UNIVERSITY OF Southampton

University of Southampton Research Repository
ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

http://eprints.soton.ac.uk

UNIVERSITY OF SOUTHAMPTON
FACULTY OF SOCIAL AND HUMAN SCIENCES
School of Mathematics

Airport Runway Optimization

by

Mohammad Mesgarpour

Thesis for the Degree of Doctor of Philosophy

March 2012

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF SOCIAL AND HUMAN SCIENCES
SCHOOL OF MATHEMATICS

Doctor of Philosophy

Aiport Runway Optimization

by Mohammad Mesgarpour

This thesis considers the scheduling of aircraft landing and take-off problems on a single runway where aircraft must respect various operational constraints. The aim is to introduce generic models and solution approaches that can be implemented in practice. Existing solution methods and techniques of airport runway optimization have been reviewed. Several solution methods such as mixed integer programming, dynamic programming, iterated descent local search and simulated annealing are proposed for the scheduling of aircraft landings in the static and dynamic environment. A multi-objective formulation is used for taking into account runway throughput, earliness and lateness, and the cost of fuel arising from aircraft manoeuvres and additional flight time incurred to achieve the landing schedule. Moreover, computational results are presented using real data from Heathrow airport as well as randomly generated problem instances which are generated based on characteristics of the real data. Later, dynamic programming, descent local search and beam search algorithms are proposed for the scheduling of aircraft take-offs in the departure holding area. Scheduling aircraft take-off is formulated as a hierarchical multi-objective problem which includes maximizing departure runway throughput and minimizing total waiting time in the holding area. Performance of the algorithms have been evaluated for three common layouts of holding area. Computational results are presented on randomly generated test data.

Contents

List of Figures

List of Tables

List of Abbreviations

ACO	Ant Colony Optimization
ALP	Aircraft Landing Problem
ALT	Average Landing Time
AMAN	Arrival Manager
ATC	Air Traffic Control
ATM	Air Traffic Management
ATP	Aircraft Take-off Problem
ATT	Average Take-off Time
BS	Beam Search
BSID	Balance of Standard Instrument Departure
CFMU	Central Flow Management Unit
COP	Combinatorial Optimization Problem
CTOT	Calculated Take-Off Time
CPS	Constrained Position Shifting
CT	Computation Time
CTS	Constrained Time Shifting
DLS	Descent Local Search
DP	Dynamic Programming
EF	Extra Fuel
ELT	Estimated Landing Time
ETT	Estimated Take-off Time
FAA	Federal Aviation Administration
FCFS	First-Come First-Served
GA	Genetic Algorithm
GP	Genetic Programming
HP	Holding Pattern

ICAO	International Civil Aviation Organization
ID	Iterated Descent
MIP	Mixed-Integer Programming
MPS	Maximum Position Shifting
ND	Number of Deviation
NATS	National Air Traffic Services
NP	Non-deterministic Polynomial
P	Polynomial
PI	Percentage of Improvement
RHC	Receding Horizon Control
RPS	Relative Position Shifting
SA	Simulated Annealing
SID	Standard Instrument Departure
SLT	Scheduled Landing Time
STT	Scheduled Take-off Time
TD	Total Deviation
TMA	Terminal Manoeuvering Area
TS	Tabu Search
TSP	Travelling Salesman Problem
TW	Time Window
TWT	Total Waiting Time
VFS	Vector-for-Space
WV	Wake Vortex

DECLARATION OF AUTHORSHIP

I, MOHAMMAD MESGARPOUR, declare that the thesis entitled Airport Runway Optimization and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University;
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- parts of this work have been published as:
    1. Bennell, J.A., Mesgarpour, M. and Potts, C.N. (2011) Invited Survey: Airport Runway Scheduling, 4OR: A Quarterly Journal of Operations Research, 9(2), pp. 115-138 (DOI: 10.1007/s10288-011-0172-x).
    2. Mesgarpour, M., Potts, C.N. and Bennell, J.A. (2010) Models for Aircraft Landing Optimization, In Proceedings of the 4th International Conference on Research in Air Transportation (ICRAT 2010), Budapest, Hungary, June 1-4, 2010, pp. 529-532.
    3. Bennell, J.A., Mesgarpour, M. and Potts, C.N. (2011) Dynamic Scheduling of Aircraft Landings, In Proceedings of the 9th Metaheuristics International Conference (MIC2011), Udine, Italy, July 25-28, 2011, pp. 543-545.
    4. Potts, C.N., Mesgarpour, M. and Bennell, J.A. (2012) Dynamic programming for Aircraft Departure Scheduling with Blocking in the Holding Area, In Proceedings of the 13th International Conference on Project Management and Scheduling (PMS 2012), Leuven, Belgium, April 1-4, 2012.

Signed: ....

Date: ....

Acknowledgements

It is my pleasure to thank who made this thesis possible. I would like to thank my supervisors, Professor Chris Potts and Professor Julia Bennell for their support, encouragement and thoughtful guidance during this research. Their technical advice was essential to the completion of this dissertation. They have taught me innumerable lessons and insights on the workings of academic research in general. I also thank my examiners, Doctor Tolga Bektas and Doctor Arthur Richards, who provided many valuable and constructive feedbacks.

This thesis was financed by the European Organisation for the Safety of Air Navigation (EUROCONTROL) in Bretigny, France and School of Mathematics, University of Southampton. I would like to thank both organizations for their generous support. I sincerely acknowledge Mr. Alan Drew, Mr. Paul Humphreys and Mr. Marc Bourgois of EUROCONTROL, Experimental Centre for their help, technical advice and guidance in the area of air traffic management. I would also like to thank Mr John Greenwood of NATS for providing the Heathrow data that form the basis for our computational evaluation of algorithms in Chapter 5.

I would like to acknowledge the friendship and support of all my friends. Last but not least, I would like to express my heart-felt gratitude to my wife, Banafsheh, and my family. I would not have contemplated this road without their love, patience, support.

To my wife and my parents

who always give me love and support.

Chapter 1

Introduction

In this thesis, our focus is on the efficient scheduling of landing aircraft, or specifically the aircraft landing problem (ALP) and impact of the holding area on aircraft take-off problem (ATP). In Section 1.1 a summary about air traffic management is provided. A brief overview of the airport runway optimization is discussed in Section 1.2 and in Section 1.3 the main contributions of this research are explained.

1.1 Air Traffic Management

According to projections, air transportation demand is expected to grow annually at rates between three and five percent in spite of the short-term economic recession (E. Grunewald and Keimel, 2007). Increasing traffic causes congestion in the terminal areas, holding delays for arriving aircraft and long queues at the holding departure areas. Given the current congestion levels in the busier airports, accommodating further flights presents a significant challenge.

Airport runway capacity is often a limiting factor when creating plans to offer additional flights at an airport. This is because improvements to the management of en-route air traffic have shifted the bottleneck from en-route airspace to the airport (Soomer and Franx, 2008), and more specifically to the runway. Although airport capacity can be increased by building a new runway, making the best usage of the existing runway(s) through careful scheduling may reduce the need to improve the infrastructure.

In addition to issues of safety, which is the responsibility of air traffic controllers (ATCs), there are other stakeholders with an interest in how aircraft landings and take-offs are scheduled. Punctuality is a priority for airlines and airports. Airport operations such as gate assignment and baggage handling require careful planning in advance, and delays to an aircraft landing and take-off may have a detrimental effect on similar operations for the subsequent aircraft. Airlines also prefer schedules that minimize the cost of fuel, and governments typically have targets for reducing CO_2 emissions. Long queues and additional manoeuvres by aircraft to create a landing and take-off sequence may increase emissions. ATCs organize the landing and take-off of aircraft to meet safety requirements and maximize throughput. Ideally, the aims of all of the various stakeholders would also be taken into account when scheduling the landings and take-off of aircraft.

Today, Air Traffic Management (ATM) is concerned about traffic optimization at the airport and in terminal manoeuvring areas for economic, environmental and capacity reasons. In this situation, air traffic controllers have to meet various challenges such as: avoiding long air and ground queues; considering the best usage of available airspace, runways, taxiways and gates; taking into account fuel efficiency; reducing noise disturbance and environmental impact; minimizing delays; and accounting for safety issues.

1.2 Airport runway optimization

One of the main factors affecting runway usage is the enforcement of minimum separations between landing/take-off aircraft that arise from safety considerations. Wake vortices are rotating masses of air that are generated by aircraft as a consequence of their lift. Without sufficient separation, wake vortices provide a hazard for the following aircraft. Wake vortices are bigger if they are created by a larger aircraft. Moreover, they have a greater impact when the following aircraft is light rather than heavy. Thus, the required minimum separations between aircraft mainly depend on the weight class of the leading and following aircraft. Consequently, effective scheduling will aim to avoid a light aircraft landing/take-off immediately after a heavy aircraft.

Our research focuses on sequencing and scheduling arrival/departure of aircraft to/from the airport. Considering the complexity of airport runway scheduling, it is hard to find the optimal solution to the problem in most cases. Thus, it draws significant attention from different scientific communities with numerous research studies carried out on modelling and developing algorithms to increase capacity at an airport. Careful sequencing and scheduling can reduce the number of long separation times thereby opening up opportunities for new landing or take-off slots.

1.3 Contribution

Most of the research on scheduling aircraft landings/take-offs deals with the static or off-line problem in which all aircraft to be scheduled are known at the outset. However, ATCs work in a dynamic or on-line environment where new aircraft enter the controller's airspace over time. In this dynamic problem, decisions about the landing/take-off of earlier aircraft have to be made without knowledge of those that may enter the airspace or may release the gate at a later time. Any system that is designed to support the decision making of ATCs should therefore consider the dynamic problem. Further, a solution of the static problem is only of theoretical interest unless it forms a component of an algorithm for the dynamic problem.

Another shortcoming of many studies in the literature is that the models do not address all of the important issues in a practical decision-making environment. For example, the objective functions within these models typically do not address the concerns of all of the stakeholders, and some of the important operational constraints are often missing. Furthermore, the solution approaches often have excessively long run-times relative to the almost instantaneous response required in a decision support system that could be of use to ATCs. Finally, many of the algorithms have not been tested using real data.

In view of the above discussion, there is a need for a model that operates in a dynamic environment and considers more of the constraints that arise in practice. Moreover, the model should adopt a multi-objective approach that considers the interests of the different stakeholders. Our aim is to develop a model that meets these requirements, and to design a dynamic/on-line scheduling algorithm that produces solutions sufficiently quickly that it would be of benefit to ATCs.

This research has been divided into two main parts: aircraft landing scheduling and take-off holding area optimization. The former part concerns finding a good and quick solution for sequencing and scheduling arrival aircraft in the terminal manoeuvering area. Static and dynamic versions of the ALP have been studied. Impacts of the take-off holding area on aircraft take-off scheduling have been investigated in the latter part.

The main contributions of the ALP are as follows:

- Majority of published works have looked at the ALP once arriving aircraft are approach the runway in the static environment while we have considered various scheduling time horizons in dynamic environment as well as static environment.

- Impact of changing the freezing time and rescheduling period have been studied in the dynamic case. In the dynamic aircraft landing problem, sequence needs to be updated in response to the arrival of the new aircraft to the planning horizon. Moreover, it is impractical to change the position of the aircraft in the sequence if it is aligned in a straight line approach to the runway threshold for landing. Therefore, rescheduling should be done on a periodic basis without updating the landing time schedule of the aircraft which are in the freezing window.

- Performance of the dynamic programming, iterative descent and simulated annealing algorithms in solving ALP has been compared with the schedules produced by the ATCs at Heathrow airport.

- Proposed algorithms have been evaluated using randomly generated test data with the characteristic of the real test data.

- Various weighted multi-criteria objective functions have been examined to take into account the interest of the various stakeholders such as ATC, airport, airlines and the government.

The main contributions of ATP are as below:

- To our knowledge, it is one of the first studies looking at the impact of the holding area on ATP.
- Effects of three common holding area layouts on the departure sequencing and runway throughput have been investigated.
- Dynamic programming algorithm as an exact method and two heuristic algorithms including beam search and descent local search are described for scheduling of aircraft take-offs.

The remainder of this thesis is organized as follows. Chapter 2 provides an overview of some of the methods used in scheduling landings and take-off aircraft at the the airport. Some basic concepts related to the ALP and ATP such as time window, separation, position shifting are explained and the description of the airport runway scheduling is given in Chapter 3. An extended literature review of the aircraft landing and take-off problem is provided in Chapter 4. The application of various exact and approximation algorithms have been also discussed in this chapter. Chapter 5 offers a complete description of the ALP including the constraints and objective functions. Moreover, this chapter continues with a description of the algorithms and experimental results for the static and dynamic problems. Chapter 6 provides the description of the ATP and associated methods for tackling this problem. Computational evaluation of the introduced algorithms for scheduling aircraft in the holding area based on three common layouts is presented in this Chapter. Finally, Chapter 7 contains the conclusion, some concluding remarks and direction for the future work.

Chapter 2

Methodologies

Optimization can be defined as the process of finding the best possible solution(s) that maximizes or minimizes a given objective function of some decision variables subject to some constraints. Section 2.1 provides background information about combinatorial optimization with focus on scheduling problems. Several solution methods for solving scheduling problems have been discussed in Section 2.2. We explain several approaches including branch-and-bound, dynamic programming, local search, simulated annealing, tabu search, beam search and genetic algorithm.

2.1 Combinatorial optimization problems

An optimization problem can be continuous (an infinite number of feasible solutions) or combinatorial (a finite number of feasible solutions). Combinatorial problems generally maximize or minimize a function of discrete variables. Therefore, combinatorial optimization problems (COPs) can be defined as allocation of limited resources (constraints) to optimally meet desired objectives when the value of some or all the variables are restricted to be integral. COPs are often referred to as integer programming problems. Such problems occur in various fields such as production, inventory control, scheduling, etc. Various methods such as linear programming, dynamic programming, branch and bound, heuristic, and local search can be used to find optimal or near-optimal solutions (Papadimitriou and Steiglitz, 1982).

The number of solutions usually grows exponentially with the number of variables and constraints in COPs. Finding an optimal or even near-optimal solution for the large-sized problem instances (mostly real-world optimization problems) poses increasing demands on time and computational resources in most cases. Therefore, using heuristic algorithms are more attractive than using exact algorithms.

2.1.1 Complexity

A decision problem can be classified into P and NP classes. The P stands for polynomial and it is the set of problems that can be solved in polynomial time on a deterministic Turing machine. The NP refers to non-deterministic polynomial and it is the class of problems that can be solved in polynomial time on a nondeterministic Turing machine. All problems in P are also in NP (Karp, 1972).

A problem is NP-hard if an algorithm to solve it in polynomial time would make it possible to solve all NP problems in polynomial time (Garey and D. S. Johnson, 1979). NP-hard problems do not have to be in NP. A problem is NP-complete if it can be proved that it is NP and it is poly-time reducible to a problem already known to be NP-complete (Karp, 1972). In other words, a problem is NP-complete if it is both NP-hard and an element of NP. NP-complete problems are the hardest problems in NP and NP-hard problems are at least as hard as NP-complete problems.

2.1.2 Scheduling

Scheduling problems are an important class of combinatorial optimization problems. The study of sequencing and scheduling dates back to 1950s (Potts and Strusevich, 2009). Scheduling has become one of the major fields within operational research which has turned to be more challenging and complex today than in the past.

2.1.2.1 Sequencing and scheduling problems

Pinedo (2008) has defined that scheduling is a decision-making process that deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives. Sequencing deals only with the specific ordering of products, items, tasks, etc; while scheduling gives a more complete description about when a particular task can start .

Resources may be machines in a workshop, nurses and practitioners at a hospital, CPU and memory in a computer system, runways at an airport, and mechanics in an automobile repair shop. Activities may be different operations in a manufacturing process, giving services to the patient at a hospital, execution of a computer program, landings and take-offs at an airport, car repairs in an automobile repair shop. Each activity may have different priorities, due date, ready time, etc. Many distinct measures are also available to optimize the problem. In general tasks have to be accomplished with the goal of minimizing or maximizing an objective or a combination of various objectives. One objective can be the minimization of the delay, whereas another objective can be the maximization of the customers satisfaction (Pinedo, 2008).

In terms of modelling uncertainty, the scheduling problems can be classified into two categories of deterministic and stochastic. In the past fifty years, scheduling problems have received an extensive amount of attention especially in deterministic scheduling. The basic assumption of deterministic scheduling is that the parameters of the problems are fixed, in which values should be known exactly (Brucker, 2007). However, many sources of uncertainty can affect the scheduling environments. In stochastic scheduling, one or more uncertainty factors are added to the problem formulation.

2.1.2.2 Multi-objective scheduling

Until the late 1980s, the majority of the scheduling research has been concentrated on a single criterion problem. However, scheduling decision should consider multiple criteria in practice to provide the decision maker with more realistic solutions. Scheduling problems become more difficult to model and solve when more than one objective (criterion) is required. In many cases, it is unlikely that various objectives would be optimized by the same choice of decision variables. In other words, an improvement in one objective is often only a gain at the expense of a detraction in other objectives. Therefore, there exists a trade-off between the multiple objectives. This type of problem is known as a multi-objective scheduling problem (T'kindt and Billaut, 2006; Hoogeveen, 2005).

Gupta et al. (2001) have classified the multi-objective scheduling problems into three different classes:

(a) *Objectives are weighted equally*: Trade-offs can be made between all efficient solutions of the problem.

(b) *Objectives are weighted differently*: Problem can be defined as a single-objective scheduling problem by defining the objective function as the sum of weighted functions.

(c) *Objectives are hierarchical*: The scheduling problem can be solved for the first priority objective by ignoring the other objectives and then be solved for the second priority objective by not changing the optimal solution of the first objective and so on.

2.2 Solution methods

For most NP-hard problems, the performance of exact methods such as branch-and-bound and dynamic programming is not satisfactory due to the huge computational time. In spite of exact algorithms which provide an optimal solution together with the proof of its optimality, heuristic provide a near-optimal or sometimes an optimal solution without proof of its optimality. On one hand, the complexity of exact methods are often too high and unacceptable for solving NP-complete or NP-hard problems. On the other hand, it is often sufficient to come up with a good solution in a reasonable time.

2.2.1 Branch-and-bound

Branch-and-bound method is used for solving integer and discrete optimization problems (Land and Doig, 1960). It is based on the enumeration of solutions which has a tree structure. The branch-and-bound method starts with the root *node*. Branching from a node represents a choice. The main idea of this method is to grow only the most promising nodes. Branch-and-bound approach attempts to omit a node based on the lower bound of the objective function called *bounding*. If the bound is worse than objective value of the trial solution, the node is *pruned*. Tightness and easy calculation of the bound in addition to the quality of the trial solution affect the efficiency of branch-and-bound method. The main disadvantage of the branch-and-bound is that it is usually time-consuming because of the large number of nodes (Pinedo, 2009).

2.2.2 Dynamic programming

Dynamic programming (DP) is a recursive optimization method that solves problems by breaking them into simpler and more trackable problems. DP is originally developed by Richard Bellman in the 1940s (Bellman, 2003). The main feature of the dynamic programming approach is dividing the optimization problem into multiple *stages* which are solved sequentially as one stage at a time.

Each stage is associated with a number of *states* of the process. At each stage the decision rule is determined by evaluating an objective function called the *recursive equation*. In other words, there is a recursive relationship between the decision at a stage and the optimum decision in the previous stage. The procedure of solving the problem starts by first solving a one-stage problem and sequentially including one stage at a time until the overall optimal solution has been found. This procedure can be either *backward*, where the first stage to be analyzed is the final stage of the problem, or *forward*, where the first stage to be solved is the initial stage of the problem.

The dynamic programming relies on a *principle of optimality*. It states that given a current state, an optimal policy for the remaining stages is independent of the policy adopted in the previous stages. In other word, the optimal solution to a problem is a combination of optimal solutions to some of its subproblems. Two main advantages of the dynamic programming are that it breaks down a complex problem into a series of interrelated subproblems and it also saves the computation time over complete enumeration. The classical DP has often been dismissed because of the *curse of dimensionality.* In fact, the number of states often grows exponentially with the dimension of the number of variables in the problem (Bellman, 2003). Approximation methods in DP is introduced to overcome this drawback by finding sub-optimal solutions.

2.2.3 Heuristics

Since NP-hard problems are unlikely to be solved in polynomial time, we have to use solution methods like heuristics and approximation algorithms to find the best possible solution. Heuristic approach is a method to find good (near-optimal) solutions at a reasonable computation time (Smith et al., 1996). Heuristics aim to provide good but not necessarily optimal solutions to difficult problems. They are especially suitable for problems arising in practice.

Heuristics can be classified as either *constructive* or *perturbative* heuristics. While constructive heuristics build the solution from scratch, perturbative heuristics start with an initial complete solution and thereafter try to iteratively improve it. Various heuristic methods have been proposed in the literature such as local search, simulated annealing, tabu search, beam search, genetic programming, etc (Burke and Kendall, 2005). An overview of the heuristic methods which have been used in this thesis are given in the following sections.

2.2.4 Local search

Local search is a class of methods that searches the solution space with the aim of improving the solution. The simplest form of the local search is called *descent local search* (DLS) which starts with an initial solution and iteratively explores neighbourhoods with lower costs. The algorithm performs a series of moves on the initial (starting) solution S_0 to find a local optimal solution S. These moves (transformations) are normally designed based on the neighbourhood structure. In each iteration, if a better solution exists, then it is selected as the current solution. The procedure is continued until no better solutions can be found in the neighbourhood of the current solution.

The final solution is a local optimum based on the neighbourhood function that is used because it stops when no improvement can be made by a single move. The main drawback of the descent local search is that it can get trapped in local optima. *Best improvement* and *first improvement* strategies can be explored. The best neighbour among all neighbour are selected in the best improvement move. In contrast, the first improvement move selects the first neighbour that improves the current solution values. Generally, the first improvement move is quicker than the best improvement move.

Iterated descent local search is a practical type of the local search methods for obtaining near-optimal solutions for a wide range of complex combinatorial optimization problems. It consists of a local search and a perturbation operator. The main advantage of the iterated descent local search is that when the local search procedure is trapped in a local optimal solution, a perturbation operator (*kick* move) is used to transform a local optima into a new starting point for the local search. The perturbation aims to effectively escape local optima without completely loosing partially optimized structure.

2.2.5 Simulated annealing

Simulated Annealing (SA) algorithm is a stochastic heuristic. SA has been developed initially as an algorithm to simulate the process of cooling and crystallization of materials in heat bath (known as the annealing process) by Metropolis et al. (1953) in thermodynamics. Kirkpatrick et al. (1983) has formally introduced the simulated annealing approach and showed that Metropolis algorithm can be applied to solve optimization problems as well. SA is a neighbourhood search algorithm. It is capable of not being stuck in local optima by allowing hill-climbing moves to reach the global optimum.

In general, it is a stochastic optimization method for minimizing a function f over a discrete domain S (Kirkpatrick et al., 1983). A standard SA procedure begins by generating an initial solution $s \in S$. At each stage, the new solution taken from the neighbourhood of the current solution $s' \in S$ is accepted as a new current solution if a solution has a lower or equal cost; if it has a higher cost it is accepted with a probability $e^{-\Delta/t}$, where Δ is the difference between the costs of the s and s' , and t is a parameter of SA referred to as *temperature*. This temperature, which is simply a positive number, is periodically reduced by a temperature scheme, so that it changes gradually from a relatively high value to near zero as the method progresses. Initially, t takes a user-defined value and it is decreased according to a function (referred to as cooling schedule) iteration-by-iteration. Thus, at the beginning of SA most of the worsening moves are accepted, but at the end only improving ones are likely to be accepted.

Apart from temperature and cooling schedule, the performance of the SA model is influenced by factors such as the stopping condition, the choice of the space of feasible solutions, the form of the objective (cost) function, and the way of choosing a neighbourhood structure (Dowsland, 1993).

2.2.6 Tabu search

Tabu search (TS) algorithm is a deterministic search method which is similar to local search algorithm. It is proposed by Glover (1989) and since then it has been applied in variety of COPs. A key aspect of tabu search is that it accepts non-improving moves if one would like to escape from a local optimal solution in spite of SA which does it with decreasing probability as search progresses. Consequently, TS maintains details of the recent moves and prevent the search cycling back to the solutions already examined by the use of short-term memory called *Tabu list*. Moreover, the long-term memory aims to diversify the search to other areas of the search space and prevents cycles in the search.

2.2.7 Beam search

Beam search (BS) method is an adaptation of branch-and-bound in which only the best β promising nodes at each level of search tree are selected to branch based on *global evaluation*, where β is the *beam width*. Then, the other nodes are pruned permanently. During filtering process, some nodes are discarded permanently based on *local evaluation* function value and it should be performed for each set of child nodes branching from the same parent node. The best α children of each beam node are retained for global evaluation step, where α is the *filter width*. BS method has been used for the first time by Lowerre (1976) for the speech recognition. Figure 2.1 illustrates an expanded beam search tree with $\alpha = 2$ and $\beta = 3$.

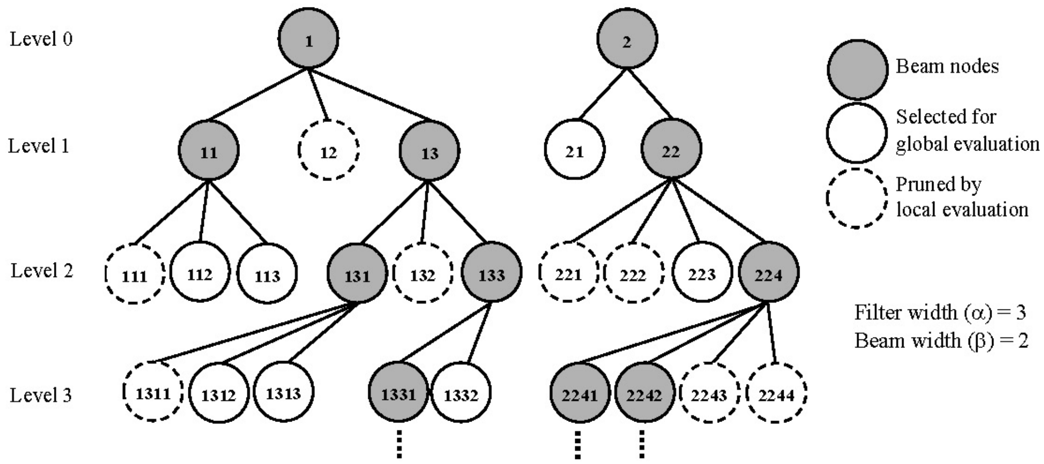

Figure 2.1: Illustration of a beam search

The local evaluation examines candidate partial solutions according to the evaluation criteria and the global evaluation attempts to estimate the minimum cost of the best solution that can be reach from the current node. Although, local evaluation functions are computationally fast, they may lead to eliminate good nodes. On the other hand, global evaluation functions are more accurate but require higher computational time. In fact, there is a trade-off between computation time and solution quality. The running time of the algorithm is polynomial for large-sized problems by restricting the search space.

2.2.8 Genetic algorithms

Evolutionary algorithms (EAs) are stochastic heuristics derived from the evolution theory. Genetic Algorithm (GA) is one of the most popular EAs which is based on natural selection and genetics. It is first introduced by Holland (1975). The GA mimics evolutionary population-based search. The solutions are represented as *chromosomes*. The population is a set of possible solutions called *individuals*. The GA starts with an initial solution called *population* and reproduces new generations by applying genetic operators such as mutation, cross-over and selection. The *selection* operator chooses the fittest individuals for reproduction. The *mutation* operator changes a random gene in an individual. The *cross-over* is for combining the selected individuals (chromosomes of parents) to obtain genetic codes of their offspring (children). The main difference between GAs and other heuristics is that GAs work on a population of possible solution rather than a single solution in their iterations.

In the next chapter, an overview of the air traffic management and description of the aircraft landing problem and aircraft take-off problem are presented.

Chapter 3

Air Traffic Management

Air Traffic Management can be defined as procedures, resources and systems that collectively have a role in safety guiding aircraft in the skies and on the ground. Some background about the air traffic control (ATC) and management as well as description of aircraft runway scheduling will be explained in this chapter. Section 3.1 describes responsibilities of air traffic controllers at different positions besides some key concepts of ATC such as time window, first-come-first-served discipline, runway capacity and separations. An overview of airport runway scheduling including objectives and modelling techniques will be presented in Section 3.2.

3.1 Air Traffic Control

An Air Traffic Management (ATM) system aims to assure safety and efficiency of air traffic flows by establishing a set of services.

3.1.1 Introduction

Three types of facilities control the aircraft between the airport used for take-off and the airport used for landing. These are the airport traffic control tower, terminal airspace control centre and en-route control centre.

The *airport traffic control tower* is responsible for ground traffic control, take-off and landing control within about 5 nautical miles (nm) and 3000 ft above ground level from the airport. The usual responsibilities of controllers in the tower are: clearance delivery, gate hold, ground control, ground planning, and runway control. The *terminal airspace control centre*, which is also called an *approach control airspace* or *Terminal Radar Approach Control* (TRACON), handles departures and arrivals up to 40 nm and 10,000 ft from the airport. The *en-route control airspace*, which is also named the *Air Route Traffic Control Centre* (ARTCC) handles the traffic flow outside the terminal manoeuvring area (see Figure 3.1). For further details, we refer to de Neufville and Odoni (2003).

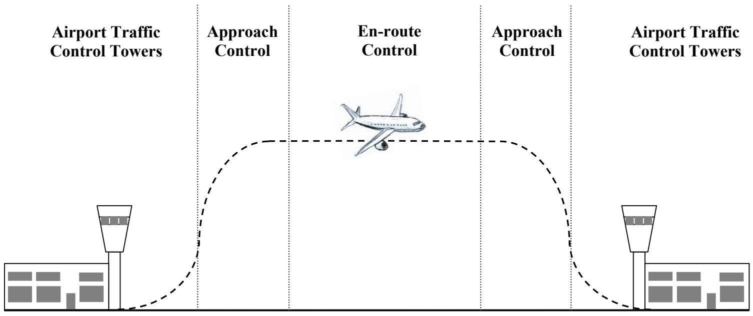

FIGURE 3.1: Air traffic control segments

The airspace is divided into a number of geographical regions of varying size, known as sectors. A sector can be defined as a volume of airspace managed by a team of controllers. Generally, sectors which handle high-flying en-route traffic are much larger than the busy sectors which handle a large amount of climbing and descending traffic. Busy airspace, such as in the London area of the UK, can be subdivided into super-low, low, high, and super-high sectors according to altitude. As a flight proceeds through the airspace, responsibility passes from one sector team to the next.

There is a limitation on the number of aircraft that can fly in each sector at any given time, which depends on several factors such as safety, flight geometry, controllers' workload, weather, surveillance equipment in use, and the training or experience of the air traffic controllers (Filar et al., 2001). It appears that dividing the airspace into smaller sectors may help in dealing with the increasing traffic demand. However, this approach requires pilots to change radio frequencies more often and increases the controllers' workload because of greater need for co-ordination between sectors (de Neufville and Odoni, 2003; Duke, 2009).

3.1.2 Decision problems

Generally, ALPs/ATPs consist of the *sequencing*, *scheduling*, and *runway-assignment* decisions. The sequencing process determines the sequence by which aircraft land or take off from the set of feasible sequences, while the aim of scheduling is to assign a scheduled landing time (SLT)/scheduled take-off time (STT) to each aircraft in the sequence, subject to maintaining operational and safety constraints (Brinton, 1992; Ernst et al., 1999). When more than one runway is available for landing or take-off, each aircraft also has to be assigned to a particular runway.

Scheduling of the landing and take-off of aircraft can be divided into three stages: creating an initial schedule, modifying the schedule, and freezing the schedule (Mesgarpour et al., 2010). For landing aircraft, the initial schedule is based on a first-come first-served (FCFS) order, which is the landing order that would result if each aircraft could proceed to the runway and land without consideration of other aircraft. The FCFS order requires updating when new aircraft enter the airport landing planner's radar range (the Extended Terminal Manoeuvring Area, or Extended TMA), about 30 to 40 minutes before touch down.

The second stage considers new aircraft entering the Extended TMA and adjusts the previous landing sequence to produce an improved schedule. Two to three minutes before landing the schedule is frozen as the aircraft is too close to the runway to make further changes to the landing order or landing time (Neuman and Erzberger, 1991). A similar concept applies for the take-off problem. The first stage may start after informing the controllers about the approximate time an aircraft will be ready to leave the gate and start its journey to the runway; this is called pushback. Modifications to the schedule are made in the second stage when precise times for pushback become available to controllers. The freezing stage begins by entering the aircraft into the holding area. Starting time and length of the freezing time window vary due to configurations of taxiway, departure holding area and runway as well as overtaking constraints.

3.1.3 Time windows

The landing/take-off time of an aircraft must lie between its earliest and latest possible landing/take-off time, which can be dependent on the technical and operational constraints such as fuel limitation, maximum allowed delay, or maximum or minimum airspeed, although other factors such as runway availability, possible manoeuvres, or meeting a connecting flight can also be taken into account. This time window should be treated as a hard constraint. Moreover, there may be a predefined landing time slot in which case controllers aim to assign a *Scheduled Landing Time* (SLT) to each aircraft so that the SLT lies within the aircraft's slot.

If an aircraft uses a congested route or is destined for a hub airport in Europe, then the Central Flow Management Unit (CFMU) of EUROCONTROL in Brussels assigns a *Calculated Take-Off Time* (CTOT). The CTOT limits the time at which aircraft enter these congested areas with the aim of smoothing the traffic in the airspace and at the airports (Atkin, 2008). Controllers aim to assign a *Scheduled Take-Off Time* (STT) to each aircraft from five minutes before to ten minutes after the CTOT. Therefore, the CTOT defines a time window for take-off of the aircraft. Ideally, CTOT is a hard constraint for departure scheduling. However, this can be treated as a soft constraint because violations are sometimes unavoidable.

In theory, there can be multiple non-overlapping time windows for an aircraft to land or take-off. Hence, an aircraft could be constrained to land or take off in any one of the collection of specified time intervals (Balakrishnan and Chandran, 2006).

3.1.4 First-come first-served (FCFS)

The simplest way of sequencing aircraft to land on a single runway is through the *first-come first-served* (FCFS) discipline. It assigns an SLT to each aircraft based upon the order generated by the estimated landing time (ELT) of the aircraft. The landing planner system calculates the ELT based on the planned arrival route, cruise speed (the most economical or preferred speed), and the standard procedure descent profile. Controllers use the ELT as a reference value in computing delays in the terminal area (Neuman and Erzberger, 1991). The order of the aircraft queueing at the holding area is the FCFS order, which provides an Estimated Take-off Time (ETT). While the FCFS rule is fair in terms of the ELTs and ETTs and it simplifies the implementation of operational constraints, FCFS does not necessarily match the preferred landing/take-off order since it does not use important information about the problem (Carr et al., 2000). Moreover, it has been established that FCFS rarely provides the best sequence in terms of runway throughput, average aircraft delay or even average passenger delay (Capri and Ignaccolo, 2004).

3.1.5 Runway capacity and assignment

As stated by Idris et al. (1998a,b), the runway provides the main constraint on capacity in an airport system. Blumstein (1959) introduces the first important analytical model for estimating the capacity of an arrival runway. The runway capacity (maximum throughput) can be defined as the maximum hourly rate of aircraft takeoff or landing operations that can reasonably be accommodated by a single or combination of runways. Capacity is generally dependent on the runway occupancy time, mix of aircraft using the runway, availability of taxiways, aircraft type/performance, spacing between parallel runways, intersection point of runways, mode of operation (segregated or mixed), performance of the ATM systems, weather conditions (visibility, wind strength and direction), and noise restrictions (Bazargan et al., 2002; de Neufville and Odoni, 2003). In segregated-mode, the runway is solely used for either landing or take-off of the aircraft, while mixed-mode allows both landing and take-off on the same runway. The airport capacity model presented by Newell (1979) shows that capacity is greater when runways are operated in mixed-mode. Since increasing the number of runways is often impractical, air traffic controllers aim to use methods and techniques to maximize the throughput from the available runways.

Airports can operate with different numbers and configurations of runways. These can be a single runway, a number of parallel or intersecting runways, or a combination of these. Assigning a runway to the landing/take-off aircraft is a decision made by controllers. The *runway assignment* is typically dependent on the airport configuration, the direction of arriving aircraft (arrival feeder gate), and departure route of the aircraft which is normally specified by the flight plan (Brinton, 1992). While an aircraft approaches the runway, adjustments can be made to the flight plan by assigning the aircraft to an alternative runway, which is known as *runway allocation*, in order to balance both the landing/take-off on each runway and the controllers' workload. Airline preferences such as parking gate location, taxi time between the runway and the gate, and controller considerations such as safety, shorter flight times, and lower workloads can lead a controller to assign a new runway to an aircraft (Isaacson et al., 1997).

3.1.6 Separation

The prime responsibility of the air traffic controllers is the safety of the flights. Standard vertical and horizontal separations that keep aircraft from becoming dangerously close comprise one of the main ATC safety considerations. The usual minimum vertical separation between civilian aircraft operating in controlled airspace is 1000 ft. The horizontal separations between aircraft vary depending on the position, type and speed of the aircraft, and possibly other considerations.

One reason for setting minimum aircraft separation is to avoid the effect of vortices generated by the aircraft as a consequence of their lift. A Wake Vortex (WV) is potentially hazardous because of the rolling moment it can impose on a following aircraft. Generally, the WV separation between consecutive aircraft depends on the airspeeds, landing/take-off routes, and types of aircraft, and therefore it is sequence dependent. For example, heavier aircraft generate a greater WV and can tolerate more turbulent air. As a result, a light aircraft following a heavy aircraft requires a greater separation than when the following aircraft is also heavy. Consequently, effective scheduling will aim to avoid a light aircraft landing immediately after a heavy aircraft.

As a consequence of WVs, the International Civil Aviation Organization (ICAO) puts into force separation standards between the leader and the follower aircraft for approach, landing, and take-off to allow safe flight operations. The WV constraints govern the minimum permissible distance between aircraft lined up in sequence on the approach to land on the runway, and the delay before an aircraft can take off from the runway. The separation standard for landing is based on distance, while the take-off separation is based on time. If the scheduling procedure for landing requires a time-based separation instead of distance, the separation distances are typically converted into separation times using a fixed landing speed for the corresponding aircraft type (Beasley et al., 2001). It can be argued that the air speed is not fixed and depends on various factors such as aircraft type, flight level and weather conditions. The separation standard must satisfy the triangle inequality:

$$s_{ac} \leq s_{ab} + s_{bc}, \qquad \text{for all aircraft classes } a, b, c, \tag{3.1}$$

where s_{ab} is the WV separation between aircraft classes a and b (Balakrishnan and Chandran, 2006).

The simple ICAO's standard international classification of aircraft is based on three weight categories (Heavy, Medium and Light), and using distance separations that are integer (or half integer) numbers of nautical miles thereby making the air traffic controllers' job simpler at the expense of reducing capacity (Tether and Metcalfe, 2003). However, in the United Kingdom, the original ICAO three-group scheme has been modified to the five groups (Heavy, Upper-Medium, Lower-Medium, Small and Light) to provide more appropriate separations for certain aircraft types. When operating at peak capacity, the WV is often a major concern. It effectively determines runway capacity, and thus limits an airport's capacity in the terminal airspace. The large asymmetries in the minimum required separation can provide an opportunity to reduce airborne delays by shifting aircraft positions in the landing sequence. Further, adjusting the take-off sequence can similarly create additional capacity.

Departure routes also impose separation constraints. Aircraft take-off along specified predefined departure routes called *Standard Instrument Departure* (SID) routes. Aircraft following the same SID route must observe the SID separation. Finally, if two consecutive aircraft belong to different speed groups, then the separation may have to be modified depending on these speed groups.

3.1.7 Holding and manoeuvres

Controllers may make an aircraft wait (hold) before landing or take-off as a result of traffic congestion, poor visibility, weather conditions, occupancy of the runway, or missed time slots. Holding an aircraft is complicated because of the restriction imposed by a predefined flight plan, congestion, capacity of the holding area, and the dependency of the aircraft's speed on its type, weather conditions, and the altitude.

For departures, aircraft can be held at stands or at specific holding points. While airborne, an aircraft can be held using a number of techniques, namely vector-for-space (VFS), holding pattern (HP), detour, shortcut, or speed control. Figure 3.2 provides an illustration of the first four techniques. VFS and HP are the main holding procedures controllers use to manage the waiting process in the terminal area (Artiouchine et al., 2008). The VFS manoeuvre consists of a deviation of the aircraft away from its original flight path for a short time so that when it rejoins the flight path the time is later than without the deviation, whereas HPs generate a constant prescribed delay for an aircraft by flying in a loop (see Figure 3.2). Several HPs may exist in each terminal area and an aircraft can enter a holding pattern several times. A common HP near airports is known as a *holding stack*, where aircraft are instructed to join a waiting loop at different altitude levels above a feeder fix point (Bianco and Bielli, 1993). When an aircraft at the lower level is cleared to leave the hold, the other aircraft are laddered down. Controllers can also use various techniques such as detour (taking a longer route by deviating from the prescribed flight path as illustrated in Figure 3.2), shortcut (taking a more direct route, again by deviating from the prescribed flight path as illustrated in Figure 3.2)), and speed up or slow down (issuing an instruction for the pilot to accelerate or decelerate) to allow aircraft to land before or after their Estimated Landing Times (ELTs).

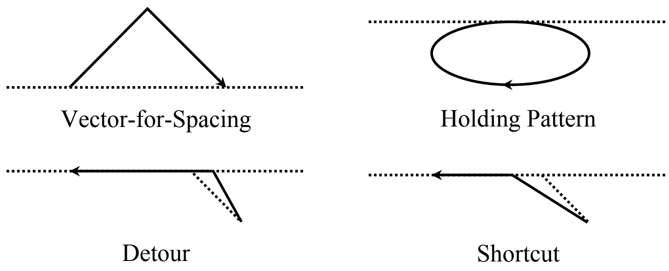

FIGURE 3.2: Holding and manoeuvring techniques

3.1.8 Position shifting

In practice, delaying or advancing an aircraft by a large number of places in the FCFS take off/landing sequence is undesirable because of the operating environment. Dear (1976) introduces *constrained position shifting* (CPS) for the ALP to limit the extent of deviations from the scheduling sequence. CPS defines the maximum number of positions an aircraft can shift in the landing sequence relative to the FCFS order. Specifically, *maximum position shifting* (MPS) uses an integer k to define the maximum deviation of the landing position of an aircraft in the selected landing sequence from its landing position in FCFS; this is also referred to as k-MPS. When k is small, an element of fairness among the aircraft is maintained by reducing the deviation from the FCFS sequence. A small value of k has the added advantage when solving the ALP or ATP that the size of the search space is reduced. As shown by de Neufville and Odoni (2003), the most undesirable landing sequences, such as those with a heavy aircraft followed by a light aircraft with an associated high separation, can be avoided, and delays can be significantly reduced, even with $k = 2$ or $k = 3$. Mesgarpour et al. (2010) consider *constrained time shifting* (CTS), which limits the deviation of landing time of an aircraft from that obtained using the FCFS sequence, where the limit can be dependent on aircraft type.

Re-sequencing becomes increasingly difficult as aircraft become closer to landing. *Relative position shifting* (RPS) takes this into account by defining the maximum number of position shifts (either backward or forward) of any aircraft in the sequence relative to the position that it occupies in FCFS. The maximum number of positions in RPS can be defined by the air traffic controllers for any subsequence of the landing/take-off sequence. Generally, this value decreases for subsequences near the beginning of the sequence and can be set to zero in the freezing stage (Bianco et al., 1997).

3.2 Airport Runway Scheduling

The aircraft landing/take-off problem is to sequence landing/take-off aircraft on/from the available runways at an airport and to assigned each aircraft a landing/take-off time subject to variety of operational constraints. The prime responsibilities of the air traffic controllers are safety of the flights and efficient planning of arriving and departing flights to and from the airport.

The aircraft landing problem (ALP) and aircraft take-off problem (ATP) models most commonly studied in the literature deal with the *static* (off-line) case, although some consider the *dynamic* (on-line or real-time) case (see Beasley et al., 2004; Moser and Hendtlass, 2007; Veidal, 2007). In the static version, the model is solved based on a given set of aircraft, where the complete information on these aircraft is assumed to be available and known. It can be argued that ignoring uncertainties leads to a reasonable approximation because information is fairly predictable close to the time of landing or take-off. Nevertheless, after solving the static problem, it is expected that these solutions are revised over time as new aircraft arrive into the system.

ALPs and ATPs differ in three important aspects. First, departure separation time minima depends on departure route and airspeed of leading and following aircraft as well as weight class of aircraft while the final approach separation distance minima depends on weight class of leading and following aircraft. Second, latest landing time constraints are hard constraints although latest take-off time constraints can be considered as soft constraints. Third, deviation of the landing sequence from FCFS sequence is limited by the approach operational constraints; however, deviation of the take-off sequence from FCFS sequence depends on configurations of taxi-out and departure holding area. Landing and departure scheduling problems are depend on separation which is the main similarity of ALPs and ATPs.

3.2.1 Objectives

Air transportation has a number of different stakeholders, including Air Traffic controllers (ATCs), airlines, airports, and government, who each have their own explicit or implicit objectives. As a result, the formulation of ALP and ATP involves the simultaneous optimization of a variety of objectives that may conflict, which is likely to lead the decision-maker into considering possible tradeoffs. The main (single) objectives are as follows (Idris et al., 1998a; Fahle et al., 2003; Lee and Balakrishnan, 2008).

(a) ATCs aim to ensure safety and efficiency of the aircraft. The following are desirable from an ATC perspective:

- maximizing the runway throughput
- minimizing the approach time of aircraft before landing
- minimizing air traffic controllers' workload
- maximizing fairness among the aircraft
- minimizing the aircraft taxi-in/taxi-out time
- minimizing the arrival/departure delay
- minimizing deviations from an appropriate balance between arrivals and departures.

(b) The airline's main objectives are:

- minimizing operating costs (especially fuel costs)
- minimizing engine run times before take-off
- maximizing punctuality with respect to landing/take-off time in published timetables
- minimizing total passenger delays
- maximizing adherence to airline priorities within their own flights
- maximizing the connectivity between incoming and outgoing flights.

c) The airport priorities are:

- maximizing punctuality relative to the operating schedule
- minimizing the need for gate changes due to delays.

d) The government preference is:

- minimizing environmental effects (noise and air pollution).

Typical scheduling objectives used in the literature, similar to those used in production scheduling, minimizes the average delay (average tardiness), average landing/take-off time (equivalent to average flow time or average completion time), and landing/take-off time of the last aircraft in the sequence (makespan). Delay is usually defined as the deviation of actual landing/take-off time from the estimated landing/take-off time calculated by the FCFS principle, and not based on the aircraft schedule. Delay is a service-based objective, whereas flow time, completion time and makespan are throughput-based objectives.

3.2.2 Modelling techniques

The literature provides a range of approaches to modelling and solving the ALP and ATP. We identify the core modelling approaches below, and we provide more details of some specific algorithms from the literature in Chapter 4.

As shown by Beasley et al. (2000) and Mesgarpour et al. (2010), the ALP and ATP can each be formulated as a mixed integer program (MIP). However, both problems are NP-hard and computation time to find an exact solution is likely to grow exponentially with the number of aircraft. As a result, solving the MIP is unattractive for real-time implementation on practical-sized instances.

Brentnall (2006) points out the relationship between the ALP and a machine scheduling problem with sequence-dependent setup times. The objective function can include makespan, and total earliness and tardiness by penalizing early and late jobs in terms of time windows or due dates. Regarding the specifics of the machine scheduling problem, each job corresponds to a landing operation of the aircraft; each machine with capacity one represents a runway; the ready time (release date) of the job corresponds to the estimated landing time (ELT) of the aircraft; the starting time of the job represents the actual landing time (ALT) of the aircraft; the completion time of the job corresponds to the time the aircraft frees the runway; and the sequence-dependent processing time between jobs represents the required separation between aircraft.

The travelling salesman problem (TSP) and the ALP also have resemblances. The classic TSP is to find a shortest tour that visits every city exactly once, starting and finishing at the same origin city (Laporte, 2010). The single-runway ALP closely resembles an open TSP with time windows, where the tour does not return to the origin. Each city corresponds to an aircraft, intercity distances represent the separations between aircraft, and the time windows for visiting each city are the landing time windows. The multiple-runway problem similarly resembles a multiple-TSP (Luenberger, 1998).

Finally, it is natural to consider the ALP/ATP as a queueing system (Pujet et al., 1999; Idris, 2001; Bauerle et al., 2007). Different classes of aircraft correspond to different customer types and the runways are servers. The service time of each customer (aircraft) is the separation time between the aircraft and its successor. Different queueing models can be used to represent the ALP/ATP problem depending on the number of runways available, the mode of operation at each runway (segregated or mixed), and the method of runway allocation by the controllers.

Applications of the operational research and management science techniques on airport runway scheduling in the literature are presented in the next Chapter.

Chapter 4

Literature Review

In this chapter, we review the main algorithmic contributions for scheduling aircraft landings and take-offs. The subsections are organized according to the main methodology used in the study. Unless stated otherwise, the problem considered is to schedule the landing/take-off of n aircraft each belongs to one of C classes, with separation times defined by the classes of the leader and follower aircraft. Section 4.1 and Section 4.2 review the literature about aircraft landing problem and aircraft take-off problem, respectively. Essential methods used in the literature are dynamic programming, branch-and-bound and genetic algorithm. The combined aircraft landing and take-off problem have been reviewed in Section 4.3. Section 4.4 includes some of our findings from the literature [1].

we would like to highlight some of our findings

4.1 The Aircraft Landing Problem

In this section, applications of various optimization methods such as dynamic programming, branch and bound, branch-and-price, genetic algorithm, ant colony optimization and queuing theory in scheduling of aircraft landings have been reviewed.

[1] The literature review has been published as an invited review paper in 4OR (see Bennell et al. (2011))

4.1.1 Dynamic programming

Dynamic Programming (DP) is a general optimization technique for making sequential decisions. Almost all ALPs can be usefully modelled as DP problems because the algorithms can evaluate current partial solutions independently of the exact sequencing decisions used to form these solutions. Beginning with the early work of Psaraftis (1978), there have been several attempts to develop efficient dynamic programming algorithm for the ALP. In many of these studies, it is assumed that all aircraft within any weight class can be sequenced. The aircraft landing problem then reduces to one of merging the individual sequences constructed for the different weight classes, with dynamic programming providing an effective approach for finding an optimal merging. The number of weight classes C is assumed to be fixed (in most practical applications, ranges between 3 and 5).

Psaraftis (1978, 1980) considers a simplified version of the ALP in which all aircraft are available to land immediately. As an objective function, he considers throughput as measured by minimizing $\mathrm{LT}_{\max}$, where $\mathrm{LT}_{\max} = \max_{j=1,\dots,n} \mathrm{LT}_j$, and delay as measured by the sum of landing times of the aircraft ($\sum_{j=1}^{n} \mathrm{LT}_j$). He develops backward dynamic programming algorithms that have as state variables the number of aircraft from each class that has not yet been scheduled to land and the class of the last aircraft to land. These algorithms essentially merge the lists of aircraft within the different classes. For the case of a single runway, the DP algorithm has a time complexity of $O(Cn^C)$, where C is the number of classes and n is the number of aircraft. For fixed C, this represents a polynomial time algorithm. Further, the DP algorithm can be adapted to handle CPS without increasing the time complexity. An extension of the algorithm to the case of two runways is also proposed.

Brentnall (2006) extends the dynamic programming approach of Psaraftis in two directions. First, he assumes that aircraft have earliest landing times. For some objective functions including the landing time of the last aircraft and the sum of landing times, he establishes that, within each class, aircraft should be sequenced in non-decreasing order of earliest landing time. Using these properties, he develops two forward DP algorithms: one minimizes the landing time of the last aircraft in $O(Cn^C)$ time, and the other minimizes the sum of landing times in $O(n^{C^2+C+1})$

time. For the second direction, he assumes that all aircraft are circling in several stacks, and the aircraft to land next is to be chosen as one at the bottom of one of these stacks. Using as state variables the number of aircraft from each stack that have been scheduled to land and the stack previously containing the last landed aircraft, forward DP algorithms requiring $O(n^{C^2+K})$ time are proposed for general objective functions, where K is the number of stacks. Brentnall and Cheng (2009) suggest four delay sharing strategies: all delay in hold, delay as late as possible, delay as early as possible, and delay evenly throughout the route. By linking the algorithms and methods to a discrete-event simulation and using several statistical methods, they analyze the output from simulations based on Stockholm Arlanda airport.

Bayen et al. (2004) propose a model that takes account of the time taken to complete a circuit in a holding stack. They assume that all aircraft belong to a single class. They develop a 5-approximation algorithm for the problem of minimizing the sum of landing time, and a 3-approximation algorithm for minimizing the landing time of the last aircraft. Their algorithms combine dynamic programming and the rounded solution of a linear program.

An alternative dynamic programming approach is introduced by Balakrishnan and Chandran (2006). They consider the problem of minimizing the landing time of the last aircraft, and impose CPS, precedence constraints between aircraft and arrival time-window constraints. The problem is formulated as a modified shortest path problem in a network with $O(n(2k+1)^{2k+2})$ arcs, where k is the maximum position shift (see Figure 4.1). In addition to a source node s and a terminal node t, the network consists of n stages, where each stage represents an aircraft position in the final sequence. A node at stage σ of the network corresponds to a subsequence of the aircraft of length $\min\{2k+1, n-\sigma+1\}$. If a node at stage σ can be followed by a node at stage $\sigma+1$, they are connected by a directed arc. The network shown in Figure 4.1 for $n = 5$ and $k = 1$ represents all the sequence combinations of possible aircraft assignments to each position. A pruned network, which is significantly smaller than the original network, can be produced by removing nodes which are not part of a path from source to sink (shown in grey) or which violate the precedence constraints.

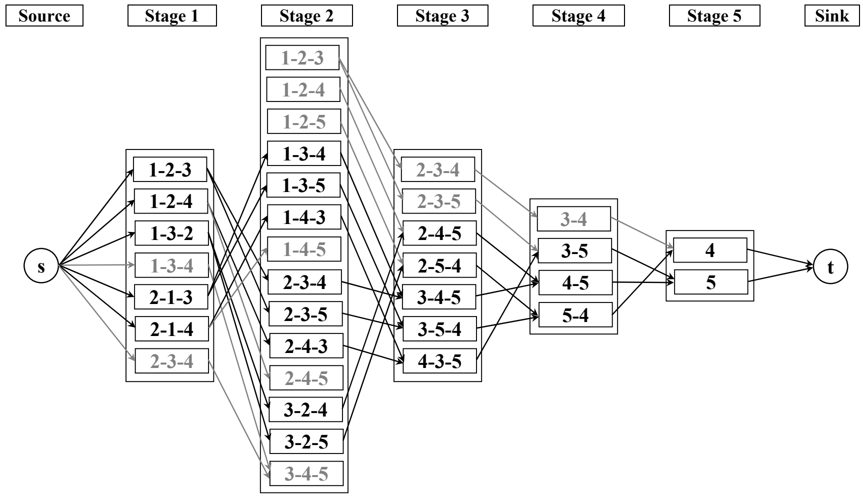

FIGURE 4.1: Pruned network for $n = 5$ and $k = 1$

Some notation is needed in our statement of the dynamic programming algorithm of Balakrishnan and Chandran (2006). Let $P(j)$ represent the set of predecessor nodes of each node j. Also, the earliest and latest landing times of the first aircraft in the sequence associated with node j are denoted by $e(j)$ and $l(j)$, respectively. The shortest path length in the network from node s to each node j of the network, which represents the landing time assigned to the first aircraft in the subsequence associated with node j, can be computed using the following forward dynamic programming recursion:

$$T(j) = \max\{e(j), \min_{i \in P(j): T(i) \leq l(i)} \{T(i) + s_{ij}\}\}.$$

Note that the density of the pruned network is significantly smaller than the worst-case complexity expression in practical instances. Also, since the basic network remains the same for any given n and k, it can be stored and retrieved when required. The computational experience of Balakrishnan and Chandran (2006) is based on an implementation of their algorithm on realistic data from the arrival flow at Denver International Airport. The algorithm exhibits small computation times for instances with up to 50 aircraft and with $k \leq 3$.

Building on the approach described above, Chandran and Balakrishnan (2007) introduce a dynamic programming algorithm to compute the tradeoff curve between the robustness and throughput. Robustness is interpreted as the probability of not violating any separation constraints. Their proposed algorithm is computationally efficient with a time complexity of $O(n(L/\epsilon)^3)$, where L is the largest difference between the latest and the earliest landing time over all aircraft, and ϵ is the interval used when discretizing time.

More recently, Lee and Balakrishnan (2008) extend the previous framework proposed by Balakrishnan and Chandran (2006) and Chandran and Balakrishnan (2007) by presenting a dynamic programming algorithm for minimizing the sum of landing costs of an arrival schedule. They use this approach firstly for minimizing total delay, which is equivalent to minimizing the sum of landing times, and secondly for minimizing fuel cost, where the strategy of speeding up some aircraft at the expense of burning extra fuel is explored. The study shows that speeding up to allow a landing up to 3 minutes earlier than normal is often advisable. By generating 1000 problem instances of 30-aircraft sequences using a Poisson distribution, the tradeoff between minimizing the average delay and maximizing the throughput as objectives, which are not necessarily aligned, is investigated. Results shows that the significant improvements in the average delay are achievable through decreasing the throughput so that the delay becomes relatively small.

4.1.2 Branch-and-bound

Brinton (1992) introduces one of the first branch-and-bound approaches for the ALP and the runway assignment problem. Static, dynamic, and depth limiting methods are used to reduce the number of tree branches that need to be searched. The objective function is a weighted combination of various costs although the proposed implicit enumeration algorithm does not depend on which costs are included. The methodology is the foundation of the Traffic Management Advisor (TMA) tool of the Center TRACON Automation System (CTAS) developed at NASA Ames Research Center. The implementation of runway and sequence optimization is discussed although there are no detailed computational results.

Abela et al. (1993) propose another branch-and-bound algorithm based on a 0-1 mixed integer programming formulation for the single-runway ALP. The objective function has a cost for each aircraft that is attributed to either speeding up or holding. Furthermore, they propose a genetic algorithm in which two heuristic operators, force feasible and squash, are used to ensure that the schedule after crossover satisfies the minimum separation times. Computational results for problem instances with up to 20 aircraft are presented.

Ernst et al. (1999) design a branch-and-bound algorithm and a local search heuristic for the ALP with single and multiple runways, where the objective function comprises penalty costs for landing before and after target times. A key component of their approach is a specialized simplex algorithm for determining a landing schedule, given on a partial order of the aircraft. They develop a heuristic-based problem space search which comprises their simplex algorithm, a constructive based heuristic to generate a good sequence, and a genetic algorithm to search the perturbation space. The heuristic algorithm and simplex method are used to obtain upper and lower bounds for the branch-and-bound algorithm. Various devices such as tightening intervals, upper-bound-based fixing, and fixing based on data are used as pre-processing methods to improve the performance of the branch-and-bound algorithm. An extended version of the algorithm can be used for multiple runways. The OR-Library data sets (Beasley, 1990) are used to evaluate the heuristic and exact algorithms on instances involving up to 50 aircraft on both a single runway and multiple runways.

In addition to providing an extensive literature overview on the ALP, Beasley et al. (2000) design branch-and-bound algorithms by employing linear programming (LP)-based tree search approaches for both single- and multiple-runway problems. They use the same objective function of Ernst et al. (1999) in which there are penalties for landing before and after target times. Their formulation of the problem is based on that introduced earlier by Abela et al. (1993). However, some additional constraints are proposed in order to reduce the zero-one space of the mixed integer formulation and strengthen the LP relaxation. The ALP is solved optimally for the problem instances found in the OR-Library (Beasley, 1990) involving up to 50 aircraft and four runways.

4.1.3 Branch-and-price

The multiple-runways version of the ALP is addressed by Wen (2005) and Wen et al. (2005) using a column generation approach. They also adopt the objective function of Ernst et al. (1999) in which there are penalties for landing before and after target times. The ALP is formulated as a set partitioning problem with side constraints. They develop and test a branch-and-price exact algorithm using the problem instances of Beasley (1990), which include 50 aircraft and four runways. Their computational results show that the linear relaxation of the set partitioning model provides a better lower bound than the linear relaxation of the mixed integer program. Further, the branch-and-price approach solves all instances while generating less than 450 columns and exploring no more than 12 nodes in the search tree.

4.1.4 Heuristics

Based on the observation that constrained position shifting can significantly reduce the number of candidate landing sequences, Dear and Sherif (1989, 1991) present an enumerative heuristic for the static and dynamic ALP. Computational results involving up to 500 aircraft and one runway show the smaller delays obtained under the heuristic than for a FCFS approach.

Bianco et al. (1999) suggest the ALP as an application of the single machine scheduling problem with release dates and sequence dependent processing times to minimize the sum of completion times (often referred to as problem $1|r_j\ seq-dep|\sum C_j$). They develop a dynamic programming formulation that is of theoretical interest because one of the state variables is the current set of scheduled jobs, and three lower bounds. Further, they propose two heuristic algorithms. The first is an $O(n^2 \log n)$ construction procedure that builds a schedule by adding jobs to the current partial sequence, and the second is based an $O(n^4)$ insertion approach. The effectiveness of the proposed heuristics is evaluated using randomly generated test instances, and two realistic ALP problem instances that include 30 and 44 commercial aircraft belonging to four weight classes. The schedules span a period lasting about 40 minutes, and are evaluated according to total aircraft landing time, and maximum and average landing delay. Because they do not consider CPS in their model, some aircraft are subject to excessive delay.

4.1.5 Genetic algorithms

Genetic algorithms (GAs) provide a popular approach for tackling sequencing problems, with many successful applications reported in the literature. Generally, a chromosome represents the order of the aircraft in a landing sequence. As an example of an alternative encoding, Beasley et al. (2001) use the assigned landing times of the aircraft.

Stevens (1995) provides one of the first and the simplest application of GA for minimizing the total penalty for landing before and after specified target times. Operational constraints include a constant three-minute separation time and an earliest landing time of three minutes before the target landing time. Results of computational tests are presented for ten problem instances involving two runways and up to 40 aircraft over a one-hour scheduling horizon.

Based on the GA of Stevens (1995), Ciesielski and Scerri (1997, 1998) compare two GA implementations for the dynamic/on-line ALP in terms of the percentage of valid solutions obtained and the best objective function value. The first algorithm builds the new schedule from scratch, and the second seeds it from the population left at the end of the last problem by removing landed aircraft and inserting new aircraft into the scheduling horizon. Their computational results are presented for two data sets involving 28 aircraft in a 37 minutes period and 29 aircraft in a 38 minutes period on two runways.

Cheng et al. (1999) design four different genetic schemes for the multiple-runway ALP. The first GA uses two chromosomes to encode the landing sequence and the runway assignment, respectively. In the second and third schemes, each chromosome forms a component of priority list for the flights. A fourth approach is based on genetic programming (GP) with chromosomes defined as mathematical operations and functions. They evaluate four approaches using one instance involving 12 aircraft and three runways. Hansen (2004) builds on the work of Cheng et al. (1999) by examining the efficiency and effectiveness of GA and GP methods in solving ALP. By applying four genetic search methods on four different test scenarios, he shows that the GP method provides the best solutions and these solutions can support controllers in real-time situations.

Beasley et al. (2001) develop a population heuristic for the static single-runway ALP with time-window restrictions. The algorithm aims to minimize the squared deviations from target landing times. Their computational results are for a single problem instance with five classes of aircraft, where the data for the instance are obtained from observations during a busy period at London Heathrow airport.

In a GA implementation of Capri and Ignaccolo (2004) for the static and dynamic ALP, three different objective function formulations are investigated for minimizing delay. Implementations to solve the static ALP consist of the GA, the GA with maximum landing time constraints, the cheapest insertion heuristic of Bianco et al. (1997), and FCFS. Results from four test problem cases proposed by Bianco et al. (1997) are used to compare the approaches. The performance of the dynamic model is evaluated using four test instances with up to 30 aircraft.

Pinol and Beasley (2006) implement two different population heuristics, scatter search and a bionomic algorithm, for the multiple-runway ALP. Solutions are represented by specifying the position of the landing time within a given time window and a runway assignment for each aircraft. Both a linear and a non-linear objective function are considered. The linear objective has penalties for landings both before and after a target landing time, while the non-linear objective has a positive quadratic penalty for landings after the target time and a negative quadratic penalty for landings before the target time. An infeasibility penalty for violations of the separation constraints between aircraft is considered separately. Computational tests with OR-Library data sets (Beasley, 1990) indicate that the relative effectiveness of the scatter search and bionomic algorithms depends on whether the linear or non-linear objective function is assumed.

Yu et al. (2009) use cellular automation (CA) to generate a landing sequence for the single-runway ALP. Improvements are made to the landing sequence using a GA with a relaxation operator. The instances of Beasley et al. (2000) are used to test the method. Bencheikh et al. (2009) propose a hybrid method to solve the ALP where ant colony optimization generates the initial population of feasible solutions for the GA. The ALP is formulated as a job shop scheduling problem with partial orders and alternative sequences through “and/or” graphs.

The dynamic ALP is studied by Hu and Chen (2005a,b), and we review their contributions below. In later work, Hu and Di Paolo (2008) introduce a new type of chromosome which defines a 0-1 value matrix based on neighbouring relationships between each pair of aircraft. The binary representation makes it easier to perform an efficient uniform crossover operator. The proposed GA based on a binary representation is compared with the GA with a permutation representation, as introduced by Hu and Chen (2005a) for the static and dynamic versions of the ALP. These authors also design a GA with uniform crossover for the multi-runway ALP using the successor relationship between aircraft to construct the chromosomes rather than the order of the aircraft in the queues (Hu and Di Paolo, 2009). They compare their method with the GP approach of Hansen (2004) and an extended version of the GA of Hu and Chen (2005a).

Hu and Chen (2005a,b) study GAs for the dynamic ALP using an approach based on receding horizon control (RHC). Figure 4.2 presents the comparison of the RHC with other optimization strategies (Hu and Chen, 2005b). The *off-line* strategy optimizes the static ALP for the entire time horizon. The *one-step-ahead adjustment* modifies the landing sequence for the current time interval given the static solution and current information. The *conventional dynamic optimization* and the *RHC* optimizes ALP over the horizon from the current time to the end of the time horizon or M time intervals ahead respectively, repeating the procedure at the beginning of each interval based on new information. For the GA of Hu and Chen (2005a), the aim is to minimize the airborne delay, which is the deviation of the actual landing time from the earliest landing time. The performance of their GA is compared with the approach of Bianco et al. (1997) and a GA that is based on conventional dynamic optimization (CDO) using test problem instances from Bianco et al. (1997). The performance of the GA based on RHC is evaluated further by Hu and Chen (2005b) under different levels of uncertainty and congestion.

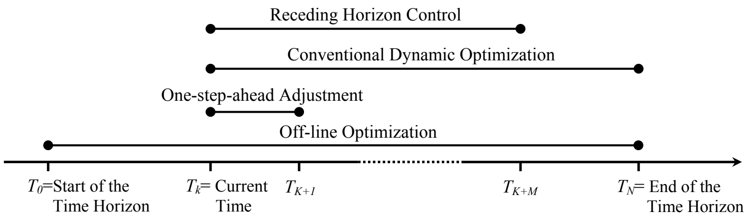

Figure 4.2: Different optimization strategies

4.1.6 Ant colony optimization

In addition to the use of ant colony optimization (ACO) by Bencheikh et al. (2009) in a hybrid method for generating initial solutions for the GA, Randall (2002) uses ACO to tackle the ALP. The objective is to minimize the total penalty associated with aircraft landing before and after specified target times as defined by Beasley et al. (2000). Moreover, there are time-window constraints. Six problem instances from Beasley (1990) are used for computational tests. Results show that the quality of the solutions is not as good as those obtained by Beasley et al. (2000).

4.1.7 Queueing theory

Bauerle et al. (2007) model the ALP as a special queueing system with the incoming aircraft corresponding to customers of different types and separation times between aircraft corresponding to customer service times. The single-runway problem is modelled as an M/SM/1 queueing system, with semi-Markov service times. In addition, they derive the stability condition and the average waiting time. Several routing heuristic strategies are studied and compared with respect to the average delay for assigning aircraft to two runways.

4.1.8 Comparative studies

There exists work on comparing different algorithms proposed for solving the ALP. Fahle et al. (2003) consider the simplest model of single-runway ALP with time-window and separation constraints in a static environment. They compare four exact methods. The first two are proposed by Beasley et al. (2000): a mixed integer programming model that uses continuous time, and an integer programming formulation that is modelled using discrete time. The other two methods consist of a constraint programming model, and a satisfiability problem formulation. In addition to these exact methods, they implement two local search heuristics based on descent and simulated annealing. These six methods are evaluated using three problem instances with up to 123 aircraft.

In another study, Beasley et al. (2004) define a displacement problem, which is used in the solution of the dynamic ALP. The rationale of this approach is that unfavourable deviations from the previous solution are penalized in the displacement problem when new information becomes available and the solution is updated. Possible solution approaches for the displacement problem are the LP-based tree search (Beasley et al., 2000), a heuristic algorithm (Beasley et al., 2000), and a population heuristic (Beasley et al., 2001). Their computational results are presented for two sets of test problems involving up to 500 aircraft and five runways.

4.2 The Aircraft Take-Off Problem

The ALP has attracted much greater research interest compared to the ATP for which studies are quite scarce. The main reason is that take-off scheduling problem is highly correlated with taxi-out scheduling problem and they cannot be solved separately. Integration of these two sub-problems makes the problem complex and difficult to solve. However, landing scheduling problem and taxi-in scheduling problem have little correlation and can be dealt with separately.

Idris et al. (1998a,b) study the flow constraints and the dynamics of airport systems. Specifically, they analyze the departure flow at Logan airport as a complex queuing system. As aircraft compete for limited resources such as gates, ramp, taxiways and runways, queues are created in various parts of the airport. They conclude that the runway is the main constraint. Pujet et al. (1999) develop an alternative queuing model of the departure system. Their model is evaluated using the runway configuration and traffic data. The intention is to relieve the departure traffic congestion on the ground.

4.2.1 Dynamic programming

Craig et al. (2001) propose a dynamic programming algorithm for sequencing the take-off of aircraft at one of the simplified holding points at London Heathrow airport. Some possible strategies for sequencing aircraft at the stands are also considered in their research. Based on their model for the ALP (Balakrishnan and Chandran, 2006), Balakrishnan and Chandran (2007) introduce a dynamic programming algorithm for the ATP. Their approach is also extended for multiple runways and active runway crossing.

4.2.2 Heuristics

Anagnostakis and Clarke (2002, 2003) investigate a two-stage heuristic algorithm for solving a runway operation planning problem. The first stage aims to maximize the throughput by generating candidate sequences of classes of aircraft, while ignoring the operational constraints. The second stage uses an integer programming model to assign aircraft to class slots in one of the sequences generated in the first phase taking into account the relevant constraints (see also Anagnostakis, 2004).

4.2.3 Metaheuristics

Atkin et al. (2004) consider an initial model of the simplified ATP at London Heathrow airport including the holding point structure. Based on the results of comparing different search heuristics, their conclusion is that tabu search performs better than simulated annealing and descent algorithms. A later and more detailed study of Atkin et al. (2006) investigates the effects of the different constraints on scheduling. A key aspect of their algorithms involves the checking of candidate take-off sequences for feasibility, taking into account the current positions of aircraft at the holding points.

Atkin et al. (2007) propose a hybrid approach that uses different search methodologies and a heuristic method to solve the static version of the ATP. Their objective function comprises a weighted sum of delay, a reordering cost (for an aircraft that is moved later in the take-off sequence than in FCFS, the cost is proportional to the number of positions moved), a non-linear cost for violation of a CTOT time window or for scheduling a take-off near to a boundary of the window, and an additional penalty cost for schedules that introduce an excessive delay on aircraft. The model is evaluated using six sets of data from London Heathrow airport. Result shows that the availability of more information about the aircraft taxiing can reduce delays and improve CTOT compliance. In a further study, Atkin et al. (2008) provide some further enhancements to their previous work (see also Atkin, 2008).

Based on Dallas Fort Worth airport, Stiverson (2009) designs a greedy heuristic and a 2-interchange heuristic for departure sequencing. Lower bounds on an optimal solution are also provided using a mixed-integer linear programming model. Performance of the heuristics has been tested using randomly generated datasets. Total delay and take-off time of the last aircraft are optimized subject to the arrival time of the aircraft to the runway, separation and runway layout. To simplify the problem, it is assumed that the possible taxi-out routes are limited and that there is no upper bound on the number of aircraft that can occupy a queue, crossing or taxiway.

4.2.4 Constraint satisfaction

Using a constraint satisfaction approach for the ATP and the ILOG solver, van Leeuwen et al. (2002) map flights onto activities, and model the taxiways, runways, and exit points of an airport as resources. Also, different type of constraints such as take-off order, time-slot, and separation are listed as temporal or resource constraints in the ILOG environment. Results of applying the model to real data from Prague airport for up to 12 aircraft in a 50-minute time interval are also presented. As the problem size gets larger, the model fails to find a solution in reasonable time. van Leeuwen and van Hanxleden Houwert (2003) introduce constraint relaxation techniques to overcome the highly complex or conflicting requirements that have to be considered in practice. The constraints are divided into different sets of soft and hard constraints according to whether they are candidates for relaxation. Additional controller-imposed constraints, time-slot constraints, and runway preferences are considered as soft constraints.

4.3 Combined aircraft landing and take-off problem

Trivizas (1998) introduces a dynamic programming approach for solving optimally the static runway scheduling problem for landings and take-offs based on the CPS concept. The mixed-mode, segregated-mode, and multiple-runway environments are considered. His computational results obtained with actual traffic data and a real airport configuration show that even a modest value such as a maximum position shift of three can increase the runway capacity up to 20% compared to FCFS sequencing.

Bianco et al. (2006) introduce static and dynamic models for scheduling the landing and take-off of aircraft in the terminal manoeuvring area (TMA). The proposed deterministic job shop scheduling model can represent several operational constraints and different runway configurations. The model considers the runway, TMA, inbound and outbound flight paths, holding stacks for landing and holding points for take-off. The solution method is based on a fast descent heuristic. Experimental results using real data of Milan Malpensa and Rome Fiumicino airports show that the average delay can be reduced by more than 40% and the TMA capacity may increase up to 30% in comparison with FCFS sequencing.

4.4 Remarks

Predictions for increasing air traffic over the next 15 years puts pressure on air navigation service providers around the world to improve safety levels, reduce delays, and cut the costs. This is the motivation behind the SESAR (Single European Sky ATM Research) and NextGen (Next Generation Air Transportation System) programs. SESAR is a European air traffic control infrastructure modernization program that aims to eliminate the fragmented approach in European air traffic management, to transform its system, to synchronize all stakeholders, and to federate resources (EUROCONTROL, 2012). NextGen is the transformation of the entire air transportation system through the use of twenty first century technology to support the current and future demand for aviation services in the

United States (FAA, 2012). After reviewing previous research studies on ALP and ATP and observing air traffic controllers in a working environment, we would like to highlight some of our findings. In our research, we aim to fill some of these gaps.

Practical vs theoretical models

As explained before, many theoretical studies may show an increase in utilization of the runway capacity, but it may not be possible to implement the models in practice. Often, some critical operational constraints in the modelling are ignored, some of the hard constraints in obtaining a solution are relaxed, or required computational resources are unreasonable.

Quick and good vs slow and optimal solution methods

In real situations, controllers can only use algorithms which can quickly (in a matter of seconds) find a good solution (near-optimal). Optimal solutions arising from lengthy computation times are of little practical use.

Defining the objective functions and constraints

Choosing an appropriate objective function for the ALP/ATP is controversial and stakeholders (air traffic control, airports, airlines, and government) may have conflicting criteria. Thus, selecting one or more objective that can satisfy the interests of all parties, or provide an acceptable compromise, is an important first step towards the model to be implemented.

Robustness and flexibility

There are different levels of uncertainty associated with the information considered within an ATP/ALP, especially in a dynamic environment. The uncertainty can be caused by weather conditions such as winds and snow, the precision of equipments, as well as the uncertainty in pushback times and taxi times for departing aircraft. However, most studies consider a static rather than a more realistic dynamic environment.

Increasing the number of separation categories

Currently, ICAO classifies aircraft into three categories of Heavy, Medium and Light. Since wake vortex separation is a primary constraint on runway throughput, refining the classification into more classes may increase runway capacity.

Integrated models

There are several models that can relatively solve problems involving individual components of airport operations effectively. However, a major challenge is to form an integrated model. Possible types of integration include integrating runway scheduling, ground movement control, and gate assignment. Another example is the scheduling of an aircraft's take-off and landing at the same time which requires runways at several airports to be scheduled simultaneously.

Throughput is the primary objective for ATC

The literature considers many different objective function criteria, whereas in general controllers are only concerned with throughput after safety considerations are taken into account. In order to balance other criteria, controllers need more information and good decision support tools to use this information.

Availability of information in advance

The accuracy and timeliness of information can improve decision making. One of the purposes of the Collaborative Decision Making (CDM) approach for airports is to provide relevant information to all parties (airport, airlines, and ATC) in advance. This helps controllers to schedule landings and take-offs with better insight into the future state of the system.

US vs Europe

There are greater research activities in airport runway scheduling in the US compared to Europe. The difference in the type of research on the ALP and ATP in the US and Europe indicate that joint research projects would provide a good opportunity for both communities to enhance their models and further develop their solution algorithms.

In the next chapter, the ALP has been defined in details and different solution methods for scheduling arrival flights to the airport have been discussed.

Chapter 5

Aircraft Landing Problem

This chapter describes the development of our solution algorithms for the static and dynamic aircraft landing problem. Our goal is to design algorithms that run in under five seconds and preferably provide solutions in less than one second. So that they can be implemented in real life as a decision support tool for the controllers. In the following subsections, we present various search algorithms for solving the static problem. These algorithms provide the core search mechanism for tackling the dynamic problem. We describe the solution procedure for the dynamic problem in the final subsection.

By its nature, there is no notion of an optimal solution for the dynamic problem because not all of the information is available when decisions start to be made. On this basis, it is not necessary to design algorithms for the static problem that guarantee optimal solutions. Instead, we are content with heuristics that provide good quality solutions at modest computational expense.

Problem has been briefly defined in Section 5.1. Landing time constraints and separation constraints have been explained in Section 5.2. Algorithms for the static problem are presented in Section 5.3. These algorithms can be regarded as building blocks because they are embedded within our proposed algorithms for the dynamic problem. The dynamic algorithms are presented in Section 5.4. Section 5.5 presents the computational experience of algorithms for static and dynamic environments. Concluding remarks have been discussed in Section 5.6

5.1 Problem definition

In our model there is a single runway that is used solely for landings and another runway for take-offs. This situation is common, although there are airports such as London Gatwick airport where both take-offs and landings are scheduled on a single runway. Associated with any schedule are landing times.

The model that is developed below contains some features associated with coordinated planning involving the various stakeholders that might be applicable in the future. In spite of the model's generality, a suitable choice of parameters makes it compatible with the criteria upon which ATCs make their decisions in a current day setting. Generally, the ALP is to sequence landing aircraft onto the available runways at an airport and to assign each aircraft a landing time, subject to a variety of operational constraints.

In the static/off-line version of the aircraft landing problem, there are n aircraft with landings to be scheduled. All data concerning these aircraft are known in advance of any decisions being made. Associated with any schedule are landing times. Specifically, in any schedule, let LT_j denote the landing time of aircraft $j \in A$, where $A = \{1, \ldots, n\}$.

However. in the dynamic/on-line version of the landing problem, aircraft arrive into an ATC's airspace over time. In practice, controllers have knowledge of an aircraft between 30 and 40 minutes before it can reach the runway. The number of aircraft is not known in advance. Further, no information is available to controllers about aircraft that have yet to arrive into their airspace. Thus, scheduling decisions have to be taken on the basis of partial data.

The model formulation that follows attempts to include an element of the type of coordinated planning to be used in the future by considering the interests of the various stakeholders. However, these days, ATCs usually schedule landings to minimize separation times between aircraft, subject to meeting safety requirements. In spite of our model's broader remit, a suitable choice of parameters maintains compatibility with the criteria upon which ATCs make their decisions in a current-day setting.

5.2 Constraints

The constraints on the aircraft landing problem are divided into two main types. There are constraints on the time that an aircraft can land, and constraints on the separation time between landings.

5.2.1 Landing time constraints

There are various constraints on the landing time of each aircraft j, for $j = 1, \ldots, n$, that take the form of time windows. First, LT_j should lie within a time window $[\mathrm{elt}_j, \mathrm{llt}_j]$, where elt_j and llt_j are the earliest and latest landing times of aircraft j. Typically, the earliest landing time is the time aircraft j takes to fly from its current location to the runway at a maximum safe speed. The latest landing time is usually the maximum possible flight time based on the fuel carried by the aircraft, although there could be reasons why an airport or airline could stipulate a smaller value of the latest landing time. Second, LT_j should lie within a time window centred around the unconstrained landing time, ult_j, of aircraft j. The value of ult_j is the time that aircraft j would be expected to land when there are no other aircraft to impede its progress to the runway. It is determined by the arrival planner system after the aircraft enters the range of the relevant radar. Aircraft j is assumed not to land before ult_j, but may land up to a maximum time shift ts_j after ult_j, which means that LT_j should lie within the time window $[\mathrm{ult}_j, \mathrm{ult}_j + \mathrm{ts}_j]$. The rationale for such a bound on the time shift is partly fairness so that no aircraft is delayed by an excessively long time, and partly workload reduction on ATCs.

The two time windows defined by the earliest/latest landing times and the deviations from the ult_j can be combined. This provides a constraint of the form

$$e_j \leq \mathrm{LT}_j \leq l_j \qquad \text{for } j = 1, \ldots, n, \tag{5.1}$$

where $e_j = \max\{\mathrm{elt}_j, \mathrm{ult}_j\}$ and $l_j = \min\{\mathrm{llt}_j, \mathrm{ult}_j + \mathrm{ts}_j\}$.

An aircraft j may also have an associated preferred landing time plt_j. The preferred landing time may be based on the aircraft's flight plan, the airlines timetable, or a time used by the airport in their plans for assigning a gate to the aircraft or for the baggage to be unloaded. However, we view the preferred landing time as a soft constraint that we address when considering the objective function.

5.2.2 Separation time constraints

Associated with each aircraft is a weight class that determines the minimum separation times between successive landings. Let C denote the number of classes. Also, let s_{bc} be the minimum separation time when an aircraft of class b lands before an aircraft of type c, for b, $c = 1, \ldots, C$. We assume that the separation times satisfy the triangle inequality so that for any aircraft of types a, b and c we have $s_{ab} + s_{bc} \geq s_{ac}$. This implies that it is sufficient to impose the separation time constraints only between successive pairs of aircraft in the landing sequence.

Due to the importance of the separation time constraints, it is sometimes convenient to use double indices for the aircraft. For any weight class c, let n_c denote the number of aircraft in this class, where $n = \sum_{c=1}^{C} n_c$. We then refer to the aircraft in each weight class c as $(1, c), \ldots, (n_c, c)$. Because an aircraft (i, b) lands either before or after any other aircraft (j, c), we obtain a separation constraint

$$\mathrm{LT}_{i,b} + s_{bc} \leq \mathrm{LT}_{j,c} \quad \text{or} \quad \mathrm{LT}_{j,c} + s_{cb} \leq \mathrm{LT}_{i,b} \tag{5.2}$$

for each pair of aircraft (i, b) and (j, c).

Note that there may be precedence constraints specifying that one aircraft must be placed before another in the landing sequence. Thus, if aircraft (i, b) must land before aircraft (j, c) according to the precedence constraints, then constraint (5.2) is replaced by $\mathrm{LT}_{i,b} + s_{bc} \leq \mathrm{LT}_{j,c}$.

5.2.3 Objective function

As previously discussed, the ALP involves a number of stakeholders with various priorities. As a result, adopting a multi-objective approach is appropriate.

The main objective of ATCs after taking into account safety is to maximize runway throughput. This naturally translates into minimizing the landing time of the last aircraft in the schedule, or minimize $\mathrm{LT}_{\max}$, where $\mathrm{LT}_{\max} = \max_{j=1,\ldots,n} \mathrm{LT}_j$. However, in a more realistic dynamic scheduling environment, there is a high likelihood that the latter part of the schedule will change due to new aircraft arriving, with the result that only the initial part of the landing schedule is implemented. Therefore, focusing only or mainly on the landing time of the last aircraft may create schedules that are less suitable when used for scheduling within a dynamic environment. Thus, we also consider the minimization of the average landing time

$$\mathrm{ALT} = \sum_{j=1}^{n} \mathrm{LT}_j / n, \tag{5.3}$$

which aims to reduce each of the landing times rather than just the last. Thus, the overall contribution to the objective function of our runway throughput measure is

$$w_1 \mathrm{LT}_{\max} + w_2 \mathrm{ALT}, \tag{5.4}$$

where w_1 and w_2 are suitably chosen non-negative weights for the maximum and average landing time, respectively.

The notion of a preferred landing time is introduced in Section 5.2.1. For each aircraft j, we define a time window $[\mathrm{plt}_j - \delta_j^e, \mathrm{plt}_j + \delta_j^l]$ within which the aircraft should ideally land, were δ_j^e and δ_j^l define allowable tolerances for earliness and lateness, respectively. If $\mathrm{LT}_j < \mathrm{plt}_j - \delta_j^e$, then there is an earliness penalty $u_j^e(\mathrm{plt}_j - \delta_j^e - \mathrm{LT}_j)$, where u_j^e is a penalty per unit of earliness with respect to the left-hand end of the time window. Similarly, if $\mathrm{LT}_j > \mathrm{plt}_j + \delta_j^l$, then there is a lateness penalty $u_j^l(\mathrm{LT}_j - \mathrm{plt}_j + \delta_j^l)$, where u_j^l is a penalty per unit of lateness with respect to the right-hand end of the time window. Generally, we would expect the model parameters to be chosen so that $u_j^l \geq u_j^e$, because lateness usually causes greater disruption then earliness. Thus, the overall penalty for violation of the time windows defined for preferred landing times is

$$\mathrm{TW} = \sum_{j=1}^{n} u_j^e \max\{\mathrm{plt}_j - \delta_j^e - \mathrm{LT}_j, 0\} + \sum_{j=1}^{n} u_j^l \max\{\mathrm{LT}_j - \mathrm{plt}_j + \delta_j^l, 0\}. \tag{5.5}$$

Lastly, the cost of using more fuel than necessary for a flight is a concern for airlines, and moreover a reduction in fuel burn is helpful in reaching government targets on CO2 emissions. Thus, another objective is the minimization of the additional fuel used to achieve a landing schedule. As a baseline, a landing time of ult_j is assumed for each aircraft j. Any later landing for aircraft j, which is defined by $\mathrm{LT}_j > \mathrm{ult}_j$, causes the aircraft to use more fuel due to being airborne for longer and also possibly through some manoeuvres requested by the ATC to delay its landing time. Recall that we do not allow any aircraft j to land before ult_j. If the permission would be given to the aircraft j to land before ult_j, either it could save fuel by taking advantage of shortening the flying route and taking the shortcut or it could cause extra fuel burn because of the increasing airspeed. If v_j^l denotes the cost per unit time of the extra fuel associated with lateness relative to ult_j, then the overall extra fuel cost is

$$\mathrm{EF} = \sum_{j=1}^{n} v_j^l \max\{\mathrm{LT}_j - \mathrm{ult}_j, 0\}. \tag{5.6}$$

Since the ALP may involve the simultaneous optimization of various dependent objectives that are not necessarily aligned, a trade-off among the objectives is required. Therefore, they need to be optimized in the form of a weighted multi-criteria objective function. Using suitable weights, we can combine the different objectives defined in (5.4), (5.5) and (5.6) to give the overall objective function

$$w_1 \mathrm{LT}_{\max} + w_2 \mathrm{ALT} + w_3 \mathrm{TWF} + w_4 \mathrm{EF}, \tag{5.7}$$

for suitably chosen non-negative weights w_3 and w_4 (as well as w_1 and w_2). This expression is to be minimized, subject to constraints (5.1) and (5.2).

Based on equation (5.7), the incremental cost of aircraft j landing at time t is given by

$$\begin{aligned} g_{j,t} \; = \; & w_2 t/n + w_3(u_j^e \max\{\mathrm{plt}_j - \delta_j^e - t, 0\} + u_j^l \max\{t - \mathrm{plt}_j - \delta_j^l, 0\}) \\ & + w_4(v_j^l \max\{t - \mathrm{ult}_j, 0\}). \end{aligned} \tag{5.8}$$

5.2.4 Assumptions

The decision variables in our model are the landing time variables LT_j for $j = 1, \ldots, n$. We assume that any selection of landing times that is chosen to satisfy (5.1) and (5.2) define a feasible solution.

One aspect of feasibility that we do not consider is runway occupancy by a landing aircraft. Suppose that the aircraft landing immediately before (j, c) is (i, b). According to constraint (5.2), aircraft (j, c) could land as early as $\mathrm{LT}_{ib} + s_{bc}$. Our model assumes that aircraft (i, b) has left the runway by this time. Thus, we do not model the blocking of the runway by any aircraft that has already landed or by any aircraft that is taxiing.

Another operational issue that does not appear in our model concerns the manoeuvres required by aircraft to achieve those landing times that correspond to the values of the decision variables. We aim to avoid the need for excessive resequencing of aircraft by imposing constraint set (5.1). On this basis, our assumption is that ATCs can achieve the desired landing times by using relevant techniques (as pointed out in 3.1.7), vectoring, detour and shortcut are used by ATCs to position aircraft according to the desired landing sequence).

5.2.5 The dynamic problem

The above formulation holds for the static problem with n chosen as the total number of aircraft, and for the dynamic problem with n chosen as the subset of aircraft available to the ATC for scheduling at a particular time. In the dynamic aircraft landing problem, aircraft are scheduled for landing using a rolling horizon approach. This means that every τ units of time, for some suitable chosen time interval τ, the previously created (provisional) schedule is updated to include new aircraft entering the system by appearing on the ATC's radar screen, and aircraft at the beginning of the schedule that land and therefore leave the system. Some aircraft that are sufficiently close to the start of the schedule cannot be rescheduled for safety reasons. Further, the likelihood of an aircraft being rescheduled reduces as it gets closer to landing. This is because any new aircraft entering the system are too far away to have a significant influence on their landing times.

We refer to τ as the *update time*. Typically, τ may be approximately five minutes. Too small a value of τ would result in too frequent updates to the schedule, possibly with only one or two additional aircraft in the system. On the other hand, if τ is too large, some of the opportunities for manoeuvres to create better landing schedules may be lost. We investigate different values of τ in our computational experiments.

5.3 Algorithms for static problem

5.3.1 Mixed Integer Programming Model

The mixed-integer programming (MIP) model has been presented in this section which is based on the MIP model introduced by Beasley et al. (2000). Since the problem is NP-hard solving instances of practical size is time-consuming. Proposed MIP model has two decision variables. Decision variable LT_j describes the scheduled landing time of aircraft j. Decision variable x_{ij} is defined to be 1 if aircraft i lands (not necessarily immediately) before aircraft j, and 0 otherwise. Moreover, parameter p_{ij} is denoted to be 1 if aircraft i must land (not necessarily immediately) before aircraft j, and 0 otherwise. M is also a big enough positive number. The MIP model is given as follow.

$$\text{Minimize} \quad w_1\mathrm{LT}_{\max} + w_2 \sum_{j=1}^{n} \mathrm{LT}_j/n + w_3\mathrm{TW} + w_4\mathrm{EF}. \tag{5.9}$$

subject to

$$x_{ij} + x_{ji} = 1 \quad \forall i, j \in A, \quad i \neq j. \tag{5.10}$$

$$\mathrm{LT}_{i,b} + s_{bc} \leq \mathrm{LT}_{j,c} + M(1 - x_{ij}) \quad \forall i, j \in A \quad i \neq j \quad b, c \in C \tag{5.11}$$

$$\mathrm{elt}_j \leq \mathrm{LT}_j \leq \mathrm{llt}_j \quad \forall j \in A \tag{5.12}$$

$$\mathrm{ult}_j - \mathrm{ts}_j \leq \mathrm{LT}_j \leq \mathrm{ult}_j + \mathrm{ts}_j \quad \forall j \in A \tag{5.13}$$

$$\mathrm{LT}_i p_{ij} < \mathrm{LT}_j \quad \forall i, j \in A, \quad i \neq j \tag{5.14}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in A \tag{5.15}$$

$$\mathrm{LT}_j \geq 0 \quad \forall j \in A. \tag{5.16}$$

In this proposed MIP model, there are $n^2 + n$ decision variables and $3n^2 + 2n$ problem constraints, where n denotes the number of aircraft. If precedence constraints are not considered, the number of constraints reduce to $2n^2 + 3n$.

The objective function (5.9) is to minimize the sum of weighted multi-objective including landing time of the last aircraft, average landing time (5.3), violation of time window (delay) (5.5) and extra fuel burn associated to earliness and lateness (5.6). Moreover, weights associated to each objective are

$$0 \leq w_k \leq 1 \quad \text{and} \quad \sum_{k=1}^{4} w_k = 1. \tag{5.17}$$

Constraint (5.10) specifies that the runway can be used by at most one aircraft at a time, so either aircraft i lands before j or vice versa. Minimum separation distance between landing aircraft is defined in constraint (5.11) to avoid turbulence caused by preceding aircraft. Earliest/latest landing time window and time shifting window are presented by constraints (5.12) and (5.13), respectively. It has to be mentioned that a time slot (time window) assigned to each landing aircraft which typically starts 5 minutes before plt_j and ends 10 minutes after plt_j does not necessarily coincide with the earliest/latest landing time window. Constraint (5.14) is the precedence constraint which shows airlines or controllers preferences on aircraft landing order. Constraints (5.15) and (5.16) guarantee that the x_{ij} are binary and landing time remains positive.

5.3.2 FCFS

In FCFS, the aircraft are sequenced in non-decreasing order of their unconstrained landing times. Thus, the landing sequence σ is chosen so that $\text{ult}_{\sigma(1)} \leq \cdots \leq \text{ult}_{\sigma(n)}$. The landing sequence effectively defines precedences between aircraft that land in succession. Thus, the actual (smallest) landing times are determined by applying constraints (5.1) and (5.2) in a straightforward way.

5.3.3 Dynamic programming

Brentnall (2006) provides several DP algorithms for sequencing of aircraft landings. He shows that the LT_{max} of the landing sequence can be optimized if aircraft of the same weight class ordered by unconstrained landing time. Moreover, he shows that an optimal landing sequence for the total lateness can be obtained if $\text{ult}_j = \text{plt}_j$ and aircraft of the same weight class is ordered by unconstrained landing time. As mentioned in Section 2.2.2, the main drawback of classical DP is that the number of states often grows exponentially by increasing the number of aircraft in ALP because of the curse of dimensionality.

Our dynamic programming algorithm assumes that, within each weight class, the aircraft are ordered in non-decreasing order of their unconstrained landing times. We index the aircraft accordingly, so that $\text{ult}_{1,c} \leq \cdots \leq \text{ult}_{n_c,c}$, for $c = 1, \ldots, C$. Our dynamic programming algorithm merges the C streams of pre-ordered aircraft $(1, c), \ldots, (n_c, c)$ for $c = 1, \ldots, C$. This DP approach does not necessarily reach the optimal solution since our objective function and operational constraints are not the same as problems considered by Brentnall (2006).

The dynamic program has state variables $(m_1, \ldots, m_C, c, t)$. This state corresponds to a landing schedule of aircraft $(1, b), \ldots, (m_b, b)$ for $b = 1, \ldots, C$, where $1 \leq m_b \leq n_b$, with t representing the scheduled landing time of aircraft (m_c, c) which has the last of the scheduled landing times. Let $f(m_1, \ldots, m_C, c, t)$ denote the minimum total cost among partial landing schedules corresponding to state $(m_1, \ldots, m_C, c, t)$. It has to be mentioned that state variable t has not been considered in the DP method proposed by Brentnall (2006).

Algorithm DP

Initialization

Set $k = 1$, and

$$f(1, \ldots, 0, 1, e_{(1,1)}) = g_{(1,1),e_{(1,1)}}$$
$$\vdots$$
$$f(0, \ldots, 1, C, e_{(1,C)}) = g_{(1,C),e_{(1,C)}}$$

where the function g is defined in equation (5.8).

Next Stage Generation

For each state $(m_1, \ldots, m_C, c, t)$ such that $\sum_{b=1}^{C} m_b = k$ and each b such that $m_b < n_b$, generate the new state $(m_1, \ldots, m_{b-1}, m_b + 1, m_{b+1}, \ldots, m_C, b, t')$, where $t' = \max\{e_{(m_b+1,b)}, t+s_{cb}\})$, together with its associated value $f(m_1, \ldots, m_C, c, t)+ g((m_b + 1, b), t')$, where $g((m_b + 1, b), t')$ is computed from equation (5.8).

Next Stage Elimination

If any state $(m'_1, \ldots, m'_C, b, t')$ is created more than once in the Next Stage Generation Step, select the one with the smallest value associated to V and set $f(m'_1, \ldots, m'_C, b, t') = V$. If $k < \sum_{b=1}^{C} n_b$, then set $k = k + 1$ and return to the Next Stage Generation step.

Select Solution

Among all states $(n_1, \ldots, n_C, b, t')$ for $b = 1, \ldots, C$ and all t', select the one with the smallest value of $w_1 t' + g_{(n_b,b),t'}$.

If the landing time of the last aircraft has many potential values in the partial schedules, then the Next Stage Elimination may not remove many states, and consequently the dynamic programming algorithm has a similar performance to that of using complete enumeration to find an optimal merging.

5.3.4 Iterated descent

We first describe a descent algorithm that provides the basic building block for our iterated descent method. Solutions are represented as a landing sequences of aircraft. Thus, each solution is defined by some aircraft sequence $\sigma = (\sigma(1), \ldots, \sigma(n))$. We use a combined *insert*, *swap* and *2-insert* neighbourhood. Soomer (2009) also introduces a local search heuristic approach using insert and swap neighbourhoods to maximize the fairness in the aircraft landing problem. The insert neighbourhood comprises all sequences that can be obtained from the current sequence by removing an aircraft from its current position and inserting it into a new position in the sequence. Thus, for $1 \leq h < i < j \leq n$, two insert neighbours of σ are

$$(\sigma(1), \ldots, \sigma(h), \sigma(i), \sigma(h+1), \ldots, \sigma(i-1), \sigma(i+1), \ldots, \sigma(n))$$

$$(\sigma(1), \ldots, \sigma(i-1), \sigma(i+1), \ldots, \sigma(j), \sigma(i), \sigma(j+1), \ldots, \sigma(n)).$$

Further, the swap neighbourhood comprises all sequences resulting from the interchange of two aircraft, so for $1 \leq i < j \leq n$ a swap neighbour of σ is

$$(\sigma(1), \ldots, \sigma(i-1), \sigma(j), \sigma(i+1), \ldots, \sigma(j-1), \sigma(i), \sigma(j+1), \ldots, \sigma(n)).$$

The 2-insert neighbourhood comprises all sequences that can be obtained by removing two adjacent aircraft having the same weight class and inserting them into a new position in the sequence. Our motivation for this move type arises from the potential benefit of batching aircraft from the same weight class in terms of separation times. Note that these neighbourhoods can create solutions that cannot be formed by a merging of streams of aircraft as in our dynamic programming algorithm.

The descent algorithm uses the FCFS sequence as the initial solution and selects to a new solution using a *best improve* strategy when searching the combined insert, swap and 2-insert neighbourhoods. Specifically, each iteration of the search generates all landing sequences that are neighbours of the current sequence, from which the corresponding (smallest) landing times are computed using (5.1) and (5.2). Any sequence that does not produce feasible landing times is not considered further, whereas other sequences with feasible landing times are evaluated using equation (5.7). The best neighbour is then selected. If it improves on the current solution, this best neighbour replaces the current solution and the search to improve the new current solution continues. If the best neighbour does not improve on the current solution, then the descent algorithm terminates with a local optimum.

Iterated descent prevents the descent algorithm from terminating at the first local optimum by applying a 'kick' to the locally optimal solution to create a new starting solution. Descent is then applied to this new solution, and the process repeats for a specified number of iterations. Our kick corresponds to k randomly generated insert moves, where any such moves that cause infeasibility due to the maximum time shift constraints are rejected and consequently replaced by other random insert moves. We investigate different values of k in our computational experiments.

5.3.5 Simulated annealing

Simulated annealing is one of a number of local search techniques that can escape from local optima through accepting non-improving moves. In brief, the search randomly selects a neighbour, evaluates it with respect to the objective function, if it is an improving move it is automatically accepted, otherwise it is accepted with a certain probability. A temperature parameter controls this probability, which dynamically changes through the search. The initial temperature is set so the probability of accepting non-improving moves is high, and as the search progresses the probability reduces. This is called the cooling schedule. Some researchers have investigated non-monotonic changes in temperature.

Fahle et al. (2003) propose a simulated annealing approach for ALP using a simple geometric cooling schedule. They implemented insert and swap neighbourhood search in their SA algorithm. Our implementation of simulated annealing follows the approach proposed by Crauwels et al. (1997) for scheduling families of jobs on a single machine, where a set-up time is required when the machine switches from processing a job in one family to a job in another family. There are some parallels with our problem, where aircraft are in families of classes and, usually, switching classes incurs a greater separation than landing consecutive aircraft from the same class. We use the same three neighbourhoods as in our iterated descent approach; insert, 2-insert and swap.

Neighbours producing the same or a better objective function values than the current solution are accepted. Neighbours producing a worse objective function value are accepted with probability $e^{-\Delta/t}$, where Δ is the amount by which the objective function increases and t is the temperature. We follow the scheme of Crauwels et al. (1997) in which the values of the temperature are periodic, rather than the usual scheme of starting with a high temperature which is gradually decreased during the course of the algorithm.

5.4 Algorithms for dynamic problem

As explained in Section 5.2.5, the dynamic problem is based on solving a static problem every τ time units, where τ is the update time. The aircraft that are available to the static scheduling algorithm depend on two parameters in addition to τ. First, we consider the *time horizon* T over which the static problem is solved. Thus, at the update time, any aircraft that are within time T of the runway are assumed to be known to the ATC and are therefore included, but those aircraft with unconstrained landing times that are more than T time units into the future are excluded. Second, we assume that there is a *freeze time* t that defines the period of time for which the previously created schedule cannot be altered. As a consequence, any aircraft that is currently scheduled to land within the next t time units cannot be rescheduled. Note that the freeze time must exceed a certain minimum level to avoid potentially dangerous manoeuvres of aircraft that are close to the runway. Also, the time horizon T is selected to include all aircraft whose appearance times would reasonably be expected to be known to the ATC.

An interesting observation is that the length of the time period $T - t$ is our main concern, rather than the specific values of t and T. Assuming that the system is empty at the start of the dynamic scheduling solution approach, then the solution provided by $T = 30$ and $t = 10$ resulting in a 20 minute scheduling window, would be the same as that for $T = 35$ and $t = 15$ under the same update time. Hence, we can set the length of the time window to be $T' = T - t$ with T' chosen such that knowledge of aircraft that are separated by more than $T - t$ time units does not significantly improve the quality of the landing schedule that is generated.

5.5 Computational experiment

5.5.1 Test data

Our computational tests use two types of data sets. The first includes all landings at Heathrow Airport, UK, over a ten day period during June 2009. The second comprises data that are randomly generated in such a way to exhibit similar characteristics of traffic volume to the Heathrow data, and cover a 40-day period. The Heathrow data are the property of NATS (National Air Traffic Services) Ltd (NATS, 2011) and subject to a non-disclosure agreement, hence motivating the generation of artificial data that can be made available to other researchers.

There are two parallel runways available for use at Heathrow airport. As the airport is situated close to residential areas, two runways generally operate in segregated mode; one for landing and one for take-off. Occasionally, landings are allowed on the nominated take-off runway to reduce delays and taxi times. Arriving aircraft approach from the east to west (westerly operation) unless the wind comes from the east in which case the landing direction is reversed so that aircraft land into the wind for safety reasons. During busy periods, controllers normally direct arriving aircraft to the top of one of four holding stacks. As aircraft reach the lowest level in their stack, controllers vector the aircraft onto the final approach and move higher aircraft down. Finally, they are merged into a single arrival stream of traffic for landing (Heathrow, 2012).

For the Heathrow data set, we extract the following information for each aircraft j to form the input for our scheduling algorithms: actual landing time, landing runway, weight class of aircraft (c_j) based on the UK's wake vortex group classification, date, time that the aircraft crosses a cordon 40nm from the airport, and unconstrained landing time (ult_j). The Arrival Manager (AMAN) tool estimates the unconstrained landing time of each aircraft and suggests the landing sequence to minimize the wake vortex separation. The UK has increased the original International Civil Aviation Organization's three wake turbulence separation groups to five, in decreasing order of weight these are Heavy (H), Upper medium (U), Lower medium (M), Small (S) and Light (L). The unconstrained landing time is calculated from the 40nm cordon crossing and provides the initial landing sequence for FCFS sequence. The crossing time of the 40nm cordon is the appearance time that defines when the flight becomes available to the controllers for scheduling. We use the landing runway data to identify and remove flights that do not land on the primary landing runway. Removing these data should not affect separation times for landing the other flights in the data set (although there may be implications on ATC workload but this is not considered in our model).

In addition to providing test instances for our algorithms, the Heathrow data are used to estimate the separation time matrix (s_{bc}) and to determine for each aircraft j its maximum time shift (ts_j). Note that air traffic controllers are required to observe standard separation distances rather than times, and therefore the time between landings of aircraft is dependent on their approach speed.

In order to estimate separation times, we first extract the times between actual landings of consecutive flights. However, not all landings are queued and consequently some separations may have greater than the minimum required. Hence, we remove any separation times that are greater than 1.2 times the standard separation distances divided by the estimated speed of aircraft immediately prior to landing at Heathrow. The remaining data are averaged by wake vortex leader/-follower categories. Unfortunately, these data cannot be used directly because some categories have insufficient observations. Instead, we determine the airspeed that, when multiplied by the standard separation distances, gives the lowest mean square error from the separation times extracted from the Heathrow data set across all wake vortex categories. The separation times in Table 5.1 arise from a landing airspeed of 149 nm per hour, which gives a mean squared error of 43.9.

TABLE 5.1: Separation times (seconds) based on an airspeed of 149 nm per hour

		Follower				
		H	U	M	S	L
Leader	H	97	121	121	145	169
	U	72	72	97	97	145
	M	72	72	72	72	121
	S	72	72	72	72	97
	L	72	72	72	72	72

Section 5.2.1 details the rationale for the maximum time shift, ts_j. Here, we define a common maximum time shift that applies to all aircraft. Analysis of the frequency of time shifts in the Heathrow data, after removing flights that land before their unconstrained landing time, show that 95% of the time shifts $\mathrm{LT}_j - \mathrm{ult}_j$ lie in the range 0-870 seconds after the unconstrained landing time. On this basis, we set $\mathrm{ts}_j = 870$ seconds for all aircraft j. Since we do not have the necessary information to determine a meaningful preferred landing time, we assume that it is equal to the unconstrained landing time and therefore set $\mathrm{plt}_j = \mathrm{ult}_j$ for all aircraft j.

5.5.2 Random test data

In order to generate the random test instances, we design a model that mimics the pattern of changes in traffic volume across the day and allows us to set different traffic intensities. As a result, we can evaluate the performance of the algorithms over a variety of problem instances. Each problem instance covers a one-day period. The appearance of the first aircraft is after 3am and the last aircraft before 10pm. Each day is divided into three periods: Morning (3am-6am), Day (6am-8pm) and Night (8pm-10pm). Fewer aircraft arrive during the Morning and Night periods. We further divide the Day period into Normal and Busy hours, where Busy hours are 6-8am, 11am-1pm and 4-7pm and the remaining hours are Normal. Table 5.2 details the average number of aircraft μ per hour and the standard deviation σ, for each time period and each traffic intensity, where Set_1

represents the lowest intensity and Set_4 the highest. Ten instances are generated for each traffic intensity level, giving forty random test instances in total.

In addition to the number of flights, we also need a mechanism to generate for each aircraft j its weight class, appearance time (ap_j) and approach direction of the flight. For any time period t, probabilities $p_t(c)$ and $q_t(d)$ for the weight class $c \in \{\text{H,U,M,S,L}\}$ and approach direction $d \in \{1, \ldots, 10\}$ of an aircraft are derived from the 10-day Heathrow data, where d is the number of the dodecant corresponding to the position the aircraft crosses a cordon 40nm from the airport (only 10 of the 12 dodecants are used for approaches). Also, a negative exponential distribution provides a good fit for the inter-arrival time of flight appearance in the Heathrow data set.

TABLE 5.2: Means and standard deviations of hourly aircraft arrivals

		Morning			Day		Night	
		3-4am	4-5am	5-6am	Normal	Busy	8-9pm	9-10pm
Set_1	μ	5	15	30	37	39	30	10
	σ	0.5	0.5	1.0	1.5	1.5	1.0	0.5
Set_2	μ	5	15	30	38	41	30	10
	σ	0.5	0.5	1.0	1.5	1.5	1.0	0.5
Set_3	μ	5	15	30	39	43	30	10
	σ	0.5	0.5	1.0	1.5	1.5	1.0	0.5
Set_4	μ	5	15	30	40	45	30	10
	σ	0.5	0.5	1.0	1.5	1.5	1.0	0.5

Algorithm 1 details the procedure for generating the test data. In brief, for each hour we generate the number of flights using the normal distribution $N(\mu, \sigma^2)$ based the means and standard deviations in Table 5.2. Then we generate the inter-arrival times between the flights using the negative exponential distribution. These times are scaled to ensure the arrivals exactly span the entire hour (with one aircraft appearing on the hour). The arrival times of the aircraft correspond directly to these values. The algorithm then computes further parameters for each aircraft j as follows. The weight class and approach direction are generated according to their respective probability distributions. Given d and the runway for landing, the remaining duration of the flight rdf_d, assuming an unimpeded passage to the runway, is estimated from the Heathrow data. Hence, we can calculate ult_j for each aircraft j. Finally, the latest landing time llt_j is found by randomly choosing a time gap of 1800, 2700 or 3600 seconds with probability 0.3, 0.5 and 0.2, respectively, and and adding it to the appearance time ap_j. Note that the size of all gaps exceeds the maximum time shift and therefore latest landing times are effectively redundant when a maximum time shift constraint is imposed.

Daily Traffic Sample Generator

Execute the following steps for each hour $h = 1, \dots, 19$ of the day, where the hours correspond to the time periods 3-4am,. . . ,9-10pm. Select the intensity Set_1, Set_2, Set_3 or Set_4 to be used, and set t to be one of the seven time periods according to the hour h and the columns of Table 5.2.

Generate Appearance Times

Generate the number of the aircraft n_h that appear during hour h from the normal distribution $N(\mu, \sigma^2)$, where μ and σ are given in Table 5.2.

Generate the gaps in seconds between aircraft appearances as follows.

A sample of unscaled inter-arrival times g_j in seconds for $j = 1, \dots, n_h$ for hour h from an exponential distribution with mean $3600/n_h$ is generated.

Compute corresponding scaled inter-arrival times $\bar{g}_j = 3600 t_j / \sum_{i=1}^{n_h} g_i$ for $j = 1, \dots, n_h$.

Assign the appearance times using $\mathrm{ap}_j = \sum_{i=1}^{j} \bar{g}_i$ for $j = 1, \dots, n_h$.

Generate Data for Each Aircraft

Execute the following statements for each aircraft j, for $j = 1, \dots, n_h$, that has an appearance time in hour h.

Generate a random number and use the the probabilities $p_t(c)$ for $c \in \{$H,U,M,S,L$\}$ to assign aircraft j a weight class.

Generate a random number and use the the probabilities $q_t(d)$ for $d \in \{1,\dots,10\}$ to assign aircraft j an approach direction (dodecant).

Generate a random number and set $l_j = \mathrm{ap}_j + 1800$, $l_j = \mathrm{ap}_j + 2700$ and $l_j = \mathrm{ap}_j + 3600$ with probabilities 0.3, 0.5 and 0.2, respectively.

Set $\text{ult}_j = \text{ap}_j + \text{rfd}_d$, $\text{plt}_j = \text{ult}_j$, $\delta_j^e = 300$, $\delta_j^l = 600$ and $\text{ts}_j = 870$.

The Generator does not guarantee that the resulting data set has a feasible schedule. Hence we check feasibility using dynamic programming and discard any set for which a feasible solution is not found. The preferred landing time and maximum time shift are assigned in the same way as for the Heathrow data.

5.5.3 Experimental design

All algorithms were coded in MS Visual C++ 2008 and run on a PC with a dual core, 2.13GHz and 2GB RAM. We refer to the first-come first-served, dynamic programming, iterated descent and simulated annealing algorithms as FCFS, Algorithm DP, Algorithm ID and Algorithm SA, respectively. For the static problem, we select three half-hour periods, three one-hour periods and one two-hour period to schedule aircraft from the 10-day Heathrow data set. These focus on time periods between 7-8am, and 5-7pm, when demand for landing is particularly high. In the case of Algorithm ID and Algorithm SA where there is a stochastic element to the search procedure, the algorithm is run $n/5$ times with different random number streams, where n is the number of flights, and the average performance is reported. For the dynamic problem, each instance corresponds to the data for one day for both the Heathrow and random data sets. Each algorithm are run once for each instance.

Both Algorithm ID and Algorithm SA require an initial solution and a termination condition. The FCFS sequence provides the initial solution for iterated descent and simulated annealing applied to the static problem. For the dynamic problem, flights do not always appear in FCFS order. Nevertheless, the initial sequence when applying iterated descent at an update is obtained by adding the newly available flights in FCFS order to the end of the previous schedule. For the static case, Algorithm ID uses a kick size of five random moves as is common within the literature, and terminates after fifty local optima are found, and Algorithm SA terminates once it has performed $n/2$ levels, where a maximum of n neighbours

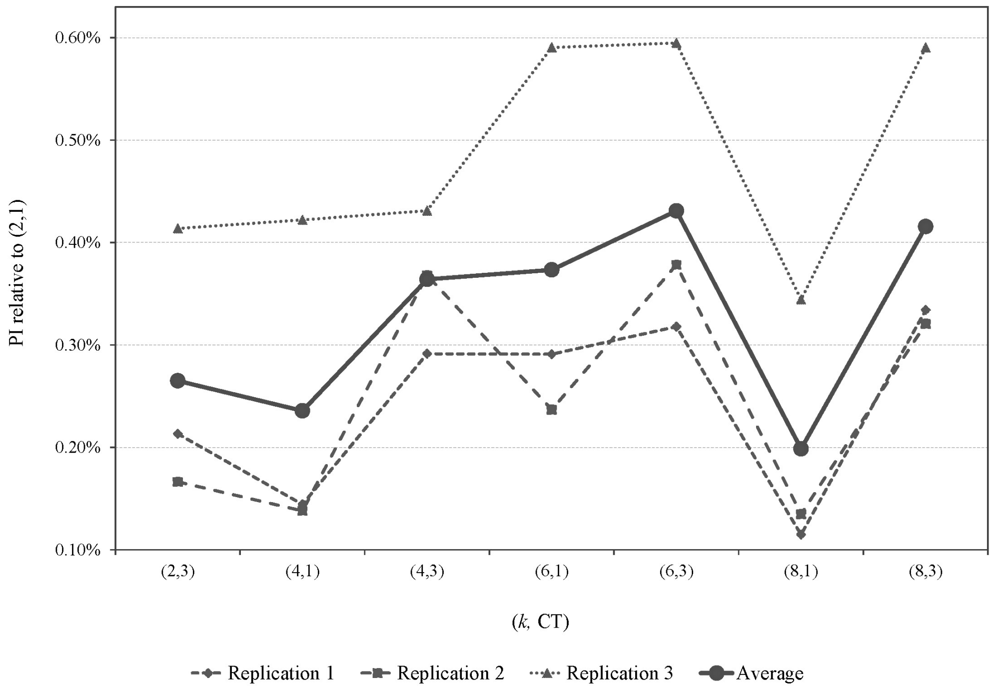

FIGURE 5.1: Average PI of Algorithm BS relative to $k = 2$ and CT $= 1$, Heathrow data, dynamic environment: Weights $(0.3, 0.5, 0.1, 0.1)$

are searched at each level. Moreover, initial experiments with Algorithm ID show that these parameters are appropriate choices for the static case.

In the dynamic case, the termination condition for each update is set to three seconds because returning a solution in a fast time is critical. Further, initial experiments with Algorithm ID show that a kick size of six is an appropriate choice. Figures 5.1 and 5.2 show the PI of Algorithm DP for objective weights $(0.3, 0.5, 0.1, 0.1)$ for different combinations of number of kicks and termination conditions using Heathrow data and random data, respectively. These results are based on three replications.

For the dynamic problem, the previous schedule is updated every τ time units. We investigate the influence of τ by considering values $\tau = 2.5, 5.0, 7.5, 10$ minutes. Scheduling starts after the freeze time that occupies the first t units of the scheduling period and considers those aircraft with unconstrained landing times that are no more than T time units into the future. As pointed out in Section 5.4, our interest is in the value of the parameter $T' = T - t$ that defines the length of the active time window. We investigate $T' = 10, 15, \ldots, 40$ minutes. For the Heathrow

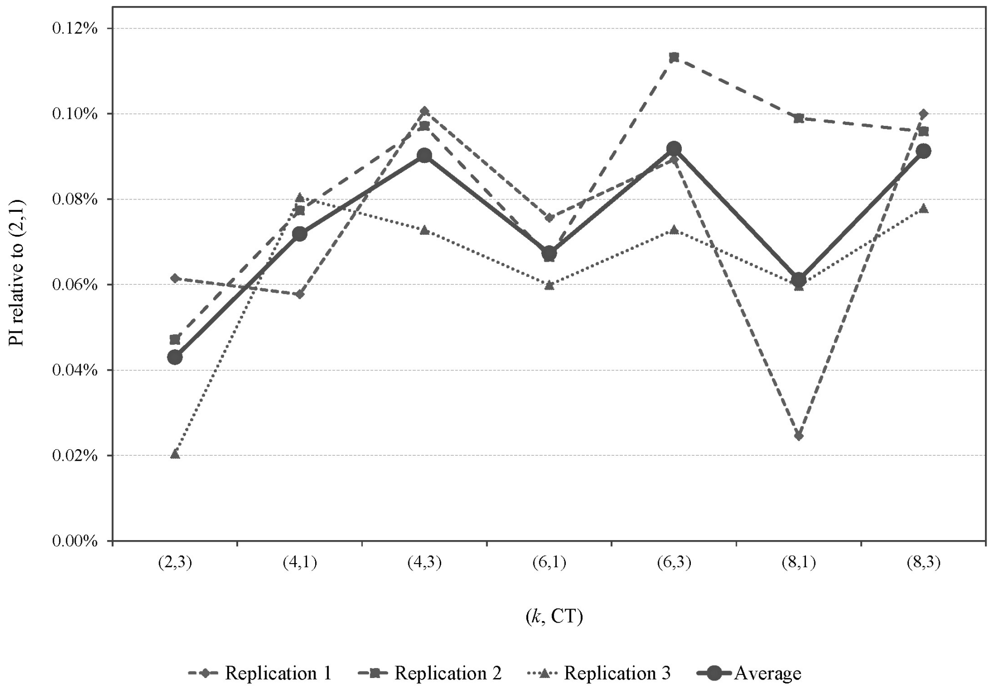

FIGURE 5.2: Average PI of Algorithm BS relative to $k = 2$ and CT $= 1$, random data, dynamic environment: Weights $(0.3, 0.5, 0.1, 0.1)$

data, appearance time is between 13 to 20 minutes before unconstrained landing time depending on the approach route. In order to study longer time windows, we subtract a constant from the appearance times.

Our experiments investigate several sets of weight vectors (w_1, w_2, w_3, w_4) for the objective function defined in equation (5.7) of Section 5.2.3. When investigating throughput, we use the objective function defined in equation (5.4) and also the single objective function $\mathrm{LT}_{\max}$. For the full multi-criteria objective function (5.7), there are penalties for time window violations and for the use of extra fuel if the unconstrained landing time is not achieved. Table 5.3 lists the unit penalty values for each weight class. In (5.5), for each aircraft j we set $\delta_j^l = 600$, where δ_j^l is expressed in seconds.

Specifically, our first weight vector is $(0.3, 0.5, 0.1, 0.1)$, which reflects the throughput considerations of ATCs with some consideration of time-window violations and extra fuel cost. The second set is $(0.2, 0.4, 0.3, 0.1)$, which gives more emphasis to time-window violations. Note that delays relative to the time windows and extra fuel costs are non conflicting. We also consider two objectives that measure

Table 5.3: Weights for time window violation and extra fuel

Weight class of j	H	U	M	S	L
u_j^l	20	17	15	12	10
v_j^l	15	13	12	10	8
u_j^e	10	8	7	5	4

throughput. The first of these is given by the weight vector $(0.4, 0.6, 0.0, 0.0)$ so that both $\mathrm{LT}_{\max}$ and ALT are considered, while the second considers only $\mathrm{LT}_{\max}$ by selecting the weight vector as $(1.0, 0.0, 0.0, 0.0)$. It is worth noting that ATCs typically prioritize throughput, and in particular $\mathrm{LT}_{\max}$, when deciding upon the landing order of aircraft. Hence, a throughput objective function is regarded as providing the best basis to compare our schedules against those designed by the controller. Although $\mathrm{LT}_{\max}$ provides the most natural measure of throughput, this objective only uses the landing time of the last aircraft and has the disadvantage of ignoring other landing times.

Our comparison of algorithms is based on the following performance statistics:

PI: percentage improvement in the solution objective function relative to the initial sequence (off-line problem) or to a specific sequence (on-line problem);

TD: total deviation of the positions in the solution landing sequence relative to FCFS;

ND: number of aircraft with changed positions in the solution landing sequence from FCFS;

SEP: sum of the standard minimum separation times in seconds between aircraft implied by the solution sequence;

CT: computation time in seconds for scheduling a given set of aircraft (off-line problem) or the available aircraft in the time horizon (on-line problem).

Max CT: maximum computation time in seconds for scheduling the available aircraft in the time horizon (on-line problem).

As well as the overall weighted objective function, we also give values of PI relative to the individual components $\mathrm{LT}_{\max}$, ALT, TW and EF, where the latter three are defined in equations (5.3), (5.5) and (5.6), respectively. The results for the Heathrow data also include those for ATC, which are the actual the actual landing schedules obtained through decisions by air traffic controllers. TD and ND quantify deviations from the FCFS landing sequence, providing a measure of the amount of intervention necessary to achieve the landing schedule. SEP can be viewed as a measure of the amount of batching used to reduce separation times. Intuitively, reducing separation times is aligned with maximizing throughput; hence ATCs implicitly use batching as a heuristic decision tool for increasing throughput.

5.5.4 Results

The tables that follow detail the average performance of the schedules arising from each of the approaches described in the earlier sections, where the objective function is defined by (5.7) for various choices of weights and the time window constraints (5.1) and separation constraints (5.2) are imposed. Results tables for the Heathrow data include the actual landing times resulting from the ATC's scheduling. As discussed previously, the ATC does not work to optimize our multi-objective function, and the data and constraints do not perfectly mirror the task that the ATC performs. Moreover, the minimum standard separations are currently based on distance (radar separation), which have been converted into time separations when used within our algorithms. Results of the algorithms have been verified to make sure that the models meet the requirements and specifications of the problem and they perform as expected.

Tables 5.4, 5.5, 5.6 and 5.7 list average results for the Heathrow data used in a static environment, where each table corresponds to an alternative objective function. The first and second columns in each table give the durations in minutes of the time windows that define the aircraft to be scheduled and the average numbers of aircraft in the data set. The third column contains row headings for the objective function criteria. Hence, for each data set, the first row gives results for the main objective used by all of the approaches and the following rows break down the objective function into its component criteria. For some entries in the

TABLE 5.4: Heathrow data, static environment: Weights $(0.3, 0.5, 0.1, 0.1)$

			ATC			Algorithm DP				Algorithm ID				Algorithm SA			
T	n	Obj.	PI	TD	ND	PI	TD	ND	CT	PI	TD	ND	CT	PI	TD	ND	CT
		Overall	−3.05			1.47				1.47				1.47			
		LT_{max}	−0.09			0.07				0.07				0.07			
30	21	ALT	−0.02	15	10	0.07	15	8	0.16	0.07	18	9	0.09	0.07	15	8	0.07
		TW	−22.78			2.65				2.65				2.65			
		EF	−7.83			8.36				8.36				8.36			
		Overall	−7.60			4.41				4.39				4.41			
		LT_{max}	−0.36			0.19				0.19				0.19			
60	42	ALT	−0.13	36	22	0.12	35	16	0.56	0.12	45	22	0.39	0.12	31	16	0.52
		TW	−87.99			16.16				16.16				16.16			
		EF	−16.25			10.50				10.50				10.50			
		Overall	−9.27			7.08				7.08				7.05			
		LT_{max}	−0.00			0.00				0.00				0.00			
120	84	ALT	−0.10	72	47	0.09	74	30	7.73	0.09	90	36	2.36	0.08	84	36	8.94
		TW	N/A			0.00				0.00				0.00			
		EF	−29.60			24.27				24.27				23.65			

TW row, a value of PI is not available (N/A) because the value of TW is zero for the initial FCFS sequence but positive for the algorithm under consideration.

The complete results for the Heathrow data used in static environment are presented in Tables A.1, A.2, A.3, and A.4 in Appendix A. Detailed results of the ATC performance are not included in these tables because of the confidentiality agreement between the NATS and research participants.

Tables 5.4 and 5.5 show an improvement in the main objective relative to the initial FCFS schedule across all approaches. The ATC schedule does not show an improvement, but this is largely due to the TW and EF cost elements. This is expected since there is no attempt by the ATC to reduce lateness or the cost of fuel. However, there is some degradation in LT_{max} and ALT, which is explored in more detail below in the discussion for dynamic environment. When considering the break-down of criteria, the improvement for LT_{max} and ALT is modest; it is clear that our approaches are producing similar throughput but with reduced cost associated with time window violations and extra fuel. The results in Table 5.5 put greater emphasis on time window violations, but this has only a small impact on the results. A much greater weight on TW may improve this criterion, but could potentially cause a poorer performance for throughput. Tables 5.6 and 5.7

Table 5.5: Heathrow data, static environment: Weights (0.2, 0.4, 0.3, 0.1)

			ATC			Algorithm DP				Algorithm ID				Algorithm SA			
T	n	Obj.	PI	TD	ND	PI	TD	ND	CT	PI	TD	ND	CT	PI	TD	ND	CT
		Overall	−2.29			1.83				1.83				1.83			
		LT_{max}	−0.09			0.07				0.07				0.07			
30	21	ALT	−0.02	15	10	0.07	15	8	0.25	0.07	21	11	0.09	0.07	13	7	0.05
		TW	−22.78			2.65				2.65				2.65			
		EF	−7.83			8.36				8.36				8.36			
		Overall	−14.20			6.30				6.30				6.30			
		LT_{max}	-0.36			0.19				0.19				0.19			
60	42	ALT	−0.13	36	22	0.12	35	16	0.71	0.12	42	20	0.51	0.12	30	15	0.52
		TW	−87.99			16.16				16.16				16.16			
		EF	−16.25			10.50				10.50				10.50			
		Overall	−12.36			8.56				8.56				8.51			
		LT_{max}	0.00			0.00				0.00				0.00			
120	84	ALT	−0.10	72	47	0.09	82	33	7.74	0.08	98	41	2.48	0.09	106	39	7.28
		TW	N/A			0.00				0.00				0.00			
		EF	−29.60			23.66				23.66				23.52			

Table 5.6: Heathrow data, static environment: Weights (0.4, 0.6, 0.0, 0.0)

			ATC			Algorithm DP				Algorithm ID				Algorithm SA			
T	n	Obj.	PI	TD	ND	PI	TD	ND	CT	PI	TD	ND	CT	PI	TD	ND	CT
		Overall	−0.05			0.10				0.10				0.10			
30	21	LT_{max}	−0.09	15	10	0.13	22	9	0.25	0.13	28	12	0.08	0.13	14	6	0.04
		ALT	−0.02			0.08				0.08				0.08			
		Overall	−0.22			0.18				0.18				0.18			
60	42	LT_{max}	−0.36	36	22	0.23	63	23	0.72	0.23	79	29	0.36	0.23	61	22	0.42
		ALT	−0.13			0.14				0.14				0.14			
		Overall	−0.06			0.05				0.05				0.05			
120	84	LT_{max}	0.00	72	47	0.00	88	34	7.64	0.00	126	46	2.08	0.00	96	34	7.14
		ALT	−0.10			0.09				0.09				0.09			

do not consider costs for time window violations or extra fuel in the objective function.

Comparing across the different solution approaches, dynamic programming has the longest and most variable computation times and iterated descent is the fastest and most consistent. For these data instances, greater computational effort does not lead to improved schedules, with all three methods having similar performance on average across all objective functions. Nevertheless, the ability of iterated descent to obtain competitive solutions with short run times provides a case for using it in preference to dynamic programming or simulated annealing.

Table 5.7: Heathrow data, static environment: Weights $(1.0, 0.0, 0.0, 0.0)$

			ATC			Algorithm DP				Algorithm ID				Algorithm SA			
T	n	Obj.	PI	TD	ND	PI	TD	ND	CT	PI	TD	ND	CT	PI	TD	ND	CT
30	21	LT_{max}	−0.09	15	10	0.13	37	13	0.26	0.13	43	16	0.05	0.13	11	5	0.05
60	42	LT_{max}	−0.36	36	22	0.23	88	31	0.73	0.23	45	16	0.25	0.23	27	9	0.47
120	84	LT_{max}	0.00	72	47	0.00	204	69	7.77	0.00	96	31	0.69	0.00	0	0	7.59

Table 5.8: MIP *vs.* Heuristic methods, Heathrow data, static environment: Weights $(0.3, 0.5, 0.1, 0.1)$

Data set	T	n	MIP PI	MIP CT	MIP Solution	Algorithm DP PI	Algorithm ID PI	Algorithm SA PI
S01	30	22	1.73	3600	Local	2.89	2.89	2.89
S02	30	21	1.10	3600	Local	1.20	1.20	1.20
S03	30	20	0.33	2	Global	0.33	0.33	0.33
S04	60	41	8.68	3600	Local	9.02	8.96	9.02
S05	30	42	2.78	3600	Local	3.51	3.51	3.51
S06	60	42	0.69	3600	Local	0.69	0.69	0.69
S07	120	84	5.16	3600	Local	7.08	7.08	7.05

One statistic of note is that all of our approaches are finding schedules of similar performance but with varying deviations from the initial FCFS sequence as measured by TD and ND. This suggests that there are many local optima with similar objective function values.

Table 5.8 presents performance of the MIP model, Algorithm DP, Algorithm ID and Algorithm SA. The Xpress IVE (FICO, 2012) optimization package has been used for solving the MIP model. The solver is stopped after 3600 seconds and the best obtained solution has been reported. The global optimal solutions have not been found for any problems instances other than data set S03. As higher values of PI are more desirable, the best reported solution by the MIP model after one hour computation time is not as good as other methods.

We now present our computational results for the dynamic environment. For these experiments we retain iterated descent as the best performing approach and dynamic programming as a benchmark, but remove simulated annealing from consideration. Note that the reported results are based on schedules created for a complete day.

We design an initial set of experiments in order to test parameters using the Heathrow data, and eight days of the random test data where two days are randomly chosen for each of the four traffic intensity levels. We first experiment with the update times $\tau = 2.5, 5.0, 7.5, 10.0$ minutes, where dynamic programming is applied to solve the resulting problem at each update and the objective function is defined by the weights $(0.3, 0.5, 0.1, 0.1)$. Table 5.9 lists average PI values relative $\tau = 2.5$ minutes, using $t = 5$ minutes and $T = 30$ minutes. It is clear from the results that $\tau = 5$ minutes provides the best strategy in terms of solution quality and it has a lower computational requirement than the next best value of $\tau = 2.5$ minutes.

TABLE 5.9: Influence of τ: Average PI relative to $\tau = 2.5$ min

	Update time τ (mins)		
Dataset	5	7.5	10
10-day Heathrow	0.011	−0.029	−0.050
8-day random data	0.000	−0.052	−0.297

Table 5.10 and Figure 5.3 present our computational results for various active time window lengths $T' = 15, 20, 25, 30, 35, 40$ minutes (in minutes), with $\tau = 5$ minutes and objective function weights $(0.3, 0.5, 0.1, 0.1)$, using dynamic programming.

TABLE 5.10: Influence of T': Average PI relative to Active $T' = 15$ mins

		Active scheduling window length T' (mins)					
Dataset	Measure	15	20	25	30	35	40
10-day Heathrow	Ave. PI		0.743	0.931	0.999	1.022	0.983
	Ave. CT	0.002	0.020	0.103	0.308	0.883	1.959
8-day random data	Ave. PI		0.245	0.761	0.855	0.973	0.980
	Ave. CT	0.002	0.012	0.052	0.157	0.380	0.831

The quality of schedules and the computation time increase as T' becomes larger. However, the improvement in solution quality becomes less with each five minute widening of T', whereas the computation time significantly increases. Hence, there are rapidly diminishing returns after $T = 25$. These results confirm the intuitive conclusion that the likelihood of new aircraft added to the end of the schedule impacting the order of the aircraft at the beginning of the schedule reduces as the

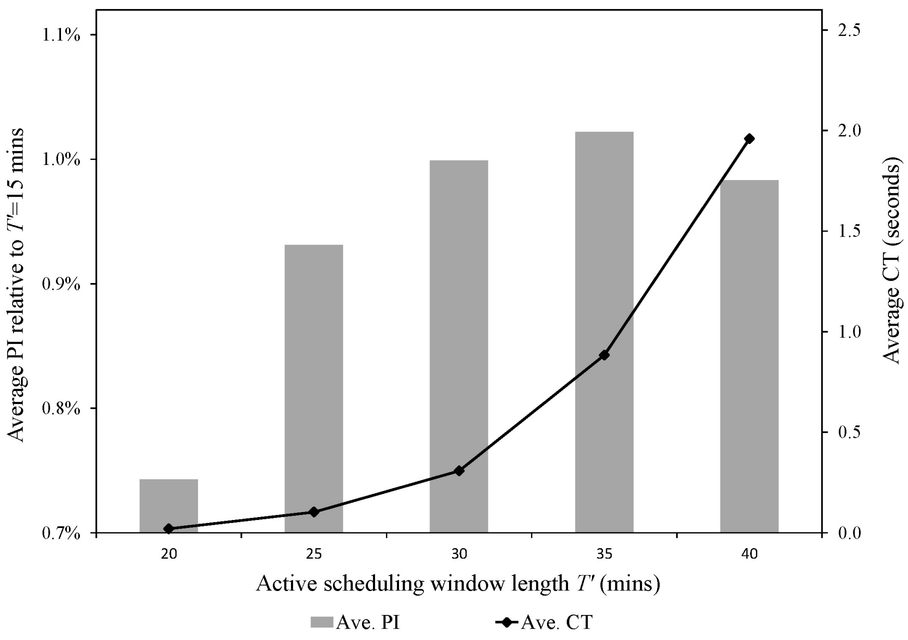

FIGURE 5.3: Influence of the T' relative to $T' = 15$mins

T' increases. As a result, we select $T' = 25$ minutes as the length of active time window.

Tables 5.11 and 5.12 detail the full results for the dynamic problem with $\tau = 5$, $T' = 25$ using Algorithm ID with $k = 6$ and a computation time limit of 3 seconds for each update and using Algorithm DP. The tables give average percentage improvements with respect to the FCFS schedule. Note that we omit objective function weights $(1.0, 0.0, 0.0, 0.0)$, which reduces to the single objective $\mathrm{LT}_{\max}$, because the landing time of the day's last aircraft is not good enough in isolation. Furthermore, it is likely that the position of the last aircraft in the partial sequence changes in the next update. However, $\mathrm{LT}_{\max}$ has not been considered as a single objective function for the dynamic environment, it has been used as a part multi-objective function for each time horizon.

The complete results for the Heathrow data used in dynamic case with respect to objective function weights $(0.3, 0.5, 0.1, 0.1)$, $(0.2, 0.4, 0.3, 0.1)$ and $(0.4, 0.6, 0.0, 0.0)$ are shown in Tables A.5, A.6 and A.7 in in Appendix A. Detailed results of the ATC performance are not included in these tables because of the confidentiality agreement between the NATS and research participants.

Table 5.11: Heathrow data, dynamic environment: Average PI relative to FCFS

		FCFS	ATC				Algorithm DP						Algorithm ID				Algorithm SA		
Weight	Obj	Sep.	PI	TD	ND	Sep	PI	TD	ND	CT	Max CT	Sep	PI	TD	ND	Sep	PI	TD	ND
$(0.3, 0.5, 0.1, 0.1)$	Overall	54151	−51.61	569	302	53354	23.76	504	218	0.11	58.60	52838	23.60	523	222	52840	22.60	556	225
	ALT		−0.20				0.14						0.14				0.13		
	TW		−3706.31				34.24						32.70				55.52		
	EF		-52.01				29.46						29.33				27.92		
$(0.2, 0.4, 0.3, 0.1)$	Overall	54151	−80.81	569	302	53354	26.32	486	216	0.10	52.90	52869	26.19	511	220	52871	25.22	534	219
	ALT		−0.20				0.13						0.13				0.13		
	TW		−3706.31				72.17						70.04				77.47		
	EF		−52.01				46.93						28.59				27.01		
$(0.4, 0.6, 0.0, 0.0)$	Overall	54151	−0.10	569	302	53354	0.05	736	272	0.09	26.40	52944	0.07	721	266	52906	0.07	742	268
	ALT		−0.20				0.11						0.15				0.14		
	TW		−3706.31				−538.30						−328.24				−377.82		
	EF		−52.01				18.14						29.37				27.18		

The complete results for the random data Set_1 used in dynamic environment are represented in Tables A.8, A.12 and A.16 in Appendix A. Moreover, Tables A.9, A.13 and A.17 in Appendix A provide the complete results for the random data Set_2. The complete results for random data Set_3 used in dynamic environment are displayed in Tables A.10, A.14 and A.18 in Appendix A. Finally, Tables A.11, A.15 and A.19 in Appendix A show the complete results for the random data Set_4.

The results for the Heathrow data in Table 5.11 show that iterated descent and dynamic programming provide schedules that improve over FCFS and ATC for all objective function components. As with the static case, the ATC schedules appear to be inferior to FCFS schedules. TW and EF play a major role in the reduction in solution quality; neither of them are used by ATC in making scheduling decisions. Our understanding is that ATC seek to maximize runway utilisation by reducing separation times. This can be achieved locally by batching aircraft. Performance measure SEP sums the minimum separation time between consecutive aircraft given the landing sequence. Using this measure we can see that ATC are successfully reducing separation times over FCFS. Iterated descent and DP also improve on FCFS and ATC by this measure. Table 5.12 shows a similar performance using the randomly generated test data. For both the Heathrow and the random data, Algorithm DP performs a little better than Algorithm ID with a lower deviation from the initial FCFS sequence. However, DP has variable computation times

Table 5.12: Random data, dynamic environment: Average PI relative to FCFS

Weight	Obj	Algorithm DP PI	TD	ND	CT	Max CT	Algorithm ID PI	TD	ND	Algorithm SA PI	TD	ND
$(0.3, 0.5, 0.1, 0.1)$	Overall	26.53	527	266	0.11	68.70	26.50	572	277	25.48	556	265
	ALT	0.20					0.20			0.19		
	TW	59.69					60.09			72.39		
	EF	29.09					29.04			27.66		
$(0.2, 0.4, 0.3, 0.1)$	Overall	31.32	503	261	0.14	77.50	31.26	544	271	30.13	533	260
	ALT	0.19					0.19			0.18		
	TW	78.81					78.66			80.27		
	EF	28.33					28.28			26.88		
$(0.4, 0.6, 0.0, 0.0)$	Overall	0.10	754	320	0.11	71.60	0.10	798	326	0.09	807	323
	ALT	0.21					0.21			0.20		
	TW	−573.23					−747.66			−737.47		
	EF	29.24					29.13			27.35		

that can sometimes be in excess of one minute, which makes it less attractive for implementation.

5.6 Concluding Remarks

This chapter has introduced models and algorithms for the static/off-line aircraft landing problem and the dynamic/on-line version of the problem. A special feature of our model is the multi-objective approach that takes into account the agendas of the various stakeholders that have an interest in the scheduling of landing aircraft.

Dynamic programming, iterated descent and simulated annealing algorithms are proposed for the static problem. Also, using a rolling horizon approach, the dynamic problem is tackled periodically updating the previous schedule with an iterated descent or dynamic programming solution approach. A thorough computational evaluation is performed using data from Heathrow airport and randomly generated test data.

Results for the static problem show that all of the proposed algorithms are effective in achieving an efficient runway throughput. In addition, algorithms are capable of finding solutions that perform well in terms of minimizing delay and minimizing the cost of extra fuel used to achieve the desired landing schedule. Iterated descent has the advantage of being faster and having more predictable run times than the other approaches, and is therefore preferred to dynamic programming and simulated annealing.

For the dynamic problem, the frequency of update time and the length of the time window when aircraft are available for scheduling are investigated. A five minute update time provides as good solutions as with a more frequent update, and has a lower computational cost. A time window of twenty-five minutes for scheduling is chosen. Wider time windows have diminishing returns and require much greater computational effort. Our overall computational results show that iterated descent and dynamic programming provide schedules that improve upon FCFS across all objective function elements. However, iterated descent is preferred to dynamic programming because of its more modest and predictable computational computational requirements.

The next chapter is dedicated to aircraft take-off problem. The impact of the departure holding area on scheduling of aircraft take-off has been investigated.

Chapter 6

Aircraft Take-off Problem

This chapter describes the departure scheduling problem. The focus is on impact of holding area on take-off scheduling. Our aim is to design algorithms to be able to sequence the aircraft according to the layout of the departure holding area. Aircraft take-off scheduling runs in under three seconds and preferably provide solutions in under one second.

Problem has been briefly defined in Section 6.1. Calculated take-off time constraints, separation time constraints and layout constraints have been discussed in Section 6.2. A description of the models and developed solution methods including dynamic programming, descent local search and beam search have been explained in Section 6.3. Section 6.4 provides the computational experiment. Finally, concluding remarks have been given in Section 6.5.

6.1 Problem definition

The take-off scheduling problem is to find a sequence and corresponding scheduled take-off times that optimizes the objective function subject to the operational constraints. Generally, ground movement controllers are responsible for giving clearance and guidance to the pilots for leaving the gate and the route for taxiing to the runway. Then, the responsibility is passed to the take-off runway controller. Therefore, the initial take-off sequence (or FCFS order) is generated by the ground movement controllers and it will be modified and finalized by the take-off runway controller. In this chapter, sequencing and scheduling of departing flights from the holding area are investigated.

Our study involves scheduling n aircraft for take-off on a single runway. The arrival times of these aircraft into the holding area are given by $\mathrm{at}_1, \mathrm{at}_2, \ldots, \mathrm{at}_n$, where the indices are chosen in a way that $\mathrm{at}_1 \leq \mathrm{at}_2 \leq \cdots \leq \mathrm{at}_n$. Our aim is to determine take-off times $T_1, \ldots, T_n$ for these aircraft.

We consider a hierarchical objective function. Minimizing the maximum take-off time (makespan or runway throughput), $T_{\max}$, where $T_{\max} = \max_{j=1,\ldots,n} T_j$ is regarded as being of primary importance by air traffic controllers, is considered as the main objective function. Minimizing the total waiting time

$$\mathrm{TWT} = \sum_{j=1}^{n} (\mathrm{T}_j - \mathrm{at}_j), \tag{6.1}$$

is chosen as the second objective, where the waiting time of an aircraft is defined as the difference between its scheduled take-off time and its arrival time into the holding area. The first objective aims to maximize the runway's utilization. Fairness among the departure flights, fuel burn, CO2 emission and delay are the main concerns of the second objective function.

Planning the taxiing of departing aircraft to the runway so that these aircraft reach the runway threshold in the right sequence and at the right time is unrealistic based on the current level of technology. Therefore, aircraft have to wait frequently in the holding area before departure. Depending on the layout of the holding area and the number of entry points to the runway, it may be possible to re-sequence the aircraft for take-off rather than using a First-Come First-Served (FCFS) sequence.

The aim is to study the impact of the runway holding area on the scheduling of aircraft take-offs. So, three layouts of the departure holding area are considered. Dynamic programming, descent local search and beam search methods for optimizing take-off schedules, subject to timing, layout and separation constraints are proposed. The performance of the proposed algorithms are evaluated by using randomly generated test data.

6.2 Constraints

6.2.1 CTOT constraints

Calculated Take-Off Time (CTOT) is an operational constraint which has to be considered as a hard constraint and cannot be violated. The CTOT which is also known as *slot time* for a flight is a period of time within which the flight is expected to get airborne. The CTOT is allocated by the Central Flow Management Unit (CFMU) to protect congested air traffic control sectors so that traffic within a sector does not reach unmanageable levels. If a slot is missed, a new slot by CFMU has to be assigned which has a big influence on the flight delay. Therefore, respecting the slot time is an important limiting factor for the airlines and controllers.

The slot for aircraft j is defined as the interval $[\mathrm{ctot}_j - \delta_1, \mathrm{ctot}_j + \delta_2]$, where typically δ_1 is set to be 5 minutes and δ_2 is set to be 10 minutes in Europe, although not all aircraft are constrained by such a slot. Thus, there is a constraint

$$e_j \leq T_j \leq l_j \quad j = 1, \ldots, n, \tag{6.2}$$

where $e_j = \max\{\mathrm{ctot}_i - \delta_1, \mathrm{at}_j\}$ is the earliest take-off time and $l_j = \mathrm{ctot}_i + \delta_2$ is the latest take-off time.

6.2.2 Separation time constraints

Another constraint which is imposed for safety reasons is the take-off separation constraint. Wake vortex generated by departing flights poses a potential risk to the following aircraft. Therefore, aircraft should maintain minimum standard separation to avoid wake turbulence hazard. The minimum standard departure separation time depends on the relative size of the consecutive aircraft, the standard instrument departure (SID) route, and the airspeed of the aircraft. A SID defines an air route out of airport to facilitate transition between take-off and en-route operations.

Generally, if the following aircraft is in a lower airspeed class category than the leading aircraft, two minutes separation is required; otherwise, a one-minute minimum separation is imposed. Moreover, the minimum separation increases by one minute if two consecutive take-off flights use the same departure route group. In addition, if two consecutive take-off flights are of different speed groups, then the route-based separation needs to be modified. Therefore, separations are different for the various combinations of departure flights. They are asymmetric and do not necessarily satisfy the triangle inequality.

If aircraft i of weight class b departs before aircraft j of weight class c, then the separation constraint is of the form

$$T_i + s_{bc} + \lambda_{ij} \leq T_j, \tag{6.3}$$

where

$$\lambda_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ have the same SID route} \\ 0 & \text{otherwise.} \end{cases}$$

assuming that aircraft j is not of a higher airspeed class than aircraft i. In general, it is sufficient to ensure that separation constraints are satisfied between each group of four consecutive departing aircraft.

6.2.3 Layout constraints

The layout and configuration of the holding area represents a major operational constraint in the take-off sequencing problem. It is the main limiting factor for the take-off runway controllers in changing the position of the aircraft in the sequence.

In this study, the impact of three main layouts of the departure holding area on take-off scheduling (Figures 6.1, 6.2 and 6.3) is investigated. These layouts are the most common configurations for the departure runway holding area. There are similarities between departure holding area of runway 30 at Cardiff International Airport (CWL), runway 24R at Los Angeles International Airport (LAX) and runway 34L at Denver International Airport (DEN) and Layout A, Layout B and Layout C, respectively.

The departure holding area can be divided into two sections; one section is used for holding and the other one is used for queueing. The *holding section* comprises the waiting positions before the entrance to the runway represented by the letter R and the positions in between shown by the letter M. It is assumed that aircraft form a queue before entering to the holding section which we refer to it as the *queueing section*. The holding positions are the focus of this research.

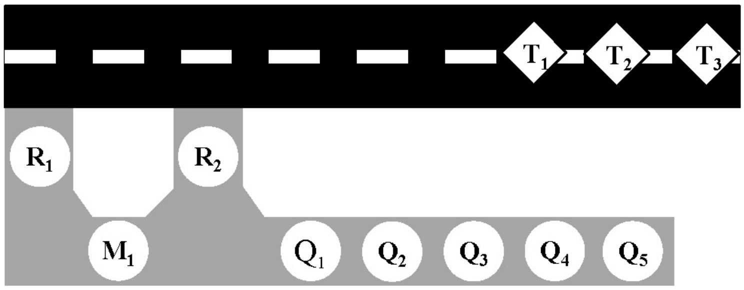

FIGURE 6.1: Holding area for Layout A

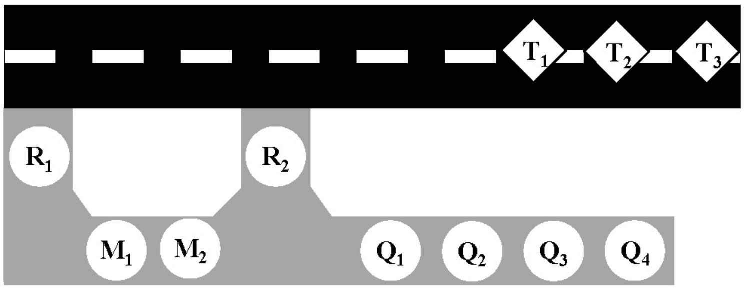

FIGURE 6.2: Holding area for Layout B

It is assumed that aircraft cannot overtake each other in the holding area. Additionally, it is assumed that heavy class aircraft must enter the runway using the last entrance (R_1).

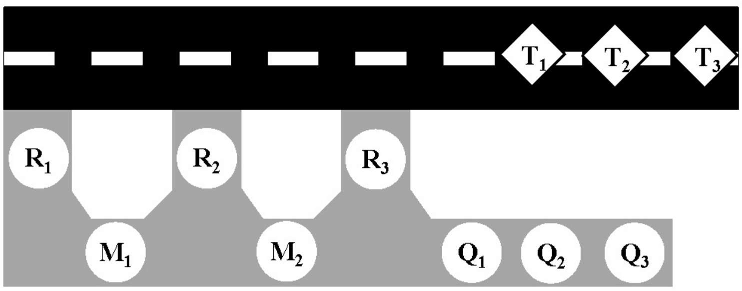

FIGURE 6.3: Holding area for Layout C

In Layout A as shown in Figure 6.1, there are three holding points, namely R_1, R_2 and M_1. However, Layout B in Figure 6.2 displays four holding points, R_1, R_2, M_1 and M_2 and Layout C in Figure 6.3 displays five holding points, R_1, R_2, R_3, M_1 and M_2.

Each layout has some specific characteristics. There is a restriction on the number of aircraft to move forward in the sequence with respect to the FCFS sequence depending on the layout of the holding area. For Layout A, each aircraft can move a maximum of two positions forward in the sequence relative to the FCFS sequence, while for Layout B the limit for moving any aircraft forward in the sequence is three positions and the limit for moving any aircraft forward in the sequence is four positions in Layout C.

Furthermore, the layout of the holding area imposes restriction on the number of aircraft which can be moved backward relative to the FCFS sequence. The position of a batch of three or more consecutive aircraft in FCFS order cannot be moved backward in Layout A. The limit for Layout B is a batch of four or more aircraft and a batch of five and more aircraft is the limit for Layout C. The size of such a batch depends on the number of holding points in holding section.

6.3 Algorithms

In this section, the FCFS, dynamic programming (DP), descent local search (DLS) and beam search (BS) algorithms have been introduced for sequencing aircraft in the departure holding area.

6.3.1 Feasibility check

One of the main challenges in designing the FCFS, DLS and BS algorithm is to test if the developed sequence can be achieved based on the holding area configurations (Atkin, 2008). It is computationally expensive to evaluate the feasibility of the departure sequence without knowing the assigned entrance to the runway for each aircraft. For this purpose a Feasibility Check (FC) algorithm has been developed.

There are two initial tests carried out in the FC algorithm. The first phase of the initial feasibility test can be performed based on the deviation of the sequence from the original FCFS sequence. A positive deviation shows a moving forward position and a negative deviation displays a moving backward position in the sequence relative to the original FCFS sequence. It has to be mentioned that the original FCFS sequence of a set of aircraft is formed by the non-decreasing order of their arrival times to the holding area and it is not necessary feasible subject to the CTOT constraint. Sequencing aircraft with FCFS method will be discussed in Section 6.3.2. Based on the deviation of the sequence from the original FCFS sequence, some feasibility rules have been derived for each layout.

For instance, in Layout A, Layout B and Layout C the maximum positive deviation of each aircraft from FCFS cannot be more than 3, 4 and 5 positions, respectively. Moreover, some consecutive order of the deviations have to be avoided. Table 6.1 displays forbidden deviation blocks for layout A, B and C. These forbidden order of deviations relative to the FCFS sequence are derived by performing experimental tests.

TABLE 6.1: Forbidden consecutive order of deviation from FCFS

Layout	Forbidden consecutive order of deviation
A	$+2, 0, -2$
B	$+2, 0, -2$
	$+3, 0, 0, -3$
	$+2, 0, +1, -3$
	$+3, +1, -1, -3$
	$+3, +1, -2, -2$
	$+3, +3, 0, -3, -3$
	$+2, +2, 0, -1, -3$
C	$+4, 0, 0, 0, -4$
	$+4, +1, 0, -1, +3, +1$
	$+4, +4, +4, 0, -4, -4, -4$

The second phase of the initial feasibility check is based on the entrance (positions next to the runway ie. R_1, R_2 and R_3) assignment. First, any entrance positions other than R_1 can be assigned to the aircraft with deviation more than +1. So in all three layouts, aircraft with deviation more than +1 cannot be assigned to position R_1 (the last entrance). Moreover, it is infeasible to assign aircraft with deviation more than +2 to position R_2 (second entrance) in Layout C. First, entrances position should be assigned to aircraft with deviation more than +1 position. Then, the entrance positions will be assigned to the aircraft starting in the first unpositioned aircraft in the sequence. If the sign of the deviation of the aircraft is the same as the following aircraft in the sequence, the same entrance position can be assigned, otherwise a different entrance should be assigned. If this condition cannot be satisfied, the sequence is infeasible.

It has to be mentioned that satisfying the initial feasibility test is necessary but not a sufficient condition for feasibility of the sequence. If the sequence can satisfy the first and second phases of the initial feasibility check, the feasibility of the sequence and assigned entrance positions are evaluated using a simulation based procedure depending on the layout of the departure holding area.

6.3.2 FCFS

In FCFS method, the aircraft are sequenced in non-decreasing order of their arrival times to the departure holding area. Therefore, the take-off sequence σ is chosen so that $\mathrm{at}_{\sigma(1)} \leq \cdots \leq \mathrm{at}_{\sigma(n)}$. Moreover, take-off times are defined by applying constraints (6.3) and (6.2). The FC algorithm is used to evaluate the feasibility of the sequence. In the case of obtaining an infeasible sequence because of considering the CTOT as hard constraints, the possibility of the modification of the sequence for satisfying the feasibility conditions is investigated. The aim is to create a feasible solution with the minimum deviation from the original FCFS sequence. It has to be mentioned that only the last runway entrance is taken into account for generating the FCFS sequence. Therefore, the FCFS sequences for all three configurations of the holding area are the same.

6.3.3 Dynamic programming

Our proposed dynamic programming algorithm has n main stages where n is the number of available aircraft to be sequenced, plus an initial stage containing a single dummy node s and a final stage containing a single dummy node t. Each transition from one stage to the next one corresponds to the take-off of one aircraft and the movement of aircraft to different holding points. It has some resemblance to the approach used by Balakrishnan and Chandran (2007) for scheduling landings with a constraint on the number of positions an aircraft can shift relative to the FCFS landing sequence. For conciseness, we only consider Layout A because Layout B and Layout C are treated analogously.

Our dynamic programming algorithm has states (r_1, r_2, m_1, t_1, t_2, t_3), where r_1, r_2 and m_1 are the aircraft in holding positions R_1, R_2, and M_1, respectively, and t_1, t_2 and t_3 are the last three aircraft to have taken off. A zero value of a state variable indicates the absence of an aircraft in a holding position or that there are less than three departed aircraft. A function value (r_1, r_2, m_1, t_1, t_2, t_3) defines the minimum take-off time of the last aircraft among all partial solutions achieving the state (r_1, r_2, m_1, t_1, t_2, t_3), where ties are broken in favour of the smallest total waiting time.

Consider state (r_1, r_2, m_1, t_1, t_2, t_3) where r_1, r_2 and m_1 are nonzero, and aircraft q_1 is the first aircraft in the queue to enter the holding area. The four possible alternative state transitions arise as follows.

- Aircraft r_1 takes off, aircraft m_1 moves from position M_1 to position R_1 and aircraft q_1 moves from the queue to position M_1 to give a new state (m_1, r_2, q_1, r_1, t_1, t_2).
- Aircraft r_1 takes off, aircraft m_1 moves from position M_1 to position R_1 but aircraft q_1 stays in the queue to give a new state (m_1, r_2, 0, r_1, t_1, t_2).
- Aircraft r_2 takes off, aircraft q_1 moves from the queue to position R_2 to give a new state (r_1, q_1, m_1, r_2, t_1, t_2).
- Aircraft r_2 takes off, and aircraft q_1 stays in the queue to give a new state (r_1, 0, m_1, r_2, t_1, t_2).

We implement the dynamic program by finding a shortest path in a network, where the nodes correspond to the states and the arcs correspond to state transitions. Given $f(r_1, r_2, m_1, t_1, t_2, t_3)$, it is straightforward to use equations 6.1 and 6.2 to compute the take-off time of the next aircraft r_1 or r_2, or to discover that the state transition is impossible. Because the aircrafts take-off slot is missed or the aircraft weight class is heavy and cannot enter to the runway via position R_2.

Since overtaking is not permitted, the holding section is the only area where runway controllers can re-sequence departing aircraft. To model the problem of sequencing departing aircraft in the holding area, we construct a network that defines allowable movements of aircraft in the departure runway holding section. The network implicitly defines the state transitions of our dynamic programming algorithm. The aim is to investigate the extent to which different network designs allow a sufficient re-sequencing of the aircraft in order to produce high-quality take-off schedules. Moreover, the impacts of the holding area layouts on runway throughput as well as minimizing delay are studied as the secondary objectives. Thus, a dynamic programming algorithm is designed for creating take-off schedules based on a given network structure.

TABLE 6.2: Number of the nodes at each network for different layouts

	Layout A		Layout B		Layout C	
Number of Aircraft	15	20	15	20	15	20
Basic Network	1.8×10^8	7.2×10^{10}	5.0×10^8	2.3×10^{11}	3.4×10^{12}	3.9×10^{16}
DP Network	1.7×10^4	4.8×10^4	8.0×10^4	3.4×10^5	2.9×10^6	1.6×10^7

Each directed arc from stage u to stage $u + 1$, for $u = 1, \ldots, n - 1$, defines the aircraft in position u of the take-off sequence. Each directed path from s to t represents a feasible take-off sequence. To provide a basis for evaluating the power of dynamic programming, we consider a basic network in which each feasible state transition creates a new node. The DP network is generated by eliminating identical nodes in the basic network. Table 6.2 shows that the DP network is significantly smaller than the basic network.

The DP network for 10 aircraft is shown in Figure 6.4. Aircraft are labelled $1, 2, \ldots, 10$ according to non-decreasing order of their arrival time into the holding area. Each node is a state with two types of information. The first row displays the aircraft in positions R_1, R_2 and M_1, respectively, where a blank entry indicates that the corresponding holding position is empty. In addition, the most recent departed aircraft is shown in the first position of the second row, the second most

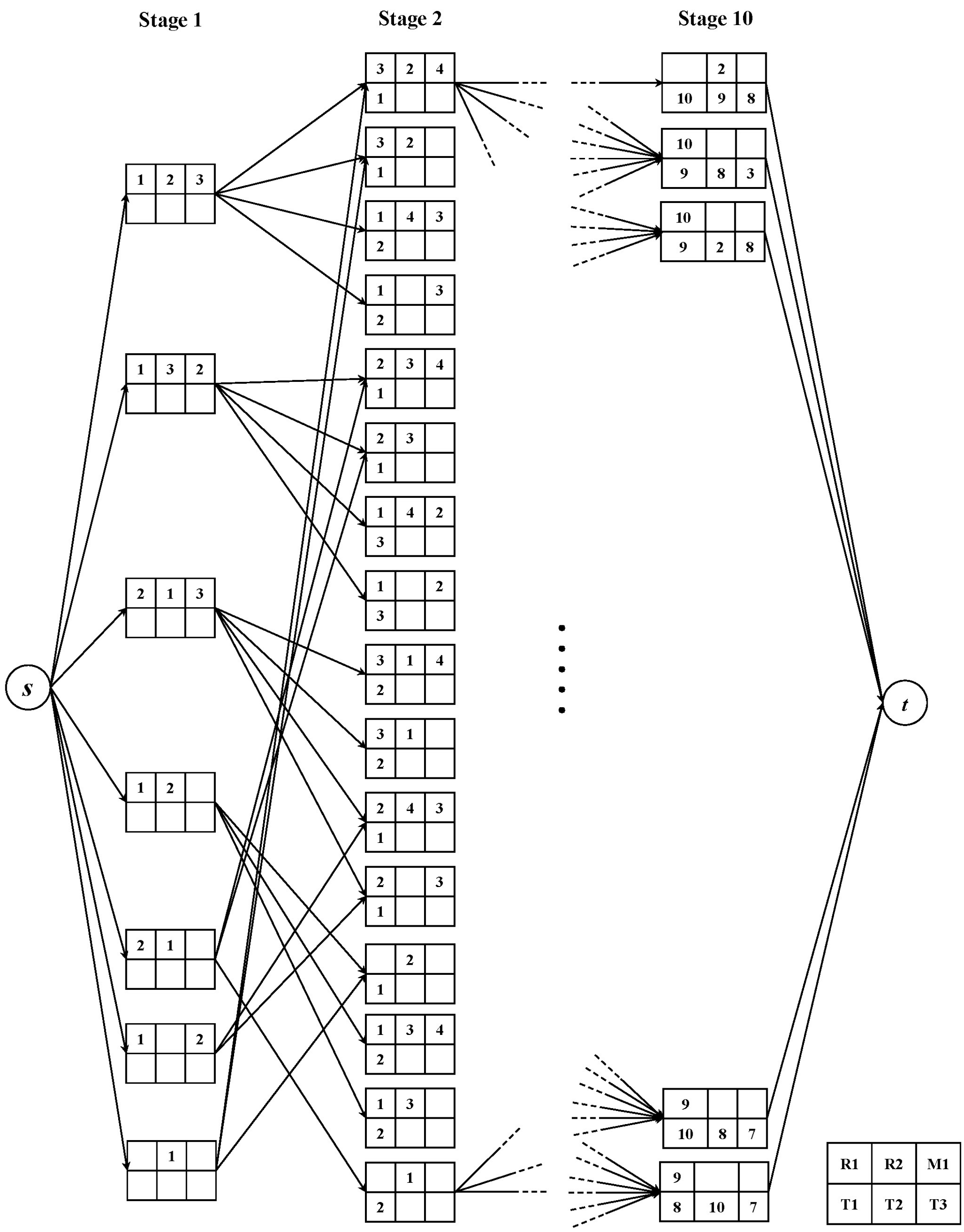

FIGURE 6.4: DP network for Layout A with 10 aircraft

recent departed aircraft is shown in the second position of the second row, and the third most recent departed aircraft is shown in the third position of the second row. Again, blank entries are allowed if less than three aircraft have departed.

The first stage corresponds to all of the feasible combinations of aircraft placement into the holding positions at the start of the process. A source node s, and a sink

Table 6.3: Generating nodes in stage $u+1$ in regarding to stage u based on Layout A, (R_1,R_2,M_1)

Stage u		Stage $u+1$		Take-off aircraft
Type	State	Type	State	
S_1	(i,j,k)	S_1	(k,j,l)	i
		S_2	$(k,j,0)$	i
		S_1	(i,l,k)	j
		S_4	$(i,0,k)$	j
S_2	$(i,j,0)$	S_3	$(0,j,0)$	i
		S_1	(i,l,m)	j
		S_2	$(i,l,0)$	j
S_3	$(0,j,0)$	S_1	(m,l,n)	j
		S_2	$(m,l,0)$	j
		S_3	$(0,l,0)$	j
S_4	$(i,0,k)$	S_1	(k,m,l)	i
		S_3	$(k,0,l)$	i

node t are also introduced to represent the beginning and the end of the sequence. Each arc joins a node at one stage to a node at the next stage and corresponds to the decision that one particular aircraft takes off, and one or more aircraft move to a different position in the holding section or from the queueing section to the holding section. Each arc and each stage of the network are generated based on the given state transition as shown in Table 6.3 for Layout A, Table 6.4 for Layout B and Table B.1, B.2 and B.3 in the appendix B for Layout C. These tables display all possible transitions from the current state to the next state for the three layouts. Each state is defined based on occupancy of holding points in the holding section. Four different types of configurations for Layout A and Layout B and fourteen different type of configurations for Layout C can be defined. Aircraft's indices are such that $\mathrm{at}_i \leq \mathrm{at}_j \leq \mathrm{at}_k \leq \mathrm{at}_l \leq \mathrm{at}_m \leq \mathrm{at}_n \leq \mathrm{at}_o \leq \mathrm{at}_p \leq \mathrm{at}_q \leq \mathrm{at}_r$.

The idea behind the dynamic programming is to find an s-t path in the network to maximize the runway throughput which is measured by the maximum take-off time. Total waiting time is used to break the ties among any identical solutions and it can be also considered as the second objective. Take-off times are subject to timing constraints (CTOT slots and arrival time into the holding area), separation (aircraft type and departure route) and layout constraints as defined by

TABLE 6.4: Generating nodes in stage $u+1$ in regarding to stage u based on Layout B, (R_1,R_2,M_1,M_2)

Stage u		Stage $u+1$		Take-off aircraft
Type	State	Type	State	
S_1	(i,j,k,l)	S_1	(k,j,l,m)	i
		S_2	$(k,j,l,0)$	i
		S_1	(i,m,k,l)	j
		S_5	$(i,0,k,l)$	j
S_2	$(i,j,k,0)$	S_3	$(k,j,0,0)$	i
		S_1	(i,m,k,n)	j
		S_2	$(i,m,k,0)$	j
S_3	$(i,j,0,0)$	S_4	$(0,j,0,0)$	i
		S_1	(i,m,n,o)	j
		S_2	$(i,m,n,0)$	j
		S_3	$(i,m,0,0)$	j
S_4	$(0,j,0,0)$	S_1	(n,m,o,p)	j
		S_2	$(n,m,o,0)$	j
		S_3	$(n,m,0,0)$	j
		S_4	$(0,m,0,0)$	j
S_5	$(i,0,k,l)$	S_1	(k,n,l,m)	i
		S_5	$(k,0,l,m)$	i

the network. Since the DP network remains the same for a given value of n, it is generated and stored off-line. The dynamic programming algorithm calls the network when required.

The dynamic programming algorithm needs to keep the take-off time of the three departure aircraft and the take-off time of the candidate aircraft for take-off in addition to the total waiting time aircraft in the partial sequence for each node. Since all the possible feasible sequence have been considered in the network, it can be concluded that DP finds the optimal solution with respect to minimizing the take-off time of the last aircraft in the sequence. The experimental results are presented in Section 6.4.

6.3.4 Descent local search

In this section, a descent local search algorithm is introduced to sequence departure flights in the holding area. The initial solution for the descent local search (DLS) is constructed based on the FCFS method which is considered as a current solution. Solutions represented as a take-off sequence of aircraft can be defined as $\sigma = (\sigma(1), \ldots, \sigma(n))$. The *best improvement* strategy has been used when searching the neighbourhoods. The DLS algorithm is based on one type of neighbourhood move. The neighbourhood move, called *insert*, has been considered which has similar concept of the insert move used in ID algorithm in Chapter 5. It compromises all feasible sequences that can be obtained from the current solution by removing an aircraft from its current position and inserting it into a new position in the sequence. Then the best feasible neighbourhood is selected by using the FC algorithm. If it improves the current solution, it replaces the current solution and search continues. If the best neighbourhood does not improve the current solution, then the algorithm terminates with a local optimal solution. The experimental results are shown in Section 6.4.

6.3.5 Beam search

The next proposed method for sequencing and scheduling departure flight in the holding area is Beam Search (BS). The BS is a heuristic method originated from branch-and-bound. Beam search saves the computational time and memory space by pruning branches of the search tree by applying local and global evaluations. The increment cost of the adding a new take-off aircraft to the partial sequence is measured by local evaluation. Therefore, it gives us a *local view* of the node. The global evaluation attempts to estimate the minimum total cost of the best solution that can be reached from the current node. Thus, it provides a *global view* of the node. This approach attempts to maximize the probability of finding a good solution with minimal effort.

The tree represents the construction of a partial sequence where each node corresponds to a new take-off aircraft. Set of starting nodes S in Level 0 includes all possible combinations of available aircraft in the holding area for take-off. Each node at Level 0 is considered to generate the offspring nodes at level 1 in the tree. Offspring nodes are generated based on the transition tables (see Tables 6.3, 6.4 and Tables B.1, B.2 and B.3 in the Appendix A) provided for each layout configuration.

All nodes at Level 1 are selected as nodes to branch from. Child nodes branching from the same parent node compete with each other based on the local evaluation cost function which is also called filtering. A subset of α value of the best nodes based on the local evaluation is selected for each parent node; α is the so-called *filter width*. Later, selected filter nodes are evaluated based on global evaluation cost function. The global evaluation procedure selects the most promising β nodes for branching in Level 2; β is the so-called *beam width* and the remaining nodes are pruned out permanently. This procedure is repeated until all the aircraft are sequenced. Number of the level is equal to the number of the aircraft, n, to be sequenced.

If the number of nodes available for local evaluation and global evaluation are fewer than α and β, respectively, then all of these nodes are chosen. It has to be mentioned that selected nodes in the local evaluation step should satisfy the CTOT constraints as well as assignment of Heavy aircraft to the last runway entrance. A larger beam width β causes the fewer nodes to be pruned and more chance of reaching the high quality solution at the expense of higher run-time.

The following four criteria have been used for the local evaluation:

$\mathrm{PT}_{\mathrm{max}}$: take-off time of the last aircraft (C_{max}) scheduled in the partial sequence;

ATT: average take-off time of the aircraft scheduled in the partial sequence;

TWT: total waiting time of aircraft scheduled in the partial sequence;

BSID: total balance between the number of aircraft from each SID category in the partial sequence relative to whole set of aircraft. The aim is to minimize deviation of percentage of the sequenced aircraft from each category in the partial sequence relative to total number of aircraft in each category from percentage of number of aircraft in the partial sequence relative to the total number of aircraft.

Minimizing the $T_{\max}$ which can be also called $C_{\max}$ is chosen as the main objective for the global evaluation and the total waiting time of the aircraft according to (6.1) is selected as the second objective. In order to perform the global evaluation, descent local search method which is explained in Section 6.3.4 has been used to construct the rest of the partial sequence. First, unscheduled aircraft are sequenced using the FCFS algorithm and the FCFS sequence is considered as an initial solution for the DLS. Then, the DLS obtain the local optimal solution for the rest of the sequence. The descent local search algorithm aims to estimate the minimum $C_{\max}$ of the sequence that can be reached from the current node.

6.4 Computational experiment

6.4.1 Random test data

In order to evaluate the performance of the developed algorithms, six data sets have been randomly generated. Each data set includes 10 problem instances. Problem instances are generated for the experimental analysis based on the following parameters in regard to the opinion of the air traffic control experts. Each problem instance includes 20 aircraft. In practice, less than 15 aircraft typically need to be considered simultaneously for sequencing in the departure holding area.

We assume that the inter-arrival times between successive aircraft reaching the holding area are exponentially distributed with arrival rate $\lambda = 1/75, 1/80$ and $1/85$, where the data set with $\lambda = 1/75$ has the highest-density traffic volume and data set with $\lambda = 1/85$ has the lowest-traffic density volume. Moreover, CTOTs are assigned to 20% of flights in Set_1, Set_3 and Set_5 and 40% of flights in Set_2, Set_4 and Set_6 (Table 6.5). If the CTOT is needed to be assigned to aircraft j, it is generated based on a uniform distribution defined as $U(at_j, at_j - 300)$ with the probability of 25% and another uniform distribution as $U(at_j + 600, at_j)$ with the probability of 75%.

Five weight classes of aircraft including Heavy (H), Upper medium (U), Lower medium (M), Small (S) and Light (L) have been considered. In addition, three departure routes (SID) are taken into account. SID categories are assigned to

TABLE 6.5: Departure data sets

	Set_1	Set_2	Set_3	Set_4	Set_5	Set_6
λ	1/75	1/80	1/85	1/75	1/80	1/85
% of CTOT	20	20	20	40	40	40

each aircraft with equal probabilities. Aircraft weight classes are assigned to each aircraft based on the probabilities given in Table 6.6.

TABLE 6.6: Weight class of departure flights

	H	U	M	S	L
Probability	0.35	0.15	0.55	0.04	0.01

The generator does not guarantee that the resulting data set has a feasible schedule. So the feasibility of each problem instance has been checked using introduced dynamic programming algorithm and any problem instance with no optimal solution has been ignored.

6.4.2 Experimental design

All the algorithms are implemented in MS Visual C++ 2008 and run on a PC with a dual core, 2.13GHz and 2GB RAM. We report on the comparison of the proposed methods using the generated test data in Section 6.4.3. We refer to the first-come first-served, descent local search, beam search and dynamic programming as FCFS, Algorithm DLS, Algorithm BS and Algorithm DP, respectively.

The minimum separation time between each pair of aircraft has been calculated based on weight classes of two aircraft and their SID routes. However, airspeed has not been taken into account. In the following, we talk about tuning of the beam search parameters regarding same performance statistics.

Beam search parameters

Beam search parameters, α and β, need to be empirically tuned for achieving acceptable performance. Considering the transition tables (Tables 6.3, 6.4 and Tables B.1, B.2 and B.3 in Appendix B), the maximum value for the filter width parameter, α, in Layout A, Layout B and Layout C are 4, 4, and 17, respectively. Therefore, performance of the BS algorithm for $\alpha = 2, 3, 4$ for Layout A and B and $\alpha = 2, 3, \ldots, 8$ for Layout C have been tested using data Set_1. Moreover, the effects of the beam width parameter on the performance of the BS algorithm have been investigated for $\beta = 80, 140, 200, 260, 320, 380$.

TABLE 6.7: Average PI of Algorithm BS relative to FCFS for Layout A

α	Local Evaluation Criteria	β					
		80	140	200	260	320	380
2	Average	17.11	17.64	17.74	17.74	17.81	17.89
	PT_{max}	17.68	18.31	18.64	18.64	18.64	18.64
	ATT	17.68	18.31	18.64	18.64	18.64	18.64
	TWT	15.55	15.85	15.57	15.57	15.85	16.17
	BSID	17.55	18.10	18.10	18.10	18.10	18.10
3	Average	19.45	19.60	19.75	19.75	19.67	19.60
	PT_{max}	19.87	19.88	19.88	19.88	19.58	19.58
	ATT	19.87	19.88	19.88	19.88	19.58	19.58
	TWT	19.06	19.33	19.92	19.92	20.20	19.92
	BSID	19.00	19.32	19.32	19.32	19.32	19.32
4	Average	19.87	20.85	20.85	20.85	20.85	20.85
	PT_{max}	19.87	20.85	20.85	20.85	20.85	20.85
	ATT	19.87	20.85	20.85	20.85	20.85	20.85
	TWT	19.87	20.85	20.85	20.85	20.85	20.85
	BSID	19.87	20.85	20.85	20.85	20.85	20.85

TABLE 6.8: Average CT of four local evaluation criteria of beam search for Layout A (in seconds)

α	β					
	80	140	200	260	320	380
2	0.33	0.52	0.63	0.76	0.92	1.20
3	0.51	0.80	1.11	1.40	1.70	2.32
4	0.58	0.88	1.21	1.59	1.88	2.55

Tables 6.7, 6.9 and 6.11 show the Average percentage improvement (PI) of the beam search for ten test data relative to FCFS solutions for various combinations of the α and β parameters, respectively for Layout A, Layout B and Layout C. The average PI for four local evaluation functions including PT_{max}, ATT, TWT and BSID as well as the average PI of these four criteria have been presented in these tables. The average computation times of beam search algorithms in seconds

for Layout A, Layout B and Layout C are also displayed in Tables 6.8, 6.10 and 6.12, respectively.

TABLE 6.9: Average PI of Algorithm BS relative to FCFS for Layout B

α	Local Evaluation Criteria	β = 80	140	200	260	320	380
2	Average	18.59	19.18	19.12	19.41	19.19	19.37
	PT_{max}	18.95	19.57	19.57	19.57	19.57	19.57
	ATT	18.95	19.57	19.57	19.57	19.57	19.57
	TWT	17.56	18.29	17.65	18.56	18.25	18.65
	BSID	18.91	19.28	19.68	19.96	19.36	19.68
3	Average	21.00	21.60	21.68	22.07	22.14	22.07
	PT_{max}	20.84	21.45	21.45	22.06	22.06	22.06
	ATT	20.84	21.45	21.45	22.06	22.06	22.06
	TWT	20.84	21.73	21.73	22.06	22.06	22.06
	BSID	21.46	21.77	22.10	22.10	22.38	22.10
4	Average	21.16	21.16	22.05	22.05	22.05	22.05
	PT_{max}	21.16	21.16	22.05	22.05	22.05	22.05
	ATT	21.16	21.16	22.05	22.05	22.05	22.05
	TWT	21.16	21.16	22.05	22.05	22.05	22.05
	BSID	21.16	21.16	22.05	22.05	22.05	22.05

Performance of the BS algorithm for four local evaluation criteria is almost similar for various combinations of the filter width and beam width parameters in Layout A other than TWT criterion which has the worst performance among all other criteria for $\alpha = 2$ (see Table 6.7). Based on the average PI and the computation time of the BS algorithms, it can be concluded that $\alpha = 4$ and $\beta = 140$ can give the best results in less than one second run-time (see Table 6.8). Since for $\alpha = 4$, all the child nodes branching from the same parent node are accepted, the local evaluation procedure has no effect on the solution.

TABLE 6.10: Average CT of four local evaluation criteria of beam search for Layout B (in seconds)

α	β = 80	140	200	260	320	380
2	0.43	0.66	0.84	1.01	1.17	1.51
3	0.61	1.00	1.36	1.73	2.06	2.87
4	0.76	1.10	1.52	1.98	2.41	3.06

Table 6.9 shows that the performance of TWT criterion is not as good as the other criteria for $\alpha = 2$ and BSID function has the best performance among the other criteria for $\alpha = 3$ for Layout B. By increasing the filter width from 3 to 4, the average performance of the algorithm has been reduced for $\beta = 260, 320, 380$. It can be due to the fact that some good solutions have been ignored because of the low acceptance rate of the nodes in global evaluation. Table 6.10 shows the computation times of different combinations of the filter width and beam width for Layout B. Based on the analysis, the conclusion is that the BS algorithm can provide the best solution using $\alpha = 3$ and $\beta = 140$ after one second run-time (see Table 6.10) for layout B.

Based on the analysis of filter width and beam width parameters presented in Table 6.11 and Figure 6.5, the performance of the BSID criterion for $\alpha = 2, 3$ is significantly better than the other local evaluation objectives. Considering the filter more than 4 does not show an improvement on the average PI and even can reduce the performance of the algorithm in some cases. According to the average PI and computation time of the BS algorithm, $\alpha = 4$ and $\beta = 80$ can reach the best solution in one second run-time (see Table 6.12). The TWT local evaluation function looks more promising than the other criteria.

6.4.3 Results

Tables 6.13, 6.14 and 6.15 present the summary performance of the schedule obtained from first-come first-served, descent local search, beam search and dynamic programming approaches described in the earlier section for Layout A, Layout B and Layout C, respectively. These tables display the average performance of algorithms using six data sets where each data set consists of 10 problem instances. Result tables for layout A and Layout B include the take-off times provided by the FCFS, DLS, BS and DP algorithms. The take-off times produced by the FCFS, DLS and BS algorithms are included in result table for Layout C. Percentage improvements (PI) of DLS, BS and DP solutions relative to the FCFS schedule are given in these tables. Results of the algorithms have been verified to make sure that the models meet the requirements and specifications of the problem and they perform as expected.

TABLE 6.11: Average PI of Algorithm BS relative to FCFS for Layout C

α	Local Evaluation Criteria	β 80	140	200	260	320	380
2	Average	17.91	19.38	20.25	20.08	20.39	20.65
	$\mathrm{PT_{max}}$	17.66	18.85	19.89	20.19	20.51	20.84
	ATT	17.66	18.85	19.89	20.19	20.51	20.84
	TWT	15.12	18.64	19.28	18.06	18.63	19.03
	BSID	21.19	21.19	21.92	21.87	21.92	21.89
3	Average	21.44	22.59	22.92	22.94	22.91	22.94
	$\mathrm{PT_{max}}$	20.54	22.17	22.54	22.54	22.54	22.54
	ATT	20.54	22.18	22.22	22.22	22.22	22.22
	TWT	21.11	21.88	22.77	23.15	23.39	22.85
	BSID	23.54	24.15	24.15	23.86	23.49	24.15
4	Average	23.62	23.62	23.70	23.86	23.86	23.86
	$\mathrm{PT_{max}}$	23.53	23.53	23.53	23.86	23.86	23.86
	ATT	23.53	23.53	23.53	23.86	23.86	23.86
	TWT	23.86	23.86	23.86	23.86	23.86	23.86
	BSID	23.54	23.54	23.87	23.87	23.87	23.87
5	Average	23.45	23.61	23.69	23.78	23.69	23.69
	$\mathrm{PT_{max}}$	23.20	23.53	23.86	23.86	23.86	23.86
	ATT	23.20	23.53	23.86	23.86	23.86	23.86
	TWT	23.86	23.86	23.53	23.53	23.53	23.53
	BSID	23.53	23.53	23.53	23.86	23.53	23.53
6	Average	23.28	23.53	23.53	23.77	23.77	23.77
	$\mathrm{PT_{max}}$	23.20	23.53	23.53	23.86	23.86	23.86
	ATT	23.20	23.53	23.53	23.86	23.86	23.86
	TWT	23.53	23.53	23.53	23.52	23.52	23.52
	BSID	23.20	23.53	23.53	23.86	23.86	23.86
7	Average	23.61	23.61	23.53	23.78	23.78	23.86
	$\mathrm{PT_{max}}$	23.53	23.53	23.53	23.86	23.86	23.86
	ATT	23.53	23.53	23.53	23.86	23.86	23.86
	TWT	23.86	23.86	23.53	23.86	23.86	23.86
	BSID	23.53	23.53	23.53	23.53	23.53	23.86
8	Average	23.53	23.53	23.53	23.53	23.86	23.86
	$\mathrm{PT_{max}}$	23.53	23.53	23.53	23.53	23.86	23.86
	ATT	23.53	23.53	23.53	23.53	23.86	23.86
	TWT	23.53	23.53	23.53	23.53	23.86	23.86
	BSID	23.53	23.53	23.53	23.53	23.86	23.86

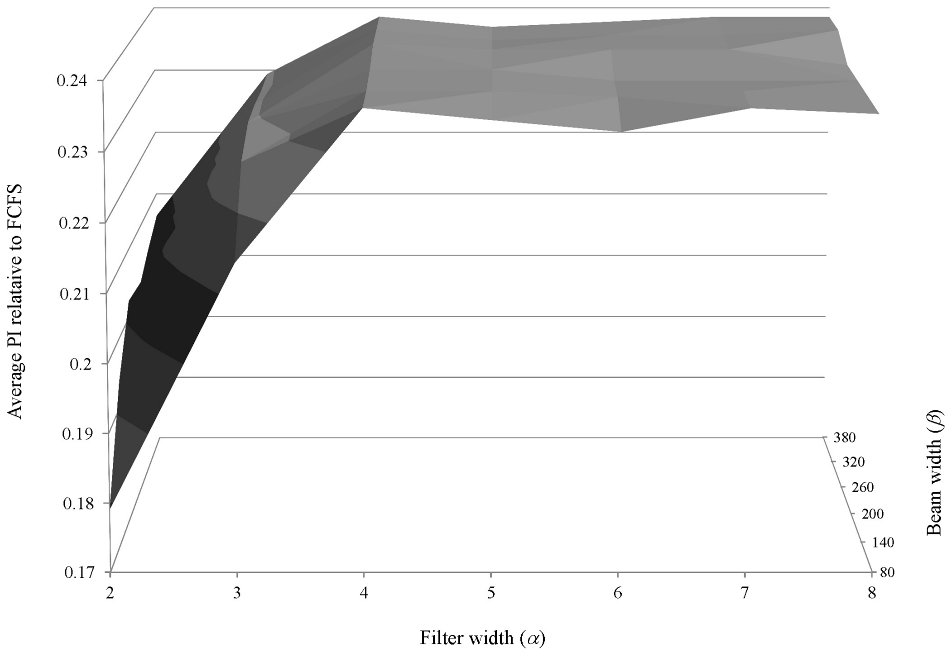

FIGURE 6.5: Average PI of Algorithm BS relative to FCFS for Layout C

TABLE 6.12: Average CT of four local evaluation criteria of beam search for Layout C (in seconds)

α	β					
	80	140	200	260	320	380
2	0.56	0.90	1.15	1.43	1.67	2.29
3	0.72	1.20	1.67	2.13	2.65	3.62
4	1.01	1.54	2.18	2.77	3.42	4.67
5	1.17	1.82	2.43	3.21	3.96	5.33
6	1.22	2.25	2.84	3.47	4.36	5.81
7	1.28	2.32	3.00	3.70	3.70	6.03
8	1.32	2.47	2.81	3.66	4.65	6.11

Algorithms are compared based on the following performance statistics:

$T_{\max}$: take-off time of the last aircraft in a given set of aircraft in seconds;

PI: percentage improvement in the solution objective function relative to the solution of the FCFS algorithm;

TWT: total waiting time of the solution take-off sequence;

CT: computation time in seconds for scheduling a given set of aircraft;

Table 6.13: Results for Layout A

Data set	λ	CTOT	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4, \quad \beta = 140$				Algorithm DP			
				$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
Set_1	1/75	20%	$\text{PT}_{\max}$	1,842	10,108	1,536	16.1	6,982	0.003	1,446	20.9	6,022	0.9	1,422	22.1	5,806	9
			ATT							1,446	20.9	6,022	0.9				
			TWT							1,446	20.9	6,022	0.9				
			BSID							1,446	20.9	6,022	0.9				
Set_2	1/75	40%	$\text{PT}_{\max}$	1,842	10,108	1,560	14.7	7,285	0.002	1,500	18.0	6,556	0.9	1,470	19.5	6,232	8
			ATT							1,500	18.0	6,556	0.9				
			TWT							1,500	18.0	6,556	0.9				
			BSID							1,500	18.0	6,556	0.9				
Set_3	1/80	20%	$\text{PT}_{\max}$	1,884	8,891	1,674	11.2	6,908	0.002	1,458	22.5	4,631	0.9	1,446	23.1	4,481	8
			ATT							1,458	22.5	4,631	0.9				
			TWT							1,458	22.5	4,631	0.9				
			BSID							1,458	22.5	4,631	0.9				
Set_4	1/80	40%	$\text{PT}_{\max}$	1,884	8,891	1,732	8.2	7,223	0.002	1,470	21.8	4,823	0.9	1,446	23.1	4,517	8
			ATT							1,470	21.8	4,823	0.9				
			TWT							1,470	21.8	4,823	1.0				
			BSID							1,470	21.8	4,823	0.9				
Set_5	1/85	20%	$\text{PT}_{\max}$	1,824	6,129	1,626	10.6	4,239	0.002	1,578	13.2	3,849	0.8	1,566	13.9	3,729	8
			ATT							1,578	13.2	3,849	0.8				
			TWT							1,578	13.2	3,849	0.8				
			BSID							1,578	13.2	3,849	0.8				
Set_6	1/85	40%	$\text{PT}_{\max}$	1,824	6,129	1,644	9.6	4,419	0.002	1,584	12.9	4,131	0.8	1,578	13.2	4,023	8
			ATT							1,584	12.9	4,131	0.8				
			TWT							1,584	12.9	4,131	0.8				
			BSID							1,584	12.9	4,131	0.8				

The first column in the result table shows the problem instance number. The second column lists the local evaluation function for the beam search algorithm. $T_{\max}$ and total waiting time of the schedule generated by each algorithm is also presented in the result table. Moreover, the computation time of the DLS, BS and DP methods are presented in Tables 6.13, 6.14 and 6.15. The complete results for 60 randomly generated problem instances for each layout are provided in Appendix B.

Tables 6.13, 6.14 and 6.15 show an improvement in the main objective relative to the FCFS schedule across all methods. Dynamic programming provides the optimal solution with respect to the $T_{\max}$. DP has the highest computation time compared to the other approaches and DLS has the shortest one. For three configurations of the departure holding area, BS approach obtains the solution in one second.

Table 6.13 presents our computational results for Layout A. Since the filter width is equal to the maximum number of the child nodes branching from each parent

Table 6.14: Results for Layout B

Data set	λ	CTOT	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4$, $\beta = 140$				Algorithm DP			
				$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
Set_1	1/75	20%	$\mathrm{PT}_{\max}$	1,842	10,108	1,524	16.8	6,854	0.003	1,434	21.4	5,614	1.0	1,404	23.2	5,440	73
			ATT							1,434	21.4	5,614	1.0				
			TWT							1,428	21.7	5,572	1.0				
			BSID							1,428	21.7	5,596	1.0				
Set_2	1/75	40%	$\mathrm{PT}_{\max}$	1,842	10,108	1,536	16.1	7,027	0.003	1,464	19.8	5,956	1.0	1,446	20.9	5,800	69
			ATT							1,464	19.8	5,956	1.0				
			TWT							1,476	19.2	6,082	1.0				
			BSID							1,482	19.0	6,166	0.9				
Set_3	1/80	20%	$\mathrm{PT}_{\max}$	1,884	8,891	1,614	14.4	6,453	0.003	1,440	23.3	4,229	1.0	1,416	24.6	4,205	69
			ATT							1,440	23.3	4,229	1.0				
			TWT							1,434	23.7	4,247	1.0				
			BSID							1,440	23.4	4,379	1.0				
Set_4	1/80	40%	$\mathrm{PT}_{\max}$	1,884	8,891	1,690	10.5	6,920	0.002	1,446	23.1	4,259	1.0	1,428	24.0	4,193	73
			ATT							1,446	23.1	4,259	1.0				
			TWT							1,440	23.3	4,241	1.0				
			BSID							1,464	22.1	4,443	1.0				
Set_5	1/85	20%	$\mathrm{PT}_{\max}$	1,824	6,129	1,626	10.6	4,239	0.003	1,566	13.9	3,813	0.9	1,566	13.9	3,627	71
			ATT							1,566	13.9	3,813	0.9				
			TWT							1,566	13.9	3,897	0.9				
			BSID							1,566	13.9	3,891	0.9				
Set_6	1/85	40%	$\mathrm{PT}_{\max}$	1,824	6,129	1,626	10.6	4,203	0.002	1,578	13.2	4,053	0.9	1,578	13.2	3,867	67
			ATT							1,578	13.2	4,053	0.9				
			TWT							1,578	13.2	4,047	0.8				
			BSID							1,578	13.2	4,083	0.9				

node, the performance of the beam search does not depend on the local evaluation criteria. BS can find the optimal solution with respect to the $T_{\max}$ in 70% of the problem instances, whereas DLS has reached the optimal solution in 11% of the cases (see Table B.4-B.9). In case of higher number of aircraft with CTOT constraints, total waiting time increases and PI decreases.

In Layout B, BS method can find the optimal solution for 80% of the problem instances and DLS approach can obtain optimal solution for 15% of test data (see Table B.10-B.15). However, beam search has almost the same performance regarding four local evaluation functions across all data sets other than TWT which has the best performance in data Set_1, Set_3 and Set_4; and BSID which has the worst performance in data Set_2. Considering average run-time, dynamic programming has the highest value which is 70 seconds. In general, PI in Layout B is 4.1% higher than PI in Layout A. Moreover, total waiting time in Layout B is 5.8% less than total waiting time in Layout A (see Table 6.14).

The required memory space for storing the DP networks of 20 aircraft for Layout A, Layout B and Layout C are 2MB, 16MB and 750MB, respectively. Therefore,

TABLE 6.15: Results for Layout C

Data set	λ	CTOT	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4, \quad \beta = 140$			
				$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
Set_1	1/75	20%	$\text{PT}_{\max}$	1,842	10,108	1,452	20.5	6,022	0.004	1,398	23.5	5,362	1.1
			ATT							1,398	23.5	5,362	1.1
			TWT							1,392	23.9	5,320	1.1
			BSID							1,398	23.5	5,386	1.1
Set_2	1/75	40%	$\text{PT}_{\max}$	1,842	10,108	1,506	17.8	6,481	0.003	1,446	20.9	5,813	0.9
			ATT							1,446	20.9	5,812	0.9
			TWT							1,446	20.9	5,839	0.9
			BSID							1,452	20.5	5,980	1.0
Set_3	1/80	20%	$\text{PT}_{\max}$	1,884	8,891	1,596	15.5	6,141	0.003	1,422	24.4	4,090	0.9
			ATT							1,422	24.4	4,090	1.0
			TWT							1,416	24.7	4,248	0.9
			BSID							1,416	24.7	4,266	0.9
Set_4	1/80	40%	$\text{PT}_{\max}$	1,884	8,891	1,684	10.9	6,776	0.002	1,422	24.3	4,121	0.9
			ATT							1,422	24.3	4,121	0.9
			TWT							1,428	23.9	4,144	0.9
			BSID							1,422	24.3	4,169	1.0
Set_5	1/85	20%	$\text{PT}_{\max}$	1,824	6,129	1,608	11.5	3,945	0.003	1,548	14.8	3,633	0.8
			ATT							1,548	14.8	3,633	0.8
			TWT							1,548	14.8	3,741	0.8
			BSID							1,560	14.2	3,849	0.8
Set_6	1/85	40%	$\text{PT}_{\max}$	1,824	6,129	1,626	10.6	4,101	0.002	1,566	13.9	3,903	0.9
			ATT							1,566	13.9	3,903	1.0
			TWT							1,566	13.9	4,023	0.9
			BSID							1,578	13.2	4,041	1.0

Table need 6.15 does not include the computational results of the dynamic programming method because of the limited computational resources. We expect the run-time of the DP method for Layout C to be more than one hour. In average, runway throughput of Layout C is higher than the one in Layout B.

6.5 Concluding remarks

In this chapter, we have defined departure scheduling problem with the focus on the impact of the departure holding area configurations on take-off scheduling. Three types of constraints, introduced in this problem, include calculated take-off time constraints, separation time constraints and layout constraints. Then, we have proposed the dynamic programming, descent local search and beam search

Table 6.16: Average PI of Algorithm BS relative to FCFS for three layouts

Data set	T_{max}			TWT		
	Layout A	Layout B	Layout C	Layout A	Layout B	Layout C
Set_1	20.85	21.60	23.62	39.82	43.76	46.36
Set_2	17.98	21.60	23.62	34.69	39.59	41.53
Set_3	22.50	23.45	24.53	46.16	51.28	53.15
Set_4	21.83	22.90	24.19	46.16	51.28	53.18
Set_5	13.24	13.88	14.69	30.87	30.87	32.02
Set_6	12.90	13.20	13.71	24.54	24.54	26.68
Average	**18.22**	**19.44**	**20.72**	**37.04**	**40.26**	**42.15**

algorithms to solve the problem. Finally, experimental test have been performed on the generated data sets.

Our experimental results show a strong performance of the beam search and dynamic programming algorithms relative to the take-off schedule based on the FCFS sequence. The average run-time of the BS method with $n = 20$ is 1 seconds. Short run-time of the algorithm, and ability to find either the optimal solution or near-optimal solution, make the BS approach suitable for implementation in practice. Performance of the DLS approach is not as good as BS or DP algorithms. But it is a desirable heuristic for estimating the objective function value in global evaluation step of beam search algorithm due to the very short run-time of DLS. Results show that the performance of the BS approach is acceptable in terms of the computation times and quality of solutions. If computation time is not an issue, then the DP method can provide the best results for Layout A and Layout B.

A comparison of Layout A and B and C indicates that adding extra holding points and entrance to the runway can increase the utilization of take-off runway as it gives more flexibility to re-order aircraft. They also reduce the total waiting time of the aircraft in the holding area (Table 6.16 and Figure 6.6). In average, value of the T_{max} and TWT increase by allocating the CTOT to more number of aircraft.

The proposed solution methods can be used to compare more complicated layouts of departure holding section. Increasing the flexibility of movements in departure holding area does not necessarily increase the runway throughput. Finding a layout that offers the highest opportunity for increasing throughput, while not causing

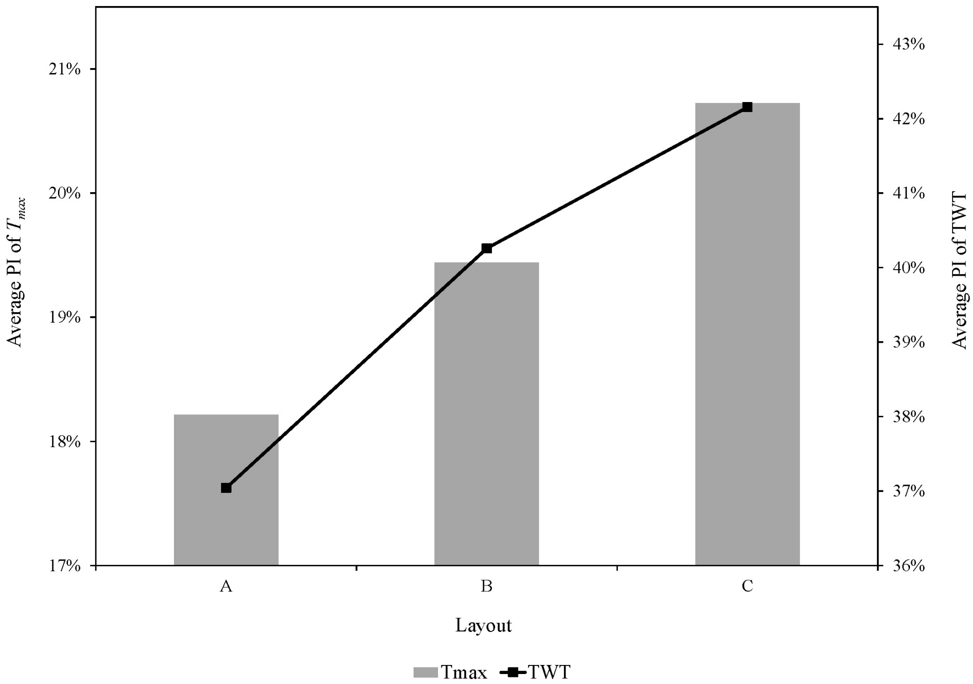

FIGURE 6.6: Average PI of Algorithm BS relative to FCFS for three layouts

air traffic controllers to make a large number of complex ground movements of aircraft remains an interesting challenge for the future.

Chapter 7

Concluding Remarks

7.1 Conclusion

This thesis looked at finding optimal or near-optimal solutions for landing and take-off problems. The aim was to design algorithms to obtain good solutions in a very short running time so that they can be used in practice. The proposed algorithms for scheduling of aircraft landings were discussed in Chapter 5. Chapter 6 presented the proposed algorithms for take-off scheduling problem.

We have introduced models and algorithms for the static/off-line aircraft landing problem and the dynamic/on-line version of the problem. A special feature of our model is the multi-objective approach that takes into account the interests of the various stakeholders that are affected by the scheduling of an airport's runway.

Dynamic programming, iterated descent and simulated annealing algorithms were proposed for the solution of the static problem. The dynamic problem was tackled using iterated descent and dynamic programming algorithms providing the solution method for periodically updating the previous schedule based on rolling horizon approach. The length of the freeze time and the time horizon were investigated. The freeze time specifies the period during which changes to the schedule are forbidden, while the time horizon defines which aircraft are to be considered whenever the previous schedule is updated.

A thorough computational evaluation is performed using data from Heathrow airport. Results for the static problem show that all of the proposed algorithms are effective in achieving an efficient runway throughput compared to FCFS which is the common approach in practice. In addition, the algorithms are capable of finding solutions that perform well in terms of minimizing delay and minimizing the cost of extra fuel which is used to achieve the desired landing schedule. Iterated descent has the advantage of faster and more predictable run-times, and is therefore preferred to dynamic programming and simulated annealing.

Iterated descent and dynamic programming are used as scheduling algorithms for the dynamic problem. Experimental tests have been performed for using 10 days data from Heathrow airport and 40 days randomly generated problem instances based on the characteristics of the real data. Results show that it is worthwhile to have a time horizon of at least 30 minutes, and a slightly longer time horizon is recommended if the freeze time is more than 5 minutes.

The second part of the research is dedicated to the aircraft take-off problem. The aim is to investigate the impact of the runway holding area on scheduling of aircraft take-off. Three common layouts of the departure holding area are considered. A hierarchical objective function has been considered. Minimizing the maximum take-off time (or runway throughput) is considered as the main objective function. Minimizing the total waiting time is chosen as the second objective.

The proposed methods obtain the sequence and schedule of take-off flights subject to timing, minimum separation and holding area configuration constraints. Performance of the algorithms and comparisons of their effectiveness for three holding area layouts are investigated using randomly generated test data. Two entrances to the departure runway and one holding position between two entrances are considered for Layout A. Layout B has the same configuration as Layout A in addition to an extra holding position between runway entrances. Layout B has more flexibility in moving the position of aircraft in the sequence relative to Layout A. Experimental results show that adding one holding position improves the quality of take-off sequence. Departure holding area with three entrances to the runway and one holding position between entrances are designed in Layout C. Layout C is more complicated than the other layouts in terms of sequencing take-off flights.

Quality of the solutions for Layout C has better quality than the other two layouts in terms of runway throughput and waiting times in the holding area.

Descent local search is the fastest algorithm among the other but the solutions are not as good as dynamic programming and beam search. Although dynamic programming method can find the optimal solution, it is expensive in terms of run-time for complicated configurations such as Layout B and Layout C. Beam search has the advantage of being fast and obtaining optimal solution or near-optimal solution. Some remarks for ALP and ATP are mentioned in the following section.

7.2 Extensions and future work

In this thesis, we have studied aircraft landing and departure scheduling problems. An static and a dynamic version of landing problems are considered. The effect of holding area on take-off scheduling is considered in take-off problem. Both problems are formulated as multi-objective models subject to operational and safety constraints. Various algorithms such as dynamic programming, iterated descent, simulated annealing, descent local search and beam search have been developed to provide optimal or near-optimal solutions in a very short time. Experimental tests show promising results for solving these problems both in terms of solution quality and run-time.

There are some directions for the future work based on this thesis which are as below.

- Integration of aircraft landing problem with en-route and taxi-in scheduling problems can be studied.
- Approaching routes constraints can also be considered for arrival scheduling problem.
- Since departure holding area has the main effect on the sequencing of aircraft take-off, more complicated configurations such as parallel holding positions

can be studied. Moreover, overtaking constraint for some holding positions can be relaxed.

- Impact of the increasing rate of the departure flights with CTOT constraint on take-off sequencing problem can be investigated.
- Patterns and relations between feasible deviations of the departure sequence from FCFS sequence for different departure layout configurations can be studied.

Appendix A

ALP: Experimental Results

Table A.1: Complete Heathrow data: static environment: Weights $(0.3, 0.5, 0.1, 0.1)$

			Algorithm DP				Algorithm ID				Algorithm SA			
#	n	Obj.	PI	TD	ND	CT	PI	TD	ND	CT	PI	TD	ND	CT
		Overall	2.89				2.89				2.89			
		LT_{max}	0.18				0.18				0.18			
P1	22	ALT	0.15	22	11	0.06	0.14	16	10	0.08	0.15	22	11	0.05
		TW	7.96				7.96				7.96			
		EF	6.27				6.27				6.27			
		Overall	9.02				8.96				9.02			
		LT_{max}	0.42				0.42				0.42			
P2	41	ALT	0.28	42	18	0.39	0.27	34	16	0.34	0.28	40	19	0.41
		TW	48.49				48.49				49.49			
		EF	11.65				11.65				11.65			
		Overall	1.20				1.20				1.20			
		LT_{max}	0.40				0.04				0.04			
P3	21	ALT	0.05	18	8	0.06	0.05	32	13	0.08	0.05	18	8	0.6
		TW	0.00				0.00				0.00			
		EF	8.03				8.03				8.03			
		Overall	3.51				3.51				3.51			
		LT_{max}	0.15				0.15				0.15			
P4	42	ALT	0.08	40	18	0.14	0.07	74	32	0.38	0.08	40	18	0.63
		TW	0.00				0.00				0.00			
		EF	15.22				15.22				15.22			
		Overall	0.33				0.33				0.33			
		LT_{max}	0.00				0.00				0.00			
P5	20	ALT	0.01	6	5	0.36	0.01	6	5	0.11	0.01	6	5	0.09
		TW	0.00				0.00				0.00			
		EF	10.77				10.77				10.77			
		Overall	0.69				0.69				0.69			
		LT_{max}	0.00				0.00				0.00			
P6	42	ALT	0.01	22	13	1.16	0.01	26	19	0.45	0.01	14	12	0.53
		TW	0.00				0.00				0.00			
		EF	4.64				4.64				4.64			
		Overall	7.08				7.08				7.05			
		LT_{max}	0.00				0.00				0.00			
P7	84	ALT	0.09	74	30	7.73	0.09	90	36	2.36	0.08	84	36	8.94
		TW	N/A				N/A				0.00			
		EF	24.27				24.27				23.65			
		Overall	3.53				3.52				3.53			
		LT_{max}	0.11				0.11				0.11			
Average		ALT	0.09	32	15	1.41	0.09	40	19	0.54	0.09	32	16	1.53
		TW	9.41				9.41				8.06			
		EF	11.55				11.55				11.46			

TABLE A.2: Complete Heathrow data: static environment: Weights $(0.2, 0.4, 0.3, 0.1)$

			Algorithm DP				Algorithm ID				Algorithm SA			
#	n	Obj.	PI	TD	ND	CT	PI	TD	ND	CT	PI	TD	ND	CT
P1	22	Overall	3.53	22	11	0.16	3.53	24	13	0.11	2.89	14	9	0.05
		LT_{max}	0.18				0.18				0.18			
		ALT	0.15				0.14				0.15			
		TW	7.96				7.96				7.96			
		EF	6.27				6.27				6.27			
P2	41	Overall	13.71	42	18	0.53	13.71	32	16	0.45	13.71	36	16	0.41
		LT_{max}	0.42				0.42				0.42			
		ALT	0.28				0.27				0.28			
		TW	48.49				48.49				48.49			
		EF	11.65				11.65				11.65			
P3	22	Overall	1.51	18	8	0.14	1.51	34	16	0.06	1.51	18	8	0.05
		LT_{max}	0.04				0.04				0.04			
		ALT	0.05				0.05				0.05			
		TW	0.00				0.00				0.00			
		EF	8.03				8.03				8.03			
P4	42	Overall	4.34	40	18	0.28	4.33	56	25	0.50	4.33	40	18	0.58
		LT_{max}	0.15				0.15				0.15			
		ALT	0.08				0.07				0.08			
		TW	0.00				0.00				0.00			
		EF	15.22				15.22				15.22			
P5	20	Overall	0.43	6	5	0.45	0.43	6	5	0.09	0.43	6	5	0.05
		LT_{max}	0.00				0.00				0.00			
		ALT	0.01				0.01				0.01			
		TW	0.00				0.00				0.00			
		EF	10.77				10.77				10.77			
P6	42	Overall	0.87	22	13	1.33	0.87	38	19	0.56	0.87	14	12	0.56
		LT_{max}	0.00				0.00				0.00			
		ALT	0.01				0.01				0.01			
		TW	0.00				0.00				0.00			
		EF	4.64				4.64				4.64			
P7	84	Overall	8.56	82	33	7.74	8.56	98	41	2.48	8.51	106	39	7.28
		LT_{max}	0.00				0.00				0.00			
		ALT	0.09				0.08				0.09			
		TW	0.00				0.00				0.00			
		EF	23.66				23.66				23.52			
Average		Overall	4.71	33	15	1.52	4.71	41	19	0.61	4.70	33	11	1.28
		LT_{max}	0.11				0.11				0.11			
		ALT	0.09				0.09				0.09			
		TW	8.06				8.06				8.06			
		EF	11.46				11.46				11.44			

Table A.3: Complete Heathrow data: static environment: Weights $(0.4, 0.6, 0.0, 0.0)$

			Algorithm DP				Algorithm ID				Algorithm SA			
#	n	Obj.	PI	TD	ND	CT	PI	TD	ND	CT	PI	TD	ND	CT
P1	22	Overall	0.36	48	17	0.16	0.36	62	21	0.06	0.36	26	13	0.05
		LT_{max}	0.36				0.36				0.36			
		ALT	0.08				0.19				0.17			
		TW	−239.32				−195.81				−52.34			
		EF	5.55				7.31				6.79			
P2	41	Overall	0.51	102	34	0.55	0.51	78	25	0.16	0.51	62	19	0.34
		LT_{max}	0.51				0.51				0.51			
		ALT	0.18				0.29				0.24			
		TW	−35.05				−12.03				7.33			
		EF	9.70				12.08				11.58			
P3	22	Overall	0.04	40	13	0.16	0.04	40	13	0.05	0.04	6	2	0.05
		LT_{max}	0.04				0.04				0.04			
		ALT	0.05				0.04				0.04			
		TW	−0.01				N/A				0.00			
		EF	1.94				6.94				6.50			
P4	42	Overall	0.19	94	31	0.31	0.19	56	24	0.36	0.19	20	9	0.53
		LT_{max}	0.19				0.19				0.19			
		ALT	0.03				0.06				0.07			
		TW	N/A				0.00				0.00			
		EF	9.27				13.34				13.48			
P5	20	Overall	0.00	22	10	0.45	0.00	26	13	0.05	0.00	0	0	0.05
		LT_{max}	0.00				0.00				0.00			
		ALT	−0.11				−0.09				0.00			
		TW	0.00				N/A				0.00			
		EF	−98.32				86.13				0.00			
P6	42	Overall	0.00	68	30	1.34	0.00	0	0	0.23	0.00	0	0	0.53
		LT_{max}	0.00				0.00				0.00			
		ALT	−0.28				0.00				0.00			
		TW	−1388.64				0.00				0.00			
		EF	−101.37				0.00				0.00			
P7	84	Overall	0.00	204	69	7.77	0.00	96	31	0.69	0.00	0	0	7.59
		LT_{max}	0.00				0.00				0.00			
		ALT	−0.43				−0.08				0.00			
		TW	N/A				N/A				0.00			
		EF	−124.88				−23.25				0.00			
Average		Overall	0.16	83	28	1.53	0.16	51	18	0.23	0.16	16	6	1.31
		LT_{max}	0.16				0.16				0.16			
		ALT	−0.08				0.06				0.07			
		TW	−415.75				−51.96				−6.43			
		EF	−42.59				−9.96				5.48			

TABLE A.4: Complete Heathrow data: static environment: Weights $(1.0, 0.0, 0.0, 0.0)$

			Algorithm DP				Algorithm ID				Algorithm SA			
#	n	Obj.	PI	TD	ND	CT	PI	TD	ND	CT	PI	TD	ND	CT
P1	22	Overall	0.26	38	15	0.16	0.26	50	19	0.05	0.26	18	8	0.05
		LT_{max}	0.36				0.36				0.36			
		ALT	0.19				0.19				0.19			
		TW	−115.31				−222.40				−134.68			
		EF	7.58				7.13				7.13			
P2	41	Overall	0.40	64	25	0.53	0.40	94	32	0.14	0.40	94	29	0.29
		LT_{max}	0.51				0.51				0.51			
		ALT	0.32				0.32				0.32			
		TW	−11.28				−34.76				−43.69			
		EF	12.76				12.68				12.89			
P3	22	Overall	0.05	18	8	0.16	0.05	24	12	0.08	0.05	14	6	0.03
		LT_{max}	0.04				0.04				0.04			
		ALT	0.05				0.05				0.05			
		TW	0.00				N/A				N/A			
		EF	8.03				8.03				8.03			
P4	42	Overall	0.12	74	26	0.31	0.12	86	32	0.48	0.12	48	19	0.53
		LT_{max}	0.19				0.19				0.19			
		ALT	0.08				0.08				0.08			
		TW	N/A				N/A				N/A			
		EF	14.37				14.97				14.97			
P5	20	Overall	0.01	10	5	0.44	0.01	10	5	0.13	0.01	10	5	0.05
		LT_{max}	0.00				0.00				0.00			
		ALT	0.01				0.01				0.01			
		TW	N/A				N/A				N/A			
		EF	8.12				8.12				8.12			
P6	42	Overall	0.01	50	19	1.33	0.01	56	22	0.47	0.01	42	17	0.48
		LT_{max}	0.00				0.00				0.00			
		ALT	0.02				0.02				0.02			
		TW	−861.45				−774.07				−637.21			
		EF	5.03				4.72				5.45			
P7	84	Overall	0.05	88	34	7.64	0.05	126	46	2.08	0.05	96	34	7.14
		LT_{max}	0.00				0.00				0.00			
		ALT	0.09				0.09				0.09			
		TW	N/A				N/A				N/A			
		EF	23.70				23.66				24.02			
Average		Overall	0.13	49	19	1.51	0.13	64	24	0.49	0.13	46	17	1.22
		LT_{max}	0.16				0.16				0.16			
		ALT	0.11				0.11				0.11			
		TW	−247.01				−343.74				−271.86			
		EF	11.37				11.33				11.51			

Table A.5: Heathrow data: dynamic environment: Weights $(0.3, 0.5, 0.1, 0.1)$

			FCFS	Algorithm DP						Algorithm ID			
#	n	Obj.	Sep.	PI	TD	ND	CT	CT_{max}	Sep.	PI	TD	ND	Sep.
D01	616	Overall ALT TW EF	52,267	14.93 0.06 82.61 21.64	322	152	0.04	1.94	51,303	14.93 0.06 82.61 21.64	332	156	51,303
D02	631	Overall ALT TW EF	53,813	22.07 0.13 −278.96 29.37	482	245	0.13	6.11	52,535	22.07 0.13 −278.96 29.37	510	248	52,535
D03	617	Overall ALT TW EF	52,557	25.37 0.14 76.41 31.64	522	221	0.45	58.60	51,226	25.38 0.14 78.2 31.6	586	244	51,226
D04	635	Overall ALT TW EF	54,199	23.53 0.14 53.48 28.26	492	221	0.08	7.22	52,897	23.77 0.14 53.48 28.57	474	222	52,897
D05	644	Overall ALT TW EF	54,891	19.78 0.09 75.58 26.4	386	196	0.06	3.30	53,732	19.78 0.09 75.58 26.4	436	215	53,732
D06	639	Overall ALT TW EF	54,731	26.93 0.15 54.26 32.26	528	206	0.18	14.5	53,404	24.12 0.14 37.72 29.53	504	201	53,499
D07	642	Overall ALT TW EF	54,903	30.66 0.19 81.43 36.65	572	231	0.04	2.37	53,670	30.87 0.19 81.43 36.91	602	240	53,647
D08	632	Overall ALT TW EF	53,811	25.31 0.13 82.28 30.20	522	196	0.05	2.15	52,599	25.31 0.13 82.28 30.20	512	190	52,599
D09	642	Overall ALT TW EF	54,456	15.78 0.08 41.72 20.37	501	209	0.02	0.76	53,178	16.50 0.09 39.43 21.34	500	196	53,154
D10	648	Overall ALT TW EF	55,885	33.25 0.27 73.63 37.77	712	302	0.05	1.48	53,835	33.31 0.27 75.26 37.79	776	307	53,811
Average		Overall ALT TW EF	54,151	23.76 0.14 34.24 29.46	504	218	0.11	9.84	52,838	23.76 0.14 32.70 29.33	523	222	52,840

Table A.6: Heathrow data: dynamic environment: Weights $(0.2, 0.4, 0.3, 0.1)$

			FCFS	Algorithm DP						Algorithm ID			
#	n	Obj.	Sep.	PI	TD	ND	CT	Max CT	Sep.	PI	TD	ND	Sep.
D01	616	Overall ALT TW EF	52,267	16.57 0.06 95.03 21.61	314	152	0.03	1.49	51,303	16.57 0.06 95.03 21.61	334	157	51,303
D02	631	Overall ALT TW EF	53,813	22.26 0.12 −4.95 27.26	422	223	0.14	6.22	52,607	22.26 0.12 −4.95 27.27	458	235	52,607
D03	617	Overall ALT TW EF	52,557	29.08 0.13 98.57 30.48	522	236	0.43	52.9	51,274	29.19 0.14 94.60 30.89	564	240	51,250
D04	635	Overall ALT TW EF	54,199	26.49 0.14 54.58 213.00	460	213	0.08	6.52	52,897	26.73 0.14 54.58 28.48	490	221	52,897
D05	644	Overall ALT TW EF	54,891	22.02 0.09 79.27 26.36	384	195	0.05	3.09	53,731	22.02 0.09 79.27 26.36	424	203	53,731
D06	639	Overall ALT TW EF	54,731	30.83 0.15 58.06 31.90	486	196	0.15	12.10	53,404	26.71 0.14 40.80 29.29	504	206	53,499
D07	642	Overall ALT TW EF	54,903	33.15 0.18 86.45 34.94	558	229	0.04	2.34	53,718	33.36 0.18 86.45 35.19	620	255	53,695
D08	632	Overall ALT TW EF	53,811	28.98 0.12 84.27 28.03	536	216	0.05	2.17	52,574	30.45 0.13 84.27 30.00	524	197	52,598
D09	642	Overall ALT TW EF	54,456	17.17 0.08 78.63 19.42	483	207	0.02	0.72	53,226	17.90 0.08 78.63 20.33	496	205	53,202
D10	648	Overall ALT TW EF	55,885	36.68 0.26 91.76 36.35	690	289	0.05	1.55	53,955	36.75 0.26 91.76 36.44	696	285	53,931
Average		Overall ALT TW EF	54,151	26.32 0.13 72.17 46.93	486	216	0.10	8.91	52,869	26.19 0.13 70.04 28.59	511	220	52,871

TABLE A.7: Heathrow data: dynamic environment: Weights $(0.4, 0.6, 0.0, 0.0)$

			FCFS	Algorithm DP						Algorithm ID			
#	n	Obj.	Sep.	PI	TD	ND	CT	Max CT	Sep.	PI	TD	ND	Sep.
D01	616	Overall	52,267	0.03	390	178	0.03	1.56	51,472	0.03	426	192	51,472
		ALT		0.07						0.07			
		TW		−649.48						−942.65			
		EF		21.51						21.44			
D02	631	Overall	53,813	0.06	544	251	0.14	6.18	52,560	0.06	586	262	52,560
		ALT		0.13						0.13			
		TW		−699.22						−654.94			
		EF		28.95						28.79			
D03	617	Overall	52,557	−0.09	1,015	315	0.27	26.40	51,540	0.07	714	271	51,227
		ALT		−0.19						0.14			
		TW		−2775.29						−60.25			
		EF		−69.28						31.14			
D04	635	Overall	54,199	0.07	716	288	0.07	6.57	52,944	0.07	702	274	52,920
		ALT		0.15						0.16			
		TW		−45.80						−53.04			
		EF		27.07						28.37			
D05	644	Overall	54,891	0.05	618	250	0.05	3.09	53,780	0.05	660	260	53,780
		ALT		0.10						0.10			
		TW		−502.52						−647.02			
		EF		25.75						25.68			
D06	639	Overall	54,731	0.08	686	242	0.15	12.20	53,429	0.08	702	246	53,428
		ALT		0.16						0.16			
		TW		6.04						8.09			
		EF		31.88						31.72			
D07	642	Overall	54,903	0.10	846	282	0.04	2.34	53,671	0.10	824	278	53,671
		ALT		0.21						0.21			
		TW		−66.88						−123.81			
		EF		38.71						38.55			
D08	632	Overall	53,811	0.04	781	285	0.06	2.28	52,768	0.07	750	255	52,744
		ALT		0.09						0.14			
		TW		−20.88						−20.46			
		EF		19.06						29.39			
D09	642	Overall	54,456	0.04	747	273	0.02	0.80	53,323	0.04	784	273	53,274
		ALT		0.09						0.10			
		TW		−550.52						−673.50			
		EF		19.39						20.53			
D10	648	Overall	55,885	0.13	1,014	354	0.05	1.51	53,956	0.13	1,064	349	53,980
		ALT		0.29						0.29			
		TW		−78.41						−114.83			
		EF		38.42						37.94			
Average		Overall	54,151	0.05	736	272	0.09	6.29	52,944	0.07	721	266	52,906
		ALT		0.11						0.15			
		TW		−538.30						−328.24			
		EF		18.14						29.37			

Table A.8: Random data, Set_1: dynamic environment: Weights $(0.3, 0.5, 0.1, 0.1)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G01	611	Overall	12.72	340	195	0.03	0.78	12.72	370	205
		ALT	0.06					0.06		
		TW	−119.26					−119.26		
		EF	18.46					18.46		
G02	608	Overall	13.48	430	220	0.06	1.92	13.48	444	224
		ALT	0.07					0.07		
		TW	22.59					22.59		
		EF	18.34					18.34		
G03	611	Overall	13.11	352	196	0.04	1.84	13.11	392	204
		ALT	0.07					0.07		
		TW	49.71					49.71		
		EF	18.07					18.07		
G04	608	Overall	16.04	436	237	0.02	0.70	16.04	450	241
		ALT	0.08					0.08		
		TW	24.42					24.42		
		EF	21.49					21.49		
G05	608	Overall	17.09	352	196	0.05	2.99	17.09	372	200
		ALT	0.09					0.09		
		TW	54.85					54.95		
		EF	18.73					18.73		
G06	614	Overall	15.65	454	227	0.06	4.93	15.65	482	236
		ALT	0.09					0.09		
		TW	73.86					76.00		
		EF	19.75					19.72		
G07	621	Overall	12.11	366	214	0.02	0.46	12.11	376	203
		ALT	0.06					0.06		
		TW	44.44					44.44		
		EF	16.46					16.46		
G08	615	Overall	17.10	428	217	0.07	4.96	17.10	444	221
		ALT	0.09					0.09		
		TW	73.57					73.57		
		EF	20.51					20.51		
G09	616	Overall	9.73	370	207	0.02	1.12	9.73	380	203
		ALT	0.05					0.05		
		TW	−96.92					−96.92		
		EF	13.88					13.88		
G10	620	Overall	14.36	394	207	0.16	7.67	14.36	420	217
		ALT	0.06					0.06		
		TW	95.18					95.18		
		EF	18.78					18.78		
Average		Overall	14.14	392	212	0.05	2.74	14.14	413	215
		ALT	0.07					0.07		
		TW	22.24					22.47		
		EF	18.44					18.44		

Table A.9: Random data, Set_2: dynamic environment: Weights $(0.3, 0.5, 0.1, 0.1)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G11	629	Overall	19.57	524	255	0.09	4.92	19.57	572	279
		ALT	0.12					0.12		
		TW	57.08					57.48		
		EF	23.31					23.29		
G12	635	Overall	19.65	544	275	0.02	0.47	19.62	564	283
		ALT	0.12					0.12		
		TW	50.51					56.44		
		EF	23.71					23.51		
G13	638	Overall	14.91	522	265	0.16	11.00	14.91	536	271
		ALT	0.09					0.09		
		TW	49.62					49.62		
		EF	17.77					17.77		
G14	628	Overall	20.29	454	247	0.03	1.86	20.29	514	271
		ALT	0.11					0.11		
		TW	90.33					90.33		
		EF	23.62					23.62		
G15	628	Overall	16.68	478	236	0.06	2.80	16.63	506	233
		ALT	0.10					0.10		
		TW	35.59					39.70		
		EF	21.12					20.97		
G16	634	Overall	14.65	436	234	0.03	0.69	14.65	446	228
		ALT	0.07					0.07		
		TW	−9.47					−9.47		
		EF	19.94					19.94		
G17	635	Overall	29.01	536	267	0.1	5.53	29.01	552	263
		ALT	0.18					0.18		
		TW	78.10					78.10		
		EF	31.32					31.32		
G18	648	Overall	36.90	522	254	0.08	2.32	36.90	562	260
		ALT	0.25					0.25		
		TW	94.33					94.33		
		EF	38.98					38.98		
G19	634	Overall	16.13	450	239	0.02	0.53	16.18	484	245
		ALT	0.09					0.09		
		TW	66.85					66.85		
		EF	20.54					20.62		
G20	634	Overall	27.75	534	273	0.06	2.06	27.77	588	291
		ALT	0.19					0.19		
		TW	67.81					67.95		
		EF	31.72					31.74		
Average		Overall	21.56	500	255	0.07	3.22	21.55	532	262
		ALT	0.13					0.13		
		TW	58.08					59.13		
		EF	25.20					25.18		

Table A.10: Random data, Set$_3$: dynamic environment: Weights $(0.3, 0.5, 0.1, 0.1)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G21	655	Overall	26.25	520	259	0.25	21.10	26.25	578	274
		ALT	0.18					0.18		
		TW	71.51					71.51		
		EF	29.06					29.06		
G22	653	Overall	25.76	580	284	0.11	5.27	25.76	684	319
		ALT	0.19					0.19		
		TW	81.16					81.16		
		EF	28.18					28.18		
G23	653	Overall	28.28	436	231	0.05	1.80	27.89	470	239
		ALT	0.20					0.20		
		TW	73.14					74.08		
		EF	31.30					30.78		
G24	650	Overall	20.52	518	268	0.17	22.50	20.54	578	281
		ALT	0.13					0.13		
		TW	53.58					53.58		
		EF	24.44					24.47		
G25	659	Overall	27.82	528	267	0.18	7.36	27.82	552	271
		ALT	0.18					0.18		
		TW	70.72					70.72		
		EF	33.09					33.09		
G26	654	Overall	30.17	578	292	0.14	5.60	30.23	642	305
		ALT	0.20					0.20		
		TW	83.74					83.74		
		EF	30.09					30.17		
G27	652	Overall	24.27	562	287	0.02	0.54	24.27	608	301
		ALT	0.17					0.17		
		TW	68.37					68.37		
		EF	25.57					25.57		
G28	657	Overall	43.64	662	308	0.18	6.07	43.64	740	329
		ALT	0.38					0.38		
		TW	94.46					94.46		
		EF	46.49					46.49		
G29	652	Overall	34.12	506	259	0.09	3.15	34.12	596	281
		ALT	0.25					0.25		
		TW	84.53					84.53		
		EF	38.89					38.89		
G30	655	Overall	27.31	562	285	0.12	3.48	27.31	626	312
		ALT	0.19					0.19		
		TW	67.42					67.42		
		EF	31.96					31.96		
Average		Overall	28.81	545	274	0.13	7.69	28.78	607	291
		ALT	0.21					0.21		
		TW	74.86					74.96		
		EF	31.91					31.87		

Table A.11: Random data, Set_4: dynamic environment: Weights $(0.3, 0.5, 0.1, 0.1)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G31	676	Overall	49.31	690	335	0.17	11.40	48.99	734	346
		ALT	0.46					0.46		
		TW	89.49					89.62		
		EF	47.03					46.58		
G32	660	Overall	26.11	596	312	0.07	2.98	26.11	680	334
		ALT	0.17					0.17		
		TW	58.54					58.54		
		EF	29.74					29.74		
G33	679	Overall	54.62	668	332	0.10	3.69	54.62	750	362
		ALT	0.62					0.62		
		TW	94.64					94.64		
		EF	50.55					50.55		
G34	679	Overall	55.97	724	351	0.10	1.87	55.97	800	360
		ALT	0.58					0.58		
		TW	95.47					95.47		
		EF	56.31					56.31		
G35	670	Overall	42.19	684	316	0.32	18.6	42.19	734	330
		ALT	0.36					0.36		
		TW	79.67					79.67		
		EF	43.57					43.58		
G36	675	Overall	57.28	698	324	0.86	68.7	57.21	772	342
		ALT	0.61					0.61		
		TW	95.40					95.15		
		EF	52.44					52.41		
G37	668	Overall	43.97	670	327	0.11	4.95	43.97	732	347
		ALT	0.36					0.36		
		TW	94.66					94.66		
		EF	39.98					39.98		
G38	672	Overall	34.03	628	300	0.05	1.44	33.99	658	294
		ALT	0.28					0.28		
		TW	86.70					87.65		
		EF	33.02					32.87		
G39	666	Overall	43.84	686	332	0.07	6.13	43.84	738	341
		ALT	0.39					0.39		
		TW	91.91					91.91		
		EF	41.22					41.22		
G40	666	Overall	39.46	678	311	0.12	5.74	39.26	734	328
		ALT	0.30					0.30		
		TW	85.09					85.77		
		EF	37.42					37.03		
Average		Overall	44.68	672	324	0.2	12.55	44.61	733	338
		ALT	0.41					0.41		
		TW	87.16					87.31		
		EF	43.13					43.03		

Table A.12: Random data, Set_1: dynamic environment: Weights $(0.2, 0.4, 0.3, 0.1)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G01	611	Overall	13.65	300	183	0.03	0.82	13.65	326	203
		ALT	0.05					0.05		
		TW	77.49					77.49		
		EF	17.55					17.55		
G02	608	Overall	14.67	420	219	0.06	1.96	14.67	432	218
		ALT	0.07					0.07		
		TW	58.00					58.00		
		EF	17.64					17.64		
G03	611	Overall	14.20	350	195	0.04	1.87	14.20	382	204
		ALT	0.07					0.07		
		TW	68.79					68.79		
		EF	18.05					18.05		
G04	608	Overall	17.65	434	241	0.02	0.73	17.65	422	231
		ALT	0.08					0.08		
		TW	49.44					49.44		
		EF	20.85					20.85		
G05	608	Overall	22.34	336	191	0.55	3.17	22.34	374	199
		ALT	0.09					0.09		
		TW	57.20					57.20		
		EF	18.32					18.32		
G06	614	Overall	18.18	438	222	0.58	5.07	18.18	460	232
		ALT	0.08					0.08		
		TW	84.73					84.73		
		EF	19.54					19.54		
G07	621	Overall	13.22	360	212	0.02	0.52	13.22	388	221
		ALT	0.06					0.06		
		TW	75.56					75.56		
		EF	16.37					16.37		
G08	615	Overall	20.50	416	220	0.07	5.38	20.50	434	225
		ALT	0.09					0.09		
		TW	92.77					92.77		
		EF	19.11					19.11		
G09	616	Overall	10.58	360	204	0.03	1.13	10.58	372	198
		ALT	0.05					0.05		
		TW	87.69					87.69		
		EF	13.79					13.79		
G10	620	Overall	17.36	386	205	0.17	8.10	17.36	406	208
		ALT	0.06					0.06		
		TW	97.69					97.69		
		EF	18.72					18.72		
Average		Overall	16.24	380	209	0.16	2.88	16.24	400	214
		ALT	0.07					0.07		
		TW	74.94					74.94		
		EF	17.99					17.99		

TABLE A.13: Random data, Set_2: dynamic environment: Weights $(0.2, 0.4, 0.3, 0.1)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G11	629	Overall	22.68	476	239	0.09	4.65	22.68	498	253
		ALT	0.11					0.11		
		TW	68.62					68.62		
		EF	22.28					22.28		
G12	635	Overall	22.51	528	271	0.01	0.44	22.51	564	278
		ALT	0.11					0.11		
		TW	61.36					61.36		
		EF	23.26					23.26		
G13	638	Overall	17.38	524	263	0.15	10.30	17.38	536	267
		ALT	0.09					0.09		
		TW	60.72					60.72		
		EF	17.33					17.33		
G14	628	Overall	24.92	448	244	0.03	1.72	24.92	532	270
		ALT	0.11					0.11		
		TW	96.82					96.82		
		EF	23.20					23.21		
G15	628	Overall	18.39	462	234	0.07	2.67	18.39	496	231
		ALT	0.10					0.10		
		TW	41.92					41.92		
		EF	20.91					20.91		
G16	634	Overall	15.80	424	228	0.03	0.67	15.80	454	228
		ALT	0.07					0.07		
		TW	54.13					54.13		
		EF	19.90					19.90		
G17	635	Overall	35.77	534	267	0.09	5.32	34.16	462	265
		ALT	0.18					0.18		
		TW	78.18					74.97		
		EF	31.31					29.82		
G18	648	Overall	45.44	504	249	0.08	2.29	45.44	524	250
		ALT	0.25					0.25		
		TW	96.45					96.45		
		EF	38.69					38.69		
G19	634	Overall	18.16	454	237	0.02	0.52	18.22	510	243
		ALT	0.09					0.09		
		TW	77.22					77.22		
		EF	20.38					20.46		
G20	634	Overall	31.84	522	279	0.07	1.98	31.31	576	298
		ALT	0.18					0.18		
		TW	74.80					73.78		
		EF	30.67					30.13		
Average		Overall	25.29	488	251	0.07	3.06	25.08	515	258
		ALT	0.13					0.13		
		TW	71.02					70.60		
		EF	24.79					24.60		

Table A.14: Random data, Set$_3$: dynamic environment: Weights $(0.2, 0.4, 0.3, 0.1)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G21	655	Overall	30.83	514	255	0.23	19.30	30.33	550	265
		ALT	0.18					0.18		
		TW	71.81					70.30		
		EF	29.03					28.61		
G22	653	Overall	30.04	570	282	0.11	4.80	30.04	644	297
		ALT	0.18					0.18		
		TW	86.32					86.32		
		EF	27.03					27.03		
G23	653	Overall	32.58	430	233	0.05	1.76	32.58	460	241
		ALT	0.20					0.20		
		TW	77.83					77.83		
		EF	30.13					30.13		
G24	650	Overall	22.94	482	250	0.15	19.80	22.94	546	263
		ALT	0.12					0.12		
		TW	80.72					80.72		
		EF	22.77					22.77		
G25	659	Overall	30.46	502	265	0.14	6.08	30.46	526	269
		ALT	0.17					0.17		
		TW	75.74					75.74		
		EF	32.05					32.05		
G26	654	Overall	38.96	540	283	0.15	5.35	39.01	612	302
		ALT	0.20					0.20		
		TW	86.67					86.67		
		EF	29.42					29.50		
G27	652	Overall	29.35	534	274	0.02	0.56	29.35	578	289
		ALT	0.17					0.17		
		TW	71.04					71.04		
		EF	24.40					24.40		
G28	657	Overall	50.41	638	305	0.16	2.98	50.41	684	314
		ALT	0.37					0.37		
		TW	98.12					98.39		
		EF	45.75					45.69		
G29	652	Overall	38.27	468	258	0.11	3.10	38.27	528	280
		ALT	0.24					0.24		
		TW	89.81					89.81		
		EF	38.00					38.00		
G30	655	Overall	30.27	552	285	0.12	3.48	30.25	596	302
		ALT	0.18					0.17		
		TW	83.23					82.82		
		EF	30.34					30.34		
Average		Overall	33.41	523	269	0.12	6.72	33.36	572	282
		ALT	0.20					0.20		
		TW	82.13					81.96		
		EF	30.89					30.85		

TABLE A.15: Random data, Set_4: dynamic environment: Weights $(0.2, 0.4, 0.3, 0.1)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G31	676	Overall	58.43	636	323	0.14	8.96	58.43	700	335
		ALT	0.43					0.43		
		TW	90.21					90.21		
		EF	44.10					44.10		
G32	660	Overall	29.88	560	303	0.07	2.85	30.09	640	314
		ALT	0.17					0.17		
		TW	61.08					61.12		
		EF	28.98					29.26		
G33	679	Overall	65.22	598	312	0.09	3.25	65.24	696	355
		ALT	0.60					0.60		
		TW	94.84					94.84		
		EF	49.24					49.27		
G34	679	Overall	64.99	682	347	0.09	1.73	64.99	752	350
		ALT	0.57					0.57		
		TW	97.57					97.57		
		EF	55.66					55.66		
G35	670	Overall	49.71	602	302	0.35	18.10	49.73	682	322
		ALT	0.35					0.35		
		TW	82.86					82.86		
		EF	42.57					42.59		
G36	675	Overall	69.12	632	314	0.99	77.50	69.12	678	336
		ALT	0.60					0.60		
		TW	96.28					96.28		
		EF	51.67					51.67		
G37	668	Overall	56.11	648	321	0.10	4.99	56.11	696	334
		ALT	0.35					0.35		
		TW	95.40					95.40		
		EF	38.81					38.81		
G38	672	Overall	42.90	564	285	0.04	0.96	42.90	632	306
		ALT	0.27					0.27		
		TW	90.53					90.53		
		EF	31.76					31.76		
G39	666	Overall	54.91	664	324	0.07	5.60	54.91	704	323
		ALT	0.39					0.39		
		TW	93.43					93.43		
		EF	40.47					40.47		
G40	666	Overall	49.99	644	309	0.13	5.00	49.99	692	325
		ALT	0.29					0.29		
		TW	86.80					86.80		
		EF	36.38					36.38		
Average		Overall	54.12	623	314	0.21	12.89	54.15	687	330
		ALT	0.40					0.40		
		TW	88.90					88.90		
		EF	41.96					42.00		

TABLE A.16: Random data, Set_1: dynamic environment: Weights $(0.4, 0.6, 0.0, 0.0)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G01	611	Overall	0.03	460	246	0.03	0.75	0.03	542	267
		ALT	0.06					0.06		
		TW	−1076.54					−1609.05		
		EF	17.25					17.41		
G02	608	Overall	0.04	584	269	0.06	1.94	0.04	560	255
		ALT	0.08					0.08		
		TW	−402.75					−514.06		
		EF	17.91					17.81		
G03	611	Overall	0.03	570	271	0.04	1.83	0.03	636	283
		ALT	0.07					0.07		
		TW	−5578.96					−6973.76		
		EF	17.30					17.34		
G04	608	Overall	0.04	528	265	0.02	0.73	0.04	544	271
		ALT	0.09					0.09		
		TW	−97.49					−182.11		
		EF	21.52					21.39		
G05	608	Overall	0.05	534	242	0.05	3.02	0.05	604	258
		ALT	0.10					0.10		
		TW	2.16					−20.68		
		EF	18.88					18.86		
G06	614	Overall	0.05	620	273	0.06	4.88	0.05	618	267
		ALT	0.10					0.10		
		TW	−188.17					−276.14		
		EF	20.59					20.55		
G07	621	Overall	0.03	566	272	0.02	0.46	0.03	516	248
		ALT	0.07					0.07		
		TW	−2276.33					−1980.74		
		EF	15.51					16.22		
G08	615	Overall	0.05	676	271	0.07	4.32	0.05	738	277
		ALT	0.10					0.10		
		TW	−165.84					−200.22		
		EF	19.21					19.65		
G09	616	Overall	0.02	456	236	0.03	1.09	0.02	504	240
		ALT	0.05					0.05		
		TW	−6732.31					−10463.08		
		EF	13.75					13.67		
G10	620	Overall	0.03	502	233	0.16	7.79	0.03	506	234
		ALT	0.07					0.07		
		TW	−112.47					−141.34		
		EF	18.22					17.97		
Average		Overall	0.04	550	258	0.05	2.68	0.04	577	260
		ALT	0.08					0.08		
		TW	−1662.87					−2236.12		
		EF	18.01					18.09		

Table A.17: Random data, Set_2: dynamic environment: Weights $(0.4, 0.6, 0.0, 0.0)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G11	629	Overall	0.06	668	293	0.09	4.92	0.06	712	299
		ALT	0.14					0.13		
		TW	−78.44					−102.61		
		EF	24.46					24.20		
G12	635	Overall	0.06	730	326	0.02	0.47	0.06	774	324
		ALT	0.13					0.13		
		TW	−74.14					−103.17		
		EF	23.39					23.18		
G13	638	Overall	0.05	782	326	0.17	11.70	0.05	788	319
		ALT	0.11					0.11		
		TW	−174.95					−220.77		
		EF	17.93					17.83		
G14	628	Overall	0.06	710	303	0.03	1.78	0.06	758	313
		ALT	0.12					0.12		
		TW	−50.10					−110.22		
		EF	23.22					22.71		
G15	628	Overall	0.05	710	304	0.05	1.95	0.05	716	299
		ALT	0.11					0.11		
		TW	−203.29					−242.62		
		EF	21.24					20.53		
G16	634	Overall	0.04	560	283	0.03	0.73	0.04	544	266
		ALT	0.08					0.08		
		TW	−5117.07					−5649.33		
		EF	18.97					18.76		
G17	635	Overall	0.09	664	302	0.09	5.64	0.09	704	300
		ALT	0.20					0.20		
		TW	41.58					26.68		
		EF	33.13					32.29		
G18	648	Overall	0.12	738	297	0.08	2.34	0.12	784	300
		ALT	0.26					0.26		
		TW	58.04					53.74		
		EF	39.17					39.16		
G19	634	Overall	0.05	668	292	0.02	0.53	0.05	704	297
		ALT	0.10					0.10		
		TW	−378.35					−482.44		
		EF	21.08					20.92		
G20	634	Overall	0.10	710	306	0.06	2.04	0.10	808	340
		ALT	0.20					0.20		
		TW	−18.76					−37.69		
		EF	31.79					31.46		
Average		Overall	0.07	694	303	0.06	3.21	0.07	729	306
		ALT	0.14					0.14		
		TW	−599.55					−686.84		
		EF	25.44					25.11		

TABLE A.18: Random data, Set_3: dynamic environment: Weights $(0.4, 0.6, 0.0, 0.0)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G21	655	Overall	0.09	792	331	0.22	19.70	0.09	822	342
		ALT	0.20					0.20		
		TW	−19.45					−15.99		
		EF	28.66					29.15		
G22	653	Overall	0.10	908	343	0.10	3.98	0.10	978	363
		ALT	0.21					0.21		
		TW	−41.03					−62.21		
		EF	29.12					28.99		
G23	653	Overall	0.10	790	334	0.05	1.82	0.10	774	322
		ALT	0.22					0.22		
		TW	−8.51					−13.18		
		EF	32.25					31.84		
G24	650	Overall	0.06	758	333	0.11	5.54	0.06	746	327
		ALT	0.13					0.14		
		TW	−163.24					−169.98		
		EF	23.16					22.74		
G25	659	Overall	0.09	736	326	0.16	7.24	0.09	748	343
		ALT	0.19					0.19		
		TW	−69.72					−97.38		
		EF	33.31					32.99		
G26	654	Overall	0.11	806	337	0.13	5.47	0.11	902	358
		ALT	0.23					0.23		
		TW	46.06					31.07		
		EF	31.85					31.85		
G27	652	Overall	0.09	896	362	0.02	0.54	0.09	970	368
		ALT	0.19					0.19		
		TW	−3.26					−21.35		
		EF	26.67					26.41		
G28	657	Overall	0.19	934	364	0.16	3.48	0.19	958	358
		ALT	0.39					0.40		
		TW	40.64					35.15		
		EF	46.95					47.34		
G29	652	Overall	0.12	692	312	0.09	3.14	0.12	776	332
		ALT	0.26					0.26		
		TW	−1.11					−22.49		
		EF	39.71					39.02		
G30	655	Overall	0.09	785	337	0.13	3.98	0.09	792	339
		ALT	0.19					0.19		
		TW	−43.67					−73.95		
		EF	32.10					31.87		
Average		Overall	0.10	810	338	0.12	5.49	0.10	847	345
		ALT	0.22					0.22		
		TW	−26.33					−41.03		
		EF	32.38					32.22		

TABLE A.19: Random data, Set$_4$: dynamic environment: Weights $(0.4, 0.6, 0.0, 0.0)$

			Algorithm DP					Algorithm ID		
#	n	Obj.	PI	TD	ND	CT	Max CT	PI	TD	ND
G31	676	Overall	0.24	920	370	0.17	11.30	0.24	996	385
		ALT	0.48					0.48		
		TW	77.59					74.34		
		EF	47.69					47.54		
G32	660	Overall	0.09	786	364	0.07	3.08	0.09	836	373
		ALT	0.18					0.19		
		TW	6.54					−10.45		
		EF	29.58					30.30		
G33	679	Overall	0.30	1,008	418	0.09	2.74	0.30	1,124	430
		ALT	0.64					0.63		
		TW	81.87					77.42		
		EF	51.31					51.03		
G34	679	Overall	0.28	1,030	395	0.09	1.65	0.28	1,094	396
		ALT	0.59					0.59		
		TW	72.58					68.72		
		EF	56.00					55.93		
G35	670	Overall	0.18	974	372	0.31	18.60	0.18	1,032	380
		ALT	0.37					0.37		
		TW	57.54					52.39		
		EF	43.92					43.76		
G36	675	Overall	0.30	976	372	0.95	71.60	0.30	1,034	394
		ALT	0.63					0.63		
		TW	84.66					83.15		
		EF	52.95					52.85		
G37	668	Overall	0.18	1,062	391	0.07	3.41	0.18	1,182	412
		ALT	0.38					0.38		
		TW	67.52					62.96		
		EF	40.23					40.13		
G38	672	Overall	0.14	1,010	380	0.04	0.96	0.14	1,120	417
		ALT	0.30					0.30		
		TW	41.55					35.85		
		EF	33.58					33.55		
G39	666	Overall	0.20	1,006	401	0.06	4.83	0.20	1,080	409
		ALT	0.42					0.42		
		TW	63.41					58.99		
		EF	41.87					41.85		
G40	666	Overall	0.15	848	348	0.14	5.27	0.15	888	350
		ALT	0.32					0.31		
		TW	62.71					60.99		
		EF	37.87					37.66		
Average		Overall	0.21	962	381	0.20	12.34	0.21	1,039	395
		ALT	0.43					0.43		
		TW	61.60					56.44		
		EF	43.50					43.46		

Appendix B

ATP: Experimental Results

TABLE B.1: Generating nodes in stage u+1 in regarding to stage u based on Layout C, (R_1,R_2,R_3,M_1,M_2)

Stage u		Stage $u+1$		Take-off aircraft
Type	State	Type	State	
S_1	(i,j,k,l,m)	S_1	(l,j,k,m,n)	i
		S_2	$(l,j,k,0,m)$	i
		S_3	$(l,j,k,m,0)$	i
		S_1	(i,m,k,l,n)	j
		S_5	$(i,0,k,l,m)$	j
		S_3	$(i,m,k,l,0)$	j
		S_1	(i,j,f,l,m)	k
		S_6	$(i,j,0,l,m)$	k
S_2	$(i,j,k,0,m)$	S_4	$(0,j,k,0,m)$	i
		S_1	(i,m,k,n,o)	j
		S_3	$(i,m,k,n,0)$	j
		S_2	$(i,m,k,0,n)$	j
		S_{10}	$(i,m,k,0,0)$	j
		S_2	$(i,j,n,0,m)$	k
		S_9	$(i,j,0,0,m)$	k
S_3	$(i,j,k,l,0)$	S_{10}	$(l,j,0,m,0)$	i
		S_8	$(i,0,l,l,0)$	j
		S_1	(i,j,l,l,o)	k
		S_3	$(i,j,l,l,0)$	k
S_4	$(0,j,k,0,m)$	S_1	(n,m,k,o,p)	j
		S_3	$(n,m,k,o,0)$	j
		S_2	$(n,m,k,0,o)$	j
		S_{10}	$(n,m,k,0,0)$	j
		S_{13}	$(0,m,k,0,0)$	j
		S_4	$(0,m,k,0,n)$	j
		S_4	$(0,j,n,0,m)$	k
		S_{11}	$(0,j,0,0,m)$	k
S_5	$(i,0,k,l,m)$	S_1	(l,n,k,m,o)	i
		S_5	$(l,0,k,m,n)$	i
		S_3	$(l,n,k,m,0)$	i
		S_8	$(l,0,k,m,0)$	i
		S_5	$(i,0,n,l,m)$	k
		S_7	$(i,0,0,l,m)$	k
S_6	$(i,j,0,l,m)$	S_1	(l,j,o,m,n)	i
		S_6	$(l,j,0,m,n)$	i
		S_9	$(l,j,0,0,m)$	i
		S_1	(i,m,o,l,n)	j
		S_6	$(i,m,0,l,n)$	j
		S_7	$(i,0,0,l,m)$	j

Table B.2: Generating nodes in stage u+1 in regarding to stage u based on Layout C, (R_1,R_2,R_3,M_1,M_2) (Continued)

Stage u		Stage $u+1$		Take-off aircraft
Type	State	Type	State	
S_7	$(i,0,0,l,m)$	S_1	(l,n,o,m,p)	i
		S_1	(l,n,p,m,o)	i
		S_6	$(l,n,0,m,o)$	i
		S_3	$(l,n,o,m,0)$	i
		S_5	$(l,0,o,m,n)$	i
		S_7	$(l,0,0,m,n)$	i
S_8	$(i,0,k,l,0)$	S_{12}	$(l,0,k,0,0)$	i
		S_1	(i,o,n,l,p)	k
		S_5	$(i,0,n,l,o)$	k
		S_8	$(i,0,n,l,0)$	k
		S_3	$(i,o,n,l,0)$	k
S_9	$(i,j,0,0,m)$	S_{11}	$(0,j,0,0,m)$	i
		S_1	(i,m,p,n,o)	j
		S_1	(i,m,o,n,p)	j
		S_6	$(i,m,0,n,o)$	j
		S_2	$(i,m,o,0,n)$	j
		S_9	$(i,m,0,0,n)$	j
		S_3	$(i,m,o,n,0)$	j
S_{10}	$(i,j,k,0,0)$	S_{13}	$(0,j,k,0,0)$	i
		S_{12}	$(i,0,k,0,0)$	j
		S_1	(i,j,n,o,p)	k
		S_3	$(i,j,n,o,0)$	k
		S_2	$(i,j,n,0,o)$	k
		S_{10}	$(i,j,n,0,0)$	k
S_{11}	$(0,j,0,0,m)$	S_1	(n,m,q,o,p)	j
		S_1	(n,m,o,p,q)	j
		S_1	(n,m,p,o,q)	j
		S_6	$(n,m,0,o,p)$	j
		S_3	$(n,m,p,o,0)$	j
		S_3	$(n,m,o,p,0)$	j
		S_{10}	$(n,m,o,0,0)$	j
		S_2	$(n,m,p,0,o)$	j
		S_2	$(n,m,o,0,p)$	j

TABLE B.3: Generating nodes in stage u+1 in regarding to stage u based on Layout C, (R_1,R_2,R_3,M_1,M_2) (Continued)

Stage u		Stage $u+1$		Take-off aircraft
Type	State	Type	State	
S_{12}	(i,0,k,0,0)	S_{14}	(0,0,k,0,0)	j
		S_1	(i,p,n,o,q)	j
		S_1	(i,o,n,p,q)	j
		S_2	(i,o,n,0,p)	j
		S_5	(i,0,n,o,p)	j
		S_8	(i,0,n,o,0)	j
		S_{10}	(i,o,n,0,0)	j
		S_3	(i,o,n,p,0)	j
		S_3	(i,p,n,o,0)	j
S_{13}	(0,j,k,0,0)	S_{14}	(0,0,k,0,0)	j
		S_1	(o,j,n,p,q)	k
		S_3	(o,j,n,p,0)	k
		S_2	(o,j,n,0,p)	k
		S_{10}	(o,j,n,0,0)	k
		S_4	(0,j,n,0,o)	k
		S_{13}	(0,j,n,0,0)	k
S_{14}	(0,0,k,0,0)	S_1	(o,p,n,q,r)	k
		S_1	(o,q,n,p,r)	k
		S_1	(p,o,n,q,r)	k
		S_5	(o,0,n,p,q)	k
		S_2	(o,p,n,0,q)	k
		S_2	(p,o,n,0,q)	k
		S_2	(o,q,n,0,p)	k
		S_3	(p,o,n,q,0)	k
		S_3	(o,q,n,p,0)	k
		S_3	(o,p,n,q,0)	k
		S_4	(0,o,n,0,p)	k
		S_8	(o,0,n,p,0)	k
		S_{10}	(o,p,n,0,0)	k
		S_{10}	(p,o,n,0,0)	k
		S_{13}	(0,o,n,0,0)	k
		S_{12}	(o,0,n,0,0)	k
		S_{14}	(0,0,n,0,0)	k

TABLE B.4: Results for Layout A, data Set_1 ($\lambda = 1/75$, 20% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4$, $\beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D01	$\text{PT}_{\max}$	1,620	11,074	1,440	11.1	9,334	0.002	1,380	14.8	8,434	0.8	1,380	14.8	8,434	9
	ATT							1,380	14.8	8,434	0.8				
	TWT							1,380	14.8	8,434	0.7				
	BSID							1,380	14.8	8,434	0.7				
D02	$\text{PT}_{\max}$	1,800	9,485	1,560	13.3	7,385	0.003	1,500	16.7	6,545	1.0	1,440	20.0	6,305	9
	ATT							1,500	16.7	6,545	1.0				
	TWT							1,500	16.7	6,545	1.0				
	BSID							1,500	16.7	6,545	1.0				
D03	$\text{PT}_{\max}$	2,040	13,415	1,500	26.5	8,255	0.003	1,560	23.5	7,835	0.9	1,440	29.4	7,475	8
	ATT							1,500	23.5	7,835	0.8				
	TWT							1,500	23.5	7,835	1.0				
	BSID							1,500	23.5	7,835	0.9				
D04	$\text{PT}_{\max}$	1,860	8,930	1,440	22.6	5,210	0.004	1,380	25.8	4,370	0.9	1,380	25.8	4,370	8
	ATT							1,380	25.8	4,370	0.9				
	TWT							1,380	25.8	4,370	0.9				
	BSID							1,380	25.8	4,370	1.0				
D05	$\text{PT}_{\max}$	2,160	10,854	1,620	25.0	5,814	0.004	1,380	36.1	3,414	0.9	1,380	36.1	3,414	8
	ATT							1,380	36.1	3,414	0.9				
	TWT							1,380	36.1	3,414	0.9				
	BSID							1,380	36.1	3,414	0.9				
D06	$\text{PT}_{\max}$	1,920	10,861	1,680	12.5	8,821	0.002	1,620	15.6	8,581	0.9	1,560	18.8	8,211	8
	ATT							1,620	15.6	8,581	0.8				
	TWT							1,620	15.6	8,581	0.9				
	BSID							1,620	15.6	8,581	0.8				
D07	$\text{PT}_{\max}$	1,860	10,867	1,560	16.1	7,507	0.002	1,380	25.8	4,627	1.0	1,320	29.0	3,427	8
	ATT							1,380	25.8	4,627	1.0				
	TWT							1,380	25.8	4,627	1.0				
	BSID							1,380	25.8	4,627	1.0				
D08	$\text{PT}_{\max}$	1,800	8,665	1,560	13.3	5,605	0.004	1,500	16.7	5,305	1.1	1,500	16.7	5,305	8
	ATT							1,500	16.7	5,305	1.0				
	TWT							1,500	16.7	5,305	1.0				
	BSID							1,500	16.7	5,305	1.0				
D09	$\text{PT}_{\max}$	1,860	9,892	1,560	16.1	6,052	0.005	1,440	22.6	5,572	0.7	1,440	22.6	5,572	9
	ATT							1,440	22.6	5,572	0.7				
	TWT							1,440	22.6	5,572	0.7				
	BSID							1,440	22.6	5,572	0.7				
D10	$\text{PT}_{\max}$	1,500	7,040	1,440	4.0	5,840	0.001	1,380	8.0	5,540	0.8	1,380	8.0	5,540	8
	ATT							1,380	8.0	5,540	0.8				
	TWT							1,380	8.0	5,540	0.8				
	BSID							1,380	8.0	5,540	0.8				
Ave.	$\text{PT}_{\max}$	1,842	10,108	1,536	16.1	6,982	0.003	1,446	20.9	6,022	0.9	1,422	22.1	5,806	8
	ATT							1,446	20.9	6,022	0.9				
	TWT							1,446	20.9	6,022	0.9				
	BSID							1,446	20.9	6,022	0.9				

Table B.5: Results for Layout A, data Set_2 ($\lambda = 1/75$, 40% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4, \quad \beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D11	$\text{PT}_{\max}$	1,620	11,074	1,500	7.4	9,334	0.002	1,440	11.1	8,434	0.9	1,440	11.1	8,434	9
	ATT							1,440	11.1	8,434	0.8				
	TWT							1,440	11.1	8,434	0.7				
	BSID							1,440	11.1	8,434	0.9				
D12	$\text{PT}_{\max}$	1,800	9,485	1,560	13.3	7,385	0.003	1,500	16.7	6,545	1.0	1,440	20.0	6,305	7
	ATT							1,500	16.7	6,545	1.2				
	TWT							1,500	16.7	6,545	1.0				
	BSID							1,500	16.7	6,545	1.2				
D13	$\text{PT}_{\max}$	2,040	13,415	1,500	26.5	8,255	0.001	1,560	23.5	8,795	0.9	1,440	29.4	7,475	8
	ATT							1,560	23.5	8,795	0.9				
	TWT							1,560	23.5	8,795	0.9				
	BSID							1,560	23.5	8,795	0.9				
D14	$\text{PT}_{\max}$	1,860	8,930	1,560	16.1	6,050	0.004	1,500	19.4	5,450	1.0	1,500	19.4	5,450	7
	ATT							1,500	19.4	5,450	0.9				
	TWT							1,500	19.4	5,450	1.0				
	BSID							1,500	19.4	5,450	0.9				
D5	$\text{PT}_{\max}$	2,160	10,854	1,620	25.0	5,814	0.001	1,440	33.3	4,434	0.6	1,440	33.3	4,434	8
	ATT							1,440	33.3	4,434	0.6				
	TWT							1,440	33.3	4,434	0.6				
	BSID							1,440	33.3	4,434	0.7				
D16	$\text{PT}_{\max}$	1,920	10,861	1,740	9.4	9,867	0.001	1,680	12.5	9,241	0.8	1,620	15.6	8,281	7
	ATT							1,680	12.5	9,241	0.8				
	TWT							1,680	12.5	9,241	0.9				
	BSID							1,680	9.4	9,241	0.9				
D7	$\text{PT}_{\max}$	1,860	10,867	1,560	16.1	7,507	0.003	1,440	22.6	4,987	1.0	1,440	22.6	4,807	8
	ATT							1,440	22.6	4,987	1.0				
	TWT							1,440	22.6	4,987	1.1				
	BSID							1,440	22.6	4,987	1.0				
D18	$\text{PT}_{\max}$	1,800	8,665	1,560	13.3	5,845	0.004	1,560	13.3	5,665	1.0	1,560	13.3	5,665	8
	ATT							1,560	13.3	5,665	1.1				
	TWT							1,560	13.3	5,665	1.0				
	BSID							1,560	13.3	5,665	1.0				
D19	$\text{PT}_{\max}$	1,860	9,892	1,560	16.1	6,952	0.003	1,500	19.4	6,472	0.7	1,440	22.6	5,932	7
	ATT							1,500	19.4	6,472	0.7				
	TWT							1,500	19.4	6,472	0.7				
	BSID							1,500	19.4	6,472	0.7				
D20	$\text{PT}_{\max}$	1,500	7,040	1,440	4.0	5,840	0.002	1,380	8.0	5,540	0.8	1,380	8.0	5,540	9
	ATT							1,380	8.0	5,540	0.9				
	TWT							1,380	8.0	5,540	0.8				
	BSID							1,380	8.0	5,540	1.0				
Ave.	$\text{PT}_{\max}$	1,842	10,108	1,560	14.7	7,285	0.002	1,500	18.0	6,556	0.9	1,470	19.5	6,232	8
	ATT							1,500	18.0	6,556	0.9				
	TWT							1,500	18.0	6,556	0.9				
	BSID							1,500	18.0	6,556	0.9				

Table B.6: Results for Layout A, data Set_3 ($\lambda = 1/80$, 20% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4$, $\beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D21	$\text{PT}_{\max}$	1,560	6,116	1,380	11.5	4,316	0.003	1,260	19.2	3,536	1.1	1,260	19.2	3,176	8
	ATT							1,260	19.2	3,536	1.1				
	TWT							1,260	19.2	3,536	1.2				
	BSID							1,260	19.2	3,536	1.1				
D22	$\text{PT}_{\max}$	1,800	8,058	1,740	3.3	7,264	0.001	1,380	23.3	3,918	0.8	1,380	23.3	3,918	8
	ATT							1,380	23.3	3,918	0.7				
	TWT							1,380	23.3	3,918	0.7				
	BSID							1,380	23.3	3,918	0.7				
D23	$\text{PT}_{\max}$	1,680	6,059	1,380	17.9	3,719	0.003	1,380	17.9	3,479	1.1	1,380	17.9	3,179	8
	ATT							1,380	23.5	3,479	1.1				
	TWT							1,380	23.5	3,479	1.2				
	BSID							1,380	23.5	3,479	1.1				
D24	$\text{PT}_{\max}$	1,980	9,891	1,800	9.1	7,640	0.001	1,500	24.2	4,491	0.8	1,500	24.2	4,491	8
	ATT							1,500	24.2	4,491	0.8				
	TWT							1,500	24.2	4,491	0.8				
	BSID							1,500	24.2	4,491	0.9				
D25	$\text{PT}_{\max}$	2,040	9,534	1,860	8.8	8,025	0.001	1,560	23.5	3,534	0.9	1,500	26.5	3,534	8
	ATT							1,560	23.5	3,534	1.0				
	TWT							1,560	23.5	3,534	0.9				
	BSID							1,560	23.5	3,534	1.0				
D26	$\text{PT}_{\max}$	1,920	7,851	1,620	15.6	4,371	0.005	1,440	25.0	2,991	0.7	1,440	25.0	2,991	8
	ATT							1,440	25.0	2,991	0.8				
	TWT							1,440	25.0	2,991	0.7				
	BSID							1,440	25.0	2,991	0.7				
D27	$\text{PT}_{\max}$	1,920	11,581	1,500	21.9	8,101	0.002	1,380	28.1	6,421	1.0	1,380	28.1	6,421	8
	ATT							1,380	28.1	6,421	1.0				
	TWT							1,380	28.1	6,421	1.1				
	BSID							1,380	28.1	6,421	1.0				
D28	$\text{PT}_{\max}$	1,980	7,653	1,680	15.2	5,433	0.003	1,560	21.2	4,593	1.1	1,560	21.2	4,593	8
	ATT							1,560	21.2	4,593	1.1				
	TWT							1,560	21.2	4,593	1.0				
	BSID							1,560	21.2	4,593	1.1				
D29	$\text{PT}_{\max}$	2,040	12,817	1,920	5.9	11,246	0.001	1,620	20.6	8,197	0.8	1,560	23.5	7,357	8
	ATT							1,620	20.6	8,197	0.8				
	TWT							1,620	20.6	8,197	0.8				
	BSID							1,620	20.6	8,197	0.8				
D30	$\text{PT}_{\max}$	1,920	9,353	1,860	3.1	8,965	0.001	1,500	21.9	5,153	1.0	1,500	21.9	5,153	8
	ATT							1,500	21.9	5,153	0.9				
	TWT							1,500	21.9	5,153	1.0				
	BSID							1,500	21.9	5,153	0.9				
Ave.	$\text{PT}_{\max}$	1,884	8,891	1,674	11.2	6,908	0.002	1,458	22.5	4,631	0.9	1,446	23.1	4,481	8
	ATT							1,458	22.5	4,631	0.9				
	TWT							1,458	22.5	4,631	0.9				
	BSID							1,458	22.5	4,631	0.9				

TABLE B.7: Results for Layout A, data Set$_4$ ($\lambda = 1/80$, 40% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4$, $\beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D31	$\mathrm{PT}_{\max}$	1,560	6,116	1,380	11.5	4,316	0.003	1,260	19.2	3,536	1.2	1,260	19.2	3,176	8
	ATT							1,260	19.2	3,536	1.1				
	TWT							1,260	19.2	3,536	1.3				
	BSID							1,260	19.2	3,536	1.2				
D32	$\mathrm{PT}_{\max}$	1,800	8,058	1,740	3.3	7,264	0.001	1,380	23.3	3,918	0.7	1,380	23.3	3,918	8
	ATT							1,380	23.3	3,918	0.7				
	TWT							1,380	23.3	3,918	0.7				
	BSID							1,380	23.3	3,918	0.7				
D33	$\mathrm{PT}_{\max}$	1,680	6,059	1,440	14.3	3,419	0.006	1,440	14.3	3,539	1.1	1,380	17.9	3,179	7
	ATT							1,440	14.3	3,539	1.1				
	TWT							1,440	14.3	3,539	1.1				
	BSID							1,440	14.3	3,539	1.2				
D34	$\mathrm{PT}_{\max}$	1,980	9,891	1,800	9.1	7,640	0.001	1,500	24.2	4,491	0.7	1,500	24.2	4,491	8
	ATT							1,500	24.2	4,491	0.8				
	TWT							1,500	24.2	4,491	0.7				
	BSID							1,500	24.2	4,491	0.7				
D35	$\mathrm{PT}_{\max}$	2,040	9,534	1,860	8.8	8,025	0.001	1,560	23.5	3,534	1.1	1,500	26.5	3,534	8
	ATT							1,560	23.5	3,534	1.0				
	TWT							1,560	23.5	3,534	1.0				
	BSID							1,560	23.5	3,534	1.2				
D36	$\mathrm{PT}_{\max}$	1,920	7,851	1,620	15.6	4,371	0.002	1,440	25.0	2,991	0.5	1,440	25.0	2,991	8
	ATT							1,440	25.0	2,991	0.5				
	TWT							1,440	25.0	2,991	0.5				
	BSID							1,440	25.0	2,991	0.5				
D37	$\mathrm{PT}_{\max}$	1,920	11,581	1,800	6.3	9,867	0.001	1,440	25.0	7,921	1.0	1,380	28.1	6,421	8
	ATT							1,440	25.0	7,921	1.1				
	TWT							1,440	18.8	7,921	1.3				
	BSID							1,440	25.0	7,921	1.0				
D38	$\mathrm{PT}_{\max}$	1,980	7,653	1,900	4.0	7,135	0.001	1,560	21.2	5,133	1.1	1,560	21.2	4,713	8
	ATT							1,560	21.2	5,133	1.1				
	TWT							1,560	21.2	5,133	1.0				
	BSID							1,560	21.2	5,133	1.1				
D39	$\mathrm{PT}_{\max}$	2,040	12,817	1,920	5.9	11,246	0.001	1,620	20.6	8,077	1.0	1,560	23.5	7,357	8
	ATT							1,620	20.6	8,077	1.0				
	TWT							1,620	20.6	8,077	1.0				
	BSID							1,620	20.6	8,077	1.0				
D40	$\mathrm{PT}_{\max}$	1,920	9,353	1,860	3.1	8,965	0.001	1,500	21.9	5,093	1.0	1,500	21.9	5,153	8
	ATT							1,500	21.9	5,093	1.0				
	TWT							1,500	21.9	5,093	1.0				
	BSID							1,500	21.9	5,093	1.0				
Ave.	$\mathrm{PT}_{\max}$	1,884	8,891	1,732	8.2	7,223	0.002	1,470	21.8	4,823	0.9	1,446	23.1	4,517	8
	ATT							1,470	21.8	4,823	0.9				
	TWT							1,470	21.8	4,823	1.0				
	BSID							1,470	21.8	4,823	0.9				

TABLE B.8: Results for Layout A, data Set_5 ($\lambda = 1/85$, 20% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4$, $\beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D41	$PT_{\max}$	1,860	4,039	1,800	3.2	3,924	0.001	1,740	6.5	3,139	0.4	1,740	6.5	3,139	8
	ATT							1,740	6.5	3,139	0.6				
	TWT							1,740	6.5	3,139	0.6				
	BSID							1,740	6.5	3,139	0.6				
D42	$PT_{\max}$	1,680	5,764	1,680	0.0	5,764	0.001	1,620	3.6	5,464	0.8	1,620	3.6	5,344	8
	ATT							1,620	3.6	5,464	1.0				
	TWT							1,620	3.6	5,464	0.8				
	BSID							1,620	3.6	5,464	0.8				
D43	$PT_{\max}$	1,740	2,746	1,740	0.0	2,746	0.001	1,680	3.4	3,646	0.6	1,620	6.9	3,346	8
	ATT							1,680	3.4	3,646	0.6				
	TWT							1,680	3.4	3,646	0.6				
	BSID							1,680	3.4	3,646	0.6				
D44	$PT_{\max}$	1,800	6,121	1,560	13.3	4,021	0.002	1,500	16.7	3,481	0.5	1,500	16.7	3,481	8
	ATT							1,500	16.7	3,481	0.5				
	TWT							1,500	16.7	3,481	0.5				
	BSID							1,500	16.7	3,481	0.5				
D45	$PT_{\max}$	1,980	7,953	1,740	12.1	5,133	0.002	1,620	18.2	3,753	0.9	1,620	18.2	3,573	9
	ATT							1,620	18.2	3,753	0.9				
	TWT							1,620	18.2	3,753	0.9				
	BSID							1,620	18.2	3,753	0.9				
D46	$PT_{\max}$	1,980	6,554	1,680	15.2	3,314	0.004	1,680	15.2	2,834	1.0	1,680	15.2	2,894	8
	ATT							1,680	15.2	2,834	1.0				
	TWT							1,680	15.2	2,834	1.0				
	BSID							1,680	15.2	2,834	0.9				
D47	$PT_{\max}$	1,680	4,581	1,620	3.6	3,501	0.002	1,620	3.6	3,561	0.9	1,620	3.6	3,501	8
	ATT							1,620	3.6	3,561	0.9				
	TWT							1,620	3.6	3,561	1.0				
	BSID							1,620	3.6	3,561	0.9				
D48	$PT_{\max}$	1,920	5,833	1,620	15.6	4,093	0.003	1,500	21.9	3,373	0.6	1,500	21.9	3,133	10
	ATT							1,500	21.9	3,373	0.6				
	TWT							1,500	21.9	3,373	0.6				
	BSID							1,500	21.9	3,373	0.6				
D49	$PT_{\max}$	1,980	10,071	1,560	21.2	6,111	0.003	1,560	21.2	5,631	1.0	1,500	24.2	5,211	8
	ATT							1,560	21.2	5,631	1.0				
	TWT							1,560	21.2	5,631	1.0				
	BSID							1,560	21.2	5,631	1.0				
D50	$PT_{\max}$	1,620	7,625	1,320	18.5	4,265	0.003	1,260	22.2	3,605	1.0	1,260	22.2	3,605	8
	ATT							1,260	22.2	3,605	1.0				
	TWT							1,260	22.2	3,605	1.0				
	BSID							1,260	22.2	3,605	1.0				
Ave.	$PT_{\max}$	1,824	6,129	1,626	10.6	4,239	0.002	1,578	13.2	3,849	0.8	1,566	13.9	3,729	8
	ATT							1,578	13.2	3,849	0.8				
	TWT							1,578	13.2	3,849	0.8				
	BSID							1,578	13.2	3,849	0.8				

TABLE B.9: Results for Layout A, data Set_6 ($\lambda = 1/85$, 40% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4$, $\beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D51	$PT_{\max}$	1,860	4,039	1,740	6.5	3,439	0.001	1,740	6.5	3,139	0.6	1,740	6.5	3,139	8
	ATT							1,740	6.5	3,139	0.6				
	TWT							1,740	6.5	3,139	0.6				
	BSID							1,740	6.5	3,139	0.6				
D52	$PT_{\max}$	1,680	5,764	1,680	0.0	5,764	0.001	1,620	3.6	5,464	0.8	1,620	3.6	5,284	8
	ATT							1,620	3.6	5,464	0.7				
	TWT							1,620	3.6	5,464	0.7				
	BSID							1,620	3.6	5,464	0.7				
D53	$PT_{\max}$	1,740	2,746	1,740	0.0	2,746	0.001	1,680	3.4	4,486	0.7	1,680	3.4	4,186	8
	ATT							1,680	3.4	4,486	0.7				
	TWT							1,680	3.4	4,486	0.7				
	BSID							1,680	3.4	4,486	0.7				
D54	$PT_{\max}$	1,800	6,121	1,620	10.0	4,861	0.002	1,560	13.3	4,681	0.6	1,560	13.3	4,681	8
	ATT							1,560	13.3	4,681	0.6				
	TWT							1,560	13.3	4,681	0.6				
	BSID							1,560	13.3	4,681	0.6				
D55	$PT_{\max}$	1,980	7,953	1,740	12.1	5,133	0.001	1,620	18.2	3,753	0.8	1,620	18.2	3,573	8
	ATT							1,620	18.2	3,753	0.9				
	TWT							1,620	18.2	3,753	0.8				
	BSID							1,620	18.2	3,753	0.8				
D56	$PT_{\max}$	1,980	6,554	1,680	15.2	3,314	0.003	1,680	15.2	3,014	1.0	1,680	15.2	3,014	8
	ATT							1,680	15.2	3,014	1.0				
	TWT							1,680	15.2	3,014	0.9				
	BSID							1,680	15.2	3,014	0.9				
D57	$PT_{\max}$	1,680	4,581	1,620	3.6	3,501	0.002	1,620	3.6	3,561	0.9	1,620	3.6	3,501	8
	ATT							1,620	3.6	3,561	0.9				
	TWT							1,620	3.6	3,561	1.0				
	BSID							1,620	3.6	3,561	0.9				
D58	$PT_{\max}$	1,920	5,833	1,620	15.6	4,093	0.001	1,500	21.9	3,973	0.6	1,500	21.9	3,973	8
	ATT							1,500	21.9	3,973	0.6				
	TWT							1,500	21.9	3,973	0.6				
	BSID							1,500	21.9	3,973	0.5				
D59	$PT_{\max}$	1,980	10,071	1,560	21.2	5,631	0.003	1,560	21.2	5,631	1.0	1,500	24.2	5,211	8
	ATT							1,560	21.2	5,631	1.0				
	TWT							1,560	21.2	5,631	1.0				
	BSID							1,560	21.2	5,631	1.0				
D60	$PT_{\max}$	1,620	7,625	1,380	14.8	5,225	0.003	1,260	22.2	3,605	0.9	1,260	22.2	3,605	9
	ATT							1,260	22.2	3,605	1.0				
	TWT							1,260	22.2	3,605	0.9				
	BSID							1,260	22.2	3,605	1.0				
Ave.	$PT_{\max}$	1,824	6,129	1,644	9.6	4,419	0.002	1,584	12.9	4,131	0.8	1,578	13.2	4,023	8
	ATT							1,584	12.9	4,131	0.8				
	TWT							1,584	12.9	4,131	0.8				
	BSID							1,584	12.9	4,131	0.8				

TABLE B.10: Results for Layout B, data Set$_1$ ($\lambda = 1/75$, 20% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 3$, $\beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D01	$\mathrm{PT}_{\max}$	1,620	11,074	1,440	11.1	9,334	0.002	1,440	11.1	8,434	0.9	1,380	14.8	8,434	75
	ATT							1,440	11.1	8,434	0.8				
	TWT							1,440	11.1	8,434	0.8				
	BSID							1,440	11.1	8,434	0.8				
D02	$\mathrm{PT}_{\max}$	1,800	9,485	1,500	16.7	6,485	0.004	1,500	16.7	6,545	1.1	1,440	20.0	5,585	74
	ATT							1,500	16.7	6,545	1.1				
	TWT							1,500	16.7	6,545	1.2				
	BSID							1,500	16.7	6,545	1.1				
D03	$\mathrm{PT}_{\max}$	2,040	13,415	1,500	26.5	8,255	0.003	1,440	29.4	7,175	1.0	1,440	29.4	7,175	71
	ATT							1,440	29.4	7,175	1.0				
	TWT							1,440	29.4	7,175	1.0				
	BSID							1,440	29.4	7,235	1.0				
D04	$\mathrm{PT}_{\max}$	1,860	8,930	1,440	22.6	5,210	0.003	1,380	25.8	4,250	1.0	1,380	25.8	4,130	71
	ATT							1,380	25.8	4,250	1.0				
	TWT							1,380	25.8	4,130	1.1				
	BSID							1,380	25.8	4,130	1.1				
D05	$\mathrm{PT}_{\max}$	2,160	10,854	1,620	25.0	5,634	0.004	1,440	33.3	3,594	1.1	1,380	36.1	3,414	75
	ATT							1,440	33.3	3,594	1.0				
	TWT							1,380	36.1	3,414	1.0				
	BSID							1,440	33.3	3,474	1.1				
D06	$\mathrm{PT}_{\max}$	1,920	10,861	1,680	12.5	8,821	0.003	1,500	21.9	7,081	1.0	1,500	21.9	7,021	73
	ATT							1,500	21.9	7,081	1.1				
	TWT							1,500	21.9	7,021	1.0				
	BSID							1,500	21.9	7,081	1.0				
D07	$\mathrm{PT}_{\max}$	1,860	10,867	1,560	16.1	7,507	0.003	1,380	25.8	3,727	1.1	1,320	29.0	3,427	72
	ATT							1,380	25.8	3,427	1.0				
	TWT							1,380	25.8	3,427	1.1				
	BSID							1,320	29.0	3,427	1.0				
D08	$\mathrm{PT}_{\max}$	1,800	8,665	1,560	13.3	5,545	0.005	1,500	16.7	5,305	1.2	1,500	16.7	5,305	73
	ATT							1,500	16.7	5,305	1.2				
	TWT							1,500	16.7	5,305	1.1				
	BSID							1,500	16.7	5,305	1.1				
D09	$\mathrm{PT}_{\max}$	1,860	9,892	1,560	16.1	6,052	0.003	1,380	25.8	4,552	0.9	1,380	25.8	4,552	72
	ATT							1,380	25.8	4,552	0.8				
	TWT							1,380	25.8	4,552	0.8				
	BSID							1,380	25.8	4,852	0.8				
D10	$\mathrm{PT}_{\max}$	1,500	7,040	1,380	8.0	5,720	0.003	1,380	8.0	5,480	0.9	1,320	12.0	5,360	71
	ATT							1,380	8.0	5,480	0.9				
	TWT							1,380	8.0	5,480	0.9				
	BSID							1,380	8.0	5,480	0.9				
Ave.	$\mathrm{PT}_{\max}$	1,842	10,108	1,524	16.8	6,854	0.003	1,434	21.4	5,614	1.0	1,404	23.2	5,440	73
	ATT							1,434	21.4	5,614	1.0				
	TWT							1,428	21.7	5,572	1.0				
	BSID							1,428	21.7	5,596	1.0				

Table B.11: Results for Layout B, data Set_2 ($\lambda = 1/75$, 40% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 3$, $\beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D11	$\text{PT}_{\max}$	1,620	11,074	1,500	7.4	9,334	0.002	1,440	11.1	8,434	0.9	1,440	11.1	8,434	71
	ATT							1,440	11.1	8,434	0.8				
	TWT							1,440	11.1	8,434	0.7				
	BSID							1,440	11.1	8,434	0.9				
D12	$\text{PT}_{\max}$	1,800	9,485	1,500	16.7	6,485	0.004	1,500	16.7	6,545	1.4	1,440	20.0	5,585	65
	ATT							1,500	16.7	6,545	1.2				
	TWT							1,500	16.7	6,485	1.2				
	BSID							1,500	16.7	6,545	1.2				
D13	$\text{PT}_{\max}$	2,040	13,415	1,500	26.5	8,255	0.001	1,440	29.4	7,595	0.9	1,440	29.4	7,355	69
	ATT							1,440	29.4	7,595	1.0				
	TWT							1,560	23.5	8,735	1.0				
	BSID							1,560	23.5	8,915	0.9				
D14	$\text{PT}_{\max}$	1,860	8,930	1,500	19.4	5,510	0.004	1,440	22.6	4,610	1.0	1,440	22.6	4,610	66
	ATT							1,440	22.6	4,610	1.0				
	TWT							1,440	22.6	4,610	1.1				
	BSID							1,440	22.6	4,610	0.9				
D15	$\text{PT}_{\max}$	2,160	10,854	1,620	25.0	5,814	0.001	1,440	33.3	4,254	0.7	1,440	33.3	4,254	68
	ATT							1,440	33.3	4,254	0.7				
	TWT							1,440	33.3	4,434	0.7				
	BSID							1,500	30.6	4,974	0.7				
D16	$\text{PT}_{\max}$	1,920	10,861	1,740	9.4	9,867	0.001	1,560	18.8	7,201	0.9	1,500	21.9	7,141	67
	ATT							1,560	18.8	7,201	0.9				
	TWT							1,560	18.8	7,201	0.9				
	BSID							1,560	18.8	7,441	0.9				
D17	$\text{PT}_{\max}$	1,860	10,867	1,560	16.1	7,507	0.003	1,440	22.6	4,087	1.1	1,440	22.6	4,807	67
	ATT							1,440	22.6	4,087	1.1				
	TWT							1,440	22.6	4,087	1.1				
	BSID							1,440	22.6	4,087	1.1				
D18	$\text{PT}_{\max}$	1,800	8,665	1,560	13.3	5,785	0.004	1,560	13.3	5,845	1.2	1,560	13.3	5,665	72
	ATT							1,560	13.3	5,845	1.2				
	TWT							1,560	13.3	5,845	1.1				
	BSID							1,560	13.3	5,725	1.1				
D19	$\text{PT}_{\max}$	1,860	9,892	1,500	19.4	5,992	0.004	1,440	22.6	5,512	0.8	1,440	22.6	5,512	72
	ATT							1,440	22.6	5,512	0.8				
	TWT							1,440	22.6	5,512	0.8				
	BSID							1,440	22.6	5,452	0.8				
D20	$\text{PT}_{\max}$	1,500	7,040	1,380	8.0	5,720	0.004	1,380	8.0	5,480	1.2	1,320	12.0	5,360	72
	ATT							1,380	8.0	5,480	1.0				
	TWT							1,380	8.0	5,480	0.9				
	BSID							1,380	8.0	5,540	0.9				
Ave.	$\text{PT}_{\max}$	1,842	10,108	1,536	16.1	7,027	0.003	1,464	19.8	5,956	1.0	1,446	20.9	5,800	69
	ATT							1,464	19.8	5,956	1.0				
	TWT							1,476	19.2	6,082	1.0				
	BSID							1,482	19.0	6,166	0.9				

TABLE B.12: Results for Layout B, data Set$_3$ ($\lambda = 1/80$, 20% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 3$, $\beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D21	$\mathrm{PT}_{\max}$	1,560	6,116	1,380	11.5	4,316	0.002	1,260	19.2	3,176	1.2	1,260	19.2	3,176	71
	ATT							1,260	19.2	3,176	1.2				
	TWT							1,260	19.2	3,176	1.3				
	BSID							1,260	19.2	3,296	1.2				
D22	$\mathrm{PT}_{\max}$	1,800	8,058	1,440	20.0	4,338	0.004	1,380	23.3	3,918	0.7	1,380	23.3	3,918	73
	ATT							1,380	23.3	3,918	0.8				
	TWT							1,380	23.3	3,918	0.8				
	BSID							1,380	23.3	4,518	0.7				
D23	$\mathrm{PT}_{\max}$	1,680	6,059	1,380	17.9	3,719	0.003	1,440	14.3	3,419	1.2	1,320	21.4	3,419	70
	ATT							1,440	14.3	3,419	1.1				
	TWT							1,380	17.9	3,539	1.2				
	BSID							1,380	17.9	3,479	1.1				
D24	$\mathrm{PT}_{\max}$	1,980	9,891	1,800	9.1	7,640	0.001	1,500	24.2	4,491	0.9	1,500	24.2	4,491	68
	ATT							1,500	24.2	4,491	0.9				
	TWT							1,500	24.2	4,491	0.9				
	BSID							1,500	24.2	4,491	0.9				
D25	$\mathrm{PT}_{\max}$	2,040	9,534	1,860	8.8	8,025	0.001	1,620	20.6	3,594	1.0	1,500	26.5	3,534	8
	ATT							1,620	20.6	3,594	1.0				
	TWT							1,620	20.6	3,594	1.1				
	BSID							1,620	20.6	3,594	1.0				
D26	$\mathrm{PT}_{\max}$	1,920	7,851	1,500	21.9	4,191	0.003	1,440	25.0	2,991	0.9	1,440	25.0	2,991	68
	ATT							1,440	25.0	2,991	0.8				
	TWT							1,440	25.0	2,991	0.9				
	BSID							1,440	25.0	3,231	0.8				
D27	$\mathrm{PT}_{\max}$	1,920	11,581	1,500	21.9	7,921	0.004	1,380	28.1	6,481	1.1	1,380	28.1	6,421	68
	ATT							1,380	28.1	6,481	1.0				
	TWT							1,380	28.1	6,481	1.1				
	BSID							1,380	28.1	6,481	1.0				
D28	$\mathrm{PT}_{\max}$	1,980	7,653	1,500	24.2	4,173	0.005	1,500	24.2	3,993	1.1	1,500	24.2	3,993	69
	ATT							1,500	24.2	4,993	1.2				
	TWT							1,500	24.2	4,173	1.1				
	BSID							1,560	21.2	4,233	1.1				
D29	$\mathrm{PT}_{\max}$	2,040	12,817	1,920	5.9	11,246	0.001	1,440	29.4	6,037	1.0	1,440	29.4	6,037	67
	ATT							1,440	29.4	6,037	1.0				
	TWT							1,440	29.4	6,037	0.9				
	BSID							1,440	29.4	6,037	1.0				
D30	$\mathrm{PT}_{\max}$	1,920	9,353	1,860	3.1	8,965	0.001	1,440	25.0	4,193	1.0	1,440	25.0	4,073	67
	ATT							1,440	25.0	4,193	1.1				
	TWT							1,440	25.0	4,073	1.0				
	BSID							1,440	25.0	4,433	1.0				
Ave.	$\mathrm{PT}_{\max}$	1,884	8,891	1,614	14.4	6,453	0.003	1,440	23.3	4,229	1.0	1,416	24.6	4,205	69
	ATT							1,440	23.3	4,229	1.0				
	TWT							1,434	23.7	4,247	1.0				
	BSID							1,440	23.4	4,379	1.0				

Table B.13: Results for Layout B, data Set_4 ($\lambda = 1/80$, 40% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 3, \quad \beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D31	$\text{PT}_{\max}$	1,560	6,116	1,380	11.5	4,316	0.002	1,260	19.2	3,296	1.4	1,260	19.2	3,176	70
	ATT							1,260	19.2	3,296	1.4				
	TWT							1,260	19.2	3,296	1.3				
	BSID							1,260	19.2	3,296	1.3				
D32	$\text{PT}_{\max}$	1,800	8,058	1,440	20.0	4,398	0.003	1,380	23.3	4,098	0.9	1,380	23.3	3,918	67
	ATT							1,380	23.3	4,098	0.8				
	TWT							1,380	23.3	3,918	0.8				
	BSID							1,380	23.3	4,518	0.8				
D33	$\text{PT}_{\max}$	1,680	6,059	1,440	14.3	3,419	0.003	1,440	14.3	3,599	1.1	1,380	17.9	3,179	66
	ATT							1,440	14.3	3,599	1.4				
	TWT							1,440	14.3	3,539	1.4				
	BSID							1,440	14.3	3,479	1.7				
D34	$\text{PT}_{\max}$	1,980	9,891	1,800	9.1	7,640	0.001	1,500	24.2	4,491	1.0	1,500	24.2	4,491	70
	ATT							1,500	24.2	4,491	0.9				
	TWT							1,500	24.2	4,551	0.9				
	BSID							1,500	24.2	4,611	0.8				
D35	$\text{PT}_{\max}$	2,040	9,534	1,860	8.8	8,025	0.001	1,620	20.6	3,594	1.0	1,500	26.5	3,534	68
	ATT							1,620	20.6	3,594	1.1				
	TWT							1,560	23.5	3,534	1.2				
	BSID							1,620	20.6	3,594	1.1				
D36	$\text{PT}_{\max}$	1,920	7,851	1,500	21.9	4,191	0.002	1,440	25.0	2,991	0.5	1,440	25.0	2,991	71
	ATT							1,440	25.0	2,991	0.6				
	TWT							1,440	25.0	2,991	0.5				
	BSID							1,440	25.0	3,171	0.5				
D37	$\text{PT}_{\max}$	1,920	11,581	1,800	6.3	9,867	0.001	1,380	28.1	6,481	1.2	1,380	28.1	6,421	92
	ATT							1,380	28.1	6,481	1.1				
	TWT							1,380	28.1	6,481	1.1				
	BSID							1,440	25.0	6,601	1.0				
D38	$\text{PT}_{\max}$	1,980	7,653	1,900	4.0	7,135	0.001	1,500	24.2	3,993	1.0	1,500	24.2	3,993	80
	ATT							1,500	24.2	3,993	1.1				
	TWT							1,500	24.2	3,993	1.0				
	BSID							1,560	21.2	4,173	1.1				
D39	$\text{PT}_{\max}$	2,040	12,817	1,920	5.9	11,246	0.001	1,500	26.5	5,977	1.0	1,500	26.5	5,979	72
	ATT							1,500	26.5	5,977	1.0				
	TWT							1,500	26.5	5,977	1.0				
	BSID							1,560	23.5	6,757	1.0				
D40	$\text{PT}_{\max}$	1,920	9,353	1,860	3.1	8,965	0.001	1,440	25.0	4,073	1.0	1,440	25.0	4,073	72
	ATT							1,440	25.0	4,073	1.0				
	TWT							1,440	25.0	4,133	1.0				
	BSID							1,440	25.0	4,133	1.0				
Ave.	$\text{PT}_{\max}$	1,884	8,891	1,690	10.5	6,920	0.002	1,446	23.1	4,259	1.0	1,428	24.0	4,139	73
	ATT							1,446	23.1	4,823	1.0				
	TWT							1,440	23.1	4,931	1.0				
	BSID							1,464	22.1	4,775	1.0				

Table B.14: Results for Layout B, data Set_5 ($\lambda = 1/85$, 20% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 3$, $\beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D41	$\text{PT}_{\max}$	1,860	4,039	1,740	6.5	3,439	0.002	1,740	6.5	3,199	0.7	1,740	6.5	3,139	70
	ATT							1,740	6.5	3,199	0.7				
	TWT							1,740	6.5	3,199	0.7				
	BSID							1,740	6.5	3,199	0.7				
D42	$\text{PT}_{\max}$	1,680	5,764	1,680	0.0	5,764	0.001	1,620	3.6	5,464	1.0	1,620	3.6	4,984	66
	ATT							1,620	3.6	5,464	0.9				
	TWT							1,620	3.6	5,464	0.9				
	BSID							1,620	3.6	5,464	0.9				
D43	$\text{PT}_{\max}$	1,740	2,746	1,740	0.0	2,746	0.001	1,620	6.9	3,586	0.7	1,620	6.9	3,346	67
	ATT							1,620	6.9	3,586	0.7				
	TWT							1,620	6.9	3,526	0.7				
	BSID							1,620	6.9	3,526	0.7				
D44	$\text{PT}_{\max}$	1,800	6,121	1,560	13.3	3,661	0.002	1,500	16.7	3,481	0.7	1,500	16.7	3,421	80
	ATT							1,500	16.7	3,481	0.7				
	TWT							1,500	16.7	3,481	0.7				
	BSID							1,500	16.7	3,421	0.7				
D45	$\text{PT}_{\max}$	1,980	7,953	1,740	12.1	5,133	0.003	1,620	18.2	3,813	0.9	1,620	18.2	3,573	76
	ATT							1,620	18.2	3,813	1.0				
	TWT							1,620	18.2	3,753	0.9				
	BSID							1,620	18.2	3,753	0.9				
D46	$\text{PT}_{\max}$	1,980	6,554	1,680	15.2	3,314	0.003	1,680	15.2	3,434	1.0	1,680	15.2	2,834	68
	ATT							1,680	15.2	3,434	1.0				
	TWT							1,680	15.2	2,254	1.0				
	BSID							1,680	15.2	3,374	1.0				
D47	$\text{PT}_{\max}$	1,680	4,581	1,620	3.6	3,501	0.002	1,620	3.6	3,741	1.0	1,620	3.6	3,561	68
	ATT							1,620	3.6	3,741	1.0				
	TWT							1,620	3.6	4,041	1.0				
	BSID							1,620	3.6	4,041	1.0				
D48	$\text{PT}_{\max}$	1,920	5,833	1,620	15.6	4,093	0.003	1,500	21.9	3,073	0.7	1,500	21.9	3,073	70
	ATT							1,500	21.9	3,073	0.7				
	TWT							1,500	21.9	3,193	0.7				
	BSID							1,500	21.9	3,193	0.7				
D49	$\text{PT}_{\max}$	1,980	10,071	1,560	21.2	6,111	0.004	1,500	24.2	4,731	1.1	1,500	24.2	4,731	72
	ATT							1,500	24.2	4,731	1.1				
	TWT							1,500	24.2	5,331	1.1				
	BSID							1,500	24.2	5,331	1.1				
D50	$\text{PT}_{\max}$	1,620	7,625	1,320	18.5	4,265	0.004	1,260	22.2	3,605	1.1	1,260	22.2	3,605	77
	ATT							1,260	22.2	3,605	1.1				
	TWT							1,260	22.2	3,725	1.1				
	BSID							1,260	22.2	3,605	1.0				
Ave.	$\text{PT}_{\max}$	1,824	6,129	1,626	10.6	4,203	0.003	1,566	13.9	3,813	0.9	1,566	13.9	3,627	71
	ATT							1,566	13.9	3,813	0.9				
	TWT							1,566	13.9	3,897	0.9				
	BSID							1,566	13.9	3,891	0.9				

TABLE B.15: Results for Layout B, data Set_6 ($\lambda = 1/85$, 40% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 3$, $\beta = 140$				Algorithm DP			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D51	$\text{PT}_{\max}$	1,860	4,039	1,800	3.2	3,924	0.001	1,740	6.5	3,859	0.5	1,740	6.5	3,139	66
	ATT							1,740	6.5	3,859	0.6				
	TWT							1,740	6.5	3,199	0.5				
	BSID							1,740	6.5	3,199	0.5				
D52	$\text{PT}_{\max}$	1,680	5,764	1,680	0.0	5,764	0.001	1,620	3.6	5,584	1.1	1,620	3.6	5,284	68
	ATT							1,620	3.6	5,584	0.9				
	TWT							1,620	3.6	5,404	0.9				
	BSID							1,620	3.6	5,704	0.8				
D53	$\text{PT}_{\max}$	1,740	2,746	1,740	0.0	2,746	0.001	1,680	3.4	4,366	0.8	1,680	3.4	4,066	66
	ATT							1,680	3.4	4,366	0.8				
	TWT							1,680	3.4	4,366	0.8				
	BSID							1,680	3.4	4,126	0.8				
D54	$\text{PT}_{\max}$	1,800	6,121	1,560	13.3	3,841	0.002	1,560	13.3	3,841	0.7	1,560	13.3	3,841	68
	ATT							1,560	13.3	3,841	0.8				
	TWT							1,560	13.3	3,841	0.7				
	BSID							1,560	13.3	3,841	0.8				
D55	$\text{PT}_{\max}$	1,980	7,953	1,740	12.1	5,133	0.002	1,620	18.2	3,753	0.9	1,620	18.2	3,573	67
	ATT							1,620	18.2	3,753	1.0				
	TWT							1,620	18.2	3,753	0.9				
	BSID							1,620	18.2	3,753	0.9				
D56	$\text{PT}_{\max}$	1,980	6,554	1,680	15.2	3,314	0.004	1,680	15.2	3,134	1.0	1,680	15.2	2,954	67
	ATT							1,680	15.2	3,134	1.0				
	TWT							1,680	15.2	3,014	1.0				
	BSID							1,680	15.2	3,134	1.0				
D57	$\text{PT}_{\max}$	1,680	4,581	1,620	3.6	3,501	0.001	1,620	3.6	3,741	1.0	1,620	3.6	3,501	66
	ATT							1,620	3.6	3,741	1.1				
	TWT							1,620	3.6	4,041	1.0				
	BSID							1,620	3.6	4,041	1.0				
D58	$\text{PT}_{\max}$	1,920	5,833	1,620	15.6	4,093	0.001	1,500	21.9	3,913	0.7	1,500	21.9	3,913	67
	ATT							1,500	21.9	3,913	0.6				
	TWT							1,500	21.9	3,913	0.6				
	BSID							1,500	21.9	3,973	0.6				
D59	$\text{PT}_{\max}$	1,980	10,071	1,500	24.2	5,451	0.004	1,500	24.2	4,731	1.3	1,500	24.2	4,731	68
	ATT							1,500	24.2	4,731	1.1				
	TWT							1,500	24.2	5,331	1.1				
	BSID							1,500	24.2	5,331	1.1				
D60	$\text{PT}_{\max}$	1,620	7,625	1,320	18.5	4,265	0.004	1,260	22.2	3,605	1.0	1,260	22.2	3,605	67
	ATT							1,260	22.2	3,605	1.1				
	TWT							1,260	22.2	3,605	1.0				
	BSID							1,260	22.2	3,725	1.0				
Ave.	$\text{PT}_{\max}$	1,824	6,129	1,626	10.6	4,203	0.002	1,578	13.2	4,053	0.9	1,578	13.2	3,867	67
	ATT							1,578	13.2	4,053	0.9				
	TWT							1,578	13.2	4,047	0.8				
	BSID							1,578	13.2	4,083	0.9				

TABLE B.16: Results for Layout C, data Set_1 ($\lambda = 1/75$, 20% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4$, $\beta = 80$			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D01	$\text{PT}_{\max}$	1,620	11,074	1,440	11.1	8,554	0.004	1,320	18.5	7,714	0.8
	ATT							1,320	18.5	7,714	0.9
	TWT							1,320	18.5	7,714	0.9
	BSID							1,320	18.5	7,714	0.9
D02	$\text{PT}_{\max}$	1,800	9,485	1,440	20.0	5,585	0.004	1,440	20.0	5,585	1.2
	ATT							1,440	20.0	5,585	1.3
	TWT							1,440	20.0	5,645	1.1
	BSID							1,440	20.0	5,585	1.2
D03	$\text{PT}_{\max}$	2,040	13,415	1,440	29.4	7,595	0.005	1,440	29.4	7,175	1.0
	ATT							1,440	29.4	7,175	1.1
	TWT							1,440	29.4	7,475	1.0
	BSID							1,440	29.4	7,055	0.9
D04	$\text{PT}_{\max}$	1,860	8,930	1,440	22.6	5,210	0.003	1,380	25.8	4,130	1.0
	ATT							1,380	25.8	4,130	1.0
	TWT							1,380	25.8	4,130	1.1
	BSID							1,380	25.8	4,130	1.0
D05	$\text{PT}_{\max}$	2,160	10,854	1,440	33.3	4,134	0.004	1,380	36.1	3,354	1.0
	ATT							1,380	36.1	3,354	0.9
	TWT							1,380	36.1	3,354	1.0
	BSID							1,380	36.1	3,354	1.2
D06	$\text{PT}_{\max}$	1,920	10,861	1,560	18.8	7,861	0.004	1,500	21.9	6,961	1.1
	ATT							1,500	21.9	6,961	1.1
	TWT							1,500	21.9	6,961	1.1
	BSID							1,560	18.8	7,921	1.4
D07	$\text{PT}_{\max}$	1,860	10,867	1,440	22.6	5,347	0.003	1,380	25.8	3,427	1.2
	ATT							1,380	25.8	3,427	1.3
	TWT							1,320	29.0	3,487	1.3
	BSID							1,320	29.0	3,427	1.3
D08	$\text{PT}_{\max}$	1,800	8,665	1,560	13.3	5,725	0.004	1,440	20.0	4,465	1.3
	ATT							1,440	20.0	4,465	1.3
	TWT							1,440	20.0	4,465	1.3
	BSID							1,440	20.0	4,465	1.3
D09	$\text{PT}_{\max}$	1,860	9,892	1,380	25.8	4,492	0.003	1,380	25.8	4,552	1.0
	ATT							1,380	25.8	4,552	1.0
	TWT							1,380	25.8	4,612	1.0
	BSID							1,380	25.8	4,852	0.9
D10	$\text{PT}_{\max}$	1,500	7,040	1,380	8.0	5,720	0.003	1,320	12.0	5,360	1.0
	ATT							1,320	12.0	5,360	1.1
	TWT							1,320	12.0	5,360	1.0
	BSID							1,320	12.0	5,360	1.1
Ave.	$\text{PT}_{\max}$	1,842	10,108	1,452	20.5	6,022	0.004	1,398	23.5	5,362	1.1
	ATT							1,398	23.5	5,362	1.1
	TWT							1,392	23.9	5,320	1.1
	BSID							1,398	23.5	5,386	1.1

TABLE B.17: Results for Layout C, data Set_2 ($\lambda = 1/75$, 20% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4$, $\beta = 80$			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D11	$PT_{\max}$	1,620	11,074	1,440	11.1	8,554	0.003	1,440	11.1	8,374	0.8
	ATT							1,440	11.1	8,374	0.9
	TWT							1,440	11.1	8,434	0.8
	BSID							1,440	11.1	8,434	0.9
D12	$PT_{\max}$	1,800	9,485	1,440	20.0	5,585	0.003	1,440	20.0	5,585	1.2
	ATT							1,440	20.0	5,585	1.2
	TWT							1,440	20.0	5,405	1.1
	BSID							1,440	20.0	5,645	1.3
D13	$PT_{\max}$	2,040	13,415	1,500	26.5	8,255	0.001	1,500	26.5	8,255	0.9
	ATT							1,500	26.5	8,255	0.9
	TWT							1,440	29.4	7,415	1.0
	BSID							1,440	29.4	8,495	0.9
D14	$PT_{\max}$	1,860	8,930	1,500	19.4	5,510	0.005	1,440	22.6	4,550	0.9
	ATT							1,440	22.6	4,550	0.9
	TWT							1,440	22.6	4,700	0.8
	BSID							1,440	22.6	4,550	0.8
D15	$PT_{\max}$	2,160	10,854	1,620	25.0	5,814	0.001	1,380	36.1	4,014	0.9
	ATT							1,380	36.1	4,014	0.9
	TWT							1,380	36.1	4,014	0.9
	BSID							1,380	36.1	4,014	0.9
D16	$PT_{\max}$	1,920	10,861	1,740	9.4	9,867	0.001	1,560	18.8	7,021	1.0
	ATT							1,560	18.8	7,021	1.0
	TWT							1,620	15.6	7,921	0.9
	BSID							1,620	15.6	7,921	0.9
D17	$PT_{\max}$	1,860	10,867	1,440	22.6	4,087	0.007	1,440	22.6	4,207	1.0
	ATT							1,440	22.6	4,207	1.0
	TWT							1,440	22.6	4,207	1.0
	BSID							1,440	22.6	4,207	1.0
D18	$PT_{\max}$	1,800	8,665	1,560	13.3	5,965	0.004	1,500	16.7	5,185	1.1
	ATT							1,500	16.7	5,185	1.1
	TWT							1,500	16.7	5,425	1.1
	BSID							1,560	13.3	5,665	1.1
D19	$PT_{\max}$	1,860	9,892	1,440	22.6	5,452	0.004	1,440	22.6	5,572	0.8
	ATT							1,440	22.6	5,572	0.8
	TWT							1,440	22.6	5,512	0.8
	BSID							1,440	22.6	5,572	0.8
D20	$PT_{\max}$	1,500	7,040	1,380	8.0	5,720	0.003	1,320	12.0	5,360	0.8
	ATT							1,320	12.0	5,360	0.9
	TWT							1,320	12.0	5,360	0.9
	BSID							1,320	12.0	5,300	0.9
Ave.	$PT_{\max}$	1,842	10,108	1,506	17.8	6,481	0.003	1,446	20.9	5,812	0.9
	ATT							1,446	20.9	5,812	0.9
	TWT							1,446	20.9	5,839	0.9
	BSID							1,452	20.5	5,980	1.0

TABLE B.18: Results for Layout C, data Set$_3$ ($\lambda = 1/80$, 20% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4$, $\beta = 80$			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D21	$PT_{\max}$	1,560	6,116	1,320	15.4	3,236	0.003	1,260	19.2	3,056	1.1
	ATT							1,260	19.2	3,056	1.0
	TWT							1,260	19.2	3,236	1.0
	BSID							1,260	19.2	3,056	1.0
D22	$PT_{\max}$	1,800	8,058	1,440	20.0	4,338	0.005	1,320	26.7	3,498	0.7
	ATT							1,320	26.7	3,498	0.8
	TWT							1,320	26.7	3,558	0.7
	BSID							1,320	26.7	3,498	0.8
D23	$PT_{\max}$	1,680	6,059	1,320	21.4	3,119	0.003	1,320	21.4	3,179	1.1
	ATT							1,320	21.4	3,179	1.0
	TWT							1,320	21.4	3,179	1.0
	BSID							1,320	21.4	3,179	1.1
D24	$PT_{\max}$	1,980	9,891	1,800	9.1	7,640	0.001	1,500	24.2	4,611	0.8
	ATT							1,500	24.2	4,611	0.9
	TWT							1,500	24.2	4,611	0.8
	BSID							1,500	24.2	4,611	1.0
D25	$PT_{\max}$	2,040	9,534	1,860	8.8	8,025	0.001	1,560	23.5	3,534	0.9
	ATT							1,560	23.5	3,534	1.0
	TWT							1,560	23.5	3,534	0.9
	BSID							1,560	23.5	3,714	1.0
D26	$PT_{\max}$	1,920	7,851	1,500	21.9	3,831	0.004	1,440	25.0	2,991	0.8
	ATT							1,440	25.0	2,991	0.9
	TWT							1,440	25.0	3,051	0.8
	BSID							1,440	25.0	3,471	0.8
D27	$PT_{\max}$	1,920	11,581	1,440	25.0	6,901	0.006	1,500	21.9	6,700	0.9
	ATT							1,500	21.9	6,700	1.0
	TWT							1,500	21.9	8,941	0.9
	BSID							1,440	25.0	7,381	0.9
D28	$PT_{\max}$	1,980	7,653	1,500	24.2	4,113	0.003	1,500	24.2	3,813	1.0
	ATT							1,500	24.2	3,813	1.1
	TWT							1,440	27.3	3,393	1.1
	BSID							1,500	24.2	3,993	1.1
D29	$PT_{\max}$	2,040	12,817	1,920	5.9	11,246	0.001	1,440	29.4	5,867	0.7
	ATT							1,440	29.4	5,867	0.8
	TWT							1,440	29.4	5,147	0.8
	BSID							1,440	29.4	5,867	0.6
D30	$PT_{\max}$	1,920	9,353	1,860	3.1	8,965	0.001	1,380	28.1	3,653	1.0
	ATT							1,380	28.1	3,653	1.1
	TWT							1,380	28.1	3,833	1.0
	BSID							1,380	28.1	3,893	1.0
Ave.	$PT_{\max}$	1,884	8,891	1,596	15.5	6,141	0.003	1,422	24.4	4,090	0.9
	ATT							1,422	24.4	4,090	1.0
	TWT							1,416	24.7	4,248	0.9
	BSID							1,416	24.7	4,266	0.9

TABLE B.19: Results for Layout C, data Set$_4$ ($\lambda = 1/80$, 40% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4$, $\beta = 80$			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D31	$\mathrm{PT}_{\max}$	1,560	6,116	1,320	15.4	3,236	0.003	1,260	19.2	3,176	1.1
	ATT							1,260	19.2	3,176	1.0
	TWT							1,260	19.2	3,236	1.1
	BSID							1,260	19.2	3,176	1.1
D32	$\mathrm{PT}_{\max}$	1,800	8,058	1,440	20.0	4,398	0.002	1,320	26.7	3,498	0.7
	ATT							1,320	26.7	3,498	0.7
	TWT							1,380	23.3	4,338	0.8
	BSID							1,380	23.3	3,738	0.7
D33	$\mathrm{PT}_{\max}$	1,680	6,059	1,440	14.3	3,419	0.003	1,440	14.3	3,359	1.1
	ATT							1,440	14.3	3,359	1.1
	TWT							1,440	14.3	3,419	1.0
	BSID							1,440	14.3	3,659	1.0
D34	$\mathrm{PT}_{\max}$	1,980	9,891	1,800	9.1	7,640	0.001	1,500	24.2	4,611	0.8
	ATT							1,500	24.2	4,611	0.9
	TWT							1,500	24.2	4,611	0.8
	BSID							1,500	24.2	4,611	1.0
D35	$\mathrm{PT}_{\max}$	2,040	9,534	1,860	8.8	8,025	0.001	1,560	23.5	3,714	0.9
	ATT							1,560	23.5	3,714	1.0
	TWT							1,560	23.5	3,714	0.9
	BSID							1,560	23.5	3,714	1.0
D36	$\mathrm{PT}_{\max}$	1,920	7,851	1,500	21.9	3,831	0.003	1,440	25.0	2,931	0.7
	ATT							1,440	25.0	2,931	0.8
	TWT							1,440	25.0	3,171	0.7
	BSID							1,440	25.0	3,471	0.8
D37	$\mathrm{PT}_{\max}$	1,920	11,581	1,800	6.3	9,867	0.001	1,440	25.0	7,141	1.0
	ATT							1,440	25.0	7,141	1.1
	TWT							1,440	25.0	7,261	1.0
	BSID							1,380	28.1	6,421	1.2
D38	$\mathrm{PT}_{\max}$	1,980	7,653	1,900	4.0	7,135	0.001	1,440	27.3	3,453	1.0
	ATT							1,440	27.3	3,453	0.9
	TWT							1,440	27.3	3,393	1.0
	BSID							1,440	27.3	3,393	1.0
D39	$\mathrm{PT}_{\max}$	2,040	12,817	1,920	5.9	11,246	0.001	1,440	29.4	5,677	1.0
	ATT							1,440	29.4	5,677	1.0
	TWT							1,440	29.4	4,641	0.8
	BSID							1,440	29.4	5,677	0.9
D40	$\mathrm{PT}_{\max}$	1,920	9,353	1,860	3.1	8,965	0.001	1,380	28.1	3,653	1.0
	ATT							1,380	28.1	3,653	0.9
	TWT							1,380	28.1	3,653	1.0
	BSID							1,380	28.1	3,833	0.9
Ave.	$\mathrm{PT}_{\max}$	1,884	8,891	1,684	10.9	6,776	0.002	1,422	24.3	4,121	0.9
	ATT							1,422	24.3	4,121	0.9
	TWT							1,428	23.9	4,144	0.9
	BSID							1,422	24.3	4,169	1.0

TABLE B.20: Results for Layout C, data Set_5 ($\lambda = 1/85$, 20% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4, \quad \beta = 80$			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D41	$\text{PT}_{\max}$	1,860	4,039	1,740	6.5	3,439	0.002	1,740	6.5	3,259	0.7
	ATT							1,740	6.5	3,259	0.7
	TWT							1,740	6.5	3,199	0.7
	BSID							1,740	6.5	3,199	0.6
D42	$\text{PT}_{\max}$	1,680	5,764	1,680	0.0	5,764	0.001	1,560	7.1	4,804	0.9
	ATT							1,560	7.1	4,804	0.9
	TWT							1,560	7.1	4,684	0.8
	BSID							1,560	7.1	4,864	0.9
D43	$\text{PT}_{\max}$	1,740	2,746	1,740	0.0	2,746	0.001	1,620	6.9	3,826	0.7
	ATT							1,620	6.9	3,826	0.7
	TWT							1,620	6.9	3,826	0.7
	BSID							1,620	6.9	3,826	0.7
D44	$\text{PT}_{\max}$	1,800	6,121	1,560	13.3	3,661	0.003	1,500	16.7	3,361	0.7
	ATT							1,500	16.7	3,361	0.6
	TWT							1,500	16.7	3,361	0.6
	BSID							1,560	13.3	3,721	0.7
D45	$\text{PT}_{\max}$	1,980	7,953	1,680	15.2	4,833	0.002	1,620	18.2	3,873	0.8
	ATT							1,620	18.2	3,873	0.8
	TWT							1,620	18.2	3,873	0.9
	BSID							1,620	18.2	3,873	0.8
D46	$\text{PT}_{\max}$	1,980	6,554	1,620	18.2	3,014	0.005	1,620	18.2	3,014	1.1
	ATT							1,620	18.2	3,014	0.9
	TWT							1,620	18.2	3,254	0.9
	BSID							1,620	18.2	3,254	0.9
D47	$\text{PT}_{\max}$	1,680	4,581	1,620	3.6	3,021	0.002	1,620	3.6	3,261	1.0
	ATT							1,620	3.6	3,261	0.9
	TWT							1,620	3.6	4,401	1.0
	BSID							1,620	3.6	3,801	0.9
D48	$\text{PT}_{\max}$	1,920	5,833	1,620	15.6	4,093	0.004	1,500	21.9	2,953	0.6
	ATT							1,500	21.9	2,953	0.7
	TWT							1,500	21.9	2,953	0.6
	BSID							1,500	21.9	3,253	0.7
D49	$\text{PT}_{\max}$	1,980	10,071	1,500	24.2	4,611	0.008	1,440	27.3	4,311	0.9
	ATT							1,440	27.3	4,311	0.9
	TWT							1,440	27.3	4,191	1.0
	BSID							1,500	24.2	5,151	1.0
D50	$\text{PT}_{\max}$	1,620	7,625	1,320	18.5	4,265	0.003	1,260	22.2	3,665	1.0
	ATT							1,260	22.2	3,665	1.2
	TWT							1,260	22.2	3,665	1.0
	BSID							1,260	22.2	3,545	1.0
Ave.	$\text{PT}_{\max}$	1,824	6,129	1,608	11.5	3,945	0.003	1,548	14.8	3,633	0.8
	ATT							1,548	14.8	3,633	0.8
	TWT							1,548	14.8	3,741	0.8
	BSID							1,560	14.2	3,849	0.8

Table B.21: Results for Layout C, data Set_6 ($\lambda = 1/85$, 40% CTOT)

#	Local Evaluation Criteria	FCFS		Algorithm DLS				Algorithm BS $\alpha = 4, \quad \beta = 80$			
		$T_{\max}$	TWT	$T_{\max}$	PI	TWT	CT	$T_{\max}$	PI	TWT	CT
D51	$\mathrm{PT}_{\max}$	1,860	4,039	1,800	3.2	3,924	0.001	1,740	6.5	3,259	0.7
	ATT							1,740	6.5	3,259	0.8
	TWT							1,740	6.5	3,199	0.7
	BSID							1,740	6.5	3,199	1.0
D52	$\mathrm{PT}_{\max}$	1,680	5,764	1,680	0.0	5,764	0.001	1,560	7.1	4,804	0.9
	ATT							1,560	7.1	4,804	1.2
	TWT							1,560	7.1	4,684	1.1
	BSID							1,560	7.1	4,864	1.3
D53	$\mathrm{PT}_{\max}$	1,740	2,746	1,740	0.0	2,746	0.001	1,680	3.4	4,426	0.8
	ATT							1,680	3.4	4,426	0.8
	TWT							1,680	3.4	4,426	0.8
	BSID							1,680	3.4	4,426	0.9
D54	$\mathrm{PT}_{\max}$	1,800	6,121	1,560	13.3	3,841	0.001	1,560	13.3	4,561	0.7
	ATT							1,560	13.3	4,561	0.7
	TWT							1,560	13.3	4,561	0.8
	BSID							1,620	10.0	4,081	0.7
D55	$\mathrm{PT}_{\max}$	1,980	7,953	1,740	12.1	5,133	0.002	1,620	18.2	3,873	1.0
	ATT							1,620	18.2	3,873	1.0
	TWT							1,620	18.2	3,753	0.9
	BSID							1,620	18.2	3,873	0.8
D56	$\mathrm{PT}_{\max}$	1,980	6,554	1,680	15.2	3,314	0.004	1,680	15.2	3,374	1.0
	ATT							1,680	15.2	3,374	1.3
	TWT							1,680	15.2	3,494	0.9
	BSID							1,680	15.2	3,854	1.1
D57	$\mathrm{PT}_{\max}$	1,680	4,581	1,620	3.6	3,021	0.002	1,620	3.6	3,261	1.0
	ATT							1,620	3.6	3,261	1.1
	TWT							1,620	3.6	4,401	1.2
	BSID							1,620	3.6	3,801	1.0
D58	$\mathrm{PT}_{\max}$	1,920	5,833	1,620	15.6	4,093	0.001	1,500	21.9	3,493	0.7
	ATT							1,500	21.9	3,493	0.6
	TWT							1,500	21.9	3,553	0.7
	BSID							1,500	21.9	3,613	0.8
D59	$\mathrm{PT}_{\max}$	1,980	10,071	1,500	24.2	4,911	0.005	1,440	27.3	4,311	1.0
	ATT							1,440	27.3	4,311	1.1
	TWT							1,440	27.3	4,311	1.1
	BSID							1,500	24.2	5,091	1.1
D60	$\mathrm{PT}_{\max}$	1,620	7,625	1,320	18.5	4,265	0.003	1,260	22.2	3,665	1.2
	ATT							1,260	22.2	3,665	1.2
	TWT							1,260	22.2	3,665	1.1
	BSID							1,260	22.2	3,665	1.2
Ave.	$\mathrm{TT}_{\max}$	1,824	6,129	1,626	10.6	4,101	0.002	1,566	13.9	3,903	0.9
	ATT							1,566	13.9	3,903	1.0
	TWT							1,566	13.9	4,023	0.9
	BSID							1,578	13.2	4,041	1.0

Bibliography

Abela, J., Abramson, D., Krishnamoorthy, M., De Silva, A., and Mills, G. (1993). Computing Optimal Schedules for Landing Aircraft. In *Proceeding of 12th National Conference of the Australian Society for Operations Research, Adelaide, Australia, pp 71-90.*

Anagnostakis, I. (2004). *A Multi-Objective, Decomposition-Based Algorithm Design Methodology and Its Application to Runway Operations Planning.* PhD thesis, Department of Aeronautics & Astronautics, MIT, USA.

Anagnostakis, I. and Clarke, J.-P. (2002). Runway Operations Planning: A Two-Stage Heuristic Algorithm. In *AIAA Aircraft, Technology, Integration and Operations Forum, Los Angeles, CA, USA.*

Anagnostakis, I. and Clarke, J.-P. (2003). Runway Operations Planning: A Two-Stage Solution Methodology. In *Proceedings of the 36th Annual Hawaii International Conference on System Science (HICSS'03), Big Island, HI, USA.*

Artiouchine, K., Baptiste, P., and Durr, C. (2008). Runway Sequencing with Holding Patterns. *European Journal of Operational Research*, 189:1254–1266.

Atkin, J. A. D. (2008). *On-Line Decision Support for Take-off Runway Scheduling with Uncertain Taxi times at London Heathrow Airport.* PhD thesis, University of Nottingham, UK.

Atkin, J. A. D., Burke, E. K., Greenwood, J. S., and Reeson, D. (2004). A Metaheuristic Approach to Aircraft Departure Scheduling at London Heathrow Airport. In *Electronic Proceedings of the 9th International Conference on Computer-Aided Scheduling of Public Transport, San Diego, California, USA.*

Atkin, J. A. D., Burke, E. K., Greenwood, J. S., and Reeson, D. (2006). An Examination of Take-Off Scheduling Constraints at London Heathrow Airport. In *Electronic proceedings of the 10th International Conference on Computer-Aided Scheduling of Public Transport, (CASPT2006).*

Atkin, J. A. D., Burke, E. K., Greenwood, J. S., and Reeson, D. (2007). Hybrid Metaheuristics to Aid Runway Scheduling at London Heathrow Airport. *Transportation Science*, 41:90–106.

Atkin, J. A. D., Burke, E. K., Greenwood, J. S., and Reeson, D. (2008). On-Line Decision Support for Take-off Runway Scheduling with Uncertain Taxi Times at London Heathrow Airport. *Journal of Scheduling*, 11:323–346.

Balakrishnan, H. and Chandran, B. (2006). Scheduling Aircraft Landings under Constrained Position Shifting. In *AIAA Guidance, Navigation and Control Conference and Exhibit, Keystone, Colorado, USA.*

Balakrishnan, H. and Chandran, B. (2007). Efficient and Equitable Departure Scheduling in Real-Time: New Approaches to Old Problems. In *USA/Europe Air Traffic Management R&D Seminar, Barcelona, Spain.*

Bauerle, N., Engelhardt-Funke, O., and Kolonko, M. (2007). On the Waiting Time of Arriving Aircraft and the Capacity of Airports with One or Two Runways. *European Journal of Operational Research*, 177:1180–1196.

Bayen, A. M., Tomlin, C. J., Ye, Y., and Zhang, J. (2004). An Approximation Algorithm for Scheduling Aircraft with Holding Time. In *43rd IEEE Conference on Decision and Control, Atlantis, Paradise Island, Bahamas.*

Bazargan, M., Fleming, K., and Subramanian, P. (2002). A Simulation Study to Investigate Runway Capacity Using TAAM. In *Proceeding of the 34th Winter Simulation Conference (WSC'02), San Diego, California, USA.*

Beasley, J. E. (1990). OR-Library: Distributing Test Problems by Electronic Mail. *Journal of the Operational Rescarch Society*, 41:1069–1072.

Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., and Abramson, D. (2000). Scheduling Aircraft Landings - The Static Case. *Transportation Science*, 34:180–197.

Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., and Abramson, D. (2004). Displacement Problem and Dynamically Scheduling Aircraft Landings. *Journal of the Operational Research Society*, 55:54–64.

Beasley, J. E., Sonander, J., and Havelock, P. (2001). Scheduling Aircraft Landing at London Heathrow using a Population Heuristic. *Journal of the Operational Research Society*, 52:483–493.

Bellman, R. (2003). *Dynamic Programming*. Princeton University Press.

Bencheikh, G., Boukachour, J., Alaoui, A. E. H., and Khoukhi, F. E. (2009). Hybrid Method for Aircraft Landing Scheduling based on a Job Shop Formulation. *International Journal of Computer Science and Network Security*, 9:78–88.

Bennell, J. A., Mesgarpour, M., and Potts, C. N. (2011). Invited Survey: Airport Runway Scheduling. *4OR*, 9(2):115–138.

Bianco, L. and Bielli, M. (1993). System Aspects and Optimization models in ATC Planning. In Bianco, L. and Odoni, A. R., editors, *Large Scale Computation and Information Processing in ATC*, pages 47–100. Springer, Berlin.

Bianco, L., Dell'Olmo, P., and Giordani, S. (1997). Scheduling Models and Algorithms for TMA Traffic Management. In Bianco, L., Dell'Olmo, P., and Odoni, A. R., editors, *Modelling and Simulation for Air Traffic Management*, pages 139–167. Springer, Berlin, Heidelberg.

Bianco, L., Dell'Olmo, P., and Giordani, S. (1999). Minimizing Total Completion Time Subject to Release Dates and Sequence-Dependent Processing Times. *Annals of Operations Research*, 86:393–415.

Bianco, L., Dell'Olmo, P., and Giordani, S. (2006). Scheduling Models for Air Traffic Control in Terminal Areas. *Journal of Scheduling*, 9:223–253.

Blumstein, A. (1959). The Landing Capacity of a Runway. *Operations Research*, 7:752–763.

Brentnall, A. R. (2006). *Aircraft Arrival Management*. PhD thesis, University of Southampton, UK.

Brentnall, A. R. and Cheng, R. C. H. (2009). Some Effects of Aircraft Arrival Sequence Algorithms. *Journal of the Operational Research Society*, 60:962–972.

Brinton, C. R. (1992). An Implicit Enumeration Algorithm for Arrival Aircraft Scheduling. In *Proceedings of the IEEE/AIAA 11th Digital Avionics Systems Conference, Seattle, WA, USA*.

Burke, E. and Kendall, G., editors (2005). *Search Methodologies. Introductory Tutorials in Optimization and Decision Support Techniques*. Springer.

Capri, S. and Ignaccolo, M. (2004). Genetic Algorithms for Solving the Aircraft-Sequencing Problem: The Introduction of Departures into the Dynamic Model. *Journal of Air Transport Management*, 10:345–351.

Carr, G. C., Erzberger, H., and Neuman, F. (2000). Fast-Time Study of Airline-Influenced Arrival Sequencing and Scheduling. *Journal of Guidance, Control, and Dynamics*, 23:526–531.

Chandran, B. and Balakrishnan, H. (2007). A Dynamic Programming Algorithm for Robust Runway Scheduling. In *Proceedings of the American Control Conference, New York, NY, USA*.

Cheng, V. H. L., Crawford, L. S., and Menon, P. K. (1999). Air Traffic Control Using Genetic Search Techniques. In *Proceedings of the IEEE International Conference on Control Applications, Hawaii, HA, USA*.

Ciesielski, V. and Scerri, P. (1997). An Anytime Algorithm for Scheduling of Aircraft Landing Times Using Genetic Algorithms. *Australian Journal of Intelligent Information Processing Systems*, 4:206–213.

Ciesielski, V. and Scerri, P. (1998). Real Time Genetic Scheduling of Aircraft Landing Times. In Fogel, D., editor, *Proceedings of The 1998 IEEE International Conference on Evolutionary Computation (ICEC'98), Anchorage, Alsaka, pp 360-364*.

Craig, A., Ketzscher, R., Leese, R. A., Noble, S. D., Parrott, K., Preater, J., Wilson, R. E., and Wood, D. A. (2001). The Sequencing of Aircraft Departures. In *40th European Study Group with Industry, Keele, UK*.

Crauwels, H., Potts, C. N., and Wassenhove, L. V. (1997). Local Search Heuristics for Single Machine Scheduling with Batch Set-Up Times to Minimize Total Weighted Completion Time. *Annals of Operations Research*, 70:261–279.

de Neufville, R. and Odoni, A. R. (2003). *Airport Systems: Planning, Design, and Management*. McGraw-Hill, New York.

Dear, R. (1976). The Dynamic Scheduling of Aircraft in the Near Terminal Area. Technical report, R76-9, Flight Transportation Laboratory, MIT, USA.

Dear, R. G. and Sherif, Y. S. (1989). The Dynamic Scheduling of Aircraft in High Density Terminal Areas. *Microelectrons and Reliability*, 29:743–749.

Dear, R. G. and Sherif, Y. S. (1991). An Algorithm for Computer Assisted Sequencing and Scheduling of Terminal Area Operation. *Transportation Research. Part A: General*, 25:129–139.

Dowsland, A. (1993). *Modern Heuristic Techniques for Combinatorial Problems*, chapter Simulated Annealing, pages 20–69. Blackwell Scientific Publishers, Oxford.

Duke, G. (2009). *Air Traffic Control, 10th ed.* Ian Allan Ltd, UK.

E. Grunewald, P. Berster, W. G. M. H. and Keimel, H. (2007). Luftverkehrsbericht 2007 - daten und kommentierungen des deutschen und weltweiten luftverkehrs verkehrsbericht. ib 326-2007/2. Technical report, PorzWahnheide/Kln, Flughafenwesen und Luftverkehr, DLR.

Ernst, A. T., Krishnamoorthy, M., and Storer, R. H. (1999). Heuristic and Exact Algorithms for Scheduling Aircraft Landings. *Networks*, 34:229–241.

EUROCONTROL (2012). EUROCONTROL's official website. Retrieved February 01, 2012, from http://www.eurocontrol.int/.

FAA (2012). FAA's official website. Retrieved February 01, 2012, from http://www.faa.gov/.

Fahle, T., Feldmann, R., Gotz, S., Grothklags, S., and Monien, B. (2003). The Aircraft Sequencing Problem. In *Computer Science in Perspective, LNCS 2598*, pages 152–166.

FICO (2012). FICO Xpress Optimimization Suite's official website. Retrieved February 01, 2012, from http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx.

Filar, J. A., Manyem, P., and White, K. (2001). How Airlines and Airports Recover from Schedule Perturbations: A Survey. *Annals of Operations Research*, 108:315–333.

Garey, M. R. and D. S. Johnson, N. Y. (1979). *Computers and Intractability, a Guide to the Theory of NP-completeness.* W. H. Freeman and Company.

Glover, F. (1989). Tabu Search - Part I. *ORSA Journal of Computing*, 1:190–206.

Gupta, J. N., Neppalli, V. R., and Werner, F. (2001). Minimizing Total Flow Time in a Two-machine Flowshop Problem with Minimum Makespan. *International Journal of Production Economics*, 69:323–338.

Hansen, J. V. (2004). Genetic Search Methods in Air Traffic Control. *Computers & Operations Research*, 3:445–459.

Heathrow (2012). Heathrow Airport's official website. Retrieved February 01, 2012, from http://www.heathrowairport.com.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Ann Arbor.

Hoogeveen, H. (2005). Multicriteria Scheduling. *European Journal of Operational Research*, 167:592–623.

Hu, X.-B. and Chen, W.-H. (2005a). Genetic Algorithm Based on Receding Horizon Control for Arrival Sequencing and Scheduling. *Engineering Applications of Artificial Intelligence*, 18:633642.

Hu, X.-B. and Chen, W.-H. (2005b). Receding Horizon Control for Aircraft Arrival Sequencing and Scheduling. *IEEE Transaction on Intelligent Transportation Systems*, 6:189–197.

Hu, X.-B. and Di Paolo, E. (2008). Binary-Representation-Based Genetic Algorithm for Aircraft Arrival Sequencing and Scheduling. *IEEE Transactions on Intelligent Transportation Systems*, 9:301–310.

Hu, X.-B. and Di Paolo, E. (2009). An Efficient Genetic Algorithm with Uniform Crossover for Air Traffic Control. *Computer & Operations Research*, 36:245–259.

Idris, H. R. (2001). *Observation and Analysis of Departure Operations at Boston Logan International Airport*. PhD thesis, Department of Mechanical Engineering, MIT, USA.

Idris, H. R., Delcaire, B., Anagnostakis, I., Hall, W. D., Clarke, J.-P., Hansman, R. J., Feron, E., and Odoni, A. R. (1998a). Observations of Departure Processes at Logan Airport to Support the Development of Departure Planning Tools. In *The 2nd USA/Europe Air Traffic Management R&D Seminar, Orlando, USA*.

Idris, H. R., Delcaire, B., Anagnostakis, I., Hall, W. D., and Pujet, N. (1998b). Identification of Flow Constraint and Control Points in Departure Operations at Airport Systems. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Boston, MA, USA*.

Isaacson, D. R., Davis, T. J., and Robinson, J. E. (1997). Knowledge-Based Runway Assignment for Arrival Aircraft in the Terminal Area. In *Proceedings of AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, USA*.

Karp, R. (1972). *Complexity of Computer Computations*, chapter Reducibility among Combinatorial Problems, pages 85–103. Plenum Press.

Kirkpatrick, S., Gellat, C. D., and Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220:671–680.

Land, A. H. and Doig, A. G. (1960). An Automatic Method for Solving Discrete Programming Problems. *Econometrica*, 28:97–520.

Laporte, G. (2010). A Concise Guide to the Traveling Salesman Problem. *Journal of the Operational Research Society*, 61:35–40.

Lee, H. and Balakrishnan, H. (2008). Fuel Cost, Delay and Throughput Tradeoffs in Runway Scheduling. In *Proceeding of American Control Conference (ACC 08), Seattle, Washington, USA*.

Lowerre, B. T. (1976). *The HAPPY Sepeech Recognition System*. PhD thesis, Carnegie-Mellon University, USA.

Luenberger, R. A. (1998). A Traveling-Salesman-Based Approach to Aircraft Scheduling in the Terminal Area. Technical report, NASA Technical Memorandum 100062.

Mesgarpour, M., Potts, C. N., and Bennell, J. A. (2010). Models for Aircraft Landing Optimization. In *Proceeding of the 4th International Conference on Research in Air Transportation (ICRAT2010), Budapest, Hungary*.

Metropolis, N., Rosenbluth, A. W., Teller, A. H., and Teller, E. (1953). Equation of State Calculation by Fast Computing Machines. *Journal of Chemistry Physics*, 21:1087–1091.

Moser, I. and Hendtlass, T. (2007). Solving Dynamic Single-Runway Aircraft Landing Problems with Extremal Optimization. In *Proceeding of the 2007 IEEE Symposium on Computational Intelligence in Scheduling (CI-Sched 2007), Honolulu, Hawaii, USA*.

NATS (2011). National Air Traffic Services' official website. Retrieved September 01, 2011, from http://www.nats.co.uk.

Neuman, F. and Erzberger, H. (1991). Analysis of Delay Reducing and Fuel Saving Sequencing and Spacing Algorithms for Arrival Traffic. Technical report, TM-103880, Ames Research Center, NASA, USA.

Newell, G. F. (1979). Airport Capacity and Delays. *Transportation Science*, 13:201–241.

Papadimitriou, C. H. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*. Upper Saddle River, NJ: Prentice-Hall, Inc.

Pinedo, M. L. (2008). *Scheduling: Theory, Algorithms, and Systems*. New York, NY, Springer.

Pinedo, M. L. (2009). *Planning and Scheduling in Manufacturing and Services*. Springer.

Pinol, H. and Beasley, J. E. (2006). Scatter Search and Bionomic Algorithms for the Aircraft Landing Problem. *European Journal of Operational Research*, 171:439–462.

Potts, C. N. and Strusevich, V. A. (2009). Fifty years of scheduling: a survey of milestones. *Journal of the Operational Research Society*, 60:41–68.

Psaraftis, H. N. (1978). A Dynamic Programming Approach to the Aircraft Sequencing Problem. Technical report, R78-4, Flight Transportation Laboratory, MIT, USA.

Psaraftis, H. N. (1980). A Dynamic Programming Approach for Sequencing Groups of Identical Jobs. *Operations Research*, 28:1347–1359.

Pujet, N., Delcaire, B., and Feron, E. (1999). Input-Output Modelling and Control of the Departure Process of Congested Airports. In *Proceedings of AIAA Guidance, Navigation and Control Conference, and Exhibit, Portland, OR, USA*.

Randall, M. C. (2002). Scheduling Aircraft Landings Using Ant Colony Optimisation. In *Proceedings of The IASTED International Conference Artificial Intelligence and Soft Computing, Banff, Canada*.

Smith, V. J. R., Osman, I. H., Reeves, C. R., and Smith, G. D., editors (1996). *Modern Heuristic Search Methods*. John Wiley & Sons.

Soomer, M. (2009). *Runway Operations Scheduling using Airline Preferences*. PhD thesis, Vrije Universiteit, Amsterdam, The Netherlands.

Soomer, M. J. and Franx, G. J. (2008). Scheduling Aircraft Landings using Airlines' Preferences. *European Journal of Operational Research*, 190:277–291.

Stevens, G. (1995). An Approach to Scheduling Aircraft Landing Times Using Genetic Algorithms. *Honours Thesis, Department of Computer Science, RMIT University, Melbourne, Australia*.

Stiverson, A. W. (2009). A Study of Heuristic Approaches for Runway Scheduling for the Dallas-Fort Worth Airport. Master's thesis, Texas A&M University, USA.

Tether, B. S. and Metcalfe, J. S. (2003). Horndal at Heathrow? Capacity Creation through Co-Operation and System Evolution. *Industrial and Corporate Change*, 12:437–476.

T'kindt, V. and Billaut, J.-C. (2006). *Multicriteria Scheduling: Theory, Models and Algorithms*. Berlin: Springer, 2nd edition.

Trivizas, D. A. (1998). Optimal Scheduling with Maximum Position Shift (MPS) Constraints: A Runway Scheduling Application. *Journal of Navigation*, 51:250–266.

van Leeuwen, P., Hesselink, H., and Rohling, J. (2002). Scheduling Aircraft Using Constraint Satisfaction. *Electronic Notes in Theoretical Computer Science*, 76:252–268.

van Leeuwen, P. and van Hanxleden Houwert, N. (2003). Scheduling Aircraft Using Constraint Relaxation. In *Proceeding of the 22nd Workshop of the UK Planning and Scheduling Special Interest Group, Glasgow, UK*.

Veidal, E. (2007). Scheduling Aircraft Landings—The Dynamic Case. Master's thesis, Department of Informatics and Mathematical Modelling, Technical University of Denmark.

Wen, M. (2005). Algorithms of Scheduling Aircraft Landing Problem. Master's thesis, Department of Informatics and Mathematical Modelling, Technical University of Denmark.

Wen, M., Larsen, J., and Clausen, J. (2005). *An Exact Algorithm for Aircraft Landing Problem*. IMM-Technical Report-2005-12, Informatics and Mathematical Modelling, Technical University of Denmark.

Yu, S., Cao, X., Hu, M., Du, W., and Zhang, J. (2009). A Real-Time Schedule Method for Aircraft Landing Scheduling Problem Based on Cellular Automaton. In *Proceedings of the 1st ACM/SIGEVO Summit on Genetic and Evolutionary Computation, Shanghai, China*.