence is committed were considered and the value of taking into consideration the taxiing aircraft was obvious. The planning horizon experiments were later repeated using the final developed system and similar results were obtained, as shown in section 8.4.

Another important piece of research was the investigation of the effects of the various constraints upon the system. The results of such an investigation were presented at the **10th International Conference on Computer-Aided Scheduling of Public Transport, 2006**. This investigation revealed the major effect that the downstream constraints have upon the sequencing, and the relatively minor effect that the physical holding area structure has upon the total system delay, given sufficient re-sequencing and the advance knowledge of taxiing aircraft. These experiments were later repeated using the final developed system and are presented in section 8.5.

Finally, a paper describing the various improvements that have been made to the system since [14], such as the follow-on search and improvement to the simulation, has been submitted for publication to the **Journal of Scheduling** [16]. This includes a detailed consideration of the effect of the taxi time uncertainty, as described in section 8.6 and uses smaller datasets for the experiments, as described in section 8.7.

9.5 Final Remarks

The primary conclusion from this thesis is that it is possible to produce the underlying algorithms for a decision support system which will react fast enough to be of use for real time decision support for the runway controller. The design of such a system is presented in this thesis. Despite the complexity of the problem to be solved, the chosen decomposition (which was based upon real controller behaviour) allowed the problem to be solved quickly enough for practical use.

A simulation was developed for the purpose of testing the decision support system design. The generated take-off sequences were examined for both performance and acceptability. Results showed a significant decrease in delay, with a simultaneous improvement in CTOT compliance compared with the sequences created live by the controllers. This was despite the inclusion of limitations such as freezing the take-off sequence a few minutes prior to take-off and freezing the allocated taxi path once aircraft enter the holding area, both of which will reduce the flexibility of the sequencing.

These benefits were greatest at the busiest times, when the controllers were busiest so had less time to consider the sequencing, but also when the decision support system has more aircraft to choose from. Despite these improvements to the sequencing, consideration of the positional shifts of aircraft showed that the sequencing was no less equitable than the real adopted sequences, once the effects of CTOTs had been taken into account. Furthermore, playback of the automatically generated sequences showed that the system did not require unacceptable paths to be allocated, and that the holding area movement was very similar to that used by the real controllers. The path allocation mechanism that is used ensures that this will always be the case.

Experimental results showed that the major benefits were attained by the system through its knowledge of aircraft on the taxiways. These results make it easy to justify the development of a decision support system for the controllers, or at least to develop a method to give the controllers the same kind of visibility of future aircraft, and the time to use that information, without increasing their workload. Such a development could potentially bring huge benefits in terms of reduced delay and fuel burn. This would be of benefit not only to airlines and passengers, but also to the environment.

References

[1] E. Aarts, J. Korst, and W. Michiels. Simulated annealing. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 187–210. Springer, 2005.

[2] J. Abela, D. Abramson, M. Krishnamoothy, 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, Australia*, pages 71–90, July 1993.

[3] I. Anagnostakis and J.-P. Clarke. A multi-objective, decomposition-based algorithm design methodology and its application to runway operations planning, 2004. Report number ICAT-2004-5, MIT, based on the PhD thesis of Ioannis Anagnostakis, MIT.

[4] 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. Report AIAA-2002-5886.

[5] I. Anagnostakis and J.-P. Clarke. Runway operations planning: a two-stage solution methodology. In *proceedings of the 36th Hawaii International Conference on System Sciences (HICSS-36), Hawaii*, volume 3, January 2003.

[6] 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), Hawaii*, January 2001.

[7] I. Anagnostakis, H. R. Idris, J.-P. Clarke, E. Feron, R. Hansman, A. R. Odoni, and W. D. Hall. A conceptual design of a departure planner decision aid. In *proceedings of the 3rd USA/Europe Air Traffic Management R&D seminar, Napoli, Italy*, June 2000.

[8] K. Andersson, F. Carr, E. Feron, and W. D. Hall. Analysis and modelling of ground operations at hub airports. In *proceedings of the 3rd USA/Europe Air Traffic Management R&D Seminar, Napoli, Italy*, June 2000.

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

[10] J. A. D. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson. Hybrid meta-heuristics to aid runway scheduling under uncertainty at London Heathrow airport (extended abstract). In *proceedings of the Models and Algorithms for Planning and Scheduling Problems (MAPSP) conference, Siena, Italy*, pages 21–24, June 2005.

[11] J. A. D. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson. A hybrid tabu search evaluation of holding point entrance allocation methods for departures at London Heathrow airport (abstract). In *proceedings of the 6th Metaheuristics International Conference (MIC), Vienna, Austria*, page 46, August 2005.

[12] J. A. D. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson. The effect of the planning horizon and the freezing time on take-off sequencing. In *proceedings of the 2nd International Conference on Research in Air Transportation (ICRAT), Belgrade, Serbia and Montenegro*, June 2006.

[13] J. A. D. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson. An examination of take-off scheduling constraints at London Heathrow airport. In *proceedings of the 10th International Conference on Scheduling of Public Transport (CASPT), Leeds, UK*, June 2006.

[14] J. A. D. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson. Hybrid meta-heuristics to aid runway scheduling at London Heathrow airport. *Transportation Science*, volume 41, number 1, pages 90–106, 2007.

[15] J. A. D. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson. A meta-heuristic approach to aircraft departure scheduling at London Heathrow airport. In M. Hickman, P. Mirchandani, and S. Voss, editors, *Computer Aided Systems of Public Transport*, Lecture Notes in Economics and Mathematical Systems. Springer, Berlin, 2008. Older version accessible as [9].

[16] J. A. D. Atkin, E. K. Burke, J. S. Greenwood, and D. Reeson. On-line decision support for take-off runway scheduling with uncertain taxi times at London Heathrow airport. *The Journal of Scheduling* (to appear), 2008.

[17] S. Atkins. Estimating departure queues to study runway efficiency. *Journal of Guidance, Control and Dynamics*, volume 25, number 4, pages 651–657, 2002.

[18] S. Atkins and C. Brinton. Concept description and development plan for the surface management system. *Journal of Air Traffic Control*, volume 44, number 1, 2002.

[19] S. C. Atkins and W. D. Hall. A case for integrating the CTAS Traffic Management Advisor and the Surface Management System. In *proceedings of the AIAA Guidance, Navigation, and Control Conference, Denver, CO*, August 2000. Report AIAA-2000-4471.

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

[21] S. Baase. *Computer Algorithms*. Addison Wesley, 1988. ISBN 0-201-06035-3.

[22] H. Balakrishnan and B. Chandran. Scheduling aircraft landings under constrained position shifting. In *proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado*, pages 21–24, August 2006. Report AIAA-2006-6320.

[23] E. Balas and N. Simonetti. Linear time dynamic-programming algorithms for new classes of restricted tsps: A computational study. *INFORMS Journal on Computing*, volume 13, number 1, pages 56–75, 2000.

[24] C. Barnhart, P. Belobaba, and A. R. Odoni. Applications of operations research in the air transport industry. *Transportation Science*, volume 37, number 4, pages 368–391, 2003.

[25] M. Bazargan, K. Fleming, and P. Subramanian. A simulation study to investigate runway capacity using taam. In *proceedings of the Winter Simulation Conference, San Diego, California, USA*, volume 2, pages 1235–1243, December 2002.

[26] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson. Scheduling aircraft landings - the static case. *Transportation Science*, volume 34, pages 180–197, 2000.

[27] J. E. Beasley, M. Krishnamoorthy, Y. M. Sharaiha, and D. Abramson. Displacement problem and dynamically scheduling aircraft landings. *Journal of the Operational Research Society*, volume 55, number 1, pages 54–64, 2004.

[28] J. E. Beasley, J. Sonander, and P. Havelock. Scheduling aircraft landings at london heathrow using a population heuristic. *Journal of the Operational Research Society*, volume 52, pages 483–493, 2001.

[29] M. Bellmore and G. L. Nemhauser. The traveling salesman problem: A survey. *Operations Research*, volume 16, pages 538–558, 1968.

[30] J. Bennell and K. A. Dowsland. Hybridising tabu search with optimisation techniques for irregular stock-cutting. *Management Science*, volume 47, number 8, pages 1160–1172, 2001.

[31] 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*, volume 86, pages 393–415, 1999.

[32] L. Bianco, P. Dell'Olmo, and S. Giordani. Scheduling models for air traffic control in terminal areas. *Journal of Scheduling*, volume 9, pages 223–253, 2006.

[33] L. Bianco, A. Mingozzi, and S. Ricciardelli. The travelling salesman problem with cumulative costs. *Networks*, volume 23, pages 81–91, 1993.

[34] L. Bianco, S. Ricciardelli, G. Rinaldi, and A. Sassano. Scheduling tasks with sequence-dependent processing times. *Naval Research Logistics*, volume 35, pages 177–184, 1988.

[35] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, volume 35, number 3, pages 268–308, 2003.

[36] A. Blumstein. Software tools to support research on airport departure planning. *Operations Research*, volume 7, number 6, pages 752–763, 1959.

[37] J. Boesel, C. X. Gladstone, J. Hoffman, P. A. Massimini, C. Shiotsuki, and B. Simmons. TAAM best practices guidelines. Available at: `http://www.mitre.org/work/tech_papers/tech_papers_01/index.html`.

[38] M. A. Bolander. *Scheduling and control strategies for the departure problem in air traffic control*. PhD thesis, University of Cincinnati, 2000.

[39] M. A. Bolender and G. L. Slater. Analysis and optimization of departure sequences. In *proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Denver, CO*, pages 1672–1683, August 2000.

[40] C. R. Brinton. An implicit enumeration algorithm for arrival aircraft scheduling. In *proceedings of the 11th Digital Avionics Systems Conference, Seattle, Washington*, pages 268–274, October 1992.

[41] R. Bronson and G. Naadimuthu. *Operations Research*. Schaum's Outlines. McGraw-Hill, New York, second edition, 1998. ISBN 0-07-008020-8.

[42] P. Brucker, editor. *Scheduling Algorithms*. Springer, Berlin Heidelberg, second edition, 1998. ISBN 0-13-02813807.

[43] R. E. Burkard, V. G. Deineko, R. van Dal, J. A. van der Veen, and G. J. Woeginger. Well-solvable special cases of the traveling salesman problem: A survey. *SIAM Review*, volume 40, pages 496–546, 1998.

[44] E. K. Burke, P. Cowling, P. De Causmaecker, and G. Vanden Berghe. A memetic approach to the nurse rostering problem. *Applied Intelligence*, volume 15, number 3, pages 199–214, 2001.

[45] E. K. Burke, P. I. Cowling, and R. Keuthen. Effective local and guided variable neighbourhood search methods for the asymmetric travelling salesman problem. In E. J. W. Boers, S. Cagnoni, J. Gottlieb, E. Hart, P. L. Lanzi, G. Raidl, R. E. Smith, and H. Tijink, editors, *Applications of Evolutionary Computing. EvoWorkshops 2001: EvoCOP,*

EvoFlight, EvoIASP, EvoLearn, and EvoSTIM, Como, Italy, April 18-20, 2001 Proceedings, volume 2037 of *Lecture Notes in Computer Science*, pages 203–212. Springer-Verlag, 2001.

[46] E. K. Burke and G. Kendall. Introduction. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 5–18. Springer, 2005.

[47] E. K. Burke and G. Kendall, editors. *Search Methodologies*. Springer, 2005.

[48] E. K. Burke, G. Kendall, and E. Soubeiga. A tabu-search hyperheuristic for timetabling and rostering. *Journal of Heuristics*, volume 9, number 6, pages 451–470, December 2003.

[49] S. Caprì and M. Ignaccolo. Genetic algorithms for solving the aircraft-sequencing problem: the introduction of departures into the dynamic model. *Journal of Air Transport Management*, volume 10, pages 345–351, 2004.

[50] F. Carr. *Robust decision-support tools for airport surface traffic*. PhD thesis, MIT ICAT, 2003.

[51] F. Carr, A. Evans, J.-P. Clarke, and E. Feron. Modelling and control of airport queuing dynamics under severe flow restrictions. In *proceedings of the American Control Conference*, volume 2, pages 1314–1319, 2002.

[52] V. Cheng, L. Crawford, and P. Menon. Air traffic control using genetic search techniques. In *proceedings of the IEEE Internation Conference on Control Applications*, volume 1, pages 249–254, 1999.

[53] P. Choroba. A vision of wake vortex research for next 20 years. In *proceedings of the 5th USA/Europe Air Traffic Management R&D Seminar, Budapest, Hungary*, June 2003.

[54] J. Cirasella, D. S. Johnson, L. A. McGeoch, and W. Zhang. The asymmetric traveling salesman problem: Algorithms, instance generators, and tests. In A. L. Buchsbaum and J. Snoeyink, editors, *Algorithm Engineering and Experimentation. Third International Workshop, ALENEX 2001, Washington, DC, USA, January 5-6, 2001. Revised Papers*, volume 2153 of *Lecture Notes in Computer Science*, pages 32–59. Springer-Verlag, 2001.

[55] J. Clayton and B. Capozzi. Dynamic airport configuration and resource scheduling. In *proceedings of the AIAA Aviation Technology Integration and Operations Forum*, volume 1, pages 160–171, 2004.

[56] G. Confessore, G. Liotta, and R. Grieco. A simulation-based architecture for supporting strategic and tactical decisions in the apron of Rome-Fiumicino airport. In *proceedings of the Winter Simulation Conference, Orlando, FL*, pages 1596–1605, December 2005.

[57] H. G. Daellenback, J. A. George, and D. C. McNiche, editors. *Introduction to Operations Research Techniques*. Allyn & Bacon, second edition, 1983. ISBN 0205079741.

[58] M. Dalal, B. Groel, and A. Prieditis. Real-time decision making using simulation. In *proceedings of the Winter Simulation Conference, New Orleans, Louisiana, USA*, volume 2, pages 1670–1676, December 2003.

[59] T. J. Davis, D. Isaacson, J. den Braven, K. Lee, and B. Sanford. Operational test results of the passive final approach spacing tool. In *proceedings of the 8th IFAC Symposium on Transportation Systems, Chania, Greece*, June 1997.

[60] T. J. Davis, K. J. Krzeczowski, and C. Bergh. The final approach spacing tool. In *proceedings of the 13th IFAC Symposium on Automated Control in Aerospace, Palo Alto, California*, September 1994.

[61] P. L. de Matos. The application of operational research to european air traffic flow management - understanding the context. *European Journal of Operational Research*, volume 123, number 1, pages 125–144, 2000.

[62] P. L. de Matosa and R. Ormerod. The application of operational research to european air traffic flow management understanding the context. *European Journal of Operational Research*, volume 123, number 1, pages 125–144, May 2000.

[63] R. G. Dear and Y. S. Sherif. The dynamic scheduling of aircraft in high density terminal areas. *Microelectronics and Reliability*, volume 29, number 5, pages 743–749, 1989.

[64] R. G. Dear and Y. S. Sherif. An algorithm for computer assisted sequencing and scheduling of terminal area operations. *Transportation Research Part A, Policy and Practive*, volume 25, pages 129–139, 1991.

[65] K. Deb. Multi-objective optimization. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 273–316. Springer, 2005.

[66] Department for Transport (White paper). The future of air transport. Appraisal framework for airports in South East and East of England, 2003. Available at : `http://www.dft.gov.uk/stellent/groups/dft_aviation/documents/pdf/dft_aviation_pdf_031504.pdf`.

[67] J. A. Díaz and E. Fernández. A tabu search heuristic for the generalized assignment problem. *European Journal of Operational Research*, volume 132, pages 22–38, 2001.

[68] G. L. Donahue and W. D. Laska. United states and european airport capacity assessment using the GMU macroscopic capacity model (MCM). In *proceedings of the 3th USA/Europe Air Traffic Management R&D Seminar, Napoli*, June 2000.

[69] G. L. Donahue and D. K. Rutishauser. The effect of aircraft wake vortex separation on air transport capacity. In *proceedings of the 4th USA/Europe Air Traffic Management R&D Seminar, Sata Fe, New Mexico*, December 2001.

[70] U. Dorndorfa, A. Drexlb, Y. Nikulinb, and E. Peschc. Flight gate scheduling: State-of-the-art and recent developments. *Omega*, volume 35, number 3, pages 326–334, June 2007.

[71] K. A. Dowsland. Classical techniques. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 19–68. Springer, 2005.

[72] G. Duke. *Air Traffic Control*. Ian Allan publishing, eighth edition, 2001.

[73] N. Durand and J.-M. Alliot. Genetic crossover operator for partially separable functions. In J. R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. B. Fogel, M. H. Garzon, D. E. Goldberg, H. Iba, and R. Riolo, editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 487–494. Morgan Kaufmann, July 1998.

[74] A. T. Ernst, M. Krishnamoorthy, and R. H. Storer. Heuristic and exact algorithms for scheduling aircraft landings. *Networks*, volume 34, pages 229–241, 1999.

[75] EUROCONTROL Experimental Centre. TAAM Operational Evaluation, EEC Report No 351, Project SIM-S-E8, 2000. Available at: `http://www.eurocontrol.int:80/eec/gallery/content/public/documents/EEC_reports/2000/EEC_report_351.pdf` [22 November 2006].

[76] EUROCONTROL Experimental Centre. Heathrow CDM information leaflet v3, June 2005. Available at : `http://www.euro-cdm.org/docs/tf11/Heathrow%20CDM%20information%20leaflet%20V3.pdf`.

[77] EUROCONTROL Experimental Centre. London Heathrow CDM WP1, EEC Note No. 03/05, February 2005. Available at : `http://www.eurocontrol.int/eec/gallery/content/public/documents/EEC_notes/2005/EEC_note_2005_03.pdf`.

[78] T. Fahle, R. Feldmann, S. Götz, S. Grothklags, and B. Monien. The aircraft sequencing problem. In R. Klein, H.-W. Six, and L. Wegner, editors, *Computer Science In Perspective. Essays Dedicated to Thomas Ottmann*, number 2598 in Lecture Notes In Computer Science, pages 152–166. Springer-Verlag Berlin Heidelberg, 2003.

[79] J. A. Filar, P. Manyem, and K. White. How airlines and airports recover from schedule perturbations: a survey. *Annals of Operations Research*, volume 108, pages 315–333, 2001.

[80] E. C. Freuder and M. Wallace. Constraint programming. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 239–272. Springer, 2005.

[81] J. Garcia, A. Berlanga, J. M. Molina, and J. R. Casar. Optimization of airport ground operations integrating genetic and dynamic flow management algorithms. *AI Communications*, volume 18, number 2, pages 143–164, 2005.

[82] M. Gendreau. An introduction to tabu search. In F. Glover and G. Kochenberger, editors, *Handbook of Metaheuristics*, pages 37–54. Kluwer Academic Publishers, 2002.

[83] M. Gendreau and J.-Y. Potvin. Metaheuristics in combinatorial optimization. *Annals of Operations Research*, volume 140, pages 189–213, 2005.

[84] M. Gendreau and J.-Y. Potvin. Tabu search. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 165–186. Springer, 2005.

[85] E. P. Gilbo. Arrival/departure tradeoff optimisation: a case study at the st. louis lambert international airport (stl). In *proceedings of the 5th USA/Europe Air Traffic Management R&D Seminar, Budapest, Hungary*, June 2003.

[86] P. C. Gilmore, E. L. Lawler, and D. B. Shmoys. Well solved special cases. In E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, editors, *The Travelling salesman problem, a guided tour of combinatorial optimization*. John Wiley and Sons Ltd, 1985.

[87] F. Glover and G. Kochenberger, editors. *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2002.

[88] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.

[89] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, volume 8, pages 156–166, 1977.

[90] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, volume 13, pages 533–549, 1986.

[91] F. Glover. Tabu search - part i. *ORSA Journal on Computing*, volume 1, number 3, pages 190–206, 1989.

[92] F. Glover. Tabu search - part ii. *ORSA Journal on Computing*, volume 2, number 1, pages 4–32, 1990.

[93] R. Gopalan and K. T. Talluri. Mathematical models in airline schedule planning: a survey. *Annals of Operations Research*, volume 76, pages 155–185, 1998.

[94] J.-B. Gotteland and N. Durand. Genetic algorithms applied to airport ground traffic optimization. In *proceedings of the Congress on Evolutionary Computation, Canberra, Australia*, December 2003.

[95] J.-B. Gotteland, N. Durand, and J.-M. Alliot. Handling CFMU slots in busy airports. In *proceedings of the 5th USA/Europe Air Traffic Management R&D Seminar, Budapest, Hungary*, June 2003.

[96] J.-B. Gotteland, N. Durand, J.-M. Alliot, and E. Page. Aircraft ground traffic optimization. In *proceedings of the 4th USA/Europe Air Traffic Management R&D Seminar, Santa Fe, New Mexico*, December 2001.

[97] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, volume 5, pages 287–326, 1979.

[98] H. Gröflin and A. Klinkert. Job shop scheduling with synchronization constraints and sequence-dependent set-up times (abstract). In *proceedings of the First Joint Research Days, EPFL, Lausanne, Switzerland*, June 2003.

[99] H. Gröflin and A. Klinkert. A tabu search for the generalized blocking job shop. In *proceedings of the 6th Metaheuristics International Conference*, August 2005.

[100] N. G. Hall and C. Sriskandarajah. A survey of machine scheduling problems with blocking and no-wait in process. *Operations Research*, volume 44, number 3, pages 510–525, 1996.

[101] C. A. Halverson, K. Harwood, T. J. Davis, and C. R. Brinton. A systems approach to design: Developing a usable automation tool for air traffic control. In *proceedings of the 11th AIAA Digital Avionics Systems Conference, Seattle, WA, USA*, October 1992.

[102] J. V. Hansen. Genetic search methods in air traffic control. *Computers and Operations Research*, volume 31, number 3, pages 445–459, 2004.

[103] S. L. M. Hockaday and B. K. Kanafani. Developments in airport capacity analysis. *Transportation Research*, volume 8, pages 171–180, 1974.

[104] T. C. Holden and F. Wieland. Runway schedule determination by simulation optimization. In *proceedings of the Winter Simulation Conference, New Orleans, Louisiana, USA*, volume 2, pages 1670–1676, December 2003.

[105] X.-B. Hu and W.-H. Chen. Genetic algorithm based on receding horizon control for arrival sequencing and scheduling. *Engineering Applications of Artificial Intelligence*, volume 18, number 5, pages 633–642, August 2005.

[106] X.-B. Hu and W.-H. Chen. Receding horizon control for aircraft arrival sequencing and scheduling. *IEEE Transactions on Intelligent Transportation Systems*, volume 6, number 2, pages 189–197, June 2005.

[107] H. R. Idris, I. Anagnostakis, B. Delcaire, R. Hansman, J.-P. Clarke, E. Feron, and A. R. Odoni. Observations of departure processes at Logan airport to support the development of departure planning tools. *Air Traffic Control Quarterly (ATCA Publications)*, volume 7, number 4, pages 229–257, 1999.

[108] H. R. Idris, J.-P. Clarke, R. Bhuva, and L. Kang. Queuing model for taxi-out time estimation. *Air Traffic Control Quarterly (ATCA Publications)*, volume 10, number 1, pages 1–22, 2002.

[109] H. R. Idris, B. Delcaire, I. Anagnostakis, W. D. Hall, J.-P. Clarke, R. 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.

[110] H. R. Idris, B. Delcaire, I. Anagnostakis, W. D. Hall, N. Pujet, E. Feron, R. Hansman, J.-P. Clarke, and A. R. Odoni. Identification of flow constraint and control points in departure operations at airport systems. In *proceedings of the AIAA Guidance, Navigation and Control conference, Boston, MA*, pages 947–956, August 1998.

[111] H. R. Idris. *Observations and Analysis of Departure Operations at Boston Logan International Airport*. PhD thesis, MIT, 2000.

[112] International Air Transport Association. Annual report 2006. Available at : `http://www.iata.org/iata/Sites/agm/file/2006/file/annual_report_06.pdf`.

[113] ir. F.R. Polak. Airport modelling: Capacity analysis of schiphol airport in 2015. In *proceedings of the 1st USA/Europe Air Traffic Management R&D Seminar, Saclay, France*, 1997.

[114] D. S. Johnson and C. H. Papadimitriou. Computational complexity. In E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, editors, *The Travelling salesman problem, a guided tour of combinatorial optimization*. John Wiley and Sons Ltd, 1985.

[115] D. S. Johnson, G. Gutin, L. A. McGeoch, A. Yeo, W. Zhang, and A. Zverovitch. Experimental analysis of heuristics for the atsp. In G. Gutin and A. P. Punnen, editors, *The Traveling Salesman Problem and Its Variations*, volume 12 of *Combinatorial Optimization*. Kluwer, 2002.

[116] D. S. Johnson and L. A. McGeoch. Experimental analysis of heuristics for the stsp. In G. Gutin and A. P. Punnen, editors, *The Traveling Salesman Problem and Its Variations*, volume 12 of *Combinatorial Optimization*. Kluwer, 2002.

[117] G. Jung and M. Laguna. Time segmenting heuristic for an aircraft landing problem. Unpublished work, previously available from `leeds-faculty.colorado.edu`, 2003.

[118] Y. C. Jung and G. A. Monroe. Development of Surface Management System integrated with CTAS arrival tool. In *proceedings of the 5th AIAA Aviation Technology, Integration, and Operations Conference, Arlington, VA*, September 2005. Report AIAA-2005-7334.

[119] G. Kendall and N. Mohd Hussin. Tabu search hyper-heuristic approach to the examination timetabling problem at university technology mara. In E. K. Burke and M. Trick, editors, *proceedings of the 5th international conference on the Practice and Theory of Automated Timetabling (PATAT), Pittsburgh, USA*, pages 199–217, August 2004.

[120] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, number 220, pages 671–680, 1983.

[121] A. Klinkert and H. Gröflin. Scheduling with generalised disjunctive graphs: feasibility issues (abstract). In *proceedings of the 15th conference of the European chapter on combinatorial optimisation, Lugano, Switzerland*, May 2002.

[122] J. R. Koza and R. Poli. Genetic programming. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 127–164. Springer, 2005.

[123] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys, editors. *The Travelling salesman problem, a guided tour of combinatorial optimization.* John Wiley and Sons Ltd, 1985. ISBN 0471904139.

[124] 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 report from the 40th European Study Group with Industry, Keele*, 2001. Available from `http://www.smithinst.ac.uk/Projects/ESGI40/ESGI40-NATS/Report` [17th September 2007].

[125] K. Lindsay, E. A. Boyd, G. Booth, and C. Harvey. The use of optimisation to perform air traffic flow management. In G. Yu, editor, *Operations Research in the Airline Industry*, International Series in Operations Research & Management Science. Kluwer Academic Publishers, 1998. ISBN 0792380398.

[126] R. A. Luenberger. A travelling-salesman-based approach to aircraft scheduling in the terminal area, 1988. NASA report TM-100062, Available at: `http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19880010040_1988010040.pdf`.

[127] A. G. Marìn. Airport management: taxi planning. *Annals of Operations Research*, volume 143, pages 191–202, 2006.

[128] A. Mascis and D. Pacciarelli. Machine scheduling via alternative graphs, 2000. Report DIA-46-2000, Dipartimento di Informatica e Automazione, Università Roma Tre, Roma, Italy.

[129] A. Mascis and D. Pacciarelli. Job-shop scheduling with blocking and no-wait constraints. *European Journal of Operational Research*, volume 143, pages 498–517, 2002.

[130] D. F. X. Mathaisel and H. Idris. Aircraft ground movement simulation. In G. Yu, editor, *Operations Research in the Airline Industry*, International Series in Operations Research & Management Science. Kluwer Academic Publishers, 1998. ISBN 0792380398.

[131] Z. Michalewicz and D. B. Fogel. *How to solve it: Modern metaheuristics*. Springer, Berlin, Heidelberg, New York, Hong Kong, London, Milan, Paris, Tokyo, second edition, 2000. ISBN 3-540-22494-7.

[132] Z. Michalewicz and D. B. Fogel. *How to solve it: Modern metaheuristics*, chapter 2, pages 35–48. Springer, Berlin, Heidelberg, New York, Hong Kong, London, Milan, Paris, Tokyo, second edition, 2000. ISBN 3-540-22494-7.

[133] L. Michel and P. V. Hentenryck. A simple tabu search for warehouse location. *European Journal of Operational Research*, volume 157, number 3, pages 576–591, 2004.

[134] N. Musliu, A. Shaerf, and W. Slany. Local search for shift design. *European Journal of Operational Research*, volume 153, pages 51–64, 2004.

[135] G. Newell. Airport capacity and delays. *Transportation Science*, volume 13, number 3, pages 201–240, 1979.

[136] A. R. Odoni, J. Bowman, D. Delahaye, J. J. Deyst, E. Feron, R. Hansman, K. Khan, J. K. Kuchar, N. Pujet, and R. W. Simpson. Existing and required modeling capabilities for evaluating ATM systems and concepts, 1997. Available at: `http://web.mit.edu/aeroastro/www/labs/AATT/aatt.html` [24 Nov 2006].

[137] A. R. Odoni and R. de Neufville. Passenger terminal design. *Transportation Research Part A: Policy and Practice*, volume 26, number 1, pages 27–35, January 1992.

[138] I. H. Osman and G. Laporte. Metaheuristics: A bibliography. *Annals of Operations Research*, number 63, pages 513–623, 1996.

[139] D. Pacciarelli. Alternative graph formulation for solving complex factory-scheduling problems. *International Journal of Production Research*, volume 40, number 15, pages 3641–3653, 2002.

[140] D. Pacciarelli and M. Pranzo. A tabu search algorithm for the runway scheduling problem. In *proceedings of the 4th Metaheuristics International Conference, Porto, Portugal*, July 2001.

[141] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*. Dover Publications Inc, Mineola, New York, 1998. ISBN 0-486-40258-4.

[142] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization*, chapter 19, pages 454–486. Dover Publications Inc, Mineola, New York, 1998. ISBN 0-486-40258-4.

[143] B. Pesic, N. Durand, and J.-M. Alliot. Aircraft ground traffic optimisation using a genetic algorithm. In *proceedings of the Genetic and Evolutionary Computation Conference (GECCO), San Francisco, California*, pages 1397–1404, July 2001.

[144] M. Pinedo. *Scheduling. Theory, Algorithms and Systems.* Prentice-Hall inc. Upper Saddle River, New Jersey, second edition, 2002. ISBN 0-13-02813807.

[145] D. E. Pitfield, A. S. Brooke, and E. A. Jerrard. A monte-carlo simulation of potentially conflicting ground movements at a new international airport. *Journal of Air Transport Management*, volume 4, number 1, pages 3–10, 1998.

[146] D. E. Pitfield and E. A. Jerrard. Monte carlo comes to rome: a note on the estimation of unconstrained runway capacity at rome fiumucino international airport. *Journal of Air Transport Management*, volume 5, number 4, pages 185–192, 1999.

[147] H. N. Psaraftis. A dynamic programming approach for sequencing groups of identical jobs. *Operations Research*, volume 28, number 6, pages 1347–1359, 1980.

[148] N. Pujet. *Modeling and control of the departure process of congested airports.* PhD thesis, MIT, 1999.

[149] N. Pujet, B. Delcaire, and E. Feron. Input-output modeling and control of the departure process of congested airports. *Air Traffic Control Quarterly*, volume 8, number 1, pages 1–32, 2000.

[150] X. Qi, J. Yang, and G. Yu. Scheduling problems in the airline industry. In J. Y.-T. Leung, editor, *Handbook of Scheduling - algorithms, models and performance analysis*, pages 50.1–50.15. Chapman & Hall, 2004.

[151] J. E. Robinson, T. J. Davis, and D. R. Isaacson. Fuzzy reasoning -based sequencing of arrival aircraft in the terminal area. In *proceedings of the AIAA Guidance, Navigation and. Control Conference, New Orleans, LA*, 1997. Report AIAA-1997-3542.

[152] P. Ross. Hyper-heuristics. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 529–556. Springer, 2005.

[153] D. Rutishauser, G. L. Donahue, and R. C. Haynie. Measurements of aircraft wake vortex separation at high arrival rates and a proposed new wake vortex separation philosophy. In *proceedings of the 5th USA/Europe Air Traffic Management R&D Seminar*, June 2003.

[154] S. Salhi. Defining tabu list size and aspiration criterion within tabu search methods. *Computers and operations research*, volume 29, pages 67–86, 2002.

[155] K. Sastry, D. Goldberg, and G. Kendall. Genetic algorithms. In E. K. Burke and G. Kendall, editors, *Search Methodologies*, pages 97–125. Springer, 2005.

[156] J. Shen, F. Zu, and P. Zheng. A tabu search algorithm for the routing and capacity assignment problem in computer networks. *Computers and Operations Research*, volume 32, pages 2785–2800, 2005.

[157] J. W. Smeltink, M. J. Soomer, P. R. de Waal, and R. D. van der Mei. Optimisation of airport taxi planning, NLR technical paper NLR-TP-2003-475, 2003. Available from National Aerospace Laboratory NLR, Amsterdam, the Netherlands, `http://www.nlr.nl`. Later version available as [158].

[158] J. W. Smeltink, M. J. Soomer, P. R. de Waal, and R. D. van der Mei. An optimisation model for airport taxi scheduling, 2004. Unpublished extension of [157], available from: `http://www.math.vu.nl/~mjsoomer/` [20 Aug 2007].

[159] C. Smith. Final approach spacing tool. In *proceedings of the 2nd USA/Europe Air Traffic Management R&D seminar, Orlando, USA*, December 1998.

[160] N. Sood and F. Wieland. Total airport and airspace model (taam) parallelization combining sequential and parallel algorithms for performance enhancement. In *proceedings of the Winter Simulation Conference, New Orleans, Louisiana, USA*, volume 2, pages 1650–1655, December 2003.

[161] M. A. Stamatopoulos, K. G. Zografos, and A. R. Odoni. A decision support system for airport strategic planning. *Transportation Research Part C: Emerging Technologies*, volume 12, number 2, pages 91–117, 2004.

[162] R. Teixeira. *An heuristic for the improvement of aircraft departure scheduling at airports.* PhD thesis, Loughborough Technical University, 1992.

[163] Thematic Network on Airport Activities (THENA). D1.4.1 Position Paper on Airport Simulation & Modelling Issues, 2002. Available at: `http://thena.aena.es/thena_public/files/WP1/04Simulation/14D01AEN10_SimulationModelling_PositionPaper.pdf` [24 Nov 2006].

[164] T. R. Thompson, S. Bradford, D. Liang, and M. Brennan. Terminal area throughput: measuring capacity and robustness. In *proceedings of the 5th USA/Europe Air Traffic Management R&D Seminar, Budapest, Hungary*, June 2003.

[165] V. Tosic. A review of airport passenger terminal operations analysis and modelling. *Transportation Research Part A: Policy and Practice*, volume 26, number 1, pages 3–26, February 1992.

[166] D. A. Trivizas. Optimal scheduling with maximum position shift (MPS) constraints: A runway scheduling application. *Journal of Navigation*, volume 51, pages 250–266, 1998.

[167] N. M. van Dijk and E. van der Sluis. Check-in computation and optimization by simulation and ip in combination. *European Journal of Operational Research*, volume 171, number 3, pages 1152–1168, June 2006.

[168] P. van Leeuwen, H. Hesselink, and J. Rohling. Scheduling aircraft using constraint satisfaction. *Electronic Notes in Theoretical Computer Science*, volume 76, 2002.

[169] H. Visser and P. Roling. Optimal airport surface traffic planning using mixed integer linear programming. In *proceedings of the AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO), Colorado, USA*, November 2003. Report AIAA-2003-6797.

[170] G. M. White, B. S. Xie, and S. Zonjic. Using tabu search with longer-term memory and relaxation to create examination timetables. *European Journal of Operational Research*, volume 153, pages 80–91, 2004.

[171] C.-L. Wu and R. E. Caves. Research review of air traffic management. *Transport Reviews*, volume 22, number 1, pages 115–132, 2002.

[172] C.-L. Wu and R. Caves. Modelling and optimization of aircraft turnaround time at an airport. *Transportation Planning & Technology*, volume 27, number 1, pages 47–66, February 2004.

[173] H. Youssef, S. M. Sait, and H. Adiche. Evolutionary algorithms, simulated annealing and tabu search: a comparative study. *Engineering Applications of Artificial Intelligence*, volume 14, pages 167–181, 2001.

[174] G. Yu, editor. *Operations Research in the Airline Industry.* International Series in Operations Research & Management Science. Kluwer Academic Publishers, 1998. ISBN 0792380398.

[175] W. Zhang. Truncated branch-and-bound: A case study on the asymmetric travelling salesman problem. In *proceedings of AAAI 1993 Spring Symposium on AI and NP-Hard Problems, Stanford, CA, March 23-25*, pages 160–166, 1993.

[176] W. Zhang. Depth-first branch-and-bound versus local search: A case study. In *proceedings of the National Conference on Artificial Intelligence*, volume 17, pages 930–936, 2000.

Appendix A

Speed-based Separation Changes

Standard (SID) route-based separations between aircraft were explained in section 2.3. These separation rules need to be modified according to the speed groups of the aircraft involved.

At present there are six different speed separation rules that can be applied to modify the standard required separation for the combination of departure routes of the aircraft. Both the initial separation and the speed rule to use depend upon the departure routes and the runway the aircraft are using. The rule number to use can be found from tables A.1 to A.4 by cross-referencing the SID-route for the leading aircraft down the left side of the table with the SID-route for the following aircraft along the top of the table. The value at the intersection in the table specifies the speed modification rule to use.

The rule number determines which of the tables A.5 to A.10 to consult to determine the modification that must be made to the separation time. The separation modification value is found by cross-referencing the speed group of the leading aircraft down the left side of the table with the speed group for the following aircraft along the top of the table. Unless an asterisk (*) is present, the value in the table should be added to the basic separation for the routes that was obtained from tables 2.1 to 2.4 in chapter 2. If an asterisk (*) is present, the value should be used instead of the original separation value.

TABLE A.1: Separation modification rule for speed groups, runway 27R, by SID route of the leading (rows) and following (columns) aircraft

SID	BPK	WOB	DET	DVR	CPT	SAM	MID	MAY
BPK	2	2	4	4	6	6	3	4
WOB	1	2	4	4	6	6	3	4
DET	4	4	2	2	4	4	5	2
DVR	4	4	2	1	4	4	5	1
CPT	6	6	4	4	2	2	4	4
SAM	6	6	4	4	2	2	4	4
MID	3	3	5	5	4	4	2	5
MAY	4	4	2	1	4	4	5	1

TABLE A.2: Separation modification rule for speed groups, runway 27L, by SID route of the leading (rows) and following (columns) aircraft

SID	BPK	WOB	DET	DVR	CPT	SAM	MID	MAY
BPK	2	2	3	3	6	6	3	3
WOB	2	1	3	3	6	6	3	3
DET	3	3	2	2	3	3	5	2
DVR	3	3	2	1	3	3	5	1
CPT	6	6	3	3	2	2	4	3
SAM	6	6	3	3	2	2	4	3
MID	3	3	5	5	4	4	2	5
MAY	3	3	2	1	3	3	5	1

TABLE A.3: Separation modification rule for speed groups, runway 09R, by SID routes of the leading (rows) and following (columns) aircraft

SID	BPK	BUZ	DET	DVR	CPT	SAM	MID	MAY
BPK	2	2	4	4	4	3	3	3
BUZ	2	2	4	4	4	3	3	3
DET	4	4	2	2	4	4	4	3
DVR	4	4	2	2	4	4	4	3
CPT	4	4	4	4	2	4	4	2
SAM	3	3	4	4	4	2	1	1
MID	3	3	4	4	4	2	2	2
MAY	3	3	3	3	2	1	1	2

TABLE A.4: Separation modification rule for speed groups, runway 09L, by SID routes of the leading (rows) and following (columns) aircraft

SID	BPK	BUZ	DET	DVR	CPT	SAM	MID	MAY
BPK	2	2	3	3	3	3	3	3
BUZ	2	2	3	3	3	3	3	3
DET	3	3	2	2	4	4	4	2
DVR	3	3	2	2	4	4	4	2
CPT	3	3	4	4	2	3	3	2
SAM	3	3	4	4	4	2	1	1
MID	3	3	4	4	4	2	2	2
MAY	3	3	2	2	2	1	1	2

TABLE A.5: Separation modification rule 1, by speed groups of leading (rows) and following (columns) aircraft

Speed group	0	1	2	3	4
0	0	60	120	180	420*
1	0	0	60	120	360*
2	0	0	0	60	300*
3	0	0	0	0	240*
4	0	0	0	0	0

TABLE A.6: Separation modification rule 2, by speed groups of leading (rows) and following (columns) aircraft

Speed group	0	1	2	3	4
0	0	60	120	180	420*
1	0	0	60	120	360*
2	-60	0	0	60	300*
3	-60	-60	0	0	240*
4	-60	-60	-60	-60	0

TABLE A.7: Separation modification rule 3, by speed groups of leading (rows) and following (columns) aircraft

Speed group	0	1	2	3	4
0	0	0	0	0	60
1	0	0	0	0	60
2	0	0	0	0	60
3	0	0	0	0	60
4	0	0	0	0	0

TABLE A.8: Separation modification rule 4, by speed groups of leading (rows) and following (columns) aircraft

Speed group	0	1	2	3	4
0	0	0	0	60	120
1	0	0	0	0	120
2	0	0	0	0	60
3	0	0	0	0	60
4	0	0	0	0	0

TABLE A.9: Separation modification rule 5, by speed groups of leading (rows) and following (columns) aircraft

Speed group	0	1	2	3	4
0	0	60	60	60	120
1	0	0	60	60	120
2	0	0	0	60	120
3	0	0	0	0	120
4	0	0	0	0	0

TABLE A.10: Separation modification rule 6, by speed groups of leading (rows) and following (columns) aircraft

Speed group	0	1	2	3	4
0	0	0	60	60	120
1	0	0	0	60	120
2	0	0	0	0	120
3	0	0	0	0	60
4	0	0	0	0	0

Appendix B

The CTOT Compliance Penalty Function

The function $C(d_i, b_i, l_i, h_i)$, was defined in section 5.6.2 by equation 5.4 and is used by the objective function to determine an appropriate penalty to apply to penalise any take-off schedule which schedules aircraft to take-off outside of the allocated CTOT take-off time-slots. Due to its complexity, an illustration of the operation of function $C(d_i, b_i, l_i, h_i)$ is given in this appendix.

Figures B.1 to B.3 illustrate how the cost that is calculated by $C(d_i, b_i, l_i, h_i)$ varies depending upon whether the aircraft takes off within its CTOT time-slot, within an allowable five minute extension to the CTOT time-slot or more than five minutes after the end of the CTOT time-slot, so too late for even an extension to be used.

The cost returned by $C(d_i, b_i, l_i, h_i)$ is always zero for any aircraft which takes off no later than the end of the take-off time-slot. The number of seconds that the take-off time is later than the end of the take-off time-slot is important and is plotted on the x-axis of figures B.1 to B.3. The contribution of the function $C(d_i, b_i, l_i, h_i)$ to the objective function is plotted on the y-axis.

Figure B.1 illustrates the three functions (labelled A, B and C in figure B.1) that are used to determine the cost. Line A, shown in red, relates to term (ii) of equation 5.4, line B, shown in green, relates to term (iii) of equation 5.4, and line C, shown in blue, relates to term (iv) of equation 5.4. When take-off is more than five minutes after the end of the CTOT time-slot, the cost is illustrated by the blue line C, and is extremely large.

When an aircraft takes off within five minutes of the end of the CTOT time-slot, the cost will be determined by either A or B depending upon how long the aircraft had been waiting in the holding area, as discussed below. Figure B.2 illustrates this relationship in more detail. As long as take-off takes place within F_H seconds of arriving at the holding area, the lower cost is applied. This is illustrated by line A in figures B.1 and B.2, which illustrates the value of term (ii) of equation 5.4. If take-off is not within F_H seconds of arriving at the holding area, then line B in figures B.1 and B.2 illustrates the cost, which is determined by term (iii) of equation 5.4.

The point X on figure B.2 relates to the value $(h_i + F_H)$ for any aircraft i which is under consideration. The exact position will change for each aircraft as it depends upon both the

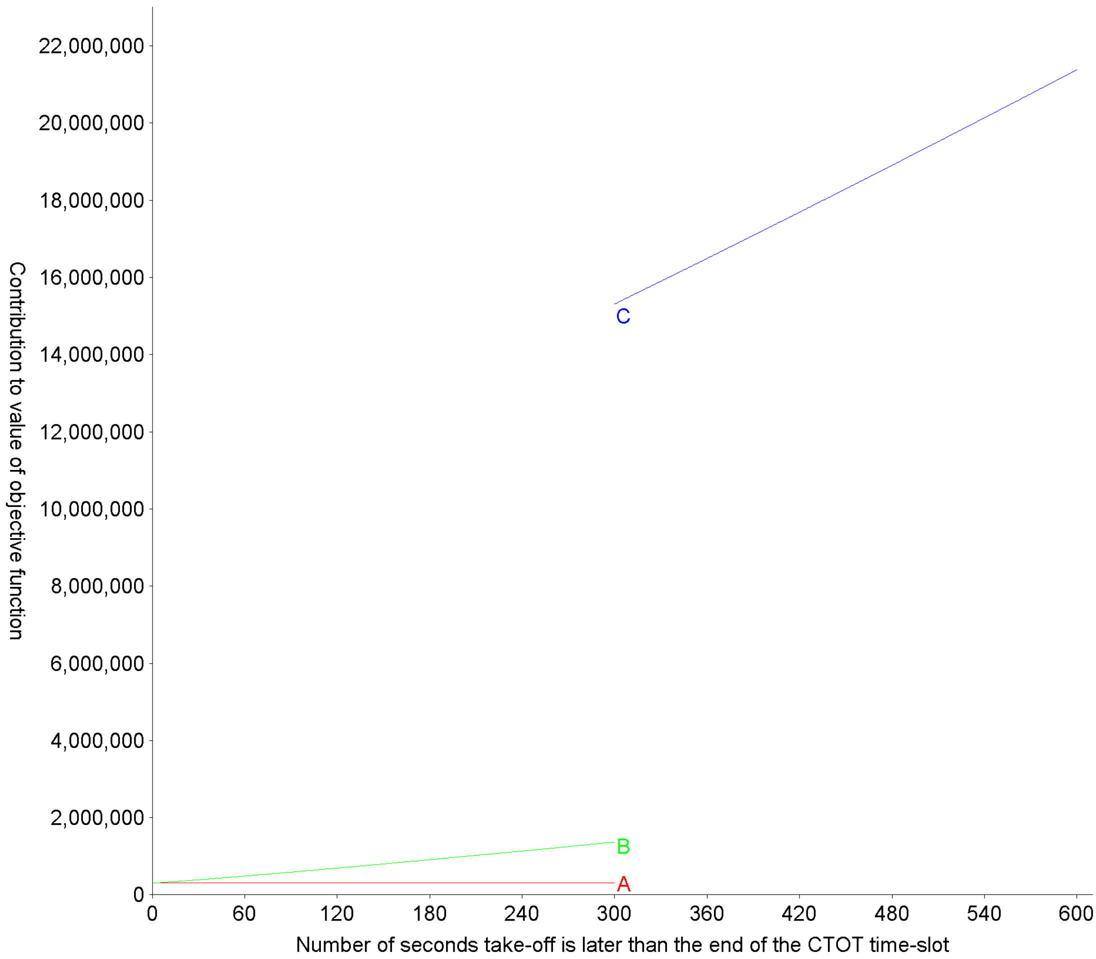


FIGURE B.1: Illustration of the function C(), illustrating how the cost varies with the delay between the end of the CTOT time-slot and the take-off time

number of seconds of leeway allowed to aircraft (specified by the constant F_H) and the holding area arrival time, h_i, of that individual aircraft, i. Until the point X, the cost for aircraft i will follow line A. Beyond point X, the cost will follow line B. If aircraft i arrives early enough at the holding area, such that $h_i \leq (l_i + F_H)$, or equivalently, $(h_i + F_H) \leq l_i$, then point X is to the left of the origin in figure B.2, and the cost for aircraft i taking off within five minutes of l_i will be illustrated entirely by line B. Otherwise, there is a discontinuity in the cost function at point X, as the cost jumps from following line A to line B at that point.

Due to the large y-axis scale in figures 1 and 2, line A may appear to be horizontal. Figure B.3 has been provided to illustrate that this is not the case, the value of term (ii) in equation 5.4 actually increases linearly as the take-off time is increased, where the gradient of line A is given by the value of ω_1, which had the value 1 in the experiments performed for this thesis. (Note that any plot of the green line B (term (iii)) on figure B.3 would appear to be almost vertical due to the scale of the y-axis compared with the rate at which term (iii) increases.)

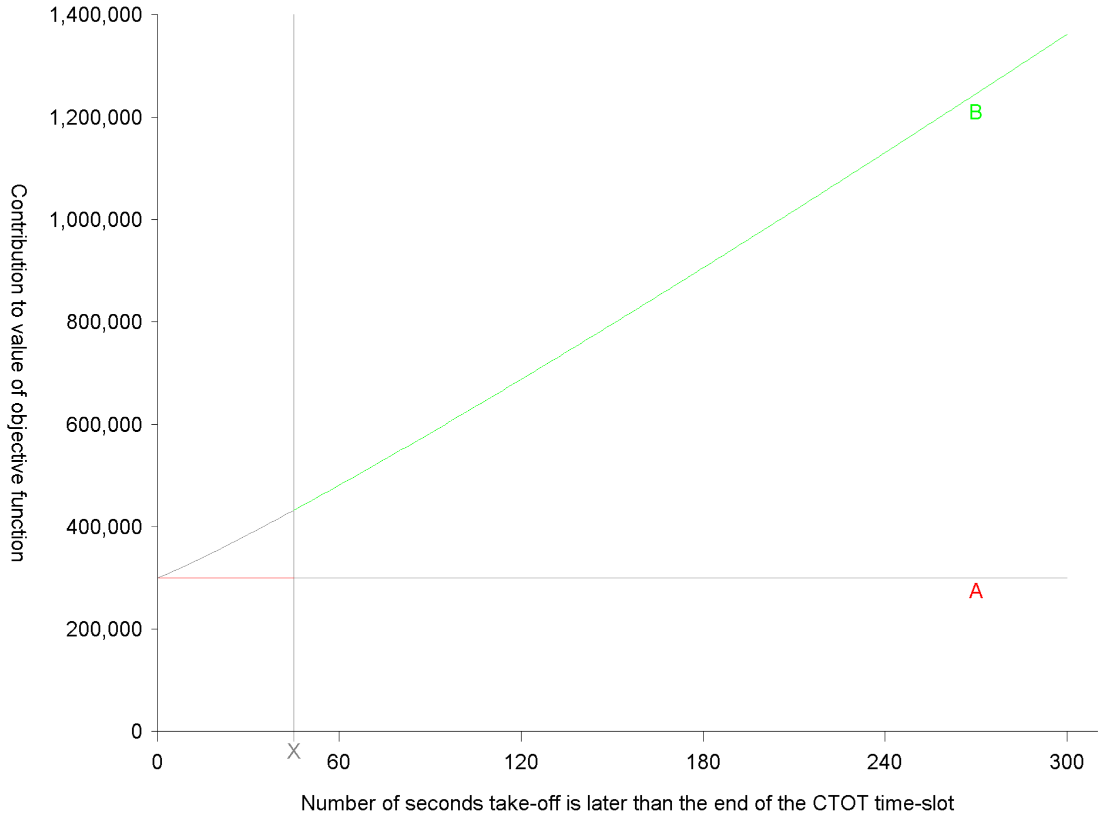


FIGURE B.2: Illustration of the function C(), showing how the terms (ii) and (iii) of equation 5.4 are related

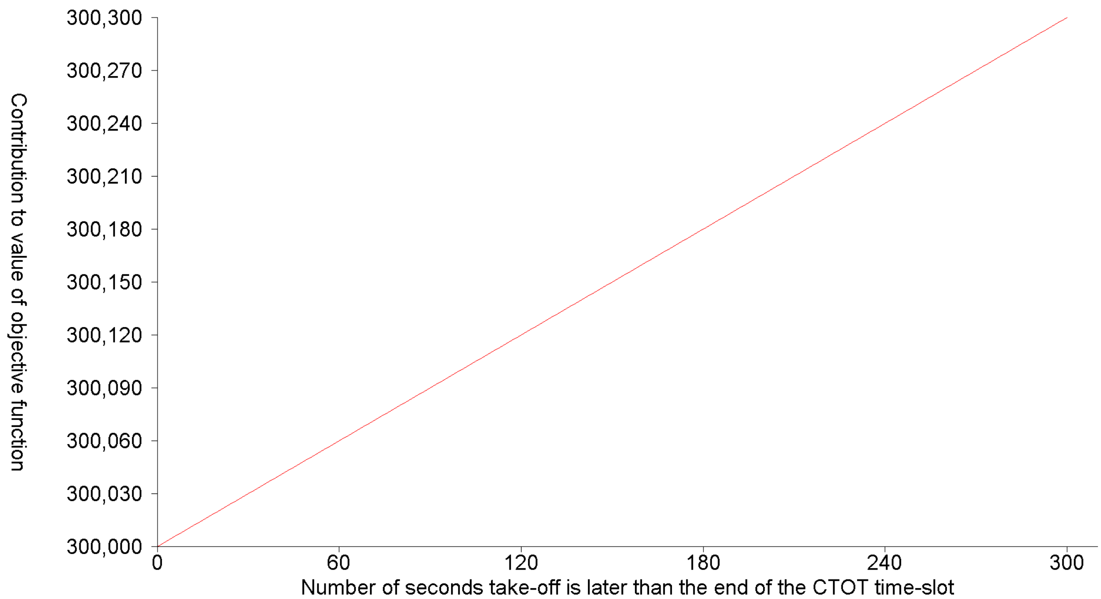


FIGURE B.3: Illustration of the function C(), showing that term (ii) increases with take-off time

Appendix C

Comparison Of Algorithms

In order to understand the search space and determine the presence of local optima, a first descent and steeper descent algorithm were applied to the take-off problem along with a tabu search and simulated annealing algorithm. Initial results of these comparisons were presented in [15]. The algorithms were also applied to the system presented in [14] and the results of doing this are presented here. The tabu search algorithm used for these results was the same as that described in chapter 5. The other three algorithms are the same as those used in [15] and are described below. In each case the method for generating new solutions, evaluating solutions and determining feasibility was exactly the same as described in chapters 5 and 6.

The termination conditions were based upon a number of iterations in each case. The first descent and simulated annealing algorithms were executed for ten thousand iterations. The steeper descent and tabu search algorithms were executed for two hundred iterations as they evaluate fifty solutions per iteration.

The simulation used to evaluate the systems was a simplified version of that described in chapter 7 and is the same as described in [14]. These results are presented only to illustrate the fact that there are (possibly shallow) local optima in the search space, even with the large neighbourhood provided to the algorithms. The use of a meta-heuristic rather than simple local search is, therefore, justified.

C.1 Definitions

In the following algorithms, let S_n denote the current solution under consideration and C_n the cost of the current solution. Let S_0 denote an initial feasible solution. This can be created in the same way as for the tabu search. Let S_B denote the best solution found so far and C_B the cost of the best solution found so far. Let S_C denote a candidate solution and C_C the cost of the candidate solution.

C.2 The First Descent Algorithm

Algorithm 18 First descent algorithm

let $S_n = S_0$
evaluate the current solution, S_n, and determine a cost, C_n
while termination condition not met **do**
  generate a new candidate solution, S_C, by random selection from the neighbourhood of S_n
  evaluate S_C and determine a cost C_C
  if $C_C \leq C_n$ **and** S_C is feasible **then**
    adopt the candidate solution as the current solution, let $S_n = S_C$ and $C_n = C_C$
  end if
end while
return S_n, the best solution found

If the only local optima in the search space are also globally optimal, then the first descent algorithm is the simplest way to obtain an optimal solution to the problem. The first descent algorithm used is described in algorithm 18. As the algorithm only accepts improving or equal valued moves, the current solution is always the best solution found so far.

C.3 The Steeper Descent Algorithm

Algorithm 19 Steeper descent algorithm

let the current solution $S_n = S_0$
evaluate the current solution, S_n, and determine a cost, C_n
set $S_B = S_n$ and $C_B = C_n$
while termination condition not met **do**
  randomly generate a set, P, of fifty new candidate solutions by random selection from the neighbourhood of S_n
  evaluate the feasibility of each solution in P and remove from P any which are infeasible
  if P is not empty **then**
    evaluate the cost for each solution in P
    determine the lowest cost solution, S_C, in P
    adopt S_C as the new current solution, let $S_n = S_C$ and $C_n = C_C$
    if $C_C < C_B$ **then**
      let $S_B = S_C$ and $C_B = C_C$
    end if
  end if
end while
return S_B, the best solution found

The implemented steeper descent algorithm was exactly the same as the tabu search algorithm with the exception that it did not have a tabu list and is described by algorithm 19. This meant it could immediately return to solutions it had recently considered, so had only a very limited ability to escape local optima, as described in section 5.2.2.

C.4 The Simulated Annealing Algorithm

Simulated Annealing was first introduced by Kirkpatrick, Gelatt and Vecchi in [120]. A recent introduction to simulated annealing can be found in [1]. Simple simulated annealing is relatively straightforward to apply. The algorithm implemented here is very similar to the first descent algorithm. However, the stochastic solution acceptance criteria give it the ability to move to worse solutions rather than being a strict descent algorithm. The implemented algorithm is described by algorithm 20.

C.4.1 Simulated Annealing acceptance criteria

Let T be a temperature, which is initially large but decreases over time. In step 9, rather than only accepting better solutions, the simulated annealing algorithm will sometimes accept moves to worse solutions, allowing it to escape local optima.

If the cost of the new solution is less than the cost of the current solution then the new solution will always be accepted.

If the cost of the new solution is more than the cost of the current solution then there is a probability that the algorithm will still accept the new solution. The acceptance criteria are such that, with a high temperature the algorithm will accept many worse solutions, but as the temperature is decreased, the simulated annealing algorithm slowly converges on the first descent algorithm over time.

This probability of acceptance is expressed in step 9 of algorithm 12 and is also related to how much worse the new solution is than the current solution, so that the algorithm is more likely to accept small increases in cost than large increases.

Algorithm 20 Simulated Annealing algorithm

1: initialise the temperature, T to 5000.
2: let the current solution $S_n = S_0$
3: evaluate the current solution, S_n, and determine a cost, C_n
4: set $S_B = S_n$ and $C_B = C_n$
5: **while** termination condition not met **do**
6:    generate a new candidate solution, S_C, by random selection from the neighbourhood of S_n
7:    evaluate S_C and determine a cost C_C
8:    **if** S_C is feasible **then**
9:       **if** $(C_C \leq C_n)$ **or** $R < e^{-\delta/T}$, where $\delta = C_C - C_n$ and R represents a uniform random variable in the range [0..1] **then**
10:          adopt the candidate solution as the current solution, $S_n = S_C$ and $C_n = C_C$
11:       **end if**
12:       **if** $C_C < C_B$ **then**
13:          let $C_B = C_C$ and $S_B = S_C$
14:       **end if**
15:    **end if**
16:    reduce the temperature T by 0.1%, $T = T * 0.999$.
17: **end while**
18: return S_B, the best solution found

The Temperature, T, decreases over time according to a cooling schedule. As the search time is so short, the standard approach of reducing the temperature very slowly is not practical here. Instead, the temperature was reduced every iteration. The initial temperature and cooling schedule parameters were obtained by applying the simulated annealing algorithm to static sequencing problems and evaluating the performance with a range of parameters. Cooling schedule parameters of 0.9, 0.95 and 0.99, 0.995, 0.999 and 0.9999 were all tested. A range of temperatures were also tested. There was very little difference in the performance of the algorithm with cooling parameters of 0.995 to 0.9999 although performance was slightly worse for faster cooling. Similarly the algorithm did not seem very sensitive to the initial temperature although much higher or lower values than those selected seemed to do slightly worse.

The lack of sensitivity to variations in parameters is not unexpected given that even the first-come-first-served algorithm performs so well on this problem. It seems that the algorithm merely needs a little ability to escape shallow local optima and that many different values of the search parameters give it this ability. This conclusion is supported by the fact that the steeper descent algorithm also performs well on the problem and this has only a very limited ability to escape local optima.

C.5 Seven Aircraft Exhaustive Search (SAES)

The seven aircraft exhaustive search (SAES) was performed by exhaustively re-sequencing the first seven aircraft in the take-off sequence for which the position has not yet been fixed. With a two-minute freezing time, this will be the first seven aircraft which are more than two minutes from their take-off time. The seven aircraft limit was determined by the fact that it can be performed in less than the second search time permitted whereas the eight aircraft search takes up to three or four seconds. A comparison of the heuristic algorithms against this algorithm shows whether it is worth considering more than seven aircraft when performing the sequencing.

It should be noted that the results in section 8.4 support the idea that it is worth considering more aircraft when possible. Furthermore, with the SAES, aircraft are only sequenced when they are close to the start of the take-off sequence and the plateau in the results for the planning horizon and observations from the sequence viewer indicate that the meta-heuristic approach often sequences the aircraft much earlier than this.

C.6 Results

Tables C.1 to C.6 present the results of the experiments that were performed using each of six datasets. In each table, the CTOT column shows the total number of CTOT slots that were missed and the delay columns the total delay (in seconds). As there is a stochastic element to the selection of solutions from the neighbourhood, each search was executed twenty times on each dataset. The tables present the mean value for the number of CTOT slots missed and the standard deviation is also given in brackets. For the delay, as well as giving the mean total delay and the standard deviation of the delay, the minimum and maximum delay is also given.

In each table, the top two rows specify the performance of the manual sequence. The 'Real Order, Actual Times' results are for the real take-off sequence, using the real take-off times. The 'Real Order, Predicted Times' are the results for the real take-off order, using the model to predict take-off times for the aircraft, with a two-minute traversal time. Following this, the results are given for the tabu search, simulated annealing, steeper descent and first descent algorithms. Finally, results are presented for the SAES search.

TABLE C.1: Comparative results for Dataset 1

Search	CTOTs missed	Total delay	Minimum delay	Maximum delay
Real, Predicted	13	110248		
Real, Actual	10	99805		
FCFS Order	94	413350		
Tabu Search	4 (0)	83913 (3)	83912	83920
Simulated Annealing	4 (0)	83734 (280)	83268	83920
Steeper Descent	4 (0)	84905 (261)	84621	85296
First Descent	4 (0)	84696 (717)	83912	85827
SAES	4	85079		

TABLE C.2: Comparative results for Dataset 2

Search	CTOTs missed	Total delay	Minimum delay	Maximum delay
Real, Predicted	9	121734		
Real, Actual	9	127891		
FCFS Order	68	671926		
Tabu Search	4 (0)	88019 (28)	87945	88041
Simulated Annealing	4 (0)	88173 (135)	88005	88485
Steeper Descent	4 (0)	88039 (50)	88005	88137
First Descent	4 (0)	88490 (189)	88077	88820
SAES	4	89239		

TABLE C.3: Comparative results for Dataset 3

Search	CTOTs missed	Total delay	Minimum delay	Maximum delay
Real, Predicted	11	121037		
Real, Actual	6	117894		
FCFS Order	94	421168		
Tabu Search	3 (0)	93508 (121)	93316	93805
Simulated Annealing	3 (0)	93429 (107)	93316	93676
Steeper Descent	3 (0)	93889 (441)	93448	95144
First Descent	3 (0)	93702 (391)	93346	94491
SAES	4	95248		

C.7 Evaluation Of The Results

Much evaluation of the decision support system was performed in chapter 8. This included an evaluation of the accuracy of the take-off time prediction system and the positional delay of aircraft. In this section the evaluation is restricted to a consideration of the relative performance of the search algorithms.

The first descent search performs well, as shown by the fact that it outperforms the seven aircraft exhaustive search (SAES) and that the best results obtained by it are close to those of the meta-heuristic searches. Therefore, the neighbourhood design is obviously successfully keeping down the number of local optima.

In one case, the first descent search even attained a better CTOT compliance than the SAES search, while simultaneously reducing the delay for aircraft. The only advantage that the first descent algorithm had over SAES was that it could consider more than just the first seven aircraft, so this added flexibility of knowledge was obviously improving the performance.

The tabu search and simulated annealing performed the best in these experiments, indicating that there were local optima in the search space. There was little difference in performance between the simulated annealing and tabu search algorithms. The tabu search algorithm was chosen for its stability and ease of tuning rather than for any improved performance over the simulated annealing algorithm.

The tabu search algorithm was usually the most reliable in terms of spread of results. For all datasets except dataset 3, tabu search had the lowest standard deviation and the smallest difference in delay between the worst and best schedules. In dataset 3 simulated annealing had the lowest standard deviation and tabu search comes second.

Although there was very little difference between the minimum delays found for each of the algorithmic searches, there was much more of a spread in the maximum delays. In many cases, the delay for the worst schedule that the simulated annealing or tabu search algorithms found was better than even the average for the steeper descent and first descent algorithms. In all cases, the worst schedule found by the simulated annealing and tabu search algorithms was better than the worst found by the first descent algorithms.

Table C.4: Comparative results for Dataset 4

Search	CTOTs missed	Total delay	Minimum delay	Maximum delay
Real, Predicted	13	124510		
Real, Actual	5	120893		
FCFS Order	55	602737		
Tabu Search	3 (0)	92719 (29)	92618	92750
Simulated Annealing	3 (0)	92789 (93)	92618	93050
Steeper Descent	3 (0)	92907 (207)	92618	93209
First Descent	3 (0)	93127 (327)	92723	93876
SAES	3	96609		

Table C.5: Comparative results for Dataset 5

Search	CTOTs missed	Total delay	Minimum delay	Maximum delay
Real, Predicted	5	112400		
Real, Actual	5	107786		
FCFS Order	53	647125		
Tabu Search	1 (0)	82958 (16)	82948	82981
Simulated Annealing	1 (0)	83355 (538)	82948	84298
Steeper Descent	1 (0)	83694 (615)	82948	84778
First Descent	1 (0)	84000 (434)	82948	84811
SAES	1	88097		

Table C.6: Comparative results for Dataset 6

Search	CTOTs missed	Total delay	Minimum delay	Maximum delay
Real, Predicted	8	131261		
Real, Actual	4	96235		
FCFS Order	33	572389		
Tabu Search	4 (0)	73581 (47)	73566	73740
Simulated Annealing	4 (0)	73689 (113)	73446	73860
Steeper Descent	4 (0)	73903 (172)	73686	74146
First Descent	4 (0)	73800 (167)	73566	74146
SAES	4	80160		

Appendix D

Results Showing The Effects Of The Planning Horizon

The results of experiments to investigate the effect of varying the planning horizon upon the performance of the system, when taxi times are known in advance are presented in the tables in this appendix. The experiments performed to generate these results were described in section 8.4.

Each row of tables D.1 to D.5 shows the performance of the system for one dataset. The columns show the mean CTOT compliance and delay for the system when considering the sequencing of the aircraft for take-off from each of the different holding area models. The configuration that was actually used in practice is displayed in bold, as these are the only problems which are really comparable to those the controllers actually had to solve.

The results in table D.1 show the performance of the system when it has no knowledge of the taxiing aircraft. Summary results for zero, one, seven and fifteen minute knowledge are shown in tables D.1 to D.5.

Full results are shown in tables D.6 to D.15 for the holding areas which were actually used in the real case. The mean, minimum and maximum CTOT compliance and delay over one hundred executions of the simulation are shown for each of the evaluated planning horizons. The rows of the tables specify the planning horizon the system was working with at the time and the columns specify the results obtained.

Table D.1: CTOT compliance and delay with no knowledge of taxiing aircraft

	27R		27L		09R	
Dataset	CTOT	Delay	CTOT	Delay	CTOT	Delay
1	**9.00**	**108461**	7.00	92255	14.00	121328
2	7.12	120473	**3.22**	**101378**	6.06	114091
3	7.00	103441	6.00	94499	**8.00**	**95233**
4	5.00	107252	**4.00**	**101437**	5.00	113713
5	**5.00**	**99937**	4.00	96260	4.00	106931
6	**2.00**	**103535**	2.00	92617	2.00	104514
7	5.00	85230	**4.00**	**81637**	4.00	91383
8	14.00	95047	10.00	86035	**17.00**	**104517**
9	**6.00**	**103828**	3.00	101138	9.00	110368
10	17.78	188675	**5.00**	**97768**	16.26	150665

Table D.2: CTOT compliance and delay with knowledge of taxiing aircraft one minute before holding area arrival

	27R		27L		09R	
Dataset	CTOT	Delay	CTOT	Delay	CTOT	Delay
1	**7.00**	**101789**	4.00	88389	5.00	104997
2	6.12	113056	**3.00**	**94370**	6.00	110251
3	6.00	98102	5.00	91609	**8.00**	**95524**
4	4.00	101172	**3.00**	**97685**	3.00	103646
5	**4.00**	**95751**	2.00	92576	3.00	100638
6	**2.00**	**92425**	1.00	90355	1.00	95650
7	4.00	76985	**4.00**	**75892**	4.00	83077
8	11.00	90089	11.00	86073	**10.00**	**85146**
9	**4.00**	**98184**	3.00	97030	6.00	105422
10	14.70	165541	**4.00**	**99130**	5.09	122339

Table D.3: CTOT compliance and delay with knowledge of taxiing aircraft five minutes before holding area arrival

	27R		27L		09R	
Dataset	CTOT	Delay	CTOT	Delay	CTOT	Delay
1	**5.00**	**86833**	4.00	85535	3.00	90612
2	4.26	103567	**3.00**	**89976**	4.98	102830
3	5.00	88189	5.00	88303	**5.08**	**87538**
4	4.00	95033	**3.00**	**92706**	3.00	98094
5	**1.00**	**86192**	1.00	84682	1.00	94712
6	**1.00**	**85214**	1.00	85214	1.00	96864
7	4.00	74609	**4.00**	**74382**	4.00	84199
8	9.00	78579	9.00	78453	**9.00**	**84223**
9	**4.00**	**96494**	3.00	94075	4.00	99979
10	4.00	100662	**4.00**	**94948**	4.00	102896

Table D.4: CTOT compliance and delay with knowledge of taxiing aircraft seven minutes before holding area arrival

	27R		27L		09R	
Dataset	CTOT	Delay	CTOT	Delay	CTOT	Delay
1	**4.00**	**87095**	4.00	85262	3.00	87704
2	4.00	101269	**3.00**	**88818**	3.03	99251
3	5.00	88037	5.00	87799	**5.18**	**87965**
4	4.00	94623	**3.00**	**91674**	3.00	96992
5	**1.00**	**84631**	1.00	84408	1.00	92693
6	**1.00**	**85116**	1.00	85116	1.00	93428
7	4.00	73839	**4.00**	**73920**	4.00	81074
8	9.00	78393	9.00	78453	**9.00**	**83887**
9	**4.00**	**96202**	3.00	93997	4.00	98955
10	4.00	99606	**4.00**	**95138**	4.00	101428

Table D.5: CTOT compliance and delay with knowledge of taxiing aircraft fifteen minutes before holding area arrival

	27R		27L		09R	
Dataset	CTOT	Delay	CTOT	Delay	CTOT	Delay
1	**4.00**	**86905**	4.00	84688	3.00	87339
2	4.00	96553	**3.00**	**88631**	3.00	97029
3	5.00	87074	5.00	86709	**5.07**	**86802**
4	4.00	91913	**3.00**	**91115**	3.00	95069
5	**1.00**	**84066**	1.00	83883	1.00	91873
6	**1.00**	**84667**	1.00	84667	1.00	93207
7	4.00	72784	**4.00**	**71995**	4.00	79924
8	9.00	78297	9.00	78303	**9.00**	**83602**
9	**4.00**	**95110**	3.00	93343	4.00	97911
10	4.00	99054	**4.00**	**93932**	4.00	98713

Table D.6: CTOT compliance and delay for dataset 1, 27R, with varying taxi knowledge

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual Sequence	22	-	-	136252	-	-
Real Times	10	-	-	99805	-	-
FCFS Sequence	99	-	-	438130	-	-
No notice	9.00	9	9	108461	108461	108461
1 minute taxi notice	7.00	7	7	101789	101789	101789
2 minute taxi notice	6.00	6	6	90858	90858	90858
3 minute taxi notice	6.00	6	6	87942	87942	87942
4 minute taxi notice	5.00	5	5	86256	86256	86256
5 minute taxi notice	5.00	5	5	86833	86745	86842
6 minute taxi notice	4.00	4	4	87599	87442	87682
7 minute taxi notice	4.00	4	4	87095	87024	87204
8 minute taxi notice	4.00	4	4	87095	87024	87204
9 minute taxi notice	4.00	4	4	87095	87024	87204
10 minute taxi notice	4.00	4	4	87030	87024	87714
11 minute taxi notice	4.00	4	4	86879	86150	87079
12 minute taxi notice	4.00	4	4	86867	86150	87104
13 minute taxi notice	4.00	4	4	86884	86150	87104
14 minute taxi notice	4.00	4	4	86888	86150	87104
15 minute taxi notice	4.00	4	4	86905	86150	87104
16 minute taxi notice	4.00	4	4	86907	86150	87104
17 minute taxi notice	4.00	4	4	86906	86150	87079
18 minute taxi notice	4.00	4	4	86903	86150	87079
19 minute taxi notice	4.00	4	4	86903	86150	87079

Table D.7: CTOT compliance and delay for dataset 2, 27L, with varying taxi knowledge

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual Sequence	19	-	-	180378	-	-
Real Times	9	-	-	127891	-	-
FCFS Sequence	68	-	-	753106	-	-
No notice	3.22	3	4	101378	101133	102251
1 minute taxi notice	3.00	3	3	94370	94370	94370
2 minute taxi notice	3.00	3	3	96684	96673	97789
3 minute taxi notice	3.00	3	3	95435	95276	97196
4 minute taxi notice	3.00	3	3	90877	90267	91836
5 minute taxi notice	3.00	3	3	89976	89842	91351
6 minute taxi notice	3.00	3	3	89826	89782	91011
7 minute taxi notice	3.00	3	3	88818	88438	90317
8 minute taxi notice	3.00	3	3	88595	88438	90030
9 minute taxi notice	3.00	3	3	88595	88438	89435
10 minute taxi notice	3.00	3	3	88617	88444	89441
11 minute taxi notice	3.00	3	3	88628	88444	89441
12 minute taxi notice	3.00	3	3	88625	88444	89441
13 minute taxi notice	3.00	3	3	88629	88444	89469
14 minute taxi notice	3.00	3	3	88631	88444	89441
15 minute taxi notice	3.00	3	3	88631	88444	89441
16 minute taxi notice	3.00	3	3	88630	88444	89441
17 minute taxi notice	3.00	3	3	88724	88444	90526
18 minute taxi notice	3.00	3	3	88718	88444	90556
19 minute taxi notice	3.00	3	3	88649	88444	90556

Table D.8: CTOT compliance and delay for dataset 3, 09R, with varying taxi knowledge

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual Sequence	17	-	-	150052	-	-
Real Times	13	-	-	128763	-	-
FCFS Sequence	61	-	-	814325	-	-
No notice	8.00	8	8	95233	95233	95233
1 minute taxi notice	8.00	8	8	95524	95524	95524
2 minute taxi notice	6.00	6	6	93773	93773	93773
3 minute taxi notice	5.00	5	5	87565	87555	88608
4 minute taxi notice	5.00	5	5	87062	87062	87133
5 minute taxi notice	5.08	5	6	87538	86822	88251
6 minute taxi notice	5.26	5	6	87800	86822	88519
7 minute taxi notice	5.18	5	6	87965	86762	88609
8 minute taxi notice	5.06	5	6	87107	86702	88139
9 minute taxi notice	5.05	5	6	86542	86160	87713
10 minute taxi notice	5.01	5	6	86498	86160	87713
11 minute taxi notice	5.01	5	6	86652	86160	90056
12 minute taxi notice	5.03	5	6	86645	86210	90046
13 minute taxi notice	5.04	5	6	86701	86171	91144
14 minute taxi notice	5.03	5	6	86665	86171	89177
15 minute taxi notice	5.07	5	6	86802	86171	90007
16 minute taxi notice	5.04	5	6	86852	86171	90046
17 minute taxi notice	5.03	5	6	86843	86171	90046
18 minute taxi notice	5.06	5	6	86880	86171	90046
19 minute taxi notice	5.04	5	6	86878	86171	90046

Table D.9: CTOT compliance and delay for dataset 4, 27L, with varying taxi knowledge

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual Sequence	12	-	-	151869	-	-
Real Times	5	-	-	120893	-	-
FCFS Sequence	51	-	-	581071	-	-
No notice	4.00	4	4	101437	101437	101437
1 minute taxi notice	3.00	3	3	97685	97685	97685
2 minute taxi notice	3.00	3	3	96443	96443	96443
3 minute taxi notice	3.00	3	3	95475	95475	95475
4 minute taxi notice	3.00	3	3	94975	94975	94975
5 minute taxi notice	3.00	3	3	92706	92706	92706
6 minute taxi notice	3.00	3	3	92298	92298	92298
7 minute taxi notice	3.00	3	3	91674	91665	91688
8 minute taxi notice	3.00	3	3	91673	91665	91688
9 minute taxi notice	3.00	3	3	91652	91340	91688
10 minute taxi notice	3.00	3	3	91663	91340	91688
11 minute taxi notice	3.00	3	3	91505	91187	91535
12 minute taxi notice	3.00	3	3	91398	91040	91535
13 minute taxi notice	3.00	3	3	91155	91040	91411
14 minute taxi notice	3.00	3	3	91110	90530	91411
15 minute taxi notice	3.00	3	3	91115	91040	91411
16 minute taxi notice	3.00	3	3	91115	91040	91411
17 minute taxi notice	3.00	3	3	91110	90613	91411
18 minute taxi notice	3.00	3	3	91110	90613	91411
19 minute taxi notice	3.00	3	3	91115	91040	91411

Table D.10: CTOT compliance and delay for dataset 5, 27R, with varying taxi knowledge

Experiment	CTOTs missed Mean	CTOTs missed Min	CTOTs missed Max	Total delay (s) Mean	Total delay (s) Min	Total delay (s) Max
Manual Sequence	16	-	-	162834	-	-
Real Times	14	-	-	140075	-	-
FCFS Sequence	56	-	-	547461	-	-
No notice	5.00	5	5	99937	99937	99937
1 minute taxi notice	4.00	4	4	95751	95751	95751
2 minute taxi notice	2.00	2	2	94518	94518	94518
3 minute taxi notice	2.00	2	2	93699	93699	93699
4 minute taxi notice	1.00	1	1	90034	89917	90505
5 minute taxi notice	1.00	1	1	86192	86188	86326
6 minute taxi notice	1.00	1	1	85578	85578	85578
7 minute taxi notice	1.00	1	1	84631	84631	84631
8 minute taxi notice	1.00	1	1	84631	84631	84631
9 minute taxi notice	1.00	1	1	84542	84541	84671
10 minute taxi notice	1.00	1	1	84543	84541	84681
11 minute taxi notice	1.00	1	1	84180	83928	84634
12 minute taxi notice	1.00	1	1	84057	83928	84504
13 minute taxi notice	1.00	1	1	84047	83928	84648
14 minute taxi notice	1.00	1	1	84066	83928	84648
15 minute taxi notice	1.00	1	1	84066	83928	84648
16 minute taxi notice	1.00	1	1	84101	83711	84648
17 minute taxi notice	1.00	1	1	84144	83567	84648
18 minute taxi notice	1.00	1	1	84150	83567	84648
19 minute taxi notice	1.00	1	1	84151	83567	84648

Table D.11: CTOT compliance and delay for dataset 6, 27R, with varying taxi knowledge

Experiment	CTOTs missed Mean	CTOTs missed Min	CTOTs missed Max	Total delay (s) Mean	Total delay (s) Min	Total delay (s) Max
Manual Sequence	7	-	-	152713	-	-
Real Times	5	-	-	107786	-	-
FCFS Sequence	55	-	-	718855	-	-
No notice	2.00	2	2	103535	103535	103535
1 minute taxi notice	2.00	2	2	92425	92425	92425
2 minute taxi notice	1.00	1	1	90638	90638	90638
3 minute taxi notice	1.00	1	1	86460	86460	86460
4 minute taxi notice	1.00	1	1	84990	84990	84990
5 minute taxi notice	1.00	1	1	85214	85214	85214
6 minute taxi notice	1.00	1	1	85160	85160	85160
7 minute taxi notice	1.00	1	1	85116	85116	85116
8 minute taxi notice	1.00	1	1	84857	84769	84914
9 minute taxi notice	1.00	1	1	84779	84769	84914
10 minute taxi notice	1.00	1	1	84676	84676	84676
11 minute taxi notice	1.00	1	1	84663	84654	84821
12 minute taxi notice	1.00	1	1	84659	84654	84676
13 minute taxi notice	1.00	1	1	84667	84662	84684
14 minute taxi notice	1.00	1	1	84667	84662	84684
15 minute taxi notice	1.00	1	1	84667	84662	84684
16 minute taxi notice	1.00	1	1	84669	84662	84684
17 minute taxi notice	1.00	1	1	84669	84662	84684
18 minute taxi notice	1.00	1	1	84669	84662	84684
19 minute taxi notice	1.00	1	1	84669	84662	84684

Table D.12: CTOT compliance and delay for dataset 7, 27L, with varying taxi knowledge

Experiment	CTOTs missed Mean	CTOTs missed Min	CTOTs missed Max	Total delay (s) Mean	Total delay (s) Min	Total delay (s) Max
Manual Sequence	17	-	-	214445	-	-
Real Times	4	-	-	96235	-	-
FCFS Sequence	33	-	-	667507	-	-
No notice	4.00	4	4	81637	81637	81637
1 minute taxi notice	4.00	4	4	75892	75892	75892
2 minute taxi notice	4.00	4	4	74735	74685	74738
3 minute taxi notice	4.00	4	4	74712	74614	74734
4 minute taxi notice	4.00	4	4	74286	74286	74286
5 minute taxi notice	4.00	4	4	74382	74382	74382
6 minute taxi notice	4.00	4	4	74381	74373	74382
7 minute taxi notice	4.00	4	4	73920	73809	74491
8 minute taxi notice	4.00	4	4	73488	73406	74004
9 minute taxi notice	4.00	4	4	72924	72343	73952
10 minute taxi notice	4.00	4	4	71935	71562	73123
11 minute taxi notice	4.00	4	4	71905	71604	72664
12 minute taxi notice	4.00	4	4	72016	71604	73165
13 minute taxi notice	4.00	4	4	71990	71604	73165
14 minute taxi notice	4.00	4	4	72021	71604	73165
15 minute taxi notice	4.00	4	4	71995	71604	73165
16 minute taxi notice	4.00	4	4	72005	71604	73165
17 minute taxi notice	4.00	4	4	72021	71604	73165
18 minute taxi notice	4.00	4	4	72005	71604	73165
19 minute taxi notice	4.00	4	4	72000	71604	73165

Table D.13: CTOT compliance and delay for dataset 8, 09R, with varying taxi knowledge

Experiment	CTOTs missed Mean	CTOTs missed Min	CTOTs missed Max	Total delay (s) Mean	Total delay (s) Min	Total delay (s) Max
Manual Sequence	20	-	-	129234	-	-
Real Times	10	-	-	99507	-	-
FCFS Sequence	57	-	-	267753	-	-
No notice	17.00	17	17	104517	104517	104517
1 minute taxi notice	10.00	10	10	85146	85146	85146
2 minute taxi notice	10.00	10	10	85554	85554	85554
3 minute taxi notice	9.00	9	9	85140	85140	85140
4 minute taxi notice	9.00	9	9	84801	84801	84801
5 minute taxi notice	9.00	9	9	84223	84223	84223
6 minute taxi notice	9.00	9	9	84175	84175	84175
7 minute taxi notice	9.00	9	9	83887	83887	83887
8 minute taxi notice	9.00	9	9	83893	83887	84043
9 minute taxi notice	9.00	9	9	83795	83779	83935
10 minute taxi notice	9.00	9	9	83843	83779	83995
11 minute taxi notice	9.00	9	9	83927	83887	84043
12 minute taxi notice	9.00	9	9	83690	82712	84055
13 minute taxi notice	9.00	9	9	83802	82712	84055
14 minute taxi notice	9.00	9	9	83654	82712	84067
15 minute taxi notice	9.00	9	9	83602	83356	84055
16 minute taxi notice	9.00	9	9	83537	82712	83947
17 minute taxi notice	9.00	9	9	83540	82712	83965
18 minute taxi notice	9.00	9	9	83554	82712	83965
19 minute taxi notice	9.00	9	9	83554	82712	83965

Table D.14: CTOT compliance and delay for dataset 9, 27R, with varying taxi knowledge

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual Sequence	17	-	-	140460	-	-
Real Times	6	-	-	117894	-	-
FCFS Sequence	100	-	-	462251	-	-
No notice	6.00	6	6	103828	103828	103828
1 minute taxi notice	4.00	4	4	98184	98184	98184
2 minute taxi notice	4.00	4	4	99141	99141	99141
3 minute taxi notice	4.00	4	4	97237	97237	97237
4 minute taxi notice	4.00	4	4	96839	96767	96929
5 minute taxi notice	4.00	4	4	96494	96494	96494
6 minute taxi notice	4.00	4	4	96151	96151	96151
7 minute taxi notice	4.00	4	4	96202	96202	96202
8 minute taxi notice	4.00	4	4	95187	94881	95679
9 minute taxi notice	4.00	4	4	95184	95130	95670
10 minute taxi notice	4.00	4	4	95299	95172	95706
11 minute taxi notice	4.00	4	4	95328	95172	95736
12 minute taxi notice	4.00	4	4	95164	94755	96900
13 minute taxi notice	4.00	4	4	95109	94595	95706
14 minute taxi notice	4.00	4	4	95112	94755	96900
15 minute taxi notice	4.00	4	4	95110	94755	95706
16 minute taxi notice	4.00	4	4	95036	94688	95706
17 minute taxi notice	4.00	4	4	95044	94688	96773
18 minute taxi notice	4.00	4	4	95039	94688	96773
19 minute taxi notice	4.00	4	4	95037	94688	96773

Table D.15: CTOT compliance and delay for dataset 10, 27L, with varying taxi knowledge

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual Sequence	20	-	-	167114	-	-
Real Times	11	-	-	120329	-	-
FCFS Sequence	46	-	-	551127	-	-
No notice	5.00	5	5	97768	97768	97768
1 minute taxi notice	4.00	4	4	99130	99130	99130
2 minute taxi notice	4.00	4	4	97511	97467	97527
3 minute taxi notice	4.00	4	4	97683	97584	97870
4 minute taxi notice	4.00	4	4	96406	96406	96406
5 minute taxi notice	4.00	4	4	94948	94848	95045
6 minute taxi notice	4.00	4	4	95287	95055	95374
7 minute taxi notice	4.00	4	4	95138	94962	95281
8 minute taxi notice	4.00	4	4	95034	94736	95281
9 minute taxi notice	4.00	4	4	94847	94736	95204
10 minute taxi notice	4.00	4	4	94297	94180	94398
11 minute taxi notice	4.00	4	4	94211	94062	94311
12 minute taxi notice	4.00	4	4	94187	93726	94311
13 minute taxi notice	4.00	4	4	93978	93726	94276
14 minute taxi notice	4.00	4	4	93929	93726	94276
15 minute taxi notice	4.00	4	4	93932	93726	94276
16 minute taxi notice	4.00	4	4	93932	93726	94276
17 minute taxi notice	4.00	4	4	93932	93726	94276
18 minute taxi notice	4.00	4	4	93933	93726	94276
19 minute taxi notice	4.00	4	4	93933	93726	94276

Appendix E

Results Showing The Effects Of Uncertainty

The tables in this appendix present the results of experiments to investigate the effects of uncertainty in taxi times upon the performance of the system. The experiments performed to generate these results were described in section 8.6.

The first column in each table indicates the experiment to which the results in that row refer, the next three show the mean, minimum and maximum number of CTOT slots missed and the final three the mean, minimum and maximum delay. The first row shows the results for the manual sequence with predicted take-off times, the next the real results the controller attained and the third the results for the first-come-first-served sequence with predicted take-off times. The fourth row shows the performance of the system with certain taxi times and the remaining rows show the performance as the taxi time uncertainty is increased.

Table E.1: CTOT Compliance and delay for dataset 1, 27R

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual Sequence	22	-	-	136252	-	-
Real Times	10	-	-	99805	-	-
FCFS Sequence	99	-	-	438130	-	-
Certain taxi times	4.00	4	4	86903	86150	87079
+/- 10% error	4.40	4	6	86449	84053	88349
+/- 20% error	4.86	4	6	86421	84426	89451
+/- 30% error	5.13	4	6	86829	84993	90475
+/- 40% error	5.20	4	7	87535	85023	91935
+/- 50% error	5.25	3	8	88726	84898	97292
+/- 60% error	5.33	3	7	90266	85330	98525
+/- 70% error	5.51	3	8	92029	86628	102383
+/- 80% error	5.98	4	10	97335	89996	109501
+/- 90% error	7.07	3	14	106423	93910	131590

Table E.2: CTOT Compliance and delay for dataset 2, 27L

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual Sequence	19	-	-	180378	-	-
Real Times	9	-	-	127891	-	-
FCFS Sequence	68	-	-	753106	-	-
Certain taxi times	3.00	3	3	88644	88444	89441
+/- 10% error	3.00	3	3	88840	88312	89823
+/- 20% error	3.00	3	3	89367	88038	92450
+/- 30% error	3.00	3	3	90087	88546	92245
+/- 40% error	3.04	3	4	90871	88648	95611
+/- 50% error	3.16	3	5	91446	88780	95445
+/- 60% error	3.21	3	5	92569	89338	96807
+/- 70% error	3.33	3	5	94457	89411	101842
+/- 80% error	3.54	3	5	98549	93439	109353
+/- 90% error	3.66	3	6	107935	97135	128175

Table E.3: CTOT Compliance and delay for dataset 3, 09R

Experiment	CTOTs missed Mean	Min	Max	Total delay (s) Mean	Min	Max
Manual Sequence	17	-	-	150052	-	-
Real Times	13	-	-	128763	-	-
FCFS Sequence	61	-	-	814325	-	-
Certain taxi times	5.05	5	6	86810	86171	88297
+/- 10% error	5.65	5	7	88164	86195	93354
+/- 20% error	5.89	5	7	89094	85935	95690
+/- 30% error	6.28	5	8	89945	85526	96963
+/- 40% error	6.47	5	9	90015	85415	98689
+/- 50% error	6.84	5	10	91149	87074	99226
+/- 60% error	6.98	5	9	92431	87433	99958
+/- 70% error	7.56	5	11	94610	88153	106035
+/- 80% error	7.93	5	11	98661	90614	113570
+/- 90% error	8.80	6	14	108621	92419	130732

Table E.4: CTOT Compliance and delay for dataset 4, 27L

Experiment	CTOTs missed Mean	Min	Max	Total delay (s) Mean	Min	Max
Manual Sequence	12	-	-	151869	-	-
Real Times	5	-	-	120893	-	-
FCFS Sequence	51	-	-	581071	-	-
Certain taxi times	3.00	3	3	91115	91040	91411
+/- 10% error	3.00	3	3	91368	90478	94011
+/- 20% error	3.00	3	3	92037	90434	95512
+/- 30% error	3.00	3	3	92856	90678	96006
+/- 40% error	3.00	3	3	93644	91373	96811
+/- 50% error	3.06	3	4	94538	91744	99384
+/- 60% error	3.08	3	4	96216	92783	100674
+/- 70% error	3.13	3	4	98537	94068	104879
+/- 80% error	3.34	3	6	102925	96954	114840
+/- 90% error	4.05	3	6	115548	100271	139249

Table E.5: CTOT Compliance and delay for dataset 5, 27R

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual Sequence	16	-	-	162834	-	-
Real Times	14	-	-	140075	-	-
FCFS Sequence	56	-	-	547461	-	-
Certain taxi times	1.00	1	1	84173	83567	84648
+/- 10% error	1.01	1	2	85183	83999	87674
+/- 20% error	1.03	1	2	86272	84106	89017
+/- 30% error	1.01	1	2	87340	84628	90547
+/- 40% error	1.01	1	2	88141	85149	92431
+/- 50% error	1.06	1	2	89465	85684	94048
+/- 60% error	1.13	1	3	90974	87102	98806
+/- 70% error	1.33	1	4	93334	88994	100104
+/- 80% error	1.47	1	4	97605	89992	110528
+/- 90% error	2.46	1	5	109148	91263	132977

Table E.6: CTOT Compliance and delay for dataset 6, 27R

	CTOTs missed			Total delay (s)		
Experiment	Mean	Min	Max	Mean	Min	Max
Manual Sequence	7	-	-	152713	-	-
Real Times	5	-	-	107786	-	-
FCFS Sequence	55	-	-	718855	-	-
Certain taxi times	1.00	1	1	84669	84662	84684
+/- 10% error	1.00	1	1	84751	83652	86276
+/- 20% error	1.00	1	1	84848	83560	86829
+/- 30% error	1.02	1	2	85361	83596	88061
+/- 40% error	1.02	1	2	86159	83924	88354
+/- 50% error	1.08	1	3	87144	84885	91482
+/- 60% error	1.10	1	2	88796	85239	93494
+/- 70% error	1.15	1	2	91277	87343	97436
+/- 80% error	1.33	1	3	96048	89146	104005
+/- 90% error	1.77	1	6	106530	91321	130336

Table E.7: CTOT Compliance and delay for dataset 7, 27L

Experiment	CTOTs missed Mean	CTOTs missed Min	CTOTs missed Max	Total delay (s) Mean	Total delay (s) Min	Total delay (s) Max
Manual Sequence	17	-	-	214445	-	-
Real Times	4	-	-	96235	-	-
FCFS Sequence	33	-	-	667507	-	-
Certain taxi times	4.00	4	4	71958	71604	73165
+/- 10% error	4.00	4	4	73035	71558	74813
+/- 20% error	4.00	4	4	73496	71907	76299
+/- 30% error	4.00	4	4	74091	72518	76319
+/- 40% error	4.00	4	4	74671	72154	77889
+/- 50% error	4.00	4	4	75603	73139	78744
+/- 60% error	4.00	4	4	76555	73389	82284
+/- 70% error	4.00	4	4	78505	74166	83843
+/- 80% error	4.03	4	5	82683	74998	93520
+/- 90% error	4.28	4	6	94259	80318	120104

Table E.8: CTOT Compliance and delay for dataset 8, 09R

Experiment	CTOTs missed Mean	CTOTs missed Min	CTOTs missed Max	Total delay (s) Mean	Total delay (s) Min	Total delay (s) Max
Manual Sequence	20	-	-	129234	-	-
Real Times	10	-	-	99507	-	-
FCFS Sequence	57	-	-	267753	-	-
Certain taxi times	9.00	9	9	83570	82712	85160
+/- 10% error	9.01	9	10	84019	82237	101395
+/- 20% error	9.25	9	11	84428	82404	87684
+/- 30% error	9.43	9	11	85219	82419	88559
+/- 40% error	9.91	9	13	86200	82357	92895
+/- 50% error	10.46	9	13	87320	83013	93160
+/- 60% error	10.85	9	14	88404	83933	94397
+/- 70% error	11.29	9	15	91411	84299	102477
+/- 80% error	12.04	9	17	96654	87712	113075
+/- 90% error	13.40	9	18	105112	90409	123729

Table E.9: CTOT Compliance and delay for dataset 9, 27R

Experiment	CTOTs missed Mean	Min	Max	Total delay (s) Mean	Min	Max
Manual Sequence	17	-	-	140460	-	-
Real Times	6	-	-	117894	-	-
FCFS Sequence	100	-	-	462251	-	-
Certain taxi times	4.00	4	4	95037	94688	96773
+/- 10% error	4.00	4	4	95461	94155	98376
+/- 20% error	4.04	4	5	95699	93298	99324
+/- 30% error	4.36	4	6	96381	93092	101574
+/- 40% error	4.53	4	7	96859	93551	100348
+/- 50% error	4.76	4	7	97922	93860	104260
+/- 60% error	4.85	4	8	98882	94726	105166
+/- 70% error	5.05	4	9	100661	94792	106642
+/- 80% error	5.45	4	9	105122	96755	120873
+/- 90% error	6.38	4	12	113603	102532	128876

Table E.10: CTOT Compliance and delay for dataset 10, 27L

Experiment	CTOTs missed Mean	Min	Max	Total delay (s) Mean	Min	Max
Manual Sequence	20	-	-	167114	-	-
Real Times	11	-	-	120329	-	-
FCFS Sequence	46	-	-	551127	-	-
Certain taxi times	4.00	4	4	93933	93726	94276
+/- 10% error	4.00	4	4	94155	93509	94910
+/- 20% error	4.00	4	4	94451	93427	95620
+/- 30% error	4.00	4	4	94962	93523	98140
+/- 40% error	4.01	4	5	95802	94015	99124
+/- 50% error	4.04	4	5	96741	94662	101933
+/- 60% error	4.09	4	5	98265	95606	104209
+/- 70% error	4.24	4	6	101042	95796	114023
+/- 80% error	4.43	4	7	105832	98354	118079
+/- 90% error	5.00	4	10	116840	103228	133628

Appendix F

Results Using Smaller Datasets

The tables in this appendix present the results of experiments to investigate the short-term performance of the system. The experiments performed to generate these results were described in section 8.7. The relevant part of the explanation given there is repeated below for clarity. There are two tables for each pair of datasets.

F.1 The CTOT Compliance and Delay Tables

These tables show the performance of the system and the characteristics of the aircraft in the datasets evaluated. The first column (labelled 'DS') shows the dataset tested and the holding area configuration that was in use. The second column (labelled 'Index') shows the starting index of the sub-dataset under test. Sixty aircraft were tested, starting from the displayed index in the dataset (providing sufficient aircraft existed in the dataset). The position in the take-off sequence and the take-off times were fixed for the first ten aircraft in each dataset.

The columns labelled 'CTOT' and 'Delay' show the number of CTOT take-off slots which were missed and the average (rather than total) delay per aircraft in minutes and seconds. As the first ten aircraft were not actually sequenced, they were ignored for the measurement of delay and CTOT compliance, so the delay is the average over fifty aircraft rather than sixty. The CTOT compliance and average delay is shown for the real take-off times, then for the predicted take-off times for the real take-off sequence and finally for the automatically produced sequences.

Following the performance details, the relative performance is shown by comparing the percentage improvement of delay between the real and predicted take-off times for the real take-off sequence (RvP), the automated sequence vs predicted times (AvP) and automated sequence vs real times (AvR). The table ends with a summary of the details of each dataset tested. The number of aircraft with CTOTs (C.), and the number of light (Lt.), medium (Med.) and heavy (Hvy.) aircraft in the dataset are shown in the table.

F.2 The Schedule Duration Tables

It is sometimes useful to see the duration of the produced take-off sequences. However, the fact that some aircraft have significant delays to await CTOT slots means that the the time at which the last aircraft takes off can be highly dependent upon the start of the CTOT slot assigned to it. Consequently, it can be even more useful to see the time at which intermediate take-offs occur, thus eliminating the dependency upon the last take-off. The durations (in minutes and seconds) of quarters of the take-off sequence are presented in the tables of schedule durations. The first quarter schedule runs from the start of the take-off sequence to the time of take-off of the 15th aircraft. The second quarter runs from the take-off of the 15th to the 30th aircraft, the third from the 30th to 45th aircraft and the fourth from the 45th to the 60th aircraft. Adding all four times will give the total schedule duration.

It is actually the total of the inter-aircraft separations that is being measured when measuring the schedule duration. As the times start at the first take-off time, the first quarter can be considered to consist of the first fourteen inter-aircraft separations (from aircraft 1 to 15) whereas every other quarter duration relates to fifteen separations (from 15 to 30, or 30 to 45 for example). Moreover, as the take-off times for the first ten aircraft are fixed, the duration of the first quarter is the sum of the first nine real separations plus the next five predicted separations in the sequence so is mainly indicative of the real schedule rather than automatically produced sequences.

TABLE F.1: CTOT compliance and delay results for partial datasets, datasets 1 and 2

		Real times		Predicted times		Automated sequence		Delay %			Dataset details			
DS.	Index	CTOT	Delay	CTOT	Delay	CTOT	Delay	RvP	AvP	AvR	C.	Lt.	Med.	Hvy.
1 27R	0	1	4:04	2	4:08	1.00	3:42	2	10	9	28	0	43	17
1 27R	25	2	4:31	3	4:30	1.00	4:01	0	11	11	29	0	39	21
1 27R	50	2	5:57	2	5:33	1.00	4:28	-7	20	25	30	0	36	24
1 27R	75	3	5:50	3	5:55	2.00	4:32	1	23	22	29	0	36	24
1 27R	100	4	5:03	6	6:57	2.00	5:09	27	26	-2	30	0	31	29
1 27R	125	2	7:22	3	9:14	1.00	5:44	20	38	22	27	0	35	25
1 27R	150	2	7:39	2	8:06	1.00	5:41	6	30	26	23	0	40	20
1 27R	175	1	6:14	1	6:36	1.00	4:46	5	28	24	18	0	35	19
1 27R	200	1	3:39	2	5:03	0.00	3:52	27	23	-6	52	0	57	3
1 27R	225	1	4:22	3	5:10	0.00	4:13	16	19	3	45	0	53	7
1 27R	250	0	4:03	1	5:23	0.00	4:10	25	22	-3	30	0	48	12
1 27R	275	1	3:47	1	4:06	0.00	3:30	8	15	8	26	0	46	14
2 27L	0	5	9:02	6	10:23	1.00	8:07	13	22	10	28	0	46	14
2 27L	25	2	5:05	3	7:30	2.00	5:37	32	25	-10	24	0	47	13
2 27L	50	1	5:59	2	7:06	1.00	4:44	16	33	21	19	0	49	11
2 27L	75	0	8:28	0	9:18	0.00	4:13	9	55	50	9	0	46	14
2 27L	100	0	8:59	0	7:22	0.00	3:58	-22	46	56	1	0	44	16
2 27L	125	0	7:41	0	8:54	0.00	4:28	14	50	42	0	0	39	21
2 27L	150	0	4:45	0	4:41	0.00	3:27	-1	26	27	0	0	25	30
2 27L	175	0	2:28	0	2:50	0.00	2:25	13	15	3	0	0	8	22
2 27L	200	2	6:03	4	7:32	0.00	5:45	20	24	5	29	0	37	23
2 27L	225	2	6:34	4	7:24	0.00	5:57	11	20	9	31	0	40	20
2 27L	250	0	7:09	0	7:40	0.00	5:50	7	24	18	27	0	43	17
2 27L	275	4	10:38	6	12:01	0.00	6:39	11	45	38	25	0	40	20

TABLE F.2: Duration results for partial datasets, datasets 1 and 2

		Real sequence duration				Manual sequence duration				Automated sequence duration			
DS.	Index	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
1 27R	0	18:24	29:38	19:23	20:31	18:28	29:52	19:00	22:00	18:28	29:52	18:00	21:25
1 27R	25	24:26	20:35	26:00	20:58	24:21	21:00	25:18	21:25	24:21	19:26	26:52	21:20
1 27R	50	25:36	22:47	22:52	19:45	24:36	24:00	22:00	20:31	25:36	22:38	19:20	21:52
1 27R	75	22:58	20:56	24:22	21:28	22:41	21:13	24:52	23:27	20:44	22:36	25:26	21:31
1 27R	100	21:18	24:14	20:34	19:03	21:18	25:55	22:00	21:00	21:10	26:07	20:00	18:00
1 27R	125	19:31	20:31	20:06	17:10	20:24	20:00	23:00	17:00	19:24	19:12	18:23	18:09
1 27R	150	21:07	19:10	20:40	18:31	20:47	20:00	21:00	18:00	19:13	18:00	22:28	18:41
1 27R	175	16:46	21:02	17:00	17:32	16:46	21:10	17:40	18:00	16:46	19:32	17:36	16:20
1 27R	200	33:00	18:00	26:01	18:59	33:00	21:00	24:00	18:00	34:00	19:00	24:14	18:46
1 27R	225	27:00	19:16	22:44	17:00	27:00	19:14	24:46	17:00	27:00	17:24	24:49	17:47
1 27R	250	20:27	20:38	23:11	20:28	22:16	20:00	23:00	20:27	20:51	19:38	23:50	19:57
1 27R	275	21:21	24:29	20:35	27:55	22:06	24:27	20:00	27:53	22:06	23:00	20:46	28:34
2 27L	0	20:03	20:15	20:08	15:09	20:26	21:00	21:00	17:00	19:41	18:00	22:58	16:17
2 27L	25	21:41	17:00	19:41	21:51	22:06	19:00	20:00	23:00	22:05	16:00	21:37	20:29
2 27L	50	16:01	24:48	27:29	16:38	17:17	23:32	28:18	18:00	17:17	23:32	22:49	18:00
2 27L	75	24:48	20:14	22:06	26:19	25:45	22:00	21:00	21:00	24:16	18:00	18:50	24:08
2 27L	100	22:48	25:58	19:04	20:17	21:40	23:00	21:26	23:16	19:40	21:42	22:44	19:00
2 27L	125	22:30	19:48	22:09	26:39	22:29	22:00	21:00	25:03	20:29	17:00	24:44	28:19
2 27L	150	18:45	24:24	31:51	24:37	18:45	24:16	31:59	24:17	18:45	19:45	36:30	23:08
2 27L	175	10:06	19:34	14:45	14:10	10:06	20:07	15:09	12:53	10:06	20:07	14:09	12:44
2 27L	200	22:24	20:49	21:25	20:32	22:20	21:47	24:00	20:00	22:09	21:50	17:36	37:21
2 27L	225	21:37	23:11	23:16	19:17	22:19	23:05	25:00	17:00	20:19	22:05	27:25	17:35
2 27L	250	24:16	20:09	23:02	18:45	23:56	21:01	21:52	21:00	23:56	21:00	19:07	19:00
2 27L	275	22:15	20:21	21:03	18:25	22:05	22:00	21:00	19:00	19:19	19:00	19:39	21:36

TABLE F.3: CTOT compliance and delay results for partial datasets, datasets 3 and 4

		Real times		Predicted times		Automated sequence		Delay %			Dataset details			
DS.	Index	CTOT	Delay	CTOT	Delay	CTOT	Delay	RvP	AvP	AvR	C.	Lt.	Med.	Hvy.
3 09R	0	5	7:05	5	7:18	2.00	5:33	3	24	22	22	0	41	19
3 09R	25	5	8:41	5	10:08	2.40	7:09	14	29	18	24	0	44	16
3 09R	50	1	6:46	1	7:39	1.00	6:32	11	15	4	25	0	46	14
3 09R	75	4	6:20	3	7:14	1.00	5:23	12	26	15	28	0	44	16
3 09R	100	6	7:59	5	7:35	2.00	4:41	-5	38	41	26	0	44	16
3 09R	125	4	10:24	4	9:47	3.00	4:31	-6	54	56	15	1	43	16
3 09R	150	1	9:46	2	10:23	0.00	7:28	6	28	24	9	1	46	13
3 09R	175	0	6:39	2	9:06	0.00	5:53	27	35	11	7	0	49	11
3 09R	200	0	4:51	0	5:50	0.00	4:50	17	17	0	5	0	49	11
3 09R	225	0	4:03	0	4:48	0.00	3:36	16	25	11	2	0	47	13
3 09R	250	0	3:17	0	3:56	0.00	3:04	17	22	6	0	0	35	25
3 09R	275	0	2:27	0	2:32	0.00	2:22	3	6	3	0	0	20	36
4 27L	0	2	5:04	2	5:15	1.00	4:49	3	8	5	36	0	57	3
4 27L	25	2	4:54	5	6:49	1.00	4:11	28	39	15	30	1	53	6
4 27L	50	0	5:43	3	7:46	0.00	4:51	26	38	15	24	1	47	12
4 27L	75	0	5:09	1	6:06	0.00	4:11	16	32	19	26	1	45	14
4 27L	100	1	5:08	1	4:58	0.00	4:06	-3	17	20	21	0	39	21
4 27L	125	1	7:30	1	6:55	0.00	4:45	-8	31	37	13	0	37	23
4 27L	150	0	8:27	0	8:36	0.00	5:44	2	33	32	6	0	37	23
4 27L	175	0	6:23	0	7:49	0.00	5:38	18	28	12	7	0	38	22
4 27L	200	1	6:10	1	7:27	1.00	4:55	17	34	20	10	0	37	23
4 27L	225	1	7:30	1	7:46	1.00	5:06	3	34	32	10	0	37	23
4 27L	250	1	5:56	1	6:26	1.00	4:59	8	23	16	10	0	34	26
4 27L	275	1	5:15	1	6:41	1.00	4:48	22	28	8	11	0	37	18

TABLE F.4: Duration results for partial datasets, datasets 3 and 4

DS.	Index	Real sequence duration 1st	2nd	3rd	4th	Manual sequence duration 1st	2nd	3rd	4th	Automated sequence duration 1st	2nd	3rd	4th
3 09R	0	22:25	27:01	20:52	18:33	22:44	26:48	20:22	21:00	22:44	25:54	18:16	18:58
3 09R	25	20:00	21:20	19:47	18:07	19:49	21:25	22:00	18:00	19:49	18:50	19:00	20:40
3 09R	50	18:36	18:04	20:14	17:10	18:47	19:00	20:18	18:00	18:47	19:00	18:26	17:29
3 09R	75	19:43	18:05	19:35	24:54	19:37	20:00	19:00	23:45	19:37	19:00	18:00	23:17
3 09R	100	17:09	27:34	25:20	19:09	18:11	26:37	24:50	19:12	17:02	22:12	24:11	23:25
3 09R	125	21:20	27:55	17:39	22:38	21:09	26:27	18:40	23:26	22:09	19:27	18:59	24:07
3 09R	150	17:29	21:48	20:32	21:13	17:36	19:14	24:50	22:00	17:36	19:07	20:29	25:00
3 09R	175	19:17	22:35	18:44	16:02	20:12	23:00	20:00	16:00	19:54	19:00	21:00	16:24
3 09R	200	17:35	18:14	24:37	20:47	19:00	18:49	22:21	22:00	19:00	17:15	23:55	20:30
3 09R	225	21:41	21:31	24:08	21:43	21:37	22:45	23:06	24:20	21:37	19:45	23:09	25:47
3 09R	250	22:07	23:46	36:09	33:11	22:08	25:20	34:34	33:30	21:05	26:31	34:26	33:30
3 09R	275	24:35	38:19	36:39	86:51	23:36	39:17	36:40	86:51	23:36	39:17	36:40	86:51
4 27L	0	34:30	24:09	23:24	19:08	32:45	24:50	25:35	21:00	32:16	27:00	23:00	18:26
4 27L	25	27:43	20:05	20:52	17:02	28:20	21:00	21:00	19:00	28:26	19:26	18:34	19:30
4 27L	50	18:21	19:40	17:56	22:24	18:51	21:27	19:00	23:00	17:51	19:27	19:11	19:00
4 27L	75	16:14	22:56	24:05	24:04	18:20	23:00	22:57	22:41	17:43	19:00	27:23	21:52
4 27L	100	22:18	29:20	22:20	21:58	22:25	27:20	23:24	21:51	22:25	26:20	24:07	20:18
4 27L	125	24:31	22:23	22:49	20:07	24:07	22:51	21:00	22:58	23:07	21:51	19:10	25:43
4 27L	150	22:41	20:09	21:49	17:44	21:50	22:02	21:00	18:54	21:50	18:59	18:21	23:43
4 27L	175	22:26	19:00	21:50	18:52	22:26	20:32	22:00	21:00	20:58	21:00	21:01	18:30
4 27L	200	15:34	23:08	22:26	21:56	17:15	24:00	21:00	22:00	18:35	20:11	19:00	23:13
4 27L	225	21:18	22:15	22:55	22:13	21:18	22:15	24:00	21:39	19:28	23:15	20:00	26:29
4 27L	250	21:50	24:29	21:39	18:43	23:14	22:41	22:36	19:00	22:14	20:44	25:33	17:44
4 27L	275	21:03	19:23	17:25	21:09	21:03	20:00	19:00	22:00	21:03	19:13	16:42	19:00

TABLE F.5: CTOT compliance and delay results for partial datasets, datasets 5 and 6

		Real times		Predicted times		Automated sequence		Delay %			Dataset details			
DS.	Index	CTOT	Delay	CTOT	Delay	CTOT	Delay	RvP	AvP	AvR	C.	Lt.	Med.	Hvy.
5 27R	0	5	7:17	4	6:45	1.00	4:21	-8	35	40	26	0	38	22
5 27R	25	4	9:10	4	8:31	0.00	4:34	-8	46	50	24	0	42	18
5 27R	50	2	9:22	2	9:04	1.00	5:27	-3	40	42	24	0	40	20
5 27R	75	6	9:10	7	10:52	1.00	8:39	16	20	6	27	1	42	17
5 27R	100	6	8:58	7	10:04	2.00	6:23	11	37	29	26	1	45	14
5 27R	125	1	8:04	2	9:12	0.00	5:47	12	37	28	19	0	47	13
5 27R	150	0	5:40	1	7:28	0.00	6:06	24	18	-8	14	0	50	10
5 27R	175	1	4:18	1	4:54	0.00	3:34	12	27	17	9	0	48	12
5 27R	200	1	7:32	1	7:23	0.00	5:13	-2	29	31	2	0	46	14
5 27R	225	0	8:11	0	8:17	0.00	5:30	1	34	33	0	0	42	18
5 27R	250	0	5:19	0	5:15	0.00	4:16	-1	19	19	0	0	28	32
5 27R	275	0	3:38	0	3:20	0.00	2:45	-9	18	24	0	0	12	39
6 27R	0	0	4:11	0	4:49	0.00	4:43	13	2	-13	39	0	57	3
6 27R	25	0	4:49	0	7:07	0.00	4:40	32	34	3	31	0	54	6
6 27R	50	0	4:35	0	7:00	0.00	3:59	34	43	13	13	0	50	10
6 27R	75	2	4:14	2	5:27	1.00	3:38	22	33	14	15	0	47	13
6 27R	100	3	6:23	2	6:18	1.00	4:28	-1	29	30	19	0	42	18
6 27R	125	2	6:41	2	7:51	0.00	5:11	15	34	22	20	1	41	18
6 27R	150	1	4:36	1	4:54	0.00	4:00	6	18	13	15	1	36	23
6 27R	175	0	3:49	0	3:54	0.00	3:35	2	8	6	11	1	34	25
6 27R	200	1	6:04	1	7:10	0.00	4:35	15	36	24	12	0	33	27
6 27R	225	1	7:52	2	10:10	0.00	5:42	23	44	28	11	0	37	23
6 27R	250	0	6:05	1	8:22	0.00	5:03	27	39	17	8	0	37	23
6 27R	275	0	4:59	0	6:23	0.00	3:33	22	44	29	5	0	39	21

Table F.6: Duration results for partial datasets, datasets 5 and 6

DS.	Index	Real sequence duration 1st	2nd	3rd	4th	Manual sequence duration 1st	2nd	3rd	4th	Automated sequence duration 1st	2nd	3rd	4th
5 27R	0	23:16	21:08	21:10	22:04	23:01	20:00	23:09	21:21	22:01	19:04	21:52	17:29
5 27R	25	18:50	25:33	23:13	19:40	19:04	24:21	23:45	17:00	19:19	18:01	23:56	19:54
5 27R	50	22:40	23:57	17:29	20:30	22:16	22:00	19:49	22:00	21:09	19:42	20:00	21:07
5 27R	75	18:55	19:58	21:56	18:35	19:16	22:00	22:00	19:00	20:26	20:00	17:26	18:51
5 27R	100	21:50	21:11	18:39	18:37	22:38	21:00	20:00	18:00	20:12	19:28	19:40	18:32
5 27R	125	18:16	20:34	18:03	21:05	18:22	20:00	20:00	22:00	16:22	18:59	18:45	23:09
5 27R	150	18:12	18:20	25:20	21:16	19:35	21:00	22:00	21:18	18:09	19:57	22:59	21:50
5 27R	175	23:15	25:04	20:22	21:04	22:25	27:16	21:00	19:00	22:25	25:50	18:58	20:39
5 27R	200	22:24	18:51	23:16	18:13	21:26	20:00	22:56	19:00	20:43	20:37	19:33	16:00
5 27R	225	20:54	21:13	20:39	21:13	20:05	22:07	21:00	22:06	19:58	17:50	21:36	25:46
5 27R	250	18:21	24:11	34:12	46:49	18:42	23:28	34:47	47:03	18:26	22:04	36:27	46:03
5 27R	275	21:49	39:30	33:36	46:44	21:54	36:23	36:43	46:33	22:06	35:38	37:16	46:33
6 27R	0	33:24	22:32	21:32	17:07	33:44	23:53	21:10	17:00	33:13	23:37	21:18	17:04
6 27R	25	22:29	17:19	24:19	19:40	22:25	19:00	26:00	20:00	22:25	18:15	22:28	19:04
6 27R	50	20:20	22:16	23:48	21:56	23:11	22:00	23:00	24:00	19:11	22:05	25:04	22:46
6 27R	75	22:36	23:59	22:30	26:05	23:07	26:46	22:00	22:00	21:13	25:19	22:24	23:59
6 27R	100	23:23	26:44	21:07	16:27	24:03	24:47	21:53	18:00	22:02	25:51	21:01	16:20
6 27R	125	22:23	20:54	18:45	17:32	22:48	21:00	20:00	18:00	21:57	19:00	17:49	21:32
6 27R	150	18:11	19:15	21:40	20:55	18:53	19:00	21:31	19:44	16:53	20:27	22:46	18:02
6 27R	175	19:31	22:15	24:08	21:46	19:31	22:31	24:14	21:41	19:31	22:31	24:14	19:44
6 27R	200	22:05	22:33	20:24	17:47	21:14	23:01	23:00	17:00	22:01	21:14	18:54	17:17
6 27R	225	20:52	19:04	19:20	20:37	22:56	19:00	20:00	21:00	21:00	16:57	18:22	22:52
6 27R	250	16:29	20:46	23:27	21:37	19:02	21:00	23:00	21:00	16:33	19:49	23:50	19:50
6 27R	275	23:21	23:05	22:29	16:23	23:51	23:00	22:00	19:00	23:06	21:06	21:32	20:06

TABLE F.7: CTOT compliance and delay results for partial datasets, datasets 7 and 8

		Real times		Predicted times		Automated sequence		Delay %			Dataset details			
DS.	Index	CTOT	Delay	CTOT	Delay	CTOT	Delay	RvP	AvP	AvR	C.	Lt.	Med.	Hvy.
7 27L	0	2	7:12	2	7:41	2.00	5:53	6	23	18	12	0	44	16
7 27L	25	0	9:09	0	10:59	0.00	5:59	17	46	35	9	0	41	19
7 27L	50	1	7:23	2	9:32	1.00	5:43	22	40	23	11	0	40	20
7 27L	75	1	4:26	3	5:59	1.00	4:09	26	30	6	12	1	44	15
7 27L	100	0	4:20	1	7:09	0.00	4:49	39	33	-11	12	1	50	9
7 27L	125	1	6:12	2	8:48	1.00	4:56	30	44	20	10	2	46	12
7 27L	150	1	6:16	1	8:07	1.00	5:15	23	35	16	10	1	44	15
7 27L	175	0	4:46	0	6:24	0.00	4:23	25	32	8	9	0	46	14
7 27L	200	0	6:13	0	7:33	0.00	4:05	18	46	34	4	0	48	11
7 27L	225	0	8:01	0	9:20	0.00	5:40	14	39	29	1	0	30	4
8 09R	0	3	4:09	6	7:02	3.00	3:53	41	45	6	47	0	54	6
8 09R	25	1	3:15	3	4:27	1.00	4:01	27	10	-23	45	0	55	5
8 09R	50	1	3:29	3	5:10	1.00	3:43	33	28	-7	40	0	50	10
8 09R	75	1	3:21	1	3:34	1.00	3:14	6	9	3	39	0	45	15
8 09R	100	1	3:42	1	3:36	1.00	3:30	-3	3	6	33	0	38	22
8 09R	125	0	5:32	1	5:18	0.00	4:11	-4	21	24	23	0	38	22
8 09R	150	3	7:00	4	8:02	2.00	4:48	13	40	31	16	0	38	22
8 09R	175	3	6:16	3	6:21	2.00	4:57	1	22	21	20	0	37	23
8 09R	200	1	5:59	1	4:52	1.00	4:01	-23	18	33	23	0	35	25
8 09R	225	1	7:15	1	8:32	1.00	6:50	15	20	6	19	0	37	23
8 09R	250	1	6:52	1	9:54	2.00	6:58	31	30	-1	10	0	38	22
8 09R	275	1	6:22	1	6:28	1.00	4:57	2	24	22	6	0	27	16

TABLE F.8: Duration results for partial datasets, datasets 7 and 8

		Real sequence duration				Manual sequence duration				Automated sequence duration			
DS.	Index	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
7 27L	0	22:23	25:28	23:10	18:35	22:52	24:28	24:08	20:00	23:44	23:13	22:00	18:15
7 27L	25	23:26	20:06	19:46	18:01	23:56	21:28	20:00	21:00	21:56	20:12	17:35	18:35
7 27L	50	20:36	19:15	19:30	17:42	20:46	20:00	21:00	20:00	20:46	16:42	19:07	20:28
7 27L	75	18:36	19:23	19:21	20:15	18:39	21:00	20:00	22:00	17:39	19:19	21:57	20:00
7 27L	100	18:33	21:40	19:47	16:03	19:41	22:00	23:00	18:00	19:41	21:19	19:01	16:34
7 27L	125	19:38	17:27	23:27	18:58	20:05	20:00	24:00	19:00	19:07	17:48	21:01	20:51
7 27L	150	18:36	22:04	18:12	19:57	18:43	22:32	22:00	19:00	17:43	21:48	20:27	16:42
7 27L	175	18:53	20:03	26:14	21:45	20:48	19:00	25:55	22:00	19:35	17:29	28:39	19:21
7 27L	200	18:12	29:26	20:48	18:27	18:05	29:18	22:00	20:00	18:05	27:39	17:58	23:11
7 27L	225	11:20	11:46	11:31	10:48	11:20	12:47	12:00	11:00	11:20	10:30	9:00	14:35
8 09R	0	39:58	18:40	21:32	20:48	41:14	22:00	20:00	20:00	37:14	20:46	22:34	20:52
8 09R	25	19:35	21:49	19:36	20:10	20:00	22:26	20:10	22:00	20:00	22:26	20:00	21:00
8 09R	50	21:37	17:43	25:22	25:01	22:36	19:00	25:00	22:24	20:58	19:26	24:41	23:32
8 09R	75	23:09	26:41	26:51	22:55	24:19	26:17	26:03	23:04	24:21	24:34	27:31	23:17
8 09R	100	25:49	24:06	22:56	19:49	26:03	23:52	21:53	20:10	26:03	23:52	21:53	20:22
8 09R	125	24:32	21:16	23:00	17:43	24:36	21:10	22:00	19:50	24:35	21:23	20:11	18:48
8 09R	150	20:45	19:58	20:10	18:12	20:27	22:05	20:00	19:00	19:27	19:26	18:11	21:28
8 09R	175	19:18	17:42	21:32	27:07	19:47	18:00	22:00	22:34	18:47	17:00	23:14	22:14
8 09R	200	20:40	29:48	19:36	21:01	21:30	24:39	22:10	22:13	20:38	24:25	23:16	20:07
8 09R	225	25:45	21:14	19:05	18:15	25:17	22:05	21:26	20:00	25:28	20:07	20:54	19:21
8 09R	250	19:00	21:11	21:41	22:11	21:21	23:00	20:00	24:00	19:21	21:01	19:10	25:41
8 09R	275	13:48	17:57	16:44	13:44	13:39	16:00	20:11	15:00	13:35	14:18	21:14	12:08

TABLE F.9: CTOT compliance and delay results for partial datasets, datasets 9 and 10

		Real times		Predicted times		Automated sequence		Delay %			Dataset details			
DS.	Index	CTOT	Delay	CTOT	Delay	CTOT	Delay	RvP	AvP	AvR	C.	Lt.	Med.	Hvy.
9 27R	0	1	5:44	2	6:45	1.00	4:59	15	26	13	44	0	55	5
9 27R	25	1	5:46	2	7:22	1.00	5:30	22	25	5	38	0	54	6
9 27R	50	0	4:00	0	4:40	0.00	3:53	14	17	3	26	0	50	10
9 27R	75	0	3:45	1	4:30	0.00	3:47	17	16	-1	20	0	48	12
9 27R	100	1	5:23	3	6:40	1.00	5:04	19	24	6	22	0	46	14
9 27R	125	1	6:46	2	6:58	1.00	5:14	3	25	23	24	0	41	19
9 27R	150	0	6:40	0	6:19	0.00	4:36	-6	27	31	24	0	35	25
9 27R	175	0	5:12	0	5:16	0.00	4:24	1	16	15	22	0	34	26
9 27R	200	0	5:25	0	4:56	0.00	4:03	-10	18	25	22	0	36	24
9 27R	225	3	7:46	5	8:06	2.00	6:01	4	26	23	30	0	40	20
9 27R	250	3	7:23	5	8:33	0.06	6:11	14	28	16	28	0	39	21
9 27R	275	1	6:11	3	7:50	0.00	4:55	21	37	20	23	0	39	21
10 27L	0	2	5:41	2	5:32	0.00	5:10	-3	7	9	28	0	42	18
10 27L	25	1	5:22	1	5:35	1.00	5:14	4	6	3	31	0	41	19
10 27L	50	5	7:30	6	8:43	3.00	6:15	14	28	17	29	0	42	18
10 27L	75	5	8:14	7	9:55	2.00	7:19	17	26	11	21	0	42	18
10 27L	100	1	7:17	5	11:36	0.00	7:45	37	33	-6	20	0	50	10
10 27L	125	1	6:10	1	6:53	0.00	5:27	10	21	12	17	0	49	11
10 27L	150	2	7:20	2	6:51	2.00	4:18	-7	37	41	15	0	46	14
10 27L	175	1	7:42	1	7:49	1.00	5:58	1	24	22	10	0	44	16
10 27L	200	0	6:32	1	7:27	0.00	4:34	12	39	30	9	0	41	19
10 27L	225	0	6:10	0	6:50	0.00	4:07	10	40	33	9	1	36	23
10 27L	250	0	3:49	0	3:48	0.00	3:20	0	12	13	5	1	26	33
10 27L	275	0	2:00	0	2:04	0.00	2:04	4	0	-4	1	1	15	40

Table F.10: Duration results for partial datasets, datasets 9 and 10

		Real sequence duration				Manual sequence duration				Automated sequence duration			
DS.	Index	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
9 27R	0	36:00	20:36	20:03	17:46	34:55	21:57	22:00	17:00	35:52	20:13	20:13	18:03
9 27R	25	18:26	20:42	17:32	22:17	20:46	20:00	18:00	20:31	19:42	17:35	20:39	21:21
9 27R	50	18:32	21:06	24:05	19:37	18:30	21:05	25:16	21:00	18:22	21:00	25:08	19:08
9 27R	75	24:19	22:22	24:09	25:43	25:01	22:01	24:15	26:35	25:00	21:08	24:41	26:27
9 27R	100	25:44	25:49	20:59	19:06	26:15	26:35	21:00	20:00	25:15	25:59	21:12	19:37
9 27R	125	23:10	21:02	21:00	19:43	23:07	22:00	20:00	19:57	22:26	21:13	19:00	19:47
9 27R	150	21:14	21:14	19:38	20:43	20:46	21:50	19:34	20:00	19:46	21:12	19:00	21:34
9 27R	175	21:18	20:23	21:07	24:32	21:32	19:16	22:48	21:44	20:05	20:05	22:13	26:13
9 27R	200	22:42	21:10	32:58	17:58	23:16	22:00	27:35	22:52	23:03	21:19	26:54	22:57
9 27R	225	29:08	23:06	19:03	16:41	24:43	28:22	21:00	16:00	26:38	25:01	18:03	18:16
9 27R	250	17:58	18:34	22:06	15:28	18:04	20:00	21:00	18:00	16:55	18:52	21:30	15:11
9 27R	275	18:55	19:20	23:21	15:45	18:55	21:00	24:00	17:00	17:54	19:49	22:51	16:00
10 27L	0	21:54	25:06	23:51	23:18	21:50	25:05	23:15	23:24	20:50	25:20	24:00	24:00
10 27L	25	23:22	24:25	28:07	20:17	23:19	24:22	28:51	21:00	24:24	23:38	28:30	24:52
10 27L	50	24:28	26:13	20:48	20:10	25:15	26:09	22:00	20:00	24:15	26:26	18:35	23:00
10 27L	75	20:08	22:25	19:05	16:00	20:21	22:00	22:00	19:00	19:21	21:00	19:32	20:00
10 27L	100	17:50	19:28	16:32	16:39	19:37	23:00	17:00	19:00	19:37	17:50	17:16	19:17
10 27L	125	15:04	17:49	20:45	27:08	15:20	19:00	21:00	25:26	15:14	17:00	19:28	29:04
10 27L	150	18:09	26:47	23:31	15:46	18:09	25:16	24:47	17:24	17:51	25:34	19:41	19:09
10 27L	175	23:19	18:11	20:18	19:25	23:06	18:24	20:18	20:00	22:06	17:05	22:30	17:03
10 27L	200	17:45	22:04	17:38	23:40	17:19	23:01	19:00	23:00	15:39	22:37	17:14	21:36
10 27L	225	18:05	22:51	25:06	25:55	19:35	22:00	24:27	26:50	17:35	20:21	25:11	29:45
10 27L	250	23:15	30:19	27:57	30:32	23:39	29:55	27:57	31:16	21:39	31:55	27:57	31:16
10 27L	275	23:48	27:34	45:40	99:33	23:48	27:34	45:40	99:33	23:48	27:34	45:40	99:33

Appendix G

Results Showing The Effects Of Constraints

The tables in this appendix present the results of experiments to investigate the short-term performance of the system. The experiments performed to generate these results were described in section 8.5. Each table shows the reduction in the number of CTOTs missed and the percentage decrease in the delay from removing different combinations of constraints from the problem considered. The four constraints which were considered for removal were the departure route separations (labelled S), the wake vortex/weight class separations (labelled W), the effects of the holding area (labelled H) and the CTOT take-off time-slots (labelled C).

The first row of results, labelled 'None', shows the performance of the system with all constraints present. Each following row shows the results obtained from removing a different combination of constraints. For example, the row labelled *'W H'* shows the improved performance of the system when both weight class and holding area effects have been removed.

Table G.1: Relative effects of the removal of constraints, dataset 1

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				**4.00**	**86903**	4.00	84698	3.00	87382
S	-	-	-	**1**	**20%**	1	19%	0	20%
-	W	-	-	**0**	**10%**	0	11%	0	12%
-	-	H	-	**0**	**3%**	0	0%	0	2%
-	-	-	C	**-**	**14%**	-	12%	-	13%
S	W	-	-	**1**	**29%**	1	27%	0	29%
S	-	H	-	**1**	**21%**	1	19%	0	21%
S	-	-	C	**-**	**33%**	-	31%	-	34%
-	W	H	-	**0**	**13%**	0	11%	0	12%
-	W	-	C	**-**	**23%**	-	21%	-	21%
-	-	H	C	**-**	**14%**	-	12%	-	14%
S	W	H	-	**1**	**29%**	1	27%	0	29%
S	W	-	C	**-**	**41%**	-	39%	-	41%
S	-	H	C	**-**	**33%**	-	31%	-	34%
-	W	H	C	**-**	**23%**	-	21%	-	21%
All				**-**	**41%**	-	39%	-	41%

Table G.2: Relative effects of the removal of constraints, dataset 2

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				4.00	96515	**3.00**	**88484**	3.00	96883
S	-	-	-	1	22%	**0**	**19%**	0	23%
-	W	-	-	0	14%	**0**	**12%**	0	14%
-	-	H	-	1	8%	**0**	**0%**	0	9%
-	-	-	C	-	19%	**-**	**13%**	-	19%
S	W	-	-	1	32%	**0**	**29%**	0	33%
S	-	H	-	1	26%	**0**	**19%**	0	26%
S	-	-	C	-	38%	**-**	**32%**	-	38%
-	W	H	-	1	19%	**0**	**12%**	0	18%
-	W	-	C	-	31%	**-**	**25%**	-	29%
-	-	H	C	-	20%	**-**	**13%**	-	20%
S	W	H	-	1	35%	**0**	**29%**	0	35%
S	W	-	C	-	46%	**-**	**41%**	-	46%
S	-	H	C	-	38%	**-**	**32%**	-	38%
-	W	H	C	-	31%	**-**	**25%**	-	29%
All				-	46%	**-**	**41%**	-	46%

TABLE G.3: Relative effects of the removal of constraints, dataset 3

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				5.00	87087	5.00	86773	**5.05**	**86810**
S	-	-	-	0	21%	0	21%	**0**	**20%**
-	W	-	-	0	8%	0	8%	**0**	**11%**
-	-	H	-	0	0%	0	0%	**0**	**4%**
-	-	-	C	-	4%	-	4%	**-**	**7%**
S	W	-	-	0	29%	0	28%	**0**	**28%**
S	-	H	-	0	21%	0	21%	**0**	**21%**
S	-	-	C	-	25%	-	25%	**-**	**25%**
-	W	H	-	0	8%	0	8%	**0**	**11%**
-	W	-	C	-	12%	-	11%	**-**	**15%**
-	-	H	C	-	5%	-	4%	**-**	**9%**
S	W	H	-	0	29%	0	28%	**0**	**28%**
S	W	-	C	-	32%	-	32%	**-**	**32%**
S	-	H	C	-	25%	-	25%	**-**	**25%**
-	W	H	C	-	12%	-	11%	**-**	**15%**
All				-	32%	-	32%	**-**	**32%**

TABLE G.4: Relative effects of the removal of constraints, dataset 4

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				4.00	91913	**3.00**	**91115**	3.00	95181
S	-	-	-	1	20%	**0**	**20%**	0	22%
-	W	-	-	0	6%	**0**	**5%**	0	6%
-	-	H	-	1	1%	**0**	**0%**	0	2%
-	-	-	C	-	12%	**-**	**11%**	-	11%
S	W	-	-	1	29%	**0**	**29%**	0	31%
S	-	H	-	1	20%	**0**	**20%**	0	23%
S	-	-	C	-	33%	**-**	**33%**	-	36%
-	W	H	-	1	6%	**0**	**5%**	0	8%
-	W	-	C	-	18%	**-**	**18%**	-	20%
-	-	H	C	-	12%	**-**	**11%**	-	15%
S	W	H	-	1	29%	**0**	**29%**	0	32%
S	W	-	C	-	43%	**-**	**42%**	-	45%
S	-	H	C	-	34%	**-**	**33%**	-	36%
-	W	H	C	-	18%	**-**	**18%**	-	20%
All				-	43%	**-**	**42%**	-	45%

Table G.5: Relative effects of the removal of constraints, dataset 5

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				**1.00**	**84173**	1.00	83995	1.00	91843
S	-	-	-	**0**	**19%**	0	19%	0	25%
-	W	-	-	**0**	**9%**	0	9%	0	10%
-	-	H	-	**0**	**0%**	0	0%	0	3%
-	-	-	C	**-**	**4%**	-	4%	-	6%
S	W	-	-	**0**	**29%**	0	29%	0	35%
S	-	H	-	**0**	**20%**	0	19%	0	26%
S	-	-	C	**-**	**25%**	-	24%	-	31%
-	W	H	-	**0**	**9%**	0	9%	0	11%
-	W	-	C	**-**	**13%**	-	13%	-	16%
-	-	H	C	**-**	**4%**	-	4%	-	7%
S	W	H	-	**0**	**29%**	0	29%	0	35%
S	W	-	C	**-**	**33%**	-	33%	-	39%
S	-	H	C	**-**	**25%**	-	24%	-	31%
-	W	H	C	**-**	**13%**	-	13%	-	16%
All				**-**	**33%**	-	33%	-	39%

Table G.6: Relative effects of the removal of constraints, dataset 6

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				**1.00**	**84669**	1.00	84671	1.00	93123
S	-	-	-	**0**	**18%**	0	18%	0	24%
-	W	-	-	**0**	**9%**	0	9%	0	8%
-	-	H	-	**0**	**0%**	0	0%	0	3%
-	-	-	C	**-**	**12%**	-	12%	-	16%
S	W	-	-	**0**	**27%**	0	27%	0	33%
S	-	H	-	**0**	**18%**	0	18%	0	25%
S	-	-	C	**-**	**29%**	-	29%	-	36%
-	W	H	-	**0**	**9%**	0	9%	0	8%
-	W	-	C	**-**	**19%**	-	19%	-	21%
-	-	H	C	**-**	**12%**	-	12%	-	17%
S	W	H	-	**0**	**27%**	0	27%	0	33%
S	W	-	C	**-**	**37%**	-	37%	-	43%
S	-	H	C	**-**	**29%**	-	29%	-	36%
-	W	H	C	**-**	**19%**	-	19%	-	21%
All				**-**	**37%**	-	37%	-	43%

TABLE G.7: Relative effects of the removal of constraints, dataset 7

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				4.00	72784	**4.00**	**71958**	4.00	79941
S	-	-	-	0	25%	**0**	**24%**	0	32%
-	W	-	-	0	12%	**0**	**11%**	0	10%
-	-	H	-	0	1%	**0**	**0%**	0	5%
-	-	-	C	-	7%	**-**	**8%**	-	5%
S	W	-	-	0	34%	**0**	**33%**	0	40%
S	-	H	-	0	25%	**0**	**24%**	0	32%
S	-	-	C	-	32%	**-**	**31%**	-	37%
-	W	H	-	0	12%	**0**	**11%**	0	13%
-	W	-	C	-	20%	**-**	**19%**	-	14%
-	-	H	C	-	11%	**-**	**10%**	-	9%
S	W	H	-	0	34%	**0**	**33%**	0	40%
S	W	-	C	-	40%	**-**	**39%**	-	45%
S	-	H	C	-	32%	**-**	**31%**	-	38%
-	W	H	C	-	20%	**-**	**19%**	-	17%
All				-	40%	**-**	**39%**	-	45%

TABLE G.8: Relative effects of the removal of constraints, dataset 8

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				9.00	78246	9.00	78371	**9.00**	**83570**
S	-	-	-	0	20%	0	20%	**0**	**25%**
-	W	-	-	0	8%	0	8%	**0**	**11%**
-	-	H	-	0	0%	0	0%	**0**	**3%**
-	-	-	C	-	9%	-	9%	**-**	**12%**
S	W	-	-	0	30%	0	30%	**0**	**34%**
S	-	H	-	0	20%	0	20%	**0**	**25%**
S	-	-	C	-	27%	-	27%	**-**	**32%**
-	W	H	-	0	8%	0	8%	**0**	**12%**
-	W	-	C	-	17%	-	18%	**-**	**21%**
-	-	H	C	-	9%	-	9%	**-**	**12%**
S	W	H	-	0	30%	0	30%	**0**	**34%**
S	W	-	C	-	36%	-	36%	**-**	**40%**
S	-	H	C	-	27%	-	27%	**-**	**32%**
-	W	H	C	-	17%	-	18%	**-**	**21%**
All				-	36%	-	36%	**-**	**40%**

Table G.9: Relative effects of the removal of constraints, dataset 9

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				**4.00**	**95037**	3.00	93238	4.00	98111
S	-	-	-	**1**	**19%**	0	19%	1	22%
-	W	-	-	**1**	**11%**	0	10%	1	10%
-	-	H	-	**1**	**2%**	0	0%	1	4%
-	-	-	C	**-**	**14%**	-	12%	-	13%
S	W	-	-	**1**	**31%**	0	29%	1	33%
S	-	H	-	**1**	**20%**	0	19%	1	23%
S	-	-	C	**-**	**29%**	-	28%	-	31%
-	W	H	-	**1**	**11%**	0	10%	1	11%
-	W	-	C	**-**	**22%**	-	20%	-	22%
-	-	H	C	**-**	**14%**	-	12%	-	15%
S	W	H	-	**1**	**31%**	0	29%	1	33%
S	W	-	C	**-**	**39%**	-	38%	-	41%
S	-	H	C	**-**	**29%**	-	28%	-	31%
-	W	H	C	**-**	**22%**	-	20%	-	22%
All				**-**	**39%**	-	38%	-	41%

Table G.10: Relative effects of the removal of constraints, dataset 10

Constraints removed				27R CTOT	27R Delay	27L CTOT	27L Delay	09R CTOT	09R Delay
None				4.00	99054	**4.00**	**93933**	4.01	98937
S	-	-	-	0	29%	**0**	**25%**	0	28%
-	W	-	-	0	17%	**0**	**16%**	0	13%
-	-	H	-	0	5%	**0**	**0%**	0	6%
-	-	-	C	-	19%	**-**	**15%**	-	17%
S	W	-	-	0	37%	**0**	**34%**	0	36%
S	-	H	-	0	29%	**0**	**25%**	0	29%
S	-	-	C	-	39%	**-**	**35%**	-	38%
-	W	H	-	0	21%	**0**	**16%**	0	18%
-	W	-	C	-	30%	**-**	**26%**	-	26%
-	-	H	C	-	19%	**-**	**15%**	-	18%
S	W	H	-	0	37%	**0**	**34%**	0	37%
S	W	-	C	-	45%	**-**	**42%**	-	45%
S	-	H	C	-	39%	**-**	**35%**	-	38%
-	W	H	C	-	30%	**-**	**26%**	-	27%
All				-	45%	**-**	**42%**	-	45%