Constructive and Evolutionary Algorithms for Airport Baggage Sorting Station and Gate Assignment Problems

Amadeo Ascó Signes, BSc., MSc.
School of Computer Science
University of Nottingham

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

December 2013

Abstract

Correct assignment of airport resources can greatly affect the quality of service which airlines and airports provide to their customers. Good assignments can help airlines and airports to keep to published schedules, by minimising changes in these schedules and reducing delays. Given the expected increases in civil air traffic, the complexities of resource scheduling and assignment continue to increase. For this reason, as well as the dynamic nature of the problems, scheduling and assignment are becoming increasingly more difficult.

The assignment of baggage sorting stations to flights is one of the resource assignment problems at an airport, and like many other real world optimisation problems, it naturally has several objectives, which conflict with each other. A model of the problem is presented, different approaches to obtaining good solutions are looked at and studied to gain an insight into their qualities. Furthermore, algorithms are studied to improve the already good solutions obtained by the approaches considered and their performance is studied where some characteristics of the problem change, such as the number of baggage sorting stations or the topology of the airport.

Changes to the flight schedule on the day of operation may invalidate previous assignments of flights to resources. These perturbations may not only affect the disrupted flights but also other flights already assigned. Some existing approaches are looked at, and others are suggested to take account of these potential perturbations at the time the assignments are generated with the aim of mitigating their detrimental effect on the day of operation.

The constructive search algorithms and robustness methods are potentially important in a wider variety of problems other than the Airport Baggage Sorting Station Assignment Problem (ABSSAP). By way of illustration, the same techniques are applied to the widely studied Airport Gate Assignment Problem (AGAP).

This thesis is dedicated to my beloved mother, Celia Signes Vidal, for her teaching, guidance, support, and constant love throughout my life.

Acknowledgements

I am grateful to my supervisors, Dr. Jason A. D. Atkin, Prof. Edmund K. Burke and Associate Prof. Dario Landa-Silva whose encouragement, guidance and support from the start has enabled me to develop an understanding of the subject. Also to NATS, especially John Greenwood, for the opportunity to take this PhD and their continuous support.

In addition to the technical and instrumental assistance above, I received equally important assistance from family and friends. My partner, Dennis Latham, who provides me with on-going support throughout the whole process, as well as technical assistance critical for advancing the project in a timely manner. My parents who instilled in me, from an early age, the desire and attitude to excel, and for their continuous support and belief in me.

Contents

Glossary

A-CDM	Airport Collaborative Decision-Making
AAP	Average Assignment Point
ABSSAP	Airport Baggage Sorting Station Assignment Problem
ABSSAP	Airport Baggage Sorting Station Problem
ACO	Ant Colony Optimisation
AGAP	Airport Gate Assignment Problem
AGAP	Airport Gate Assignment Problem
AOA	Airport Operators Association
ARS	Area of Reduction in Service
ASP	Activity Selection Problem
ATC	Air Traffic Control
ATM	Air Traffic Management
ATRS	Act Tangent Reduction in Service
B&B	Branch and Bound
B&C	Branch and Cut
BA	British Airways
BAA	British Airports Authority
BILP	Binary Integer Linear Programming
BSARS	Base Sub-Area Reduction in Service
BSS	baggage sorting station
BSS	Baggage Sorting Station
BSSAA	Baggage Sorting Station Assignment Algorithm
BSSS	Baggage Sorting Station Selection
C1P	1-point crossover
C2P	2-point crossover
CDA	Continuous Descent Approach
CGA	Canonical Genetic Algorithm

CnP	n-point crossover
DSEMO	Dummy Single Exchange Mutation Operator
DSMMO	Dummy Single Move Mutation Operator
EA	Evolutionary Algorithm
ECJ	Evolutionary Computation Java library
EDD	Earliest Due Date First
ERS	Exponential Ranking Selection
ES	Evolutionary Strategy
ES	Elitist Selection
F&R	Fix and Relax
FAA	Federal Aviation Administration
FBT	Flexible Buffer Times
FCFS	First Come First Serve
FIFO	First In First Out
GA	Genetic Algorithm
GMC	Ground Movement Control
GRASP	Greedy Randomised Adaptive Search Procedure
ICAO	International Civil Aviation Organization
ILP	Integer Linear Programming
ILS	Instrument Landing System
IS1ES	Index Selection with Elitist Selection and a group size of 1
IS1SUMS	Index Selection with Stochastic Universal Modified Sampling and group size of 1
ISPEA2	Improved Strength Pareto Evolutionary Algorithm 2
ISxES	Index Selection with Elitist Selection
ISxSUMS	Index Selection with Stochastic Universal Modified Sampling
ISxy	Index Selection
LIFO	Last In First Out
LMAP	Lower Maximum Assignment Point
LMAP$_p$	Lower Maximum Assignment Point with Parking
LRS	Linear Ranking Selection
MA	Memetic Algorithm
MEBPFNRn	Multi Exchange By Pier between a Fixed Number of Resources
MEBPFNR3	Multi Exchange By Pier between a Fixed Number of 3 Resources

MEBPRNRn	Multi Exchange By Pier between a Random Number of Resources
MEBPRNR10	Multi Exchange By Pier between a Random Number of 10 Resources
MEBPRRNRxy	Multi Exchange By Pier between a Range Random Number of Resources
MEFNRn	Multi Exchange between a Fixed Number of Resources
MEFNR3	Multi Exchange between a Fixed Number of 3 Resources
MERNRn	Multi Exchange between a Random Number of Resources
MERRNRxy	Multi Exchange between a Range Random Number of Resources
MILP	Mixed Integer Linear Programming
MIP	Mixed Integer Programming
MOEGA	Multi-Objective Evolutionary Genetic Algorithm
MOOP	Multi-Objective Optimisation Problem
NSGA	Nondominated Sorting Genetic Algorithm
NSGA-II	Elitist Nondominated Sorting Genetic Algorithm II
OBT	Order Between Times
OBTLI	Order Between Times Lookahead and Improvement
ODT	Order by Departure Time
ODTLI	Order by Departure Time Lookahead and Improvement
OS	Operating System
OST	Order by Starting Time
PAES	Pareto Archived Evolution Strategy
PCBG	Probability of Conflict Based on the Gap
PDMOEA	Primal-Dual Multiobjective Optimisation Algorithm
PSI	Particle Swarm Intelligence
PSMO	Probability Single Multi Operator
RAM	Random Access Memory
RDEAMO	Remote Dummy Exchange All Mutation Operator
RDMAMO	Remote Dummy Move All Mutation Operator
RDSEMO	Remote Dummy Single Exchange Mutation Operator
RHC	receding horizon control
RISxyz	Range Index Selector
RITA	Research and Innovative Technology Administration
RMEBPFNRn	Range Multi Exchange By Pier between Fixed Number of Resources

RMEBPRNRn	Range Multi Exchange By Pier between Random Number of Resources
RMEBPRNRxy	Range Multi Exchange By Pier between Range Random Number of Resources
RMEFNRn	Range Multi Exchange between Fixed Number of Resources
RMEFNR2	Range Multi Exchange between Fixed Number of 2 Resources
RMERNRn	Range Multi Exchange between Random Number of Resources
RMERRNRxy	Range Multi Exchange between Range Random Number of Resources
RNC	Reduction in the Number of Conflicts
RSS	Remainder Stochastic Sampling
RWMS	Roulette Wheel Member Selection
SA	Simulated Annealing
SARS	Sub-Area of Reduction in Service
SC1P	1-point serial crossover
SC1P(2)	1-point serial crossover generating two children
SESAR	Single European Sky ATM Research
SMC	Surface Movement Control
SO	Sequential Operator
SPEA	Strength Pareto Evolutionary Algorithm
SSEA	Steady State Evolutionary Algorithm
SSEA1	Steady State Evolutionary Algorithm with $\ell = 1$
STAR	Standard Terminal Arrival Route
SUMS	Stochastic Universal Modified Sampling
SUS	Stochastic Universal Sampling
TMA	Terminal Manoeuvring Area
TMS	Tournament Member Selection
TRS	Total Reduction in Service Time
TS	Tabu Search
UESRS	Unsupervised Estimated Stochastic Reduction in Service
UMAP	Upper Maximum Assignment Point
UMAP$_p$	Upper Maximum Assignment Point with Parking
VEGA	Vector Evaluated Genetic Algorithm

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and motivation

The Federal Aviation Administration (FAA) predicted that, in spite of all of the challenges suffered during the last few years and from which we are still recuperating, the number of passengers travelling continues to grow over the long term, showing the importance of air transportation (Federal Aviation Administration (2010) and Federal Aviation Administration (2012)). The predicted growth in airport traffic will further increase the already high density of operations in some airports, especially large hub airports. These increases urge to consider the problem from different points of view, from the optimisation of the different individual areas to the analysis and study of the overall air traffic problem. Different initiatives currently exist directed at tackling some of the problems identified in a global view, for example the Single European Sky ATM Research (SESAR) (Commission (2010)) project which includes the Airport Collaborative Decision-Making (A-CDM) initiative. iFly (Keinrath et al (2008)) for en-route traffic aims to develop an advanced airborne self-separation design for European airspace. Similarly in the USA the NextGen-Airspace project (Swenson et al (2006)) intends to integrate the currently increasing optimal assignment of ground and air automation technologies.

A series of assignment problems must be solved before aircraft can arrive at or depart from an airport such as baggage sorting stations (BSSs) and gates, in addition to the performance of the multiple intermediate activities linked to these operations. Many parties are involved, each planning their own schedule. This leads to uncertainty and unreliability which may result in suboptimal solutions for the operations required to run an airport successfully. Airport partners set up schedules without knowing exactly where and when such resources as gates, baggage systems or aircraft, are going to be available. Some may later have technical difficulties or meteorological changes could occur, which are aggravated by the interdependency between the different airports in which the disturbed services operate. Creating a revised schedule using input from all stakeholders could potentially reduce the time required to recuperate from disruptions and provide more accurate departure times and a better view of available ground resources. This in turn emphasises the need to share data in order to improve decisions based on more accurate information.

According to Mueller and Chatterji (2002), only 16% of the air traffic delays are attributed to the point at which an aircraft is airborne, with 26% from taxi-out and 8% from taxi-in, the remaining delays derive from delays when the aircraft are at a gate (50%), which shows a greater potential for improvement in those operations assigned around an aircraft when it is at the gate (stand). This indicates that 84% of the delay relates to ground operations, which are defined as those operations performed on and around the ground in an airport.

Even where many different resources are involved in the daily operation of an airport, most of the research to date has been concentrated on a few types of resource, with the assignment of gates to flights being one type. Whereas the Airport Gate Assignment Problem (AGAP) has been widely studied the same cannot be said of the Airport Baggage Sorting Station Assignment Problem (ABSSAP) which also contributes to the successful operation of an airport, having a particularly significant influence on the satisfaction and opinion of passengers.

The mishandling of airport baggage in airports has been one of the more important issues for passengers for several years, both in Europe and the USA. It was ranked third in complaints after cancellations and delays in the report of the Air Transport Users Council (2009) and its importance was further emphasized in the April 2010 report of the Office of Aviation Enforcement and Proceedings (U.S. Department of Transportation (2010)), where over a hundred thousand baggage reports were logged, ranking baggage complaints as the second most common complaint. The expected increases in civil air traffic which are predicted by ICAO (2010) and the Federal Aviation Administration (2010) will continue to increase the complexity and difficulty of these problems.

I have studied both the ABSSAP and the AGAP, where planning and scheduling may contribute significantly to a reduction in airport delays. The aims are not merely to provide optimal or near optimal plans and schedules, but to ensure that these can cope with disruptions at the time of their implementation, removing or reducing the impact of such disruptions upon the daily operations at an airport.

1.2 Aims

The Aviation Network Management is expected to be an important element in improving the En-Route Air Traffic Services for which NERL (NATS En-route plc) is responsible in the UK, and Air Traffic Management (ATM) 'Network Management' is also strongly relevant to airport operations, since the airport operations occur prior to take-off, where more options for Network Planning exist. Related work in Europe, such as in the Eurocontrol Airport-CDM Initiative (A-CDM, Noël et al (2009)), suggests that airport operations should be considered as a whole, and that improvements in Air Traffic Control (ATC) will be limited if they merely cover the period following the pilot's calls for push and start.

This thesis considers the development of methods for co-ordinating land-side (passengers, baggage and gates) and airside (aircraft and tugs) planning in a collaborative decision-making airport system, i.e. the Eurocontrol Airport-CDM Initiative. Key goals are to develop techniques for building and maintaining plans, and investigating potential improvements to departure time predictions. The effect of sharing information, specifically during the arrival and departure processes, has already been studied by Andersson et al (2000), Böhme et al (2007) and Burgain et al (2009) which followed the same philosophy as in the A-CDM.

Given the above, it is advisable that the assignment of both resources, namely BSSs and gates (stands), be incorporated in the solution. The expectations are; minimising the deviation from identified ideal values such as Companies Aims (economic, statutory, image, etc.), Customers Expectations (satisfaction, price, etc.) and Employees Objectives (Satisfaction, fairness, pride, etc.).

1.3 Summary of the contributions of this thesis

The contributions of this thesis are summarised below.

Firstly, this thesis presents new models for both the ABSSAP and AGAP, which are used throughout the thesis. Whereas the AGAP and models have already been presented in the literature, my models represent a new approach which considers towing operations with extra constraints.

Secondly, the thesis provides insight into the differing behaviour of some constructive algorithms for these resource assignment problems, particularly where service time reduction is permitted. This allows for the generation of improved initial solutions when used with perturbative algorithms, enhancing the solution quality they can reach within a very limited search time.

Thirdly, a Steady State Evolutionary Algorithm has been designed to search the space of solutions in order to find solutions of high quality, which may be used in single objective or multi-objective resource assignment problems. New operators have been designed which generate feasible solutions at a high speed, providing improvement over those solutions obtained when using tools such as CPLEX, Gurobi and meta-heuristics such as Tabu Search (TS) and the Canonical Genetic Algorithm (CGA) for the problems considered. Additionally, the solutions reached have been shown to be of a higher quality than the initial solutions and good quality when compared with the Upper Bound provided by CPLEX. New replacement strategies were designed to improve on the solutions obtained, and these are shown to assist the search significantly in reaching statistically significantly fitter solutions when compared with other standard selectors. There is potential for combining all of these components in order to further improve the solutions reached from the point of view of both fitness and search speed. Insights into the effects of the different components of this algorithm are presented, and the way in which they affect the search is shown.

Fourthly, the thesis presents new approaches to taking account of the detrimental effects of delays within the solutions obtained on the day of operation. The grade in which the reduction of such detrimental effects is achieved on the day of operation is an indication of the robustness of the assignments. These approaches are then compared with other methods typically used in resource assignment problems, thus providing an insight into a wider range of approaches and their characteristics.

1.4 Overview of this thesis

Chapter 2 introduces some of the ground operation problems and the approaches which has been used in the literature, providing an overview of recent contributions in the various fields. Special attention is paid to the assignment of flights to gates which, together with the assignment of baggage sorting stations to airport flights, is presented in more detail in the following chapters.

Chapter 3 provides a detailed description of the ABSSAP studied and introduces the mathematical model which underpins the approaches presented in this thesis. The assignment of flights to baggage sorting stations within an airport is presented and the similarities and differences between this and the AGAP are also likewise presented.

Chapter 4 presents some constructive algorithms belonging to the group of exact methods which consider the topology of the airport in different situations. This is followed by an investigation of the results obtained when using the constructive algorithms to generate solutions for the airport baggage sorting station problem and their contribution to the different objectives of the problem.

Chapter 5 develops an evolutionary algorithm for solving resource assignment problems. New operators and selectors are presented to search the space of solutions and others are modified for use with the model already presented in Chapter 3. The algorithm is compared with other heuristic algorithms and the results from applying CPLEX and Gurobi solvers to an Integer Linear Programming (ILP) model of the problem, the results of which are summarised in the final section of the chapter.

Chapter 6 presents some approaches to building robust assignments for the problems studied. These approaches are compared amongst themselves and also with some initially disrupted schedules.

Chapter 7 provides a detailed description of the AGAP studied and introduces the mathematical model which is used in the following chapter. A variant of the constructive algorithms which were initially presented in Chapter 4 is presented and their applicability to the related problem of flight assignment to airport gates is considered. This is followed by an investigation of the results obtained when using the constructive algorithms to generate solutions for the airport baggage sorting station problem and their contribution to the different objectives of the problem.

Chapter 8 extends the proposed evolutionary algorithm presented in Chapter 5 and the robustness approaches presented in Chapter 6 to the AGAP, and considers their applicability to the related problem of flight assignment to airport gates. The modified evolutionary algorithm is studied and compared with other heuristic algorithms. Similarly, the robustness approaches are compared amongst themselves and also with some initially disrupted schedules. Finally all of the results are summarised in the final section of the chapter.

Chapter 9 presents the conclusions, lists the contributions and summarises the problem-specific results. Suggestions and recommendations for future research then follow.

1.5 Publications

The work in this thesis has previously been presented in the following full papers, abstracts and posters.

Full papers

- **Amadeo Ascó**, J. A. D. Atkin, E. K. Burke. The airport baggage sorting station allocation problem. In: Proceedings of the 5th Multidisciplinary International Scheduling Conference (MISTA), Phoenix, Arizona, USA. MISTA, August 2011.

- **Amadeo Ascó**, J. A. D. Atkin, E. K. Burke. An evolutionary algorithm for the over-constrained airport baggage sorting station assignment problem. In: 9th Intl. Conf. on Simulated Evolution And Learning, SEAL2012, Lecture Notes in Computer Science, vol. 7673, pp. 3241. Springer Berlin Heidelberg. Hanoi, Vietnam, December 2012.

- **Amadeo Ascó**, J. A. D. Atkin, E. K. Burke. An analysis of constructive algorithms for airport baggage sorting station assignment. Submitted to the Journal of Scheduling, resubmit prior to VIVA 2013.

Abstracts, posters and presentations

- **Amadeo Ascó**, J. A. D. Atkin, E. K. Burke. Resource Allocation at Airports and Dispatching Rules. 2nd Student Conference on Operational Research. Nottingham, UK, April 2010.

- **Amadeo Ascó**, J. A. D. Atkin, E. K. Burke. A comparison of constructive algorithms for baggage sorting station allocation. In: Proceedings of the 24th European Conference on Operational Research. Lisbon, Portugal, July 2010.

- **Amadeo Ascó**. Poster for the LANCS Initiative Advisory Board Meeting. Cumberland Lodge, Windsor Great Park, November 2011.

- **Amadeo Ascó**, J. A. D. Atkin, E. K. Burke. Airport resource allocation using constructive algorithms. In: Proceedings of the Operational Research 53 Annual Conference, Nottingham, UK, September 2011.

- **Amadeo Ascó**, J. A. D. Atkin, E. K. Burke. Over-constrained Airport Baggage Sorting Station Assignment Problem. 3rd Student Conference on Operational Research. Nottingham, UK, April 2012.

Chapter 2

Literature Review

This chapter starts with a non-technical introduction to the different operations which are considered at an airport. The non-technical introduction is followed by a review of the literature on the different identified areas in an airport operations, from when an aircraft arrives at the airport to when it departs. The content covered in this chapter provides the reader with the necessary background to put into context the problems considered and to better understand the work presented in this thesis.

2.1 Overview

The overall civil aviation problem is composed of the en-route, airport ground operations and airline problems. The en-route problem covers the time during which the aircraft is airborne and away from the airport vicinity. The airline problem consists of scheduling the flights and fleets. On the other hand, Airport Ground operations are those operations which are performed on and around the airport grounds, usually at the stand/gate, on the taxi ways, or at the runways. A stand is an area on the ground where aircraft are parked, with stands next to the airport gates, normally also called gates. The Airport Ground Operations Problem has been approached in many different ways to date, and has been subdivided into different subproblems in order to simplify the complexity and assist in achieving a solution, especially as different companies solve different parts of it.

The overall airport problem begins when an aircraft leaves the en-route phase, in the vicinity of an airport, where the Standard Terminal Arrival Route (STAR) starts (a STAR normally covers the phase of a flight that connect the cruise or en-route of a flight and the final approach to a runway for landing). The Air Traffic Control (ATC) guides the aircraft through both this and the approach phase until it lands safely. Once the aircraft has landed, it is guided through the taxi ways by the Ground Controllers until the aircraft reaches its assigned gate where the passengers disembark. Ground Controllers are responsible of the movement on the airport ground, such as taxiways, runways, holding areas and intersections. The passengers' baggage is then unloaded and transported to the assigned baggage sorting stations ready to be collected by their owners or transferred to their owners next flight. During the time the aircraft is at the gate, it is cleaned, re-supplied with fuel and food, and the appropriate safety checks are also successfully carried out. Near the time when the aircraft is due to depart, the baggage belonging to the next passengers is transported from the baggage sorting stations to the aircraft, where it is loaded. The passengers embark before the aircraft is permitted to leave the gate and join the departure sequencing, through which the aircraft will progress until it reaches the entrance to the runway ready to depart. Finally, the aircraft leaves the airport and it is guided by the Air Traffic Management (ATM) until it leaves the airport airspace.

Some approaches consider the overall airport in a more general way, for example in Janic (2007) where the allocation of airport runway capacity to expected demand is studied using a heuristic based upon a greedy algorithm, which was designed to minimise the cost of arrival and departure flight delays. Janic (2007) simple approach seems to have potential, as it provided sufficiently close empirical results to those obtained when using already established optimisation methods based on integer linear programming, although there are some concerns given the limited number of scenarios considered.

An overview of some of the ground operations and approaches used for each of these problems is presented in the following sections.

2.2 Scheduling flight arrivals

At an airport the arrivals of aircraft within the STAR have to ensure safe separation by controlling the speed, height and length of routing prior to the aircraft turning for its final approach, which is directed by the ATC onto the Instrument Landing System (ILS) when landing commences (Arrivals fact sheet Heathrow (2010)). It is required to take account of safety constraints maintaining standards by considering aircraft separation and controller workload. An aircraft generates wake vortices (turbulence) which may affect the aerodynamic stability of the following aircraft depending on the distance between them. This distance depends on the aircraft weight class and its speed, which have recently been extended to take account of the wing configuration (Tittsworth et al (2012), Administration (2012)). When the airport is busy and approach delays are expected, aircraft arriving may be held by ATC in 'holding stacks' before being instructed to make their final approach, Figure 2.1. Aircraft in the 'holding stack' circle at different heights until the way is clear for them to commence their final approach, or to move to a lower cycle ready to land. Nevertheless a Continuous Descent Approach (CDA) is preferred, given its advantages in that it provides a reduction in noise and fuel consumption, thereby reducing costs, cutting emissions, and providing overall environmental benefits, Clarke et al (2004) and Alam et al (2010). Thus the traffic movements have to be carefully planned to limit peaks of activity and assure smooth operations, but even so a CDA does not always happen (Arrivals fact sheet Heathrow (2010)).

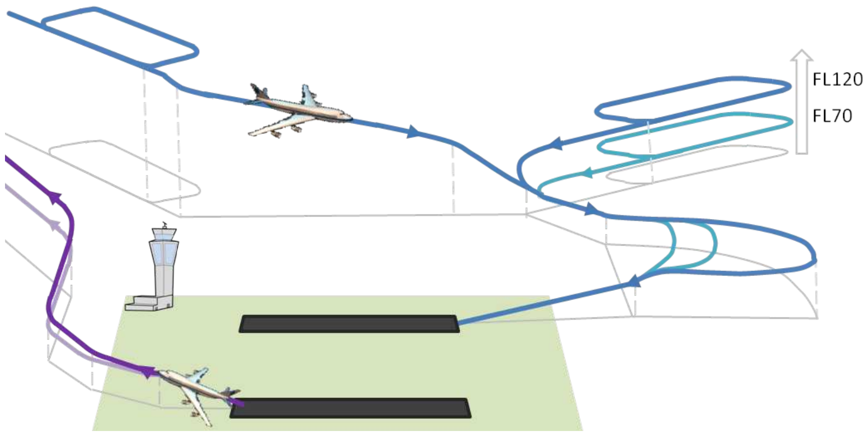


Figure 2.1: Overview of an airport departures and arrivals with stacks approach.

Hansen (2004) extended the study conducted in Cheng et al (1999) which used Genetic Algorithms (GAs) with two ways of representing the problem of scheduling arriving aircraft to available runways: one genome and the multiple genomes to define a complete runway assignment, sequencing and scheduling, to minimise delays satisfying safety constraints. The problem was simplified by sub-dividing it into arriving groups or banks of flights within which the problem is solved, so the solution may be optimal within a bank of flights but may not be so on the overall day of operation. Coverage of the arriving flight by the bank of flights may be increased by enlarging it which will in turn increase the time consumed in finding solutions, but means that results approximate more closely to those for the overall arrival problem. The results in Hansen (2004) corroborated the original suggestions in Cheng et al (1999), highlighting the potential of GAs in solving this problem. Hu and Chen (2005) introduced the concept of a receding horizon control (RHC) to the problem of scheduling and sequencing arrivals, and investigated the effects on airborne delays and computational burdens. Similarly as with the bank of flights, the RHC solves the problem for all flights entering the STARs, in this case, within a time window which is shifted forward and the process repeated until the overall problem time period is covered, i.e. all of the flights have been sequenced and scheduled. For simplicity, aircraft waiting to land are classified in a relatively small number of distinct categories, according to speed, capacity, weight, and other technical characteristics which are then used to perform position shifting (PS) with the main objective of minimising total airborne delays, Figure 2.2. One of the important parameters in establishing this approach is the size of the horizon considered, which has a direct effect on the speed of the approach, and is very important for online systems. Furthermore, the online updating of information, which is then fed back into the following horizon calculations, improves the decision-making and increases the robustness of the solutions obtained. This approach is equally applicable to both problems studied in this thesis: the Airport Baggage Sorting Station Assignment Problem (ABSSAP) and the Airport Gate Assignment Problem (AGAP), in order to reduce the computational time required to reach potentially good solutions.

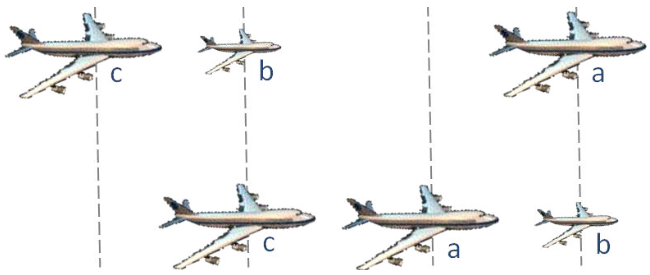


Figure 2.2: Position shifting (PS) base on aircraft separation.

Xiangwei et al (2010) suggested a GA where chromosomes are constructed as a permutation of the categories of aircraft arriving, reducing the encoding space such that the search speed is improved when compared with an aircraft order based GAs, but this only considers the static problem and one runway.

Relying on the stack delaying mechanisms described previously, a set of time windows in which landing is possible can be associated with each aircraft entering the airport airspace. These windows were used by Artiouchine et al (2008) in their Mixed Integer Linear Programming (MILP) representation of the runway sequencing using a fixed window size for the whole problem, which gave good results, and it was extended to consider different window sizes using a hybrid algorithm with a Branch and Cut (B&C) mechanism. A MILP corresponds to the minimisation or maximisation of a problem with linear objectives and subject to linear constraints where some variables in the model are real and some of the variables are integer. B&C involves running a Branch and Bound (B&B) algorithm and using cutting planes. B&B is an exact approach to find the optimal solution to a problem, where the entire set of feasible solutions is divided by partitioning it into smaller and smaller subsets, and the best possible solution of a subset of solutions is computed. When the provably best possible solution in a sub–tree is found then there is no need to further branch that sub-tree, which speeds the search, and a further description may be found in Wolsey (1998) and Burke and Kendall (2005). This thesis uses the B&B algorithm for the ABSSAP, which uses my Integer Linear Programming (ILP) representation of the problem.

Caprì and Ignaccolo (2004) considered both arrival and departure scheduling in both the static and dynamic problems when using a GA, and Böhme et al (2007) looked at the co-ordination of airport arrival and departure management when applying mixed mode operations, where a runway is used for both arrivals and departures, using an algorithm based on fuzzy rules, and moving from a minimum separation sequencing for arrivals to a time-based scheduling. Böhme et al (2007) experimental results for Frankfurt Airport indicated that the total throughput of aircraft was enhanced and, in the case of departures, increased in punctuality with only a minor extension of flight arrival times.

It is widely agreed that given the cost involved in flying an aircraft, which greatly depends on the time expended in the air, it would be preferable to keep it at the original airport gate rather than keep it in the air (stacks) waiting to land, which would in turn facilitate the implementation of the CDA. In order to achieve this, good communications are required and data shared between airports, which would improve decisions, improve the airports efficiency and reduce airspace congestion which are some of the main objectives of the overall problem.

The increase in the time flights stay at their assigned gate or are delayed in reaching that gate may have detrimental effects for the remaining flights assigned to the same gates, and may also have a potential effect on other resources, such as baggage sorting stations (BSSs). Flights which require a considerable extension of their stay on the ground awaiting permission to depart may be moved to remote stands or holding areas where they can await clearance to depart without affecting other flights and resources. Nevertheless, those flights which extend their stay at the gate originally assigned to them or which arrive late may have detrimental effects on other flights and resources. Any uncertainty as to arrival time may also have a potential effect on the other assignments, such as those of flights to gates and BSSs. Thus, it would be advantageous if these instances could in some way be taken into account in their assignment so that where they occur these such detrimental effects are reduced or completely removed. Such assignments are in this thesis said to be robust. Robustness is studied in Chapter 6 for the ABSSAP, and also Section 8.4 for the AGAP.

2.3 Ground movement

Ground movement concerns the movement of vehicles, particularly aircraft, on the ground around the airport, Figure 2.3. This generally includes all holding areas, taxiways, inactive runways, and some intersections and transitional aprons where aircraft arrive, having vacated the runway or stands. Any bottleneck in the aircraft flow on the taxiways could therefore increase the ground delays and decrease the airport capacity. The Ground Movement Control (GMC), also called Surface Movement Control (SMC), is responsible for the strategic assignment of aircraft to a runway, with the main aim being a reduction in delays, whilst operating within regulations and constraints based on traffic volume and weather conditions. Some of the regulations and constraints refer to the aircraft ground movements in taxiways, such as aircraft separation, cross points occupancy and procedural constraints, as used in Capozzi (2003). Different areas of potential economic and environmental interest have been identified in ground movement by Gelinas and Fan (1979) and Miller and Clarke (2004). An overview, categorisation and critical examination of previous research in ground movement is presented in Atkin et al (2010).

An overview of some of the ground movement operations and approaches used are presented in the following subsections, starting with Taxiing, followed by the scheduling and routing of trucks, de-icing/anti-icing machines and finishing with a short view of other ground movement operations.

2.3.1 Taxiing

Taxiing relates to the routing of flights from their entrance point via the airport ground infrastructure to their assigned stand (either local stands at a gate or remote stands on an apron) and back to the departure area. As such it links together the main airport operations.

Some research has taken place on understanding and solving the ground problem, examples of which are throughput, congestion and terminal volume (amount of traffic), an example of this being the CSD (1999). Objectives evaluated were how to reach the destination as swiftly as possible, meet safety requirements and maximise

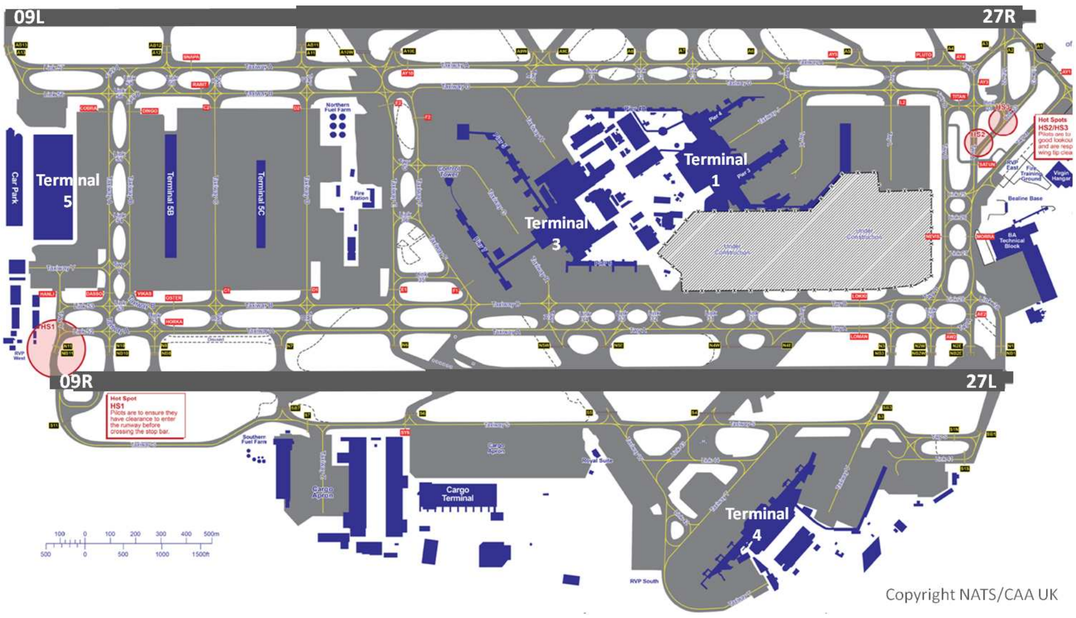


Figure 2.3: London Heathrow airport grounds and taxi ways.

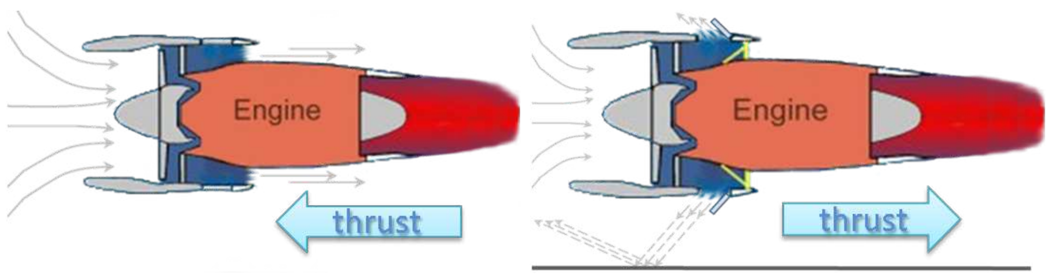
utilization of the taxi ways while avoiding conflicts. Similarly more recently, Roussos and Kyriakopoulos (2009) presented an approach using a 3D aircraft collision avoidance system implementation, which uses repulsion fields, which could be adjusted for airport taxiing. I nevertheless anticipate some concerns regarding the time taken in solving the taxiing problem, since the potential fields have to be calculated per aircraft, and where the potential movements of an aircraft take account of other aircraft. An opportunity to use concurrency arises when obtaining the fields, and some speed is expected to be gained when passing from the original 3D problem to a 2D problem. Clare and Richards (2009) showed that in average taxi times can be reduced when using RHC approach compared to a First Come First Serve (FCFS) approach. Gotteland and Durand (2003) used GA for the minimisation of taxiing time, and Marn (2006) used B&B and Fix and Relax (F&R) methodologies for the aircraft routing and scheduling on the airport ground.

The importance of robustness when assigning flights to gates is emphasised by the fact that ground movement links together arrivals/departures runway sequencing with gate assignment, such that uncertainty and disruptions at these stages are likely to propagate to the gate assignments with potential undesirable consequences such as reassignment of flights to other gates or even cancelation of flights.

2.3.2 Scheduling and Routing of towing trucks

A stand is an area in the airport grounds where aircraft are parked, with a remote stand being one which is not located immediately beside the airport buildings, whereas gates correspond to those stands which are located at the airport terminal buildings. Aircraft need to travel from their assigned stand to the departure holding areas ready for departure. This operation may be executed by using the aircraft's engines or by towing trucks. Whereas many aircraft are now able to move backwards on the ground using reverse thrust, the jet blast from the engines may cause damage to the terminal building and equipment, with the added hassle provided by engines close to the ground which may blow sand and debris forward and then suck it back, causing damage to the engine, Figure 2.4b. This does not happen when using normal thrust given that the air flow enters the front of the engine and leaves from the rear, whereas in reverse thrust the air also enters from the front but leaves from the lateral parts of the engine, Figure 2.4. This makes the towing trucks a preferable alternative to pushback when moving from gates to the departure holding areas. Additionally, the high price of fuel and an increase in environmental concerns has revitalised interest in using different means of reducing these factors, rendering previous research especially relevant such as Gelinas and Fan (1979); Fan (1990); Miller and Clarke (2004). This trend has been confirmed by the UK Airport Operators.

On 30^{th} June 2010 the Airport Operators Association (AOA) launched new guidelines to reduce aircraft ground emissions, which amongst other initiatives, outlined aims to increase taxiing with no engine, which may increase the use of towing trucks during taxiing. Merlin (1983) and UNIQUE (2005) both considered the use of towing for taxiing aircraft as a means of reducing contamination. However the following disadvantages, which were mentioned by controllers, were not considered:



a Forward thrust. b Reverse thrust.

Figure 2.4: Engine thrust.

- The speed of loaded towing trucks is lower than the speed of aircraft using their own propulsion systems, thus increasing congestion on the ground.

- The propulsion system of an aircraft requires reaching a specified temperature before take-off, which means that when it reaches the departure sequencing point the aircraft has to start its engines and wait until the engine take-off temperature is reached. This may entail some extra work for the GMC in the instruction to start engines, and introduce uncertainty as to when the aircraft is ready for take-off. Any uncertainty may affect current flight assignments to BSSs and gates.

The assignment of towing trucks (also called tugs) to aircraft is pre-calculated, based upon the planned stand allocations and arrival/departure times. Perturbation in the flight arrival sequences may affect their stand allocations which may therefore have to be re-allocated. It is believed that towing truck assignment may not always be re-allocated in these circumstances, i.e. the stand allocation moves but the tow plan does not. One area of investigation is the effect of this lack of re–planning. Importantly, how often does stand re-planning mean that the departure times for these towed aircraft are no longer achievable?

Other points for consideration at London Heathrow airport are presented below.

- Towing truck teams are qualified in specific aircraft types, so there are issues if they are re-allocated.

- Towing trucks which are not towing aircraft control themselves (driver), with the responsibility of keeping out of the way of other aircraft.

- Towing trucks which are towing respond to instructions from the tower, GMC.

- There are eleven handling agents and over ninety airlines involved at London Heathrow airport.

- Aircraft from remote stands normally do not require towing.

No previous work directly related to scheduling and routing of towing trucks was found, although various scheduling papers could be relevant, such as Du et al (2008) which considered the assignment of flights to oil tank trucks that may have different fuel capacities and Kolischa and Hartmann (2006) investigated heuristics for the resource constrained project scheduling. The scheduling presents similarities to and has interdependencies with the AGAP and ABSSAP, while the routing has similarities to the routing of taxiing aircraft. The model for the AGAP presented in Chapter 7 considers the use of towing trucks. Also the minimisation of the number of towing operations is one of the objectives in the AGAP model presented in Chapter 7.

2.3.3 Scheduling and Routing of de-icing/anti-icing machines

The predictions of cooler winters in the UK, Lockwood et al (2010); Seidenkrantz et al (2009), have prompted some airport operators to order more de-icing machines revealing an increasing influence of these resources on the ground operations of airports in the UK. Norin et al (2007) describe a decision tool for the de-icing process which uses a Greedy Randomised Adaptive Search Procedure (GRASP) whereas in Norin et al (2009) an optimisation algorithm to schedule de–icing trucks is developed which is integrated within a simulation model in which the results show a reduction in flight delays and waiting times.

Flights needing de–icing/anti–icing require this operation to be scheduled, which is normally conducted at the gate, but may also be executed in a remote location. In both cases, these operations need to be considered when assigning flights to gates as they may affect the assignment and length of time the flight needs to spend at the gate.

2.3.4 Other ground operations

Several operations need to be completed at the gate before a flight is ready to start its departure process. Aircraft may be fuelled by oil tankers or underground pipelines. Aircraft located at gates without pipelines have to be fuelled by oil tankers which may have different fuel capacities. Du et al (2008) presented an Ant Colony Optimisation (ACO) algorithm with Max-Min and Rank-based Ant System with an heuristic called Earliest Due Date First (EDD) to solve the multi-objective assignment problem of oil tankers to flights with a minimisation of the number of oil tankers required, the total start time for servicing flights, and the total flow time of oil tankers. Similarly, flight catering requires vehicles to transport and place the required supplies next to the aircraft in readiness for its next flight. These operations increase the airport ground traffic, and if they were to be considered in conjunction with the other operations presented in this thesis, will in turn further increase the complexity of the overall problem.

2.4 Scheduling and assignment of flights to stands

Many parking positions are located beside the airport buildings, next to the gate from which passengers will board the aircraft. There are often other parking positions off the terminal (on the apron) which are also called stands, where aircraft may be parked for longer periods, but when used to embark passengers then they must be transported by mobile lounge or bus, thus increasing the congestion on the ground. Gates, however, correspond to those stands which are located in the airport terminal buildings, as shown in Figure 2.5 numbered in blue. Those stands not located immediately by the airport buildings are called remote stands. The gates provide extra services to those that are provided at a remote stand. A description of the AGAP and a model is presented in Chapter 7.

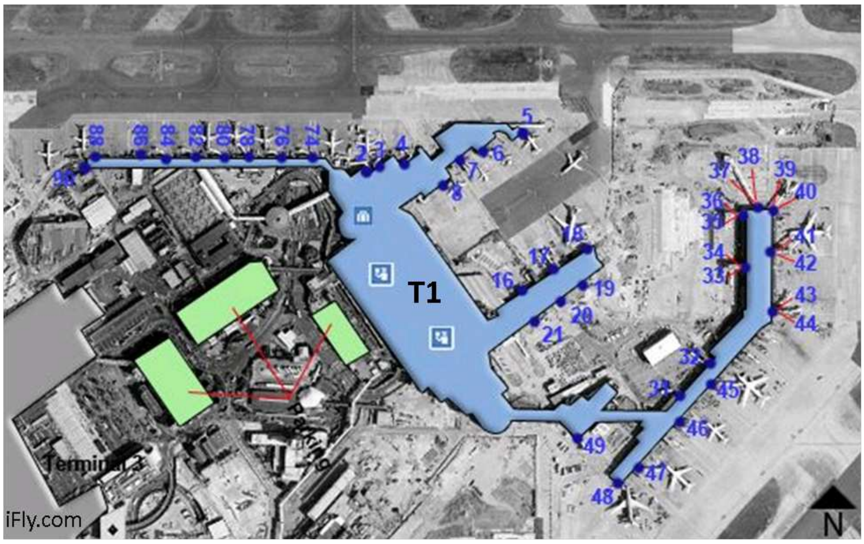


Figure 2.5: London Heathrow airport Terminal 1 with gate location.

According to Mueller and Chatterji (2002) whereas only 16% of the air traffic delays are attributable to the point at which the aircraft is airborne with 26% from taxi-out and 8% from taxi-in, the remaining delays may be derived from delays where the aircraft are at a gate (50%), which reveals the gates to be of considerable importance in reducing overall airport delays.

The assignment of gates is planned in advance for seasonal flight schedules, which equates to the static problem, whereas the stand planners prepare and modify plans on the day of operation, which plans are frequently updated to accommodate disruptions and delays on the day, when the time available to achieve good changes is greatly limited. This is sometimes called "dynamic scheduling" and frequently results in suboptimal schedules. Exact methods are more appropriate for the off-line static problems, given the greater solution time available. Babic et al (1984) and Bolat (1999) attempted to reduce the required algorithm execution time by using a one-pass constructive heuristic requiring times of up to 209.6 sec, considerably lower than B&B but not sufficiently fast to solve the dynamic problem. In the real-time problem the time available to execute the algorithms is greatly reduced, Ding et al (2005); Lim et al (2005); Dorndorf et al (2008); Drexla and Nikulina (2008); Wei and Liu (2009).

The gate assignment problem is normally presented as a multi-constraint and multi-objective problem where different objectives have been used in different papers but not always together, and in some cases not all of the objectives were used, to reduce the complexity of the problem, a model is presented in Chapter 7. In the Airport Gate Assignment Problem (AGAP) different approaches have been followed. Babic et al (1984) used B&B with the object of minimising passenger walking distance, with some enhancements to accelerate the computation. Mangoubi and Mathaisel (1985) took account of the transfer of passengers using linear programming relaxation and greedy algorithms, and Bihr (1990) used Binary Integer Linear Programming (BILP) to solve the minimum walking distance, whereas a GA was used by Lim et al (2005). Where traditional ACO used pheromone trail information to construct complete solutions to the AGAP, Pintea et al (2008) used a hybrid ant-local search system where pheromone trail information is used to perform modifications on AGAP solutions.

The objective most used in the current AGAP literature corresponds to the improvement in service satisfaction, assured to be achieved by reducing passenger walking distance inside the terminal building. When considering the passengers walking distance the AGAP can be modelled by analogy with the NP-hard quadratic assignment problem, Obata (1979), Pardalos et al (1994) and Cela (1998), which is a facility location problem where the cost of assigning a flight to a gate depends on the assignment of other resources and the transport volume between two resources (see also Lawler (1963)).

Various techniques have been applied to solve this problem, for instance, Baron (1969) uses simulation to analyse the effects of passengers' walking distance resulting from different gate usage strategies where both local and transfer passengers are considered, Babic et al (1984) and Bihr (1990) use linear binary programming, whereas Xu and Bailey (2001) use mixed 0 - 1 quadratic integer programming and Tabu Search (TS). Gu and Chung (1999) make use of a genetic algorithm, multi-objective programming is used by Yan and Huo (2001), Ding et al (2004) uses Simulated Annealing, Ding et al (2003) presents a Simulated Annealing approach and a hybrid of Simulated Annealing and Tabu Search, and Lim et al (2005) use both a TS and a Memetic Algorithm (MA) which their results appear to improve on those obtained from CPLEX and a GA. A survey on AGAP is presented in Dorndorf et al (2007a). Usually the improvement in service satisfaction created by reducing passenger walking distance inside the terminal building is considered by airlines owning part of the terminal, and are in charge of producing the scheduling (e.g. British Airways (BA) at London Heathrow airport Terminal 5), whereas at some international airports the stand planning is performed by the airport authorities (e.g. London Heathrow airport terminals 1, 3 and 4). As a consequence the stand planners often do not have complete passenger data, and importantly they do not have all transfer patterns. Although in some airports the planners consider passenger walking distances by other means, such as locating flights with numerous passengers close to the terminal building, they are not in a position to perform a full minimisation of passenger walking distance. This is not the case for London Heathrow airport where the flight assignment first identify the stands that physically can hold the aircraft, following by the type of flight International, Domestic or CTA (Irish), and finally it is considered the airlines or handler preference usually based on lounges or equipment locations etc.

Gosling (1990) considers that it may be more important to the assignment that the distance the passengers have to travel is reduced rather than minimising the walking distance. Other approaches to the problem which do not consider the walking distance are presented in Gu and Chung (1999) and more recently in Diepen (2012).

From the point of view of robustness, by the nature of the objective function, the proposed procedures (mostly heuristics) usually assign most of the flights to a few attractive stands. The assignments to heavily utilised stands will easily be disrupted even by minor changes in flight schedules, as discussed in Mangoubi and Mathaisel (1985). Bertsimas and Sim (2004) investigated ways to decrease what they called the price of robustness; the trade–off between the optimality of the solution for some given objectives and its robustness. A survey in theoretical and applied robust optimisation is presented in Bertsimas et al (2011).

Terms commonly used in robust assignments are ‘Idle time’, which refers to the time between two consecutive assignments to the same resource (e.g. BSS or gate) where no activity is assigned to the resource, and buffer time, which refers to a predetermined amount of time introduced at the beginning and/or end of an activity. Whereas the buffer time implies a preference for a particular gap size between consecutive assignments to the same resource, an ‘idle time’ does not.

Since the total time available to stands and the total ground time of flights are constant, the total 'idle time' is constant and independent of the way in which the flights are assigned. Bolat (1999) has proposed models to utilize the gates as uniformly as possible so that the assignments are robust enough to absorb minor changes in flight schedules, by introducing a model to minimise the range of the idle times, being the difference between the maximum and the minimum idle times. The variance of idle times at the gates using B&B and dynamic priority functions are developed to guide the assignment process in Bolat (2000). Dorndorf et al (2007b) propose an approach for obtaining robust gate assignments based on recovery strategies, and discuss several robustness related concepts based on resource-switching. A review of how disturbances at a given airport could be handled and a survey on recovery from airline schedule perturbations is presented in Filar et al (2001).

Another technique used to improve robustness of the scheduling is the application of buffer times to the assignments at the expense of reducing aircraft productivity in order to minimise system costs caused by operational uncertainties. The objective is to absorb small perturbations on the day of the schedule implementation. A novel encoding which uses fixed buffer times (without any dependence on the type of flight or airline) where conflicts in the objective functions are allowed, and does not consider the passengers' walking distance, is presented in Li (2009). Simplex with improved variables, linear relaxation, B&B and buffer times is also presented in Yan and Huo (2001) which may be too time consuming in the case of very large problems using B&B. A simulation framework is proposed in Yan et al (2002) which is able to analyze the effects of stochastic flight delays on static gate assignments, and evaluate flexible buffer times with just one objective, the passenger walking distance.

Stochastic Optimisation Models are another means of considering the uncertainty inherently present in the AGAP. An extension of deterministic robust approaches based on the buffer and idle times is presented in Seker and Noyan (2012) which is used to develop stochastic optimisation models and a TS which uses swap and insertion neighbourhood strategies to find solutions.

A different approach with three components, a stochastic gate assignment model, a real-time assignment rule, and two penalty adjustment methods which consider both the planning and the real-time stages, is presented in Yan and Tang (2007). Wei and Liu (2009) model the AGAP using fuzzy where the idle times of flight to gate are regarded as fuzzy variables using a modified genetic algorithm, the results of which are compared with those obtained when using buffer times. Similar techniques for robustness are utilised in this thesis in Chapter 6 and Section 8.4.

Whereas in many instances the problem, which corresponds to a multi-objective problem with conflicting objectives, has been restricted to optimising the most important objective (mainly the passengers' walking distance as in Bihr (1990); Haghani and Chen (1998) or by aggregating the individual objectives into a single scalar (the weighted sum of the individual objectives as in Cheng (1997); Ding et al (2004); Lim et al (2005); Wei and Liu (2009)), there are more recent studies which maintain the multi-objective essence of the problem by generating diverse and equally distributed sets of high quality trade-off solutions in a single run of the algorithm as in Hu and Di Paolo (2007); Wei and Liu (2013). A survey which concentrates on continuous Multi-Objective Evolutionary Genetic Algorithms (MOEGAs) is presented in Zhou et al (2011).

Other studies considering stands have brought to light the advantages of absorbing some of the inevitable aircraft delay at the stands, as presented in Atkin et al (2011) which shows a significant reduction in take–off delays.

2.5 Scheduling of baggage sorting stations to flights

A BSS is the part of the baggage system where the passengers' baggage is collected and temporarily stored ready to be sorted and transported to the side of the aircraft where it is loaded onto, Figure 2.6. The scheduling of baggage sorting station is presented in more details in Chapter 3.

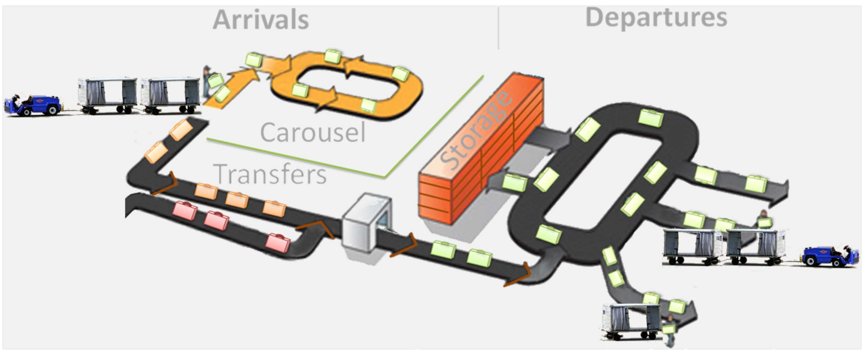


Figure 2.6: Overview of airport baggage system with storage.

This area seems to have been neglected until recently, and the earliest study found in the literature is Abdelghany et al (2006) which uses the activity selection algorithm, modified to satisfy different operational requirements, to study the trade-off between different operational constraints and requirements to reach a satisfactory near optimal solution. Frey et al (2010) consider the storage, BSSs and other required resources such as carts and parking, and use decomposition, where the problem is split in different sub-problems which can be modelled as different Mixed Integer Programming (MIP), to solve the problem. They also provide a model and a proof of its NP-hardness (where the sub-problems are also NP-hard). Barth and Pisinger (2011) also consider a baggage system with internal storage capacity and this considers the problem to be not merely one of assigning BSSs but also the BSSs starting time to be used, with the application of two methods GRASP and a decomposition approach based on different MIP for the static problem. It was concluded that as the GRASP is faster it could potentially be used in the dynamic problem. A model for the ABSSAP is presented in Section 3.3.

Another study not directly related to ABSSAP but which gives an insight into the problem was presented in Robinson (1969) which used simulation models to evaluate alternative designs of hypothetical baggage handling systems for large-capacity aircraft under different baggage-per-passenger conditions. Pitt et al (2002) concentrate on airport configurations and available types of some resources, providing general conclusions based on the configuration, size and expected expansion of airports. Rijsenbrij and Ottjes (2007) present new concepts for baggage transportation to and from narrow-body aircraft and estimate the time required to service a flight from the point of view of baggage handling, also depending on the resources used to upload/download the baggage. It gives an idea of the current methods of baggage handling and implies that automatic scheduling improves the process. Finally Johnstone et al (2010) specifically refers to dynamic baggage routing, baggage handling system (BHS) control, with the use of a status-based routing algorithm which applies learning methods to select criteria based on routing decisions.

2.6 Scheduling flight departures

Once the passengers are onboard the aircraft and all of the required operations have been successfully completed, the aircraft is ready to proceed with its departure. The ground control gives clearance to the aircraft to proceed to the holding areas at the end of the take-off runway, Figure 2.7, where they wait in queues for permission to take-off. A runway controller guides the aircraft through the holding areas whilst attempts are made to find the best order for aircraft take-off simultaneously, taking the necessary safety requirements into account, such as sequence-dependent separation rules which depend upon aircraft size, departure route and speed group.

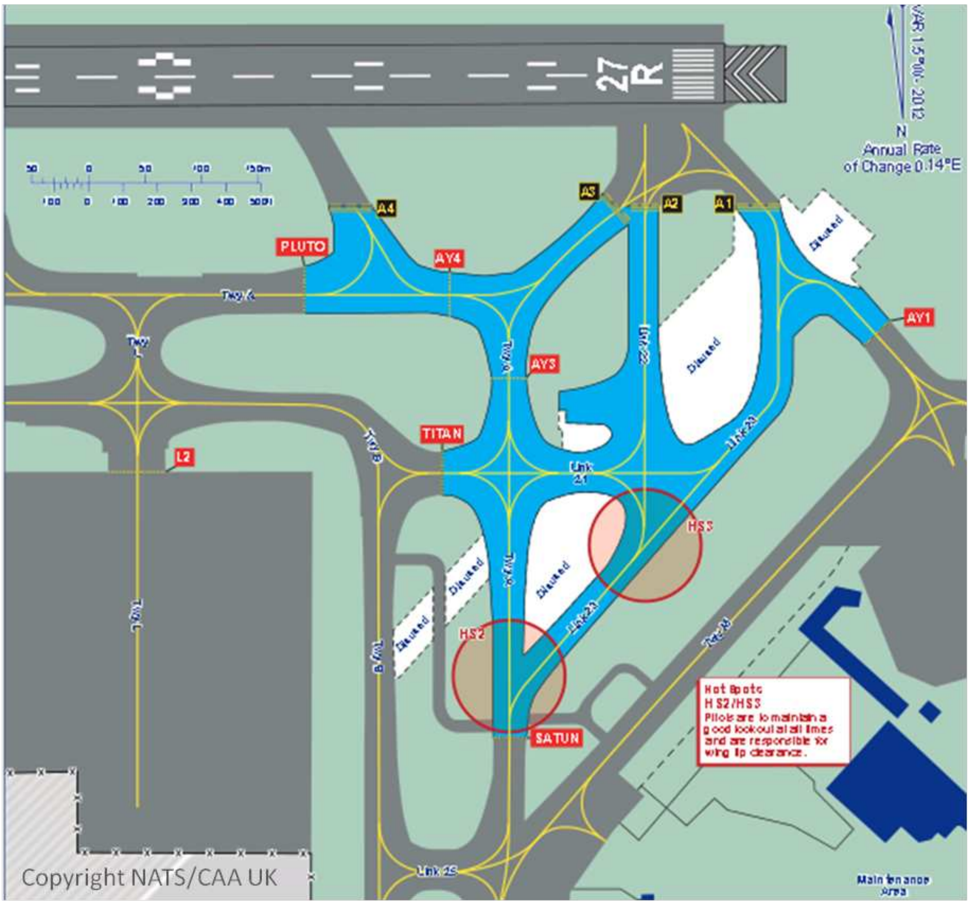


Figure 2.7: London Heathrow airport holding areas for RWY 27R.

Bolender (2000) considers the construction of optimal departure sequences for the aircraft being queued, using several greedy search algorithms which were compared with a GA for the static problem. This concluded the need for a queue assignment algorithm. Anagnostakis and Clarke (2003) presented a system structure and a formulation of runway operations, planning problems, and more specifically, departures, for the static problem where the airport layout makes runway crossing necessary. This shows how the geometry of an airport, particularly the runways, creates interdependencies between the scheduling of arrivals and departures. However, the real systems are dynamic and there is usually significant uncertainty associated with any prediction, partly because the information required is not always available at the time it is required, but this uncertainty is reduced as the time between the prediction and the implementation shortens.

Atkin et al (2007, 2008) presented a decision support system which considers the taxiing aircraft in addition to those already at the holding area, increasing the available information, which could help to improve significantly the departure sequence at busy times of the day. The advisory system is based on Hybrid Metaheuristics to help obtain take-off orders that would improve the throughput and reduce delays at London Heathrow airport. Empirical results for real data from London Heathrow airport corroborated the potential of this approach and highlights the dependency on the volume of traffic and the accuracy of the estimated taxing times. Böhme et al (2007) looked at mixed mode operations and took into account both the situation of the departure traffic on the ground and the arrival situation in the Terminal Manoeuvring Area (TMA) using Fuzzy Reasoning. Bianco et al (2006) presented heuristics with a job-shop model to solve the problem of arrival and departure sequencing and scheduling allowing for different runway configurations, and Xiujuan et al (2008) implemented a hybrid algorithm composed of Particle Swarm Intelligence (PSI) optimisation combined with Simulated Annealing (SA). Similar to GAs, PSI is also based on a fitness function that is optimised through population mutation and crossover, but the focus lies not only with the optimisation of a global fitness function but also the maximisation of local neighbourhood fitness where individual particles also communicate their fitness locally to other particles Kennedy and Eberhart (1995); Eberhart et al (2001).

A study of the effects of the constraints, using a simulation of the London Heathrow airport departure system, is presented in Atkin et al (2009) where some physical and operational changes in the way the departure system currently operates are suggested. The interested reader is directed to Atkin et al (2007, 2008); Atkin (2008) for a more extended study of London Heathrow airport departures.

Delays in the departure of aircraft may extend the stay of the aircraft at the gates with potential detrimental repercussions to those flights already assigned to the same gates and those resources also assigned to those flights such as BSSs. It would be desirable if the assignment of flights to both BSSs and gates could cope with some of the uncertainty inherently present in the departure such that delayed departing flights may be able to stay at the gates for longer without any or minimum detrimental effects to the rest of the schedule and assignments.

2.7 Evolutionary Algorithms

An Evolutionary Algorithm (EA) is a population-based mechanism inspired by biological evolution, such as reproduction, mutation, recombination (population selection), and parent selection (member selection), which are based on the Darwin and Wallace (1858) theory of natural selection as developed in the former's classic foundational work Origin of Species (On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life, Darwin (1859)) and Mendelian genetics (Experiments in Plant Hybridisation, Mendel (1865)), which are recognised as the foundation of evolutionary biology.

GAs have been used in the solution of a wide range of problems and are one of the methodologies belonging to the population-based model of EAs. GAs are population based approaches which encode the problem solutions on a chromosome-like data structure, the population being composed of solutions. Solutions are then selected, based on the reproductive allocated opportunities, following which recombination operators are applied in order to produce new solutions in the solution search space. The genetics principles were taken from biology and then applied to artificial systems, based on the work of Holland (1975) and DeJong (1975) which constituted the origin of GAs. The early theoretical studies of GAs included such works as Vose and Liepins (1991) which aimed at achieving a better understanding of the Simple Genetic Algorithm (which is alternatively titled the Canonical Genetic Algorithm (CGA)) using the support of matrices (Walsh matrix), and Prugel-Bennett and Shapiro (1994), which applied a statistical mechanics-style approach in order to explain behaviour. Hinton and Nowlan (1987) investigated the way in which learning can mould the fitness landscape, since an individual's fitness will consist of a genetic contribution, referred to as crossover, and a learned contribution known as mutation. Goldberg (1990), Whitley (1991) and Holland (1975) explored the problems of exploiting linkage and the recombination of tagged representations. Eiben et al (1995), Tsutsui and Jain (1998), and Eiben (2003) studied both the effect of using multiple parents and multiple crossover points. These studies emphasises the importance of operators. Blickle and Thiele (1996) presented an analysis of some different selection schemes, with the objective of overcoming the premature convergence problem, wherein offspring are never superior to their parents. Some typical selection operators are shown in Section 2.7.1.

Theoretical studies of the GAs however were and still are based on a binary problem representation which arguably restricts its applicability, but undoubtedly assist an overall understanding of GA workings.

The terms phenotype and genotype are typically used in genetics to assist in the explanation and comparison of individuals. The phenotype is the observable realisation of an individual (in this thesis an individual is the equivalent of a solution), where the genotype refers to the makeup of the same individual. For example, when considering the two genes determining the organism's gender (X and Y), two of these genes are necessary to represent the gender (genotype), so that XX represents a female and YX represents a male. The genotype is the combination of these genes, that is to say YX and XX, and being male or female is the phenotype, as shown in Figure 2.8.

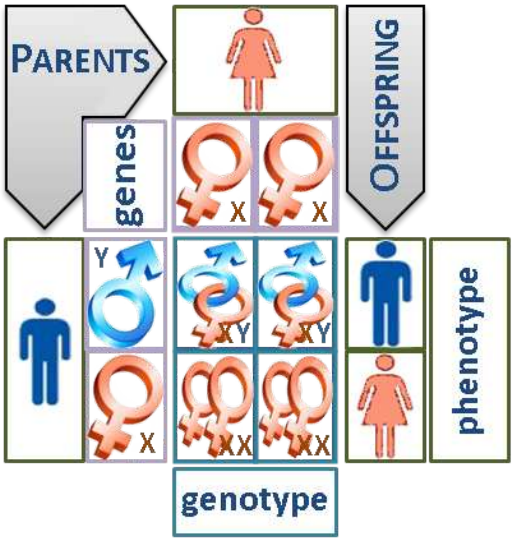


Figure 2.8: Examples of phenotype and genotype.

GAs differ from other methods in that they search among a population of solutions (often called a population based algorithm), and work with the encoded parameter set, which constitutes the genotype, rather than using the parameter values themselves.

The CGA was introduced by Holland (1975), using a binary model, and the Schema theorem was then developed to explain it. The next population of solutions of a predetermined size is then generated by applying a replacement strategy to the current population, here referred to as population selector. A replacement strategy selects solutions from a given population to take part in creating the next population. The members from the next population will be used as parents in producing a new population of solutions. The selection of the parents in generating a new solution is called parent selection or member selector. A crossover operator with a certain high probability is then applied to all solutions taken from the next population (which constitute the parent solutions) to produce the new solutions, which may be modified once more by application of a mutation operator with a low probability, finally constituting the current population. The process described represents one generation. These operations are repeated until one of the stopping conditions is reached, whereupon the new solutions are assessed for use in the final solutions, as demonstrated in Figure 2.9.

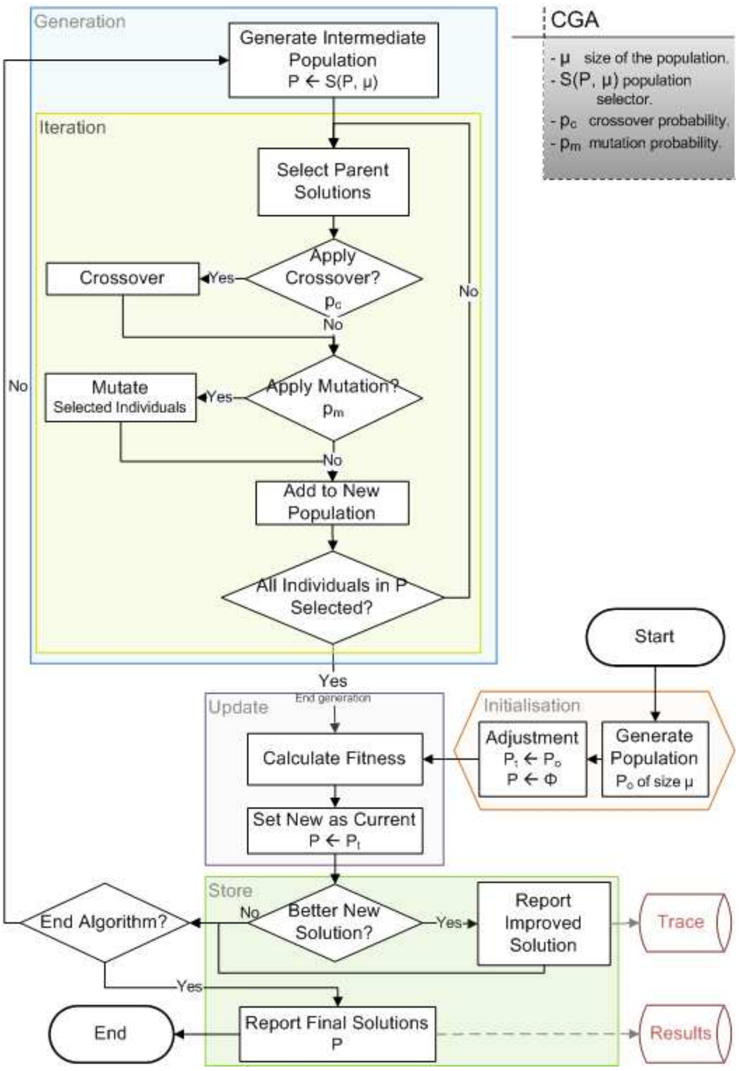


Figure 2.9: Flowchart of the Canonical Genetic Algorithm (CGA).

Evolutionary Strategies (ESs) are a sub-class of nature-inspired search methods belonging to the class of EAs and are based on the work of Rechenberg (1971). The canonical versions of the ESs are denoted by (μ, λ)-ES and $(\mu + \lambda)$-ES. Where μ is the number of parents and λ is the number of offspring. The (μ, λ)-ES is closer to the generational model used in CGA where offspring replace the parents and take part in the next generation, $\lambda \geq \mu$. In the $(\mu + \lambda)$-ES, μ parents produce λ offspring and the new population of μ parents are selected from the combined population of offspring and the parents.

2.7.1 Selection Approaches

The fitness function defines a scalar value for each individual used by the selection method to compare individuals. The loss of different fitness values in the population leads to a reduction in the selection pressure on individuals having the same fitness.

Some common selection approaches (selectors) are presented here and are used in the study conducted in this thesis.

Elitist Selection

The Elitist Selection (ES) selects the fittest μ population members from the current population.

Roulette Wheel Parent Selection

The Roulette Wheel Member Selection (RWMS) was originally used by Holland (1975), where the probability of a solution being selected is assigned to each solution in the population of λ solutions ($1 \leq i \leq \lambda$), which is proportional to their fitness (f_i), Equation 2.1. A section of a roulette wheel is assigned to each of the solutions based on their corresponding probability, where $s_0 = 0$, $s_i = \sum_{j=1}^{i} p_j$ and $[s_{i-1}, s_i)$ $\forall i \in [1 \ldots \lambda]$.

$$p_i = \frac{f_i}{\sum_{i=1}^{\lambda} f_i} \text{ for } i \in [1 \ldots \lambda] \tag{2.1}$$

A random number between zero (included) to one (excluded) is obtained, which is represented here as $rnd[0,1)$, so the section within which the random number falls, identifies the solution to select, e.g. for $s_{i-1} \leq rnd[0,1) < s_i$ solution i is selected.

One spin of the roulette ($rnd[0,1)$) is required per solution to be selected, whereas in the Stochastic Universal Sampling (SUS) with only one spin all of the required solutions are obtained. Given that the selections are independent of each other, in both the Tournament Member Selection (TMS) and the RWMS, Blickle and Thiele (1996) showed that there is a relatively high mean variation in the outcome of selecting the solutions in a population, which can be almost eliminated completely by using SUS (Baker (1987)).

Blickle and Thiele (1996) looked at different selection methods for discrete and continuous problems, and their selection variance (fitness before and after selection) concluding, based on the assumption that higher variance is advantageous, that Roulette Wheel Selection method is not appropriate as a selection scheme and the Exponential Ranking Selection is the best selection schema. They also pointed out that for a better understanding of the behaviour it is necessary to consider the operators used.

Tournament Selection

In the TMS, commonly called Tournament Selection, a few individuals from the population are chosen at random, where all members of the population have the same probability of being selected, Goldberg (1990); Goldberg and Deb (1991). The fittest is finally selected from among the chosen individuals.

Stochastic Universal Sampling

The SUS was introduced by Baker (1987) to reduce bias and inefficiency in the selection of individuals. The SUS exhibits less bias and spread (range of possible values for the number of an individual's offspring) than the Roulette Tournament Selection. The λ members of the population are mapped by sections, as in the Roulette Tournament Selection, in the range $[p_{i-1}, p_i)$ $\forall i \in [1 \dots \lambda]$, with $p_0 = 0$, based on their fitness f_i ($p_i = \frac{\sum_{j=1}^{i} f_j}{\sum_{j=1}^{\lambda} f_j}$). μ individuals are selected by obtaining an initial random number within $[0, \frac{1}{\mu})$, i.e. $r_0 = rnd[0, \frac{1}{\mu})$, and subsequent ones spread $\frac{1}{\mu}$ from the previous one. The solution i is selected once for each $p_{i-1} \leq \frac{j-1}{\mu} + r_0 < p_i$ $\forall j \in [1 \dots \mu]$.

Remainder Stochastic Sampling

Remainder Stochastic Sampling (RSS) is based upon the ratio between the fitness of a solution and the average population fitness. In Remainder Stochastic Sampling with Replacement (RSSR), the fractional relative fitness values are used to calculate weights in a roulette wheel selection which is then used to produce the remaining population.

In Remainder Stochastic Sampling Without Replacement (RSSWR), the fractional part of an individual is set to zero where it has been selected during the fractional phase of the selection. According to Goldberg (1989), RSSR has a greater probability of population diversity than the roulette wheel technique and provides zero bias (similarly to Stochastic Universal Modified Sampling (SUMS) and SUS).

Linear Ranking Selection

The Linear Ranking Selection (LRS) was first suggested by Baker (1989), Whitley (1989) and Bäck and Hoffmeister (1991). For a population ordered in ascending fitness, the probability assigned to an individual i for a population of size λ is provided by Equation 2.2 where p_1 is the probability of the worst individual being selected and p_λ is the probability of the best individual being selected.

$$p_i = \left(p_1 + (p_\lambda - p_1) * \frac{i-1}{\lambda - 1}\right) \;\forall\; i \in [1 \ldots \lambda],\; p_\lambda = (\frac{2}{\lambda} - p_1) \text{ and } 0 \leq p_1 \leq 1 \quad (2.2)$$

All individuals have a different rank so all receive a different probability, even if they are of the same fitness.

Exponential Ranking Selection

The Exponential Ranking Selection (ERS) differs from LRS only in that the assigned probabilities are exponentially weighted, Equation 2.3. Blickle and Thiele (1996) discussed the meaning and the influence of the parameter c in detail.

$$p_i = \frac{c^{\lambda - i}}{\sum_{j=1}^{\lambda} c^{\lambda - j}} \;\forall\; i \in [1 \ldots \lambda] \text{ and } 0 < c < 1 \quad (2.3)$$

Given that $\sum_{i=1}^{\lambda} p_i = 1$ the Equation 2.3 can be written as

$$p_i = \frac{c-1}{c^\lambda - 1} c^{\lambda - i} \text{ for } i \in [1 \ldots \lambda] \text{ and } 0 < c < 1 \quad (2.4)$$

2.7.2 Multi-objective Optimisation

Multi-objective optimisation has been applied in many areas where decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. The difficulty appears because in the case of a non-trivial Multi-Objective Optimisation Problem (MOOP), a single solution which simultaneously optimises each objective does not exist. Given this trade-off between two or more conflicting objectives, a solution is known as non-dominated, Pareto optimal (Pareto (1909), Tarascio (1968)), where there are objective(s) which cannot be improved without degrading one or many of the other objectives. The non-dominated solutions constitute what is known as the Pareto front. So it is necessary to find as many Pareto-optimal solutions (non-dominated solutions) as possible, Michalewicz and Fogel (2002) and Burke and Kendall (2005).

The MOOP has been solved as a single-objective optimisation problem where a single fitness function is used, i.e. a weighted sum of all the objectives (Prem Kumar and Bierlaire (2013), Dorndorf et al (2010), Hu and Di Paolo (2009), Pesch et al (2008), Dorndorf et al (2007a), Lim et al (2005) in the AGAP, and Ascó et al (2013), Ascó et al (2012), Ascó et al (2011) and Abdelghany et al (2006) in the ABSSAP). In a single-objective optimisation problem, the aim is to find one solution which optimises the combined fitness function. The aim is more than merely finding optimal solutions for each objective in MOOPs. The objective function in multi-objective problems constitutes a multi-dimensional space (the objective space), in addition to the decision variables space common to all optimisation problems. Although the search process of an algorithm takes place in the decision variable space, multi-objectives EAs use the objective space information in their search operators. In a multi-objective approach the aims are commonly convergence to the Pareto front and maintenance of a set of maximally-spread Pareto-optimal solutions. Most multi-objective optimisation algorithms use the idea of dominance in their search for solutions to reach and build the Pareto front.

The weighted sum approach is a commonly used classical multi-objective optimisation approach, which consists of converting the multi-objective problem into a single objective as the combined weighted sum of each objective. Its conceptual simplicity is complicated by the need to determine appropriate weights, the answer to which is not unique, as it depends on the importance given to each objective. This approach of combining multiple objectives into a single one is used in this study. Another classic approach is the ϵ-Constraint introduced in Haimes et al (1971) which keeps one objective whilst restricting the remaining objectives.

EAs combine methodologies which allow an efficient means of finding multiple Pareto-optimal solutions in a single run. Srinivas and Deb (1994) introduced the Nondominated Sorting Genetic Algorithm (NSGA) which was later modified in Kalyanmoy Deb and Meyarivan (2002) which introduced the Elitist Nondominated Sorting Genetic Algorithm II (NSGA-II), with the intention of overcoming some of the problems of the original NSGA. More recently Hanne (2009) in their GA, known as Primal-Dual Multiobjective Optimisation Algorithm (PDMOEA), considered the infeasible solutions and uses populations of variable size. Their results show that by extending the search to infeasible regions, the population may more easily reach new parts of the Pareto front.

The Strength Pareto Evolutionary Algorithm (SPEA) was introduced in Zitzler and Thiele (1999) and further improved in Zitzler et al (2001) (SPEA2) which incorporates a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method. The Improved Strength Pareto Evolutionary Algorithm 2 (ISPEA2) presented in Sheng et al (2012) is a more recent extension of the SPEA. Other multi-objective optimisation approaches are Vector Evaluated Genetic Algorithm (VEGA) (Schaffer (1984) and Schaffer (1985)), and Pareto Archived

Evolution Strategy (PAES) (Knowles and Corne (2003), Knowles and Corne (2000), Knowles and Corne (1999b) and Knowles and Corne (1999a)) which uses a simple (1+1) local search evolution strategy.

Coello et al (2007) provided a comprehensive survey of EA for multi-objective optimisation, and the survey in Castillo Tapia and Coello Coello (2007) concentrated on multi-objective optimisation in the areas of economics and finance.

2.7.3 Diversity

The population diversity of an EA is an important factor in avoiding premature convergence Michalewicz (1996). For many EAs a key obstacle to finding the global optimal solution is insufficient solution diversity, causing the algorithm to become trapped in a local optimum. The solution diversity can be influenced by the algorithm parameters such as population size, operators and diversity preservation approaches. One of the diversity preservation approaches corresponds to the selection methods, some of which are presented in Section 2.7.1. A survey of measures used to capture diversity in genetic programming was provided in Burke et al (2004).

Other approaches used to promote diversity are as follows:

Ageing: Syswerda (1990) uses ageing to help maintain diversity in the population. Arabas et al (1994) and Kubota and Fukuda (1997) used ageing approaches to resolve the premature convergence problem. Ghosh et al (1998) incorporated an ageing approach where new individuals begin with a zero age and at every iteration their age increases, which age is then used to calculate their effective fitness value, which changes dynamically.

Island model: this model considers the geographical distribution of individuals, Martin et al (1997). This model is used in parallel distributed GA, surveyed in Knysh and Kureichik (2010) and Cant-Paz (1998).

Crowding technique: this was introduced by DeJong (1975) as a technique for preserving population diversity and preventing premature convergence. Crowding is applied to generate the next generation in GAs. The next generation is composed of the individuals selected using the crowding technique among those in the current population and their offspring. Crowding is composed of two main stages: pairing and replacement. In the pairing stage, offspring individuals are paired with individuals in the current population according to a similarity metric. In the replacement stage, it is decided for each pair of individuals which of them will remain in the population. A review of crowding approaches for GAs can be found in Mengshoel and Goldberg (2008).

Genotype sharing: this considers behaviour and structural similarities and measures the inter-chromosomal Hamming distance, Deb and Goldberg (1989). NSGA uses genotype sharing whilst MOEGA usually prefers a phenotypic sharing since it seeks a global trade-off surface in the objective function space.

2.8 Conclusions

This chapter has introduced some of the different scheduling problems and research fields in airport resource scheduling and routing, provided a detailed overview of the different approaches currently used, and gives a brief introduction to a number of related fields. The area covered in this chapter helps the reader to understand the context of the work presented in this thesis, the motivation, scope, and value of its contribution.

Chapter 3

Problem Description

This chapter introduces the Airport Baggage Sorting Station Assignment Problem (ABSSAP), defines its scope, and introduces a mathematical model to represent it which is used in subsequent chapters. References to the relevant literature are included for completeness. The final section of this chapter provides some conclusions.

3.1 Overview

The transportation of passengers in an airport begins with the arrival of passengers at the terminal from which their flight will depart. They then proceed to the check-in desks to which their flights have been assigned, where they leave their baggage for processing. The baggage at this point enters the baggage system which delivers it to the ground side for processing at the corresponding baggage sorting stations (BSSs), where the baggage handling workforce sorts and places the baggage onto trailers, in readiness for transportation by cart to the airside next to the aircraft. Here the ground workforce places it in the cargo hold of the aircraft, in readiness for travel to their destination.

The passengers then proceed to passport control prior to entering the international area of the airport, and make their way to the gate assigned to their flight departure, where the aircraft is being made ready for take-off. Some operations need to be completed before the flight commences its departure process, such as boarding of the passengers, cleaning the aircraft, refuelling and loading baggage and catering supplies. The passengers then board the flight and the aircraft captain requests permission to start the departure process which commences when the aircraft leaves the gate and makes its way to the departure queue system in readiness for take-off. The flight next enters the national air control space where it is guided by the en-route services until

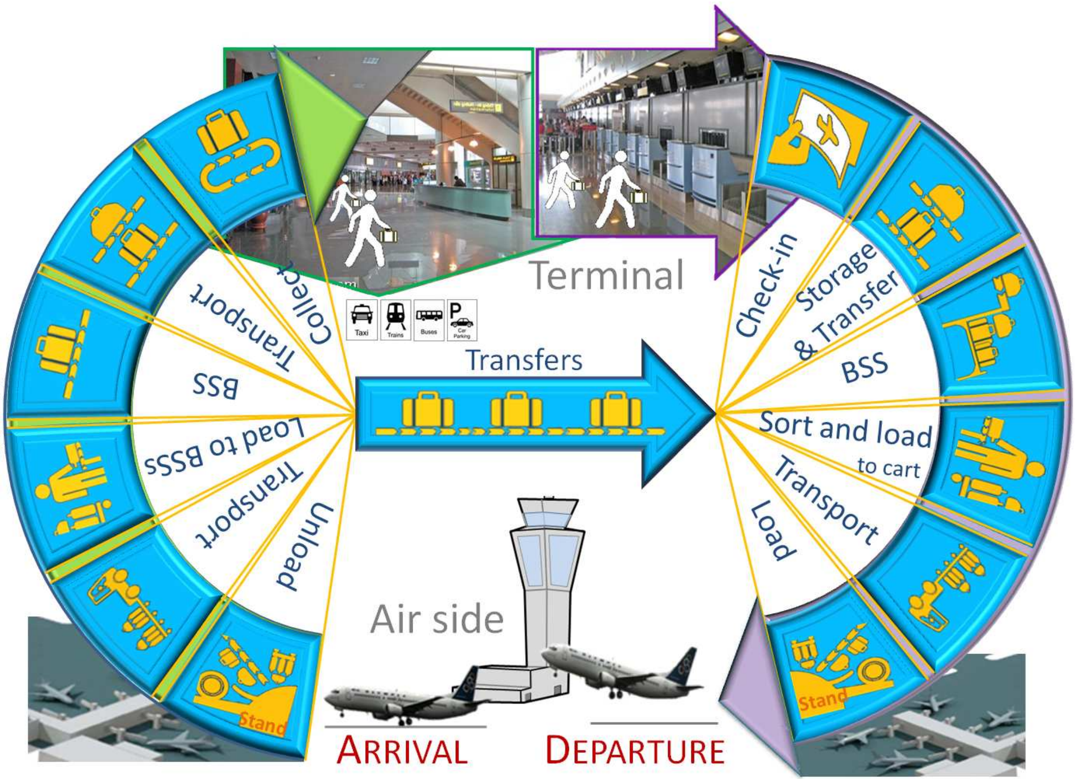


Figure 3.1: General view of the overall process in an airport.

the flight reaches the airspace of the destination airport, where the airport tower takes control and guides the flight to land, Air Traffic Control (ATC). Once the flight has landed, control is passed to the ground management team, which will guide the flight through the taxi ways to the arrival gate assigned to that flight, where passengers disembark, those leaving the airport make their way to the baggage collection point, the others continuing their journey by means of the gate assigned to their next flight, Figure 3.1. The aircraft may then be moved to a parking stand, awaiting the time to be moved to the assigned departure gate at which the process commenced.

The process is reversed on arrival at the destination, with passengers disembarking. The passengers' baggage is removed from the cargo area by the ground workforce and placed onto baggage trolleys ready for transportation by carts to a place next to the baggage sorting stations, where the handling force transfers it from the trolleys or containers onto the baggage sorting stations assigned. At this point the baggage enters the baggage system which delivers it to the carousel assigned to their flight, ready for collection by the corresponding owners, who will either leave the airport, or transfer to another flight and continue their journey. The baggage of these passengers is delivered to the BSSs assigned to their next flight, Figure 3.2. The sorting station assigned to a flight arrival normally corresponds to a carousel sorting station, which is the same as that used to process the baggage belonging to passengers leaving the airport, and only transfer baggage re-enters the baggage system for delivery to the sorting stations assigned to their departure next on the itinerary of their owners.

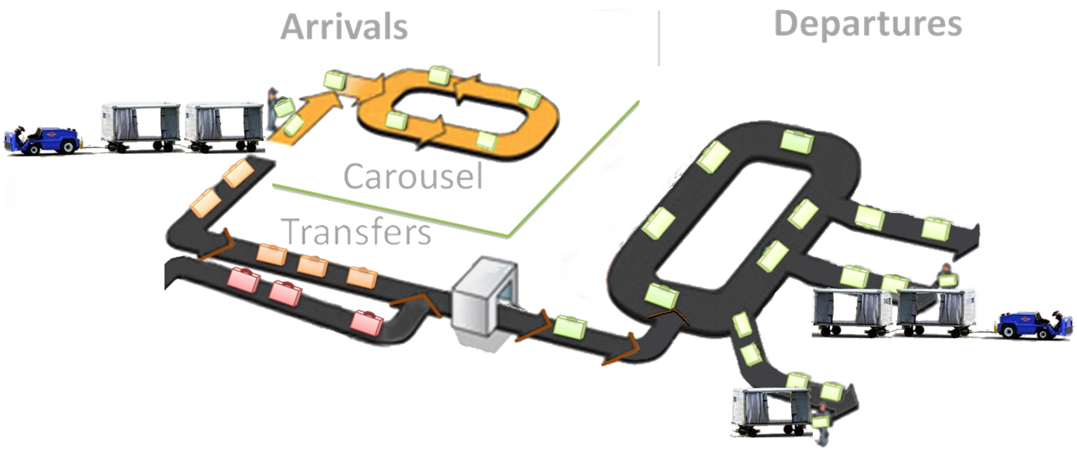


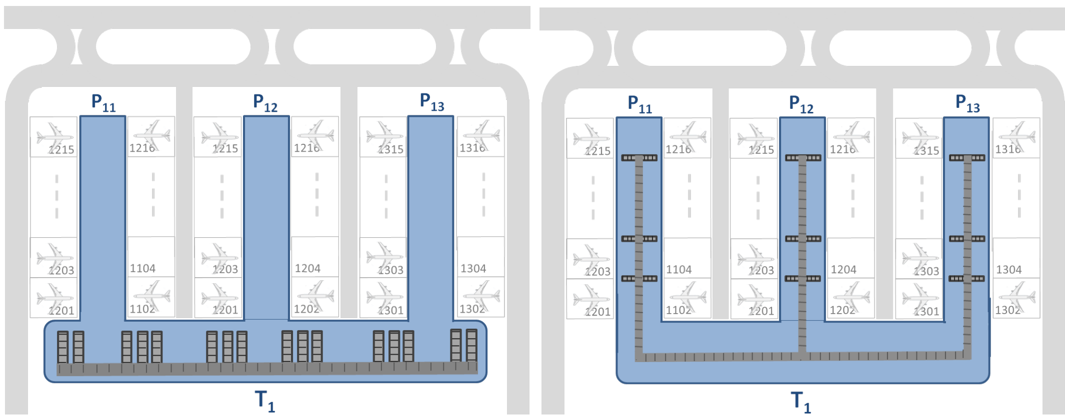
Figure 3.2: Arrival/departure baggage system.

3.2 Airport Layout

Airport geometry plays an important role in the assignment of resources and the safety of airport operations. An overview of the airport configurations and technology relevant to the transportation of passengers and baggage was presented by Pitt et al (2002) who concentrated on airport configurations and the availability of different types of resources. Rijsenbrij and Ottjes (2007) provided an overview of different elements of the baggage handling system and gave a description of the way in which baggage is currently handled, identifying potential areas of improvement.

An airport pier is that section of an airport terminal where the gates and associated stands are distributed along the sides of the building; outside in the case of stands and inside the building in the case of the gates through which passengers board the flights. Whereas the stands are allocated at the side of a pier, the baggage sorting station may be placed in different positions in respect to the piers.

Some examples of topologies based on the position of the BSSs are represented in Figure 3.3 for a terminal with three piers, where the sorting stations are shown in the diagrams as a small set of rectangles with T_i representing the terminal i and P_{ij} representing the pier j in terminal i.



a Perpendicular BSSs to piers

b Parallel BSSs to piers

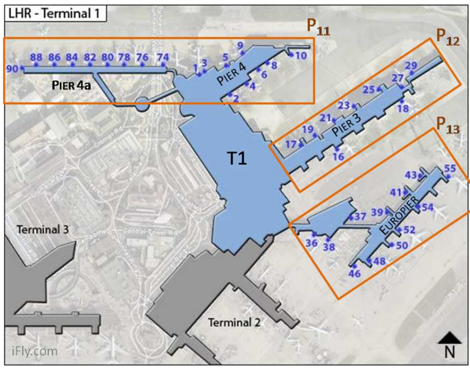
Figure 3.3: Example of airport topologies with one terminal.

Rather than being specific to the example layouts described above, the models utilised in this thesis are appropriate for any airport where there are groupings of aircraft/gates which enforce a baggage sorting station group preference (as when aircraft are at piers) and where there is a distance or cost metric for the assignment of a baggage sorting station to a flight. For example, at some airports, the sorting stations may be between the gates, in which case the distance/cost preference when assigning flight-sorting station pairs may be much stronger, whereas the group/pier preference may not be so strong.

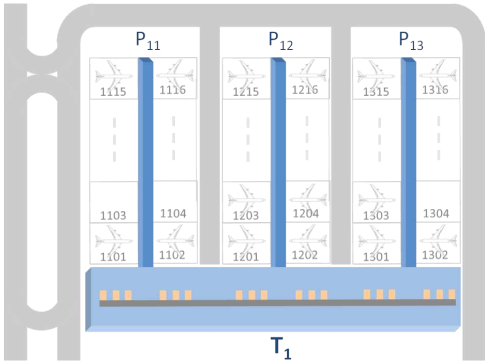
The topologies considered in this thesis correspond to those shown in Figures 3.4.

3.3 Airport Baggage Sorting Station Assignment Model

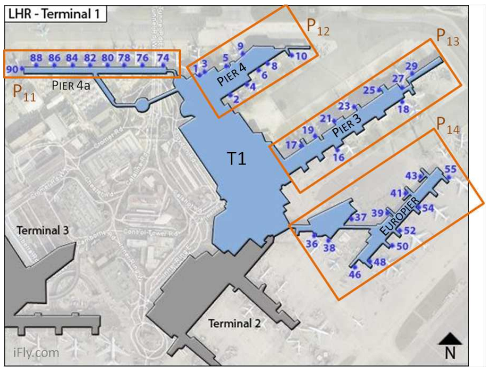
The checked-in baggage at a passenger airport first enters the baggage system where it is processed and delivered to the ground side, an overview of the process being provided in Figure 3.5. The baggage is then transported by conveyor belt to the baggage system's security hall where it is individually scanned. Most baggage will continue straight on, but if at the scanning stage suspicions were aroused concerning the baggage then it is diverted to the security checking area where it will be further checked by one of the security personal and, if clear, will rejoin the normal journey with the rest of the baggage. The baggage will then continue (on conveyor belts) to the baggage hall and be transported to the baggage sorting station assigned to it, where the baggage accumulates ready for the workforce to sort and place on trolleys or into special containers which go directly into the aircraft, ready for transportation by



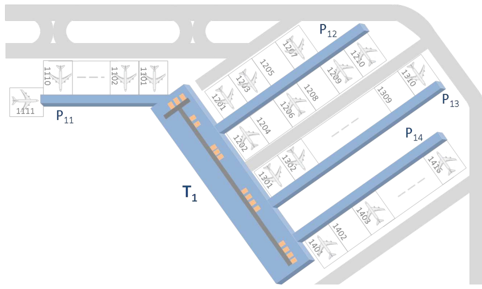
a London Heathrow airport Terminal 1 with 3-pier



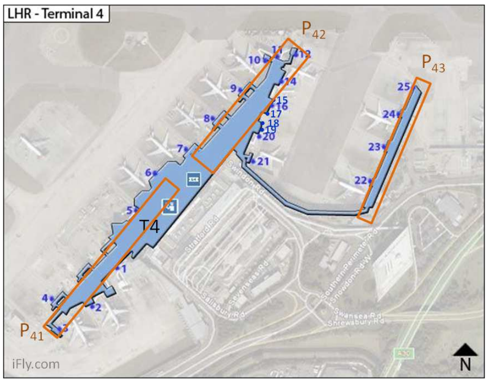
b 3-pier topology.



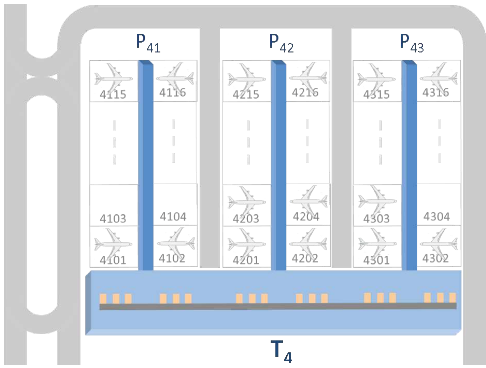
c London Heathrow airport Terminal 1 with 4-pier



d 4-pier topology



e London Heathrow airport Terminal 4 with 3-pier



f 3-pier topology

Figure 3.4: Representations of the terminal topology used.

cart and placed next to the aircraft on the airside where the baggage is loaded into the aircraft hold by the ground workforce, ready to travel to its destination. Containers are used to transport the baggage on wide fuselage aircraft, for long distance flights, which are directly placed into the hold of the aircraft. Trolleys are used in the narrow fuselage aircraft and the baggage is individually loaded into the hold of the aircraft by the handling workforce, who use conveyor belts to lift the baggage from the trolleys to the level of the aircraft's hold. Johnstone et al (2010) looked at the routing of the baggage within the baggage system with the aim to provide additional insight into how agents can learn to route in a baggage handling system, which experiments showed that the learning method performs better than the search method.

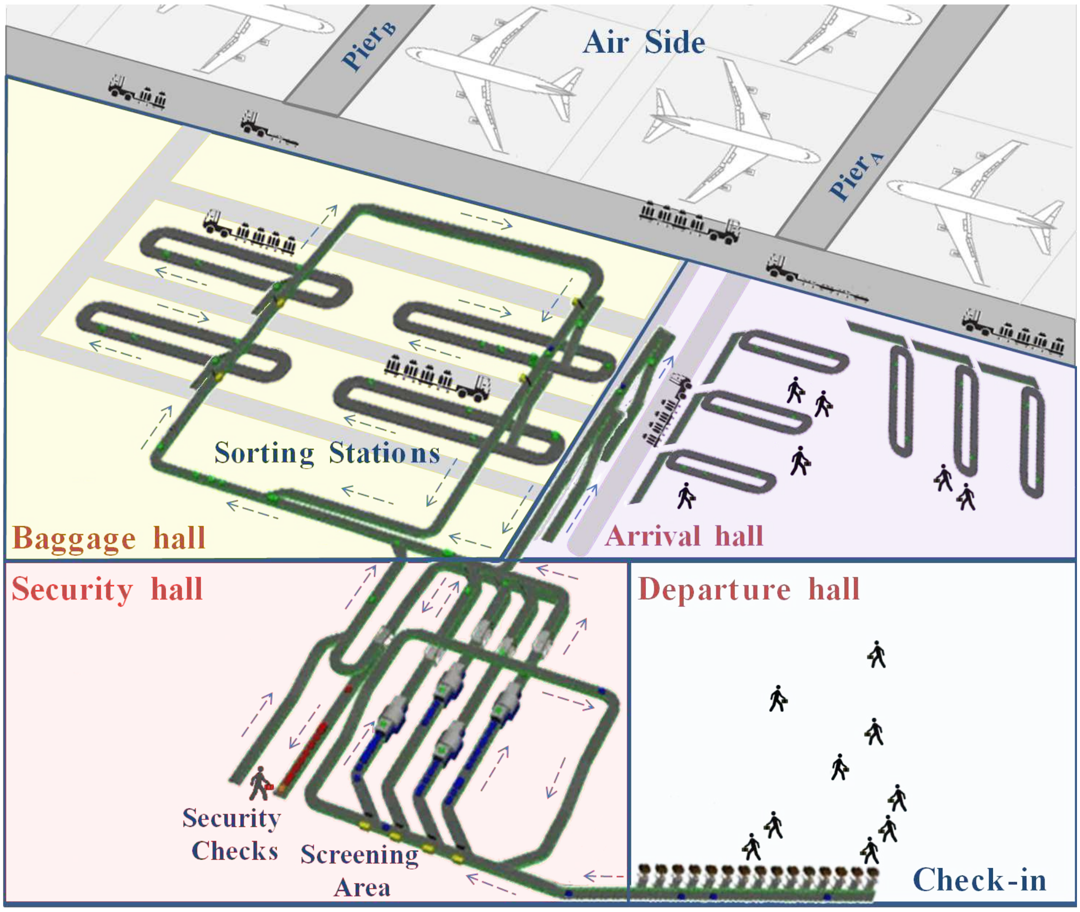


Figure 3.5: Baggage System.

On reaching the destination airport the process is reversed, so that the ground workforce removes the baggage from the cargo area of the aircraft and place it directly on baggage carts (open trolleys, onto which baggage is separately loaded and protected with a canvas cover) or load in baggage containers onto dollies (trailers, on which baggage containers are loaded) ready to be transported by cart to the baggage sorting stations assigned. Here the handling force transfers it from the trolleys or containers onto the baggage sorting stations for transportation to the ground side of the arrival hall. The baggage then enters the baggage system which delivers the baggage to the carousel to which the flight is assigned, in readiness for collection by the corresponding owner, and will then leave the airport. In the case of transfer passengers, their baggage is delivered to the baggage sorting station assigned to their next flight, Figure 3.6. The sorting station used by a flight arrival is normally directly linked to the carousel assigned to the passenger flight for the given destination, and only the transfer baggage re-enter the baggage system for delivery to the sorting station assigned to its next departure, as shown in the 'Arrival hall' in Figure 3.5. The transfer baggage does not usually need to be directed through the security hall, given that it should already have been checked at the original airport.

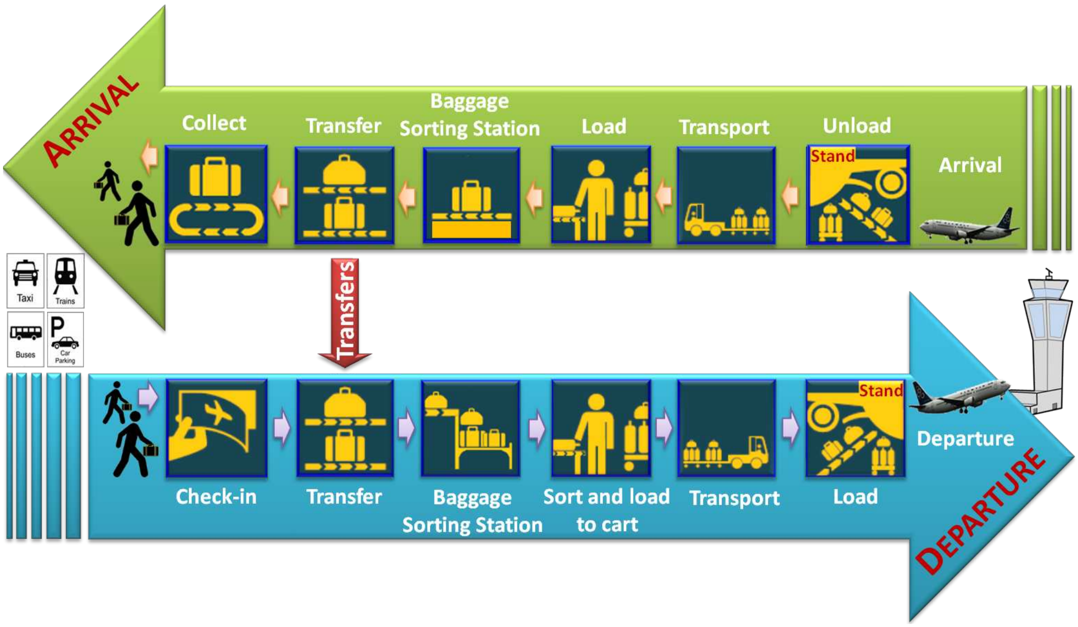


Figure 3.6: General view of the baggage process in an airport.

The different parts of the overall configuration tend to be distributed on several levels, whereas the check-in (departures) is normally located on the upper level of the airport. Passengers on flights arriving at the airport (arrivals) collect their baggage from the carousels normally placed on the lower level. The baggage system is placed at a lower level than that of the arrivals, and may also be underground in some airports.

Where airports have several terminals it would be unrealistic to assume that baggage from a flight at a terminal stand is serviced by a baggage sorting station in another terminal (e.g. passengers usually go through security and board flights from the terminal at which they checked their baggage in). This may not be the case for transfers where passengers and their baggage arrive at an airport terminal and perhaps leave the airport by another flight departing from a different terminal.

The ABSSAP involves the assignment of BSSs to flights already scheduled. These previously scheduled flights have already been assigned to stands, which are the areas allocated for parking aircraft, and the stand is required from the time of arrival to the time of departure, whereas gates are the areas in a terminal where passengers access the aircraft. In the Airport Gate Assignment Problem (AGAP) practitioners refer to the assignment of flights to gates, which is equivalent to the stands associated with these gates, normally located at a pier next to the gate. Examples of flight assignments to gates are presented in Figure 3.7 which uses a type of GANTT chart, where the vertical axis represents the stands and the horizontal axis shows when the stands are in use. For example, for 219 flights the top row shows five flights assigned to stand 1101. Here the first digit of the full stand number refers to the terminal number (1), the second digit is the pier number (1), and the last two digits are the stand identification. A pier is the area around which stands are grouped. This problem was originally studied by Abdelghany et al (2006), which used an activity selection algorithm and considered a sufficient number of BSSs for assignation to all flights, whereas Ascó et al (2011) studied the same problem and assessed different scenarios with different topologies, preferences and numbers of BSSs, examining the trade-off between objectives when using different constructive algorithms.

There are also terminals at which the baggage system has some general storage capacity, as shown in Figure 2.6, which reduces the time BSSs are required by flights, which in turn depends on the system capacity and overall load of this general storage at the time of use, which Frey et al (2010) took into account in their model.

Rijsenbrij and Ottjes (2007) provided an analysis of different elements of baggage handling, whereas Pitt et al (2002) presented a broader view of baggage systems and technologies.

In summary, airport baggage processing mainly concerns the baggage taken from both arrival and departure flights. The baggage arrival is processed based on type, while the baggage relating to transfers is assigned and processed by specialised baggage sorting stations (such as laterals) which usually take the baggage into the baggage system at a point beyond the security area, since their clearance should have taken place in the airport of origin. These BSSs may be assigned to several flights simultaneously if the sorting station capacity limit is maintained. The baggage of those passenger arrivals leaving the airport is assigned to a carousel BSS, but the same BSS may be assigned to baggage from different overlapping flights. Baggage associated with departing flights presents multiple and conflicting requirements which increase the difficulty and interest of the problem, as described in the following sections and

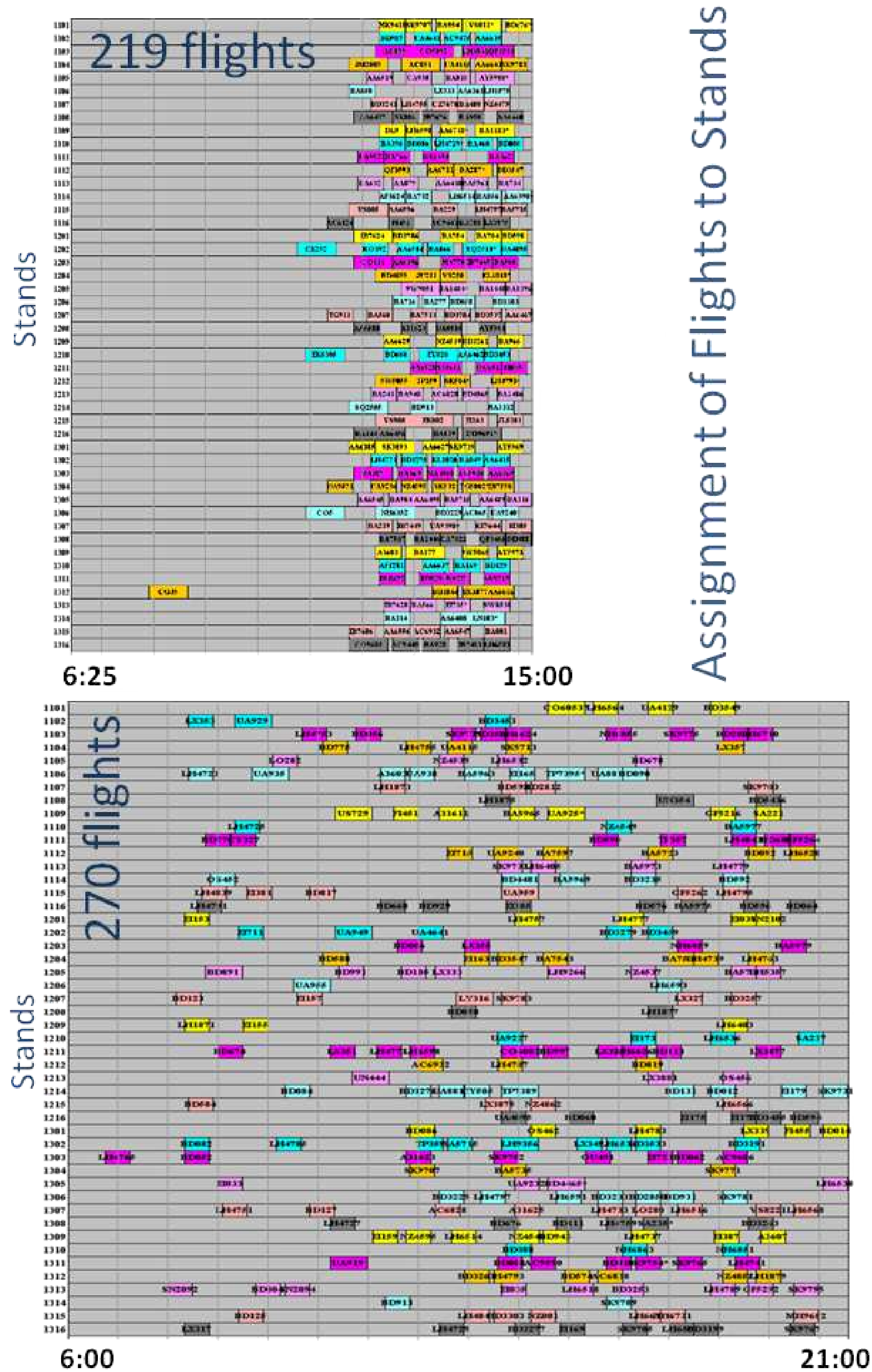


Figure 3.7: A random assignment of flights to stands for the data sets taken from the British Airports Authority (BAA)'s website.

studied in subsequent chapters.

3.3.1 Model

The problem under consideration is a multi-constraint and multi-objective one, which may be summarised as the assignment of available Baggage Sorting Stations (BSSs) to previously scheduled flights. As mentioned before, an airport pier is the protruding section of a terminal building at which aircraft park, so that passengers may embark and disembark. Baggage sorting stations are normally associated with piers according to the airport topology.

A Base Service Period is associated with each flight departure, during which the baggage for that flight is accumulated at the assigned baggage sorting station and normally loaded onto baggage carts for transfer to the flight. This service period may (optionally) be extended by applying an extra time (the buffer time), since it is preferable to have a gap between the servicing of consecutive flights by the same baggage sorting station.

The problem is presented as an Integer Linear Programming (ILP), where a flight j is composed of P_j activities each of which must be serviced by a different BSS. The objective here is to find appropriate values for the y_{ijp} Boolean variables, which take a value of 1 if activity p of flight j is assigned to baggage sorting station i or zero otherwise. The target service time represents the time in which a BSS is expected to be assigned to a flight. The reduction in service time has a detrimental effects on the capability of a solution to absorb real-life delays. Therefore the amount of reduction in the target service time for the assignment of an activity p for flight j is represented by r_{jp}, which is calculated in seconds (as an integer). The constants and variables of the model are shown in Tables 3.1 and 3.2. The full model is presented in the following sections.

3.3.2 Input Data, Constants and Decision Variables

The various constants used in the model are summarised in Table 3.1. A flight may require more than one BSS, since baggage on large aircraft may be split between several BSSs, e.g. one BSS is responsible for the baggage-claim, the second for baggage arrivals to be transferred and also one BSS may be assigned to the baggage belonging to each class of passenger on a departing flight, where P_j $(0 < P_j \leq N)$ denotes the total number of sorting stations required by flight j.

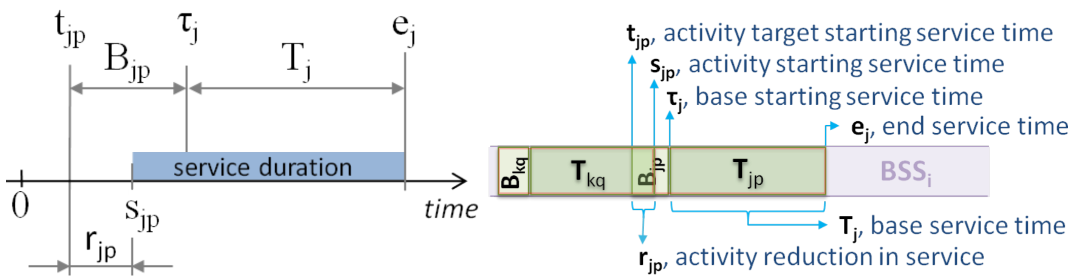
Additionally, buffer time is applied between two consecutive flights on the same

Name	Description
N	The total number of BSSs under consideration.
M	The total number of flights to which sorting stations should be allocated.
P_j	The total number of activities to be serviced by baggage service stations for a given flight j, which also equates to the total number of sorting stations required to fully service flight j, $P_j > 0$.
T_j	The base service duration for flight j.
B_{jp}	The desired buffer time for flight j and activity p ($p \in [1 \dots P_j]$), introduced in Section 2.4.
e_j	The end service time for flight j.
τ_j	The base starting service time for flight j, $\tau_j = e_j - T_j$.
t_{jp}	The target starting service time for flight j and activity p, $t_{jp} = \tau_j - B_{jp}$, assuming the full buffer time is available. Target service duration is the difference between the end service time and the target starting service time, $e_{jp} - t_{jp}$.
C_{jp}	A flight specific constant representing the amount of baggage to be processed for flight j and its activity p. This determines the difficulty involved in allocating the flight to a more distant sorting station. For example, this may represent the number of delivery trips required to move the baggage from the sorting station to the aircraft. In the absence of baggage load figures, it is used $C_{jp} = 1$ for all activities and flights.
d_{ij}	The distance between baggage sorting station i and flight j.
d'_{ik}	The distance between the baggage sorting stations i and k.

Table 3.1: List of the constants and input values for the ABSSAPs model.

baggage sorting station in order to absorb small disturbances in the real system behaviour. Buffer times are a common means of increasing robustness in order to absorb small delays, as studied by Nikulin (2006) and Mulvey et al (1995). Buffer times were used in the scheduling of baggage sorting stations by Abdelghany et al (2006), and Wu and Caves (2004) used them in the optimisation of the aircraft turnaround process. The AGAP has some characteristics similar to the baggage sorting station assignment problem, and buffer times have been commonly considered for the AGAP by Hassounah and Steuart (1993), Yan and Chang (1997), Bolat (2000), Yan et al (2002) and Wu and Caves (2004). Yan and Huo (2001) provided a sensitivity analysis for the AGAP buffer time, noting that the length of buffer time significantly influences the gate assignment process, so a reasonable minimum value should be used. Yan et al (2002) looked at the suitability of Flexible Buffer Times (FBT) where, given low delays, short FBTs usually improve real-time objectives, such as the reassigning of an incoming aircraft at a minimum distance. Wei and Liu (2009) showed the feasibility and effectiveness of using a fuzzy model in conjunction with fixed buffer times for the AGAPs. Ascó et al (2011) used buffer times to cope with small perturbations in the ABSSAP, and several constructive algorithms were also studied. Wu and Caves (2000) and Wu and Caves (2004) showed the significance of a correct use of scheduled buffer time in maintaining schedule punctuality and performance by balancing the trade-offs between schedule punctuality and aircraft utilisation. The position of buffer time for a given flight service time may have an impact on the problem, as is the case when buffer time is not the same for all flights, a point covered in Chapter 6.

The flight activity service time is the duration from activity starting time (s_{jp}) to end service time (e_j), and the target service time is the duration from activity target starting service time (t_{jp}) to e_j. The relationship between the timing values is illustrated in Figures 3.8.



a Representation of the different times. b Flights times.

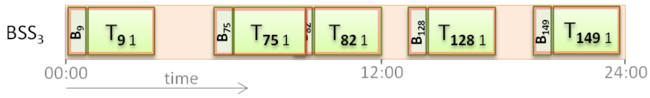
Figure 3.8: View of the time taken to service a flight.

The decision variables which are used in this model are presented in Table 3.2. The solution algorithms will attempt to find values of y_{ijp} and r_{jp} such that the constraints in Section 3.3.3 are met, and the relevant objectives (e.g. maximising assignments and minimising reduction in service times) in Section 3.3.4 are improved. An example

Name	Description
y_{ijp}	Specifies the assignment of flights to sorting stations. $y_{ijp} = 1$ if baggage sorting station $i \in [1 \dots N]$ is allocated to flight $j \in [1 \dots M]$ for $p \in [1 \dots P_j]$, and 0 otherwise.
r_{jp}	Specifies the necessary reduction in service time for activity $p \in [1 \dots P_j]$ of flight $j \in [1 \dots M]$, given the service starting time allocated, s_{jp}.
s_{jp}	The service starting time allocated to activity $p \in [1 \dots P_j]$ of flight $j \in [1 \dots M]$ and given that a sorting station can only service one flight at a time. s_{jp} can be determined from r_{jp} since $s_{jp} = t_j - r_{jp}$.

Table 3.2: List of the decision variables used in this ABSSAPs model.

of assignments for the baggage sorting station 3 and the reduction in service time for



$y_{3j} = 1 \ \forall \ j \in (9, 75, 82, 128, 149)$

$r_j = 0 \ \forall \ j \in (9, 75, 128, 149)$ and $r_{82} = 15$min

$y_{3k} = 0 \ \forall \ k \in (9, 75, 82, 128, 149)$

$y_{iq} = 0 \ \forall \ i \in (1, 2, 4, \ldots, N)$ and $q \in (9, 75, 82, 128, 149)$

Figure 3.9: Example of assignments of flights with $P_j = 1 \forall j \in [1 \ldots M]$ to baggage sorting station 3.

those assigned flights is shown in Figure 3.9.

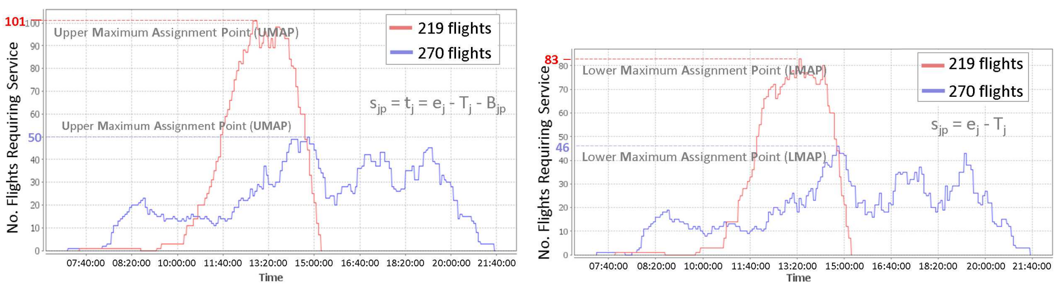
Flights which cannot be assigned to any BSS are assigned to the dummy BSS, an approach widely used in the Airport Gate Assignment Problem (AGAP), as shown in Yan and Huo (2001), Drexla and Nikulina (2008) and Tang et al (2009), as well as other areas of optimisation such as vehicle dispatching, Ichoua et al (2006). Furthermore the dummy BSS may be assigned to overlapping flight activities, where $i = 0$ is used to represent the dummy BSS.

The following two points were defined and will be observed to be useful later when interpreting the results for both the ABSSAP and the AGAP.

The **Lower Maximum Assignment Point (LMAP)** is the number of resources required to service a certain number of activities when the service starting time (s_{jp}) coincides with the target starting service time (t_{jp}).

The **Upper Maximum Assignment Point (UMAP)** is the number of resources required to service those activities when the service starting time (s_{jp}) coincides with the base starting service time (τ_j).

Two examples of both points are shown in Figure 3.10.



a Representation without reduction in service time, UMAP.

b Representation with maximum reduction in service time, LMAP.

Figure 3.10: Flights service distribution for the data sets obtained from the BAA's website.

3.3.3 Constraints

Various constraints apply to the assignment of BSSs and may be summarised as follows.

Assignment Limits

Each flight must be assigned to at most P_j BSSs, as expressed by Inequality (3.1). In normal operations, each flight should be assigned to exactly P_j sorting stations, in which case Inequality 3.1 would become an equality. However, in extreme situations, where there are insufficient sorting stations (as discussed in this thesis) there may be no feasible assignment of flights to sorting stations such that all flights can be allocated, hence the inequality. However, when assignments to the dummy BSS are also included the inequality become an equality: $\sum_{i=0}^{N} y_{ijp} = P_j$.

$$\sum_{i=1}^{N} y_{ijp} \leq P_j \ \forall j \in [1 \ldots M] \text{ and } \forall p \in [1 \ldots P_j] \tag{3.1}$$

Complete Assignment

When $P_j > 1$ the activities corresponding to the same flight must either all be assigned or none should be assigned, as expressed by Formula 3.2.

$$\sum_{i=1}^{N} y_{ijp} = \sum_{i=1}^{N} y_{ij(p+1)} \ \forall j \in [1 \ldots M] \text{ and } \forall p \in [1 \ldots P_j - 1] \tag{3.2}$$

Reduction in Service

BSSs can only be used by one flight at a time, so it may be necessary to reduce the flight service time (usually by reducing the buffer times between flights, shown in Figures 3.8b and 3.9) in order to assign flights to the same sorting station. The principal objective is usually to maximise assignment of BSSs to flights, as expressed by Formula 3.7.

For any pair of different flights where service times overlap, if the overlap in service times is greater than the maximum reduction allowed (B_{lq} for activity q of flight l), then both flight activities cannot be assigned to the same BSS. Thus, Inequality 3.3 applies to any such pair of flights, j and l ($j \neq l$), where $t_{lq} < e_j \leq e_l$ and $(e_j - t_{lq}) > B_{lq}$.

$$y_{ijp} + y_{ilq} \leq 1 \tag{3.3}$$

They may otherwise be assigned to the same BSS as long as the service duration of flight l is sufficiently reduced to remove the overlap. Inequality 3.4 applies to any such pair of flights, j and l ($j \neq l$), and their activities p and q respectively, where $t_{lq} < e_j \leq e_l$ and $(e_j - t_{lq}) \leq B_{lq}$. One objective is to minimise these service time reductions, as discussed later.

$$r_{lq} \geq (y_{ijp} + y_{ilq} - 1) * (e_j - t_{lq}) \tag{3.4}$$

Limit of Service Reduction

The reduction in service duration may not exceed a limit, as expressed by Inequality 3.5.

$$0 \leq r_{jp} \leq B_{jp} \quad \forall j \in [1 \ldots M] \text{ and } \forall p \in [1 \ldots P_j] \tag{3.5}$$

3.3.4 Objectives

A number of objectives concerning this problem need consideration, and there is a trade-off to be made amongst them. The various objectives considered in this section are:

Maximise Assignment of Baggage Sorting Stations

The first and most important objective is to maximise the number of flights assigned to BSSs, as expressed by any of the Formulas 3.6, 3.7 and 3.8. In airport practice, this objective would probably be a hard constraint at most times, since all flights would normally have to be serviced, but we wish to observe the performance of the algorithms when there are too few sorting stations, as well as when there are sufficient or plentiful.

$$max \sum_{i=1}^{N} \sum_{j=1}^{M} \left(\frac{\sum_{p=1}^{P_j} y_{ijp}}{P_j} \right) \tag{3.6}$$

The objective representation in Formula 3.6 refers to maximisation of the quantity of assigned flights, which gives preference to flights having a lower number of BSSs required, i.e. those with a lower P_j value. To increase the importance of obtaining fully serviced flights, the objective may be represented as shown in Formula 3.7. In this case preference to assign activities to BSSs is given to flights with a higher P_j. Finally, assignment to a flight could be spread wider between different values of P_j as expressed by Formula 3.8, where more weight is given to flights with greater P_j but not so great as in Formula 3.7.

$$\max \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{p=1}^{P_j} y_{ijp} \tag{3.7}$$

$$\max \sum_{i=1}^{N} \sum_{j=1}^{M} \left(\left(1 + \alpha * \left(0.5 - \frac{1}{P_j + 1} \right) \right) * \frac{\sum_{p=1}^{P_j} y_{ijp}}{P_j} \right) \tag{3.8}$$

When $P_j = 1 \; \forall j \in [1 \ldots M]$ both Formulas 3.7 and 3.8 are the same and Formula 3.8 should also be the same, irrespective of the value of α, giving a value for the constant within the brackets multiplied by α of 0.5. Furthermore, an $\alpha = 0$ changes the formula to the same as Formula 3.6, whereas for $P_j = P \; \forall j \in [1 \ldots M]$ and a value of $\alpha = 2 * (P + 1)$ makes it the same as Formula 3.7.

Robustness

Delays on the day of operation may render some assignments unfeasible which need to be re-assigned. It is therefore desirable to account for potential delays on the day of operation when generating the flight assignments to BSSs in the planning stage, such that the final flight assignments differ little or not at all from the original assignments on the day of operation. The degree to which this is achieved is an indication of the solution robustness, so a solution which requires less re-assignments is said to be more robust than those solutions requiring more re-assignments. Robustness is the ability of assignments to resist changes consequence of perturbations by reducing or removing the need to re-assign current assignments.

There are different ways of increasing robustness depending on the intended effect. One of the most simple and widely used methods is the introduction of a buffer time between assignments which allows absorption of small disturbances. Wu and Caves (2000) showed the significance of a proper use of schedule buffer times in maintaining schedule punctuality. Yan and Huo (2001) applied buffer times to the AGAPs and concluded that the length of buffer time significantly influences the gate assignment process. Thus a reasonable value should be used. The 'idle time' refers to the time between two consecutive assignments to the same BSS, from the end time of one activity to the base starting time of the following assignment, also called gap. Other approaches for improving the robustness make use of the distribution of 'idle time', and the reduction of the number of reassignments of the disrupted schedules. A selection of these is presented below and introduced in more detail in Chapter 6, along with a review of where these methods have been used in the past.

1. Minimise Reduction in Service: Given the detrimental effect that the reduction in service time has for the robustness of the assignment as against real-life delays, it is advisable to minimise the total reduction in service time, thus maximising total buffer time. This objective can be expressed by Formula (3.9) and it is further described in Section 6.3.1.

$$\min \sum_{j=1}^{M} \sum_{p=1}^{P_j} r_{jp} \tag{3.9}$$

2. Distribute Idle Time: Bolat (1999) proposed to distribute the 'idle time' uniformly between the gates. In the case of the Airport Baggage Sorting Stations Problem, this corresponds to the uniform distribution of 'idle time' between BSSs. This is described in greater detail in Section 6.3.2.

3. Reduce Reassignment on Disruption: The ability to reassign all flights directly affected by a disruption is desirable, without the need to reassign other flights also. The intention here is to generate schedules which take this objective into account, e.g. count the number of assignments between which a reassignment could be placed. This is also described in detail in Section 6.3.3.

4. Area of Reduction in Service: The flight density is defined as the number of flights requiring to be serviced each time. In order to account for the importance of the time wherein the reduction in service is performed, a greater penalisation of the reduction of the service time of those assignments with higher flight density is used. This may be achieved by a new approach which uses the area of flight density for the period of reduced service time. This is described further in Section 6.3.4.

5. Sub-Area of Reduction in Service: A new approach is proposed to also account for the service load at different times of the day in the robustness and uses the area between both flight densities, where no reduction in service is applied, and when base service duration is considered. This is described further in Section 6.3.5.

6. Unsupervised Estimated Stochastic Reduction in Service: Lim and Wang (2005) proposed that a new robustness strategy be used for the AGAP, whereby an unsupervised estimation function is applied. This estimates the mean probability of conflict between flights, thus taking account of the potential impact of future disruptions in the flight schedule. It assumes that the larger BSS gaps result in

a lesser probability of BSS conflicts. This is discussed further in Section 6.3.6.

7. Reduction in the Number of Conflicts: Yan and Tang (2007) used randomly generated delay scenarios in the 'Planning Stage' which represent potential disruptions in the schedule, and are used to obtain the anticipated semi-deviation risk measure (Ruszczynski and Shapiro (2003)) used to account for the robustness. Similarly, delay scenarios could be used to calculate the average number of conflicts a solution has when considering all those delay scenarios, which could be regarded as an indication of the performance of such a solution on the day of operation. The delay scenarios may be obtained in different ways, such as randomly as in Yan and Tang (2007), based on historic data, or generated by known distribution(s) drawn from information available at the time of producing the schedule. Collisions are defined as those flights which cannot be serviced by the BSS assigned, because the service time of a flight previously assigned to the same BSS has an overlapping base service duration. This is further discussed in Section 6.3.7.

8. Probability of Conflicts Based on the Gap: Previous approaches normally required a large number of perturbed data sets, which made their application very slow. Given that we are still interested in reducing the number of conflicts, though without the heavy cost in speed, it is deemed advisable to use the probability of a conflict in a given gap, which is applied to each flight. This can easily be obtained if the delay distribution is known. This is described further in Section 6.3.8.

Minimise Distance

The distance between the baggage sorting stations which are assigned to the flights and the flights to which they are assigned should be as short as possible. This objective aims to minimise the inconvenience, work and time involved in getting baggage to the aircraft, and could reflect preferences rather than distances. One way to handle this objective would be by expressing it as in Formula (3.10) where $\sum_{i=1}^{N}(y_{ijp} * d_{ij})$ represents the distance between flight j and its allocated BSS for activity p.

$$\min \sum_{j=1}^{M} \sum_{p=1}^{P_j} \left(C_{jp} * \sum_{i=1}^{N} (y_{ijp} * d_{ij}) \right) \tag{3.10}$$

For any stand in the perpendicular topology model, shown in Figure 3.3a, it would be preferable to assign luggage to the baggage sorting stations on the same side of the same pier. Alternatively, more distant sorting stations could be used, but these are less preferable. Whereas, in Figure 3.3b the baggage sorting stations are placed next to the stands within the pier. A 'cost' can be associated with a stand-sorting station pairing, and one aim is to reduce this cost by assigning as many flights as possible to their preferred sorting stations.

Consecutive Assignments

When a flight j requires servicing by more than one BSS, $P_j > 1$, then the BSSs assigned should be adjacent to one another. This helps the workforce assigned to a flight to perform their duties by reducing the need for them to move between BSSs. Formula 3.11 aims to reduce this distance.

$$\min \sum_{j=1}^{M} \left(\sum_{p=1}^{P_j} \sum_{q=p+1}^{P_j} \left(\sum_{i=1}^{N} \sum_{k=1}^{N} \left(y_{ijp} \cdot y_{kjq} \cdot d'_{ik} \right) \right) \right) \tag{3.11}$$

Where:
d_{ij} is the distance between sorting station i and flight j.
d'_{ik} is the distance between BSSs i and k, see Figure 3.11.

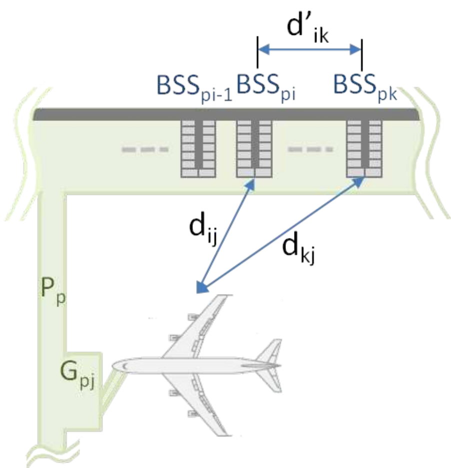


Figure 3.11: Flight distance to sorting stations.

The distance between a sorting station and itself is always zero, $d'_{ii} = 0$, whereas the distance between BSSs i and k is the same as the distance between BSSs k and i, symmetry; $d'_{ik} = d'_{ki}$.

Fair Workload

The quantity of baggage assigned to different BSSs should be comparable. This could be interpreted as minimisation of the total deviation in the actual usage of each BSS from the mean usage of all BSSs. This is expressed by Formula (3.12), where $e_j - s_{jp}$ represents the actual service duration for the flight j, which is the usage duration of the BSS. This objective aims to find a fairer assignment across BSSs, as discussed in Abdelghany et al (2006).

$$\min \sum_{i=1}^{N} \left| \underbrace{\sum_{j=1}^{M} \sum_{p=1}^{P_j} (y_{ijp} * (e_j - s_{jp}))}_{\text{workload for station i}} - \underbrace{\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{p=1}^{P_j} (y_{ijp} * (e_j - s_{jp}))}{N}}_{\text{mean workload over all stations}} \right| \tag{3.12}$$

If the quantity of baggage per flight activity which requires processing is different, then the quantity of baggage to be processed needs to be taken into account. b_j is the total number of passengers/baggage for flight j, and b_{jp} represents the quantity of baggage required in servicing activity p, $p \in [1 \dots P_j]$, where $\sum_{p=1}^{P_j} b_{jp} = b_j$. The average baggage per sorting station, average sorting station usage, is $\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{p=1}^{P_j} (y_{ijp} * b_{jp})}{N}$ which may be simplified as $\frac{\sum_{j=1}^{M} b_j}{N}$ when all of the activities are assigned to sorting stations.

Taking into account the different baggage requirements corresponding to each flight, the fairness objective corresponds to the minimisation of the usage deviation of all the sorting stations from the sorting station average usage, as represented by Equation 3.13.

$$\min \sum_{i=1}^{N} \left| \underbrace{\sum_{j=1}^{M} \sum_{p=1}^{P_j} (y_{ijp} * b_{jp})}_{\text{sorting station i usage}} - \underbrace{\frac{\sum_{j=1}^{N} \sum_{j=1}^{M} \sum_{p=1}^{P_j} (y_{ijp} * b_{jp})}{N}}_{\text{sorting station average usage}} \right| \tag{3.13}$$

The fairness objective encourages opening all of the available sorting stations, which may not be ideal as it may also relate to an increase in operational costs when only some of the sorting stations are required to fully service all flights.

Preferred Piers

Flights may have preferred piers, and this should be taken into account when assigning BSSs. It may be considered preferable to allocate sorting stations to each flight on the same pier. This preference should have already been taken into account when assigning a flight to a gate, so to reflect this preference, BSSs closer to the gate assigned, i.e. the pier, are preferable, Formula (3.10), and it is therefore unnecessary to consider this separately.

Flights to the Same Destination

It is preferable that flights from the same carrier to the same destination be assigned to the same BSS so that, for example, any delayed baggage could be transported on the next flight. However, flights would normally also be allocated to stands according to carrier, and potentially according to destination (or at least long-haul vs. short-haul). It is necessary to consider the time differential between assignments to the same destination, given that early departures to the same destination may be preferable, e.g. baggage unable to reach its flight on time is left at the BSS ready to be taken to the next flight to the same destination. Similarly flights to the same destination assigned to sorting stations closer to the current sorting station assigned are preferable to those further away, as the baggage should take less time and inconvenience to transport to the flight. There may also be restrictions as to which carriers may be considered appropriate for these selections, as own flights to the same destination would be regarded as preferable to those of other carriers.

The following constants are defined:

1. $D_{jk} = 1$ if the destination of flight j and k are the same, or $D_{jk} = \Gamma_d$ otherwise, $j,\ k \in [1 \dots M]$. Γ_d is a constant which is sufficiently large to discourage the assignment of sorting stations close to flights with different destinations.

2. $A_{jk} = 1$ if flights j and k belong to the same airline or $A_{jk} = \Gamma_a$ otherwise, $j,\ k \in [1 \dots M]$. Γ_a corresponds to a number sufficiently large to discourage the assignment of sorting stations close to flights from different airlines.

3. $\Psi(i, l, j, k)$ is the cost of assigning flight k to BSS l when flight j is assigned to sorting station i. If the flights are ordered by ascending departure time then $e_j <= e_k\ \forall j < k$ for $j, k \in [1 \dots M]$.

$$e_j \leq (e_k - T_k)\ \ \forall\ i, l \in [1 \dots N] \text{ and } j, k \in [1 \dots M] \tag{3.14}$$

$$\Psi_{jk} = \sum_{i=1}^{N-1} \sum_{l=1}^{N} (\Psi(i, l, j, k) * y_{ij} * y_{lk} * D_{jk} * A_{jk}) \text{ for } j \neq k \tag{3.15}$$

$$\Psi(i, i, j, k) \cong 0 \text{ for } D_{jk} = 1 \text{ and } A_{jk} = 1 \tag{3.16}$$

Equation 3.15 states that for two flights assigned from the same carrier to the same destination, when they comply with Inequality 3.14 (meaning that times do not overlap), there is a cost associated with assigning the flights to different sorting stations.

Formula 3.16 means that if there is a flight k from the same airline to the same destination as flight j, which is also assigned to the same sorting station, then there will still be a small cost, part of the time effect, as another flight may exist with the same attributes as flight j but which departs earlier than k, and which would be preferable. Furthermore, when this does not occur then $\Psi(i, i, j, k) = 0$ irrespective of the flight k. The objective is represented by Formula 3.17.

$$\min \sum_{j=1}^{M-1} \sum_{k=j+1}^{M} \Psi_{jk} \tag{3.17}$$

In certain cases it may also be appropriate for any delayed baggage which did not reach its flight in time to be transported on the next flight, even when this does not belong to the same carrier. To consider these cases, an extension of the above formulation requires the re-definition of A_{jk} in order to take account of cases where it is acceptable that baggage left from flight j is taken by flight k, even when flight k does not belong to the same airline ($A_{jk} \geq 1$). Where the value of 1 is given to a pair of flights j and k which belong to the same airline and $\Gamma_a > A_{jk} > 1$ when neither flight belongs to the same airline but there is a certain agreement between these airlines to cover such occurrences, otherwise $A_{jk} = \Gamma_a$. Similarly, $\Psi(i, l, j, k) \geq d'_{il}$ where flights assigned to closer sorting stations are preferred to those further apart, $d'_{il} = 0$ for $i = l$ and $d'_{il} > 0$ for $i \neq l$, Figure 3.11. The reason for the $\geq$ in the value of $\Psi(i, l, j, k) \geq d'_{il}$ corresponds to the time effect previously mentioned which may be taken into account by using the departure times of both flights j and k, e.g. $\Psi(i, l, j, k) = d'_{il} * \frac{e_k}{e_j}$ with $e_j \leq e_k$.

Other Objectives

Other objectives may also be considered, such as a reduction in the number of sorting stations open (to reduce the number of baggage handlers required). These are, however, in direct conflict with considerations of equity and reduction in service.

3.4 Conclusions

This chapter provided an extensive view of the ABSSAP and presents the model as an ILP which is used in the subsequent chapters for the ABSSAP. The time an aircraft expends parked at a gate has a considerable affect on the operations which take place up-stream in the overall operation of an airport. Delays in starting the departure sequencing may have important effects on the departure itself, which in turn may also require other aircraft to extend their holding time at the gates. This could well affect other flights arriving which also have those gates assigned to them. Any operation which may effect on the holding time at a gate does have the potential to disrupt the full airport operation. One of these operations correspond to the ABSSAP presented in this chapter.

Regarding the ABSSAP, the effect also extends to those cases where the baggage is lost, misplaced or does not reach the departing flight in time, but which may affect passengers satisfaction. These effects may be reduced by bearing these in mind when planning and assigning the resources involved in the baggage processing.

The ABSSAP is extensively examined in the following chapters which uses the mathematical representation presented in this chapter, commencing with a review of some constructive algorithms in Chapter 4. In Chapter 5 an new Evolutionary Algorithm (EA) is designed and compared with other algorithms and in Chapter 6 various different methods of accounting for robustness are presented and studied, where some of them are new approaches.

Chapter 4

Constructive Algorithms for the Airport Baggage Sorting Station Assignment Problem

This chapter introduces constructive algorithms for the Airport Baggage Sorting Station Assignment Problem (ABSSAP) presented in Chapter 3, introduces a new framework which facilitates the adaptation of these constructive algorithms to other problems, carries out a rigorous analysis of their parameter settings and compares their results with those provided by other approaches. References to the relevant literature are included for completeness. The final section of this chapter provides some conclusions.

4.1 Overview

The ABSSAP may be seen as a multi-objective resource constrained assignment problem, where the aim is to assign the limited baggage handling resources to the various flights which have to be serviced. Research into a similar problem was set out in Abdelghany et al (2006), but various questions were left unanswered. This chapter aims to answer these questions and to perform a rigorous analysis of the effects and benefits of the various different constructive algorithms applied to the problem, with a view to utilising these to provide initial solutions to other search methods. The intention is not to develop the 'perfect' algorithm for constructing a baggage sorting station assignment, but to understand the effects and trade-offs resulting from different choices.

The problem is represented as an Activity Selection Problem (ASP), which corresponds to the scheduling of resources amongst several competing activities within a given time frame. It is similar to the Airport Gate Assignment Problem (AGAP), but with some characteristics differing from the AGAP, some of which are:

1. The root of the problem in baggage sorting station assignment is that baggage sorting stations are required for a longer period than flights, so there can be no one-to-one correspondence between baggage sorting stations and stands, and ideal locations cannot be guaranteed. The service time is the period of time an activity is assigned to a resource, which was introduced in Section 3.3.2. Flights typically require the services provided at a gate from 25 to 45 minutes whereas for Baggage Sorting Stations (BSSs) this is usually between 1 to 3 hours.

2. A flight may require more than one BSS in order to be serviced, which is represented by the 'Assignment Limits' (Section 3.3.3) and 'Complete Assignment' (Section 3.3.3) constraints.

3. The objectives to be considered for a BSS also differ from those normally considered in the AGAP problem, as presented in Section 3.3.4.

As presented in Section 3.3.2, ideally there should also be a buffer time between sorting station usages, to reduce the risk of small perturbations affecting assignment and mixing of baggage between flights, but the contention for baggage sorting stations (BSSs) means that this sometimes has to be reduced or eliminated. One purpose of this chapter is to better understand the way in which the potential reduction in buffer times affects the various constructive algorithms.

There are a number of objectives to consider in the ABSSAP (for example, maximising the assignments, maximising available buffer times and assigning flights to the closest sorting stations, as discussed in Section 3.3.4) and these are in obvious conflict with each other. Any solution method needs to take this into account. In particular, different constructive algorithms will be observed to perform better for differing objectives. Hybridisation of the algorithms themselves or the appropriate utilisation or recombination of solutions from different algorithms may potentially lead to assignments which better reflect the overall objectives and this will be considered later, in Chapter 5.

This chapter is structured as follows: Firstly, the algorithms considered are described in Section 4.2, followed by a description of the problem data in Section 4.3. The results from the application of the algorithms herein to the problem are then provided and various observations are made and explanations given, in Section 4.4. Finally some conclusions are presented in Section 4.5.

4.2 Algorithms

The constructive algorithms considered here assign baggage sorting stations to flights one at a time until no further assignments are possible, as shown in Algorithm 1. Flights are first ordered according to one of the flight ordering methods under consideration, then a sorting station is selected for each in turn. The flight ordering and baggage sorting station assignments are considered below.

Algorithm 1: Constructive Algorithms Overview

```
Order flights for assignment (Section 4.2.1);
Determine the sets of BSSs to be considered (Section 4.2.2);
foreach flight do
    Select a set of feasible BSSs;
    repeat
        if the set of feasible BSSs is not empty then
            Select a BSS from the current set based on certain criteria (Baggage
            Sorting Station Selections);
            Assign flight to the BSS;
        end
    until flight was assigned OR there are no more sets to choose from;

    if flight was not assigned then
        Assign to the dummy;
    end
end
```

4.2.1 Flight Ordering Methods

The flight ordering method determines the order in which flights are selected for assignment. The different sorting approaches are considered below:

1. **Order by Starting Time (OST)**. This orders flights into ascending order by their target starting time, t_j values. When two flights have the same service target starting times, they are then sorted by their service end time, e_j. From the algorithm pseudo code presented therein, this appears to be that previously used in Abdelghany et al (2006).

2. **Order by Departure Time (ODT)**. This was previously used by Ding et al (2005) for the Airport Gate Assignment Problem (AGAP). This orders flights into ascending order according to their departure time, e_j. When two flights have the same service end times, they are sorted by their target starting time, t_j. Where service time reductions are not permitted, sorting by service end times provides maximum assignments when using the Last In First Out (LIFO)

baggage sorting station selection, and not constraining the set of sorting stations from which to select (see Section 4.2.2).

The ODT flight ordering method could potentially perform badly on the objectives other than the maximisation of assignments. In particular, the re-use of sorting stations can lead to an extremely inequitable assignment across BSSs. A new flight ordering method called Order by Departure Time Lookahead and Improvement (ODTLI) looks ahead when allocating sorting stations, to improve one of the other objectives while maintaining the maximal assignment of flights to BSSs. The lookahead and improvement is further described in Section 4.2.3.

3. **Order Between Times (OBT)**. OBT orders the flights based on a point positioned between the target starting time and the end time of each flight. The point is identified by the parameter $\alpha \in [0, 1]$, which is considered a constant, Equation 4.1.

$$t_j^B = t_j + \alpha * (e_j - t_j) \tag{4.1}$$

The ordering corresponds to OST when $\alpha = 0$ and ODT when $\alpha = 1$. The other values of α provide intermediate orderings to these two. These orderings are only useful where there are flights with two or more different service periods, otherwise they all correspond to the same single ordering, which also applies to ODT, ODTLI and OST.

4.2.2 Baggage Sorting Station Assignment

Once the flight to be assigned has been identified, the next stage is to determine to which sorting station it is to be assigned. The Baggage Sorting Station Assignment involves two stages. The first decision is upon which sets of BSSs to consider for assignment and in what order. In particular, whether only those stations for the same pier should be considered first, and whether service time reductions should be considered. The second decision involves the ranking of BSSs within each set, to enable selection of an individual BSS for assignment.

Baggage Sorting Station Assignment Algorithms

The Baggage Sorting Station Assignment Algorithm determines which sets of baggage sorting stations (for example only those on the same pier, or on all piers) are to be considered, in which order, and at what point reductions in service times are considered within each set. The baggage sorting stations within each set are then considered according to the selection priority given, namely the 'Baggage Sorting Station Selections'.

Three baggage sorting station assignment algorithms are presented here, which are named Algorithms 'A', 'C' and 'E' (two more are described in Appendix B.1), and represent a variation in the level of restrictions where Algorithm 'A' corresponds to the most restrictive and the others are less restrictive with 'E' being without any restriction.

Algorithms 'A' to 'E' express different priorities. Algorithm 'A' will attempt to assign all aircraft to their own piers before considering assigning any aircraft to other piers. Algorithm 'C' is similar to 'A' but considers alternative piers or reductions in service time for the current aircraft prior to considering the next aircraft, giving a much weaker pier preference overall. Algorithm 'E' does not impose any restriction on which piers to use.

Algorithm 'A': Baggage Sorting Station Assignment Algorithm **'A'** (strong pier preference)

```
begin
    Order all flights based on the current flight choice algorithm (Section 4.2.1);
    forall the flights do
        if feasible BSS exists on flight's own pier then
            Select a BSS using the selection algorithm;
            Assign the flight to the BSS;
        else
            Reduce the flight service time by the maximum reduction allowed;
            if feasible BSS exists on flight's own pier then
                Select a BSS using the selection algorithm;
                Assign the flight to the BSS;
            end
        end
    end
    forall the unassigned flights do
        if feasible BSS exists in the airport then
            Select a BSS using the selection algorithm;
            Assign the flight to the BSS;
        else
            Reduce the flight service time by the maximum reduction allowed;
            if feasible BSS exists in the airport then
                Select a BSS using the selection algorithm;
                Assign the flight to the BSS;
            else
                Assign the flight to the dummy BSS;
            end
        end
    end
end
```

Algorithm '*C*': Baggage Sorting Station Assignment Algorithm '*C*' (partial pier preference)

```
begin
    Order all flights based on the current flight choice algorithm (Section 4.2.1);
    forall the flights do
        if feasible BSS exists on flight's own pier then
            Select a BSS using the selection algorithm;
            Assign the flight to the BSS;
        else
            Reduce the flight service time by the maximum reduction allowed;
            if feasible BSS exists on flight's own pier then
                Select a BSS using the selection algorithm;
                Assign the flight to the BSS;
            else
                Reset the service time without reduction;
                if feasible BSS exists in the airport then
                    Select a BSS using the selection algorithm;
                    Assign the flight to the BSS;
                else
                    Reduce the flight service time by the max. reduction allowed;
                    if feasible BSS exists in the airport then
                        Select a BSS using the selection algorithm;
                        Assign the flight to the BSS;
                    else
                        Assign the flight to the dummy BSS;
                    end
                end
            end
        end
    end
end
```

In each case, once the algorithm has determined the set of sorting stations for consideration, the appropriate sorting station to be assigned from amongst those available at the time is determined by the baggage sorting station selection method currently being used, which is presented next.

Baggage Sorting Station Selections

The Baggage Sorting Station Selection method determines which of the baggage sorting stations in the current set should be assigned to the current flight. The following methods are considered:

1. **First In First Out (FIFO)**: The baggage sorting station with the earliest free service time is selected from all of the baggage sorting stations in the set under consideration. Initially this will keep opening new service stations, while they

Algorithm 'E': Baggage Sorting Station Assignment Algorithm **'E'** (no pier preference)

```
begin
    Order all flights based on the current flight choice algorithm (Section 4.2.1);
    forall the flights do
        if feasible BSS exists in the airport then
            Select a BSS using the selection algorithm;
            Assign the flight to the BSS;
        else
            Reduce the flight service time by the maximum reduction allowed;
            if feasible BSS exists in the airport then
                Select a BSS using the selection algorithm;
                Assign the flight to the BSS;
            else
                Assign the flight to the dummy BSS;
            end
        end
    end
end
```

exist, since a new one would always be that with least recent use. This is useful in meeting the fairness objective expressed by Formula 3.12 and the reduction in service time objective expressed by Formula 3.9.

2. **Last In First Out (LIFO)**: The baggage sorting station most recently used amongst those in the set is selected. This selection reduces the number of baggage sorting stations in use at any one time, since a new baggage sorting station is only opened when the previous ones cannot be assigned to the flight.

   When flights are ordered by their departure times, service time reductions are not permitted and assignment Algorithm 'E' is used (so that all sorting stations are considered, rather than only those on the preferred pier). This selection method guarantees the maximum assignments (maximising the objective expressed by Formula 3.7), by minimising the wasted/idle time between flights, Ding et al (2004) and Cormen et al (2001).

3. **Closest**: The BSS with the least distance to the current flight is selected from those in the set. This consists of both new sorting stations and those used previously. This method is useful for meeting the distance reduction objective expressed by Formula 3.10. With the measure of distance used in here, this objective will ensure that flights are assigned to sorting stations on their own pier by preference. Where sorting stations are at the same distance, a LIFO or FIFO method is used to break the ties. When LIFO is used it corresponds to

minimising the number of open BSSs, whereas FIFO corresponds to maximising the number of open BSSs which equates to increasing the fairness.

4. **Random**: A random BSS is selected from the BSSs in the aforesaid list. This sorting station selection approach does not take account of any particular objective.

4.2.3 Lookahead and Improvement

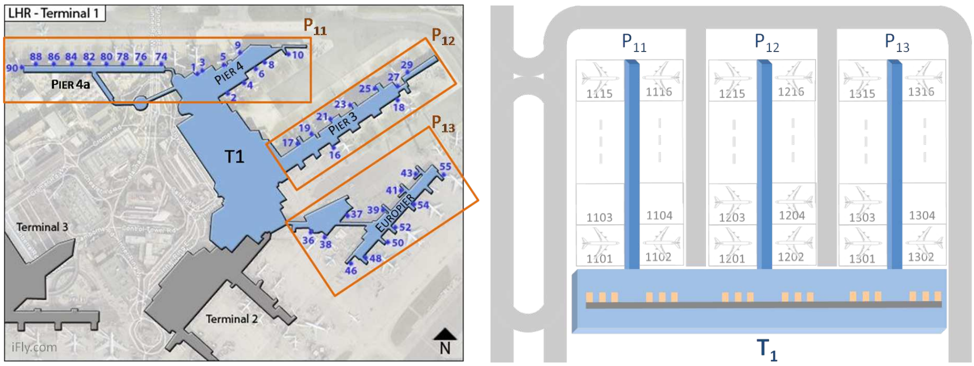
Haralick and Elliott (1980) considered the concept of "Lookahead and anticipate the future in order to succeed in the present" and "Lookahead to the future in order not to worry about the past". A type of lookahead was also used in Voß et al (2005). The ODT flight ordering method could potentially perform badly on the maximisation of assignments. The aim of the ODTLI is to retain the ODT flight ordering, but to look ahead when assigning sorting stations, thus potentially improving the assignment objective. The developed ODTLI method maintains a list of available sorting stations for this flight. It looks ahead to find out whether the selection of any of the available sorting stations may render a future flight infeasible. If this is the case, and there are other available sorting stations from which to select, this sorting station will be removed from the list. At the improvement stage of the process, sorting stations which have been removed will be reconsidered and may be exchanged for a station in the list if this improves the current selection method used.

4.3 Problem Data

Since it would be unrealistic to assume that the baggage from a flight at a terminal stand is serviced by a baggage sorting station in another terminal (e.g. passengers usually go through security and board flights from the same terminal at which they checked their baggage in), the following analysis is centred on a single terminal.

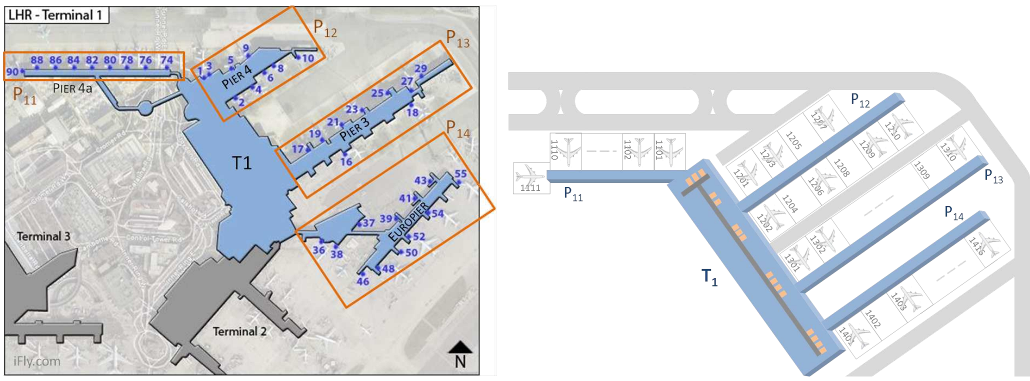
Two data sets obtained from the British Airports Authority (BAA) website were originally considered, being for December 2009 and March 2010 and composed of 219 and 270 flights respectively. No stand allocation information was available at the time. These data sets were studied for a terminal with three piers as shown in Figure 4.1 and presented in Ascó et al (2011). Subsequently NATS provided more detailed data for London Heathrow airport, which also contains details of the flight assignments to stands and of cancelations for both days. The two new data sets consist of 194 flights for the 16^{th} December 2009 and 163 flights for the 1^{st} March 2010, all departing from London Heathrow airport Terminal 1. More detailed information on Terminal 1 at

London Heathrow airport was also provided, which prompted the study of a 4-pier topology as a closer representation of London Heathrow airport Terminal 1, Figure 4.2. These two topologies gave an opportunity to look at some of the effects the topologies had upon the solutions obtained.



a Terminal 1 overall view. b 3-pier topology.

Figure 4.1: London Heathrow airport Terminal 1.

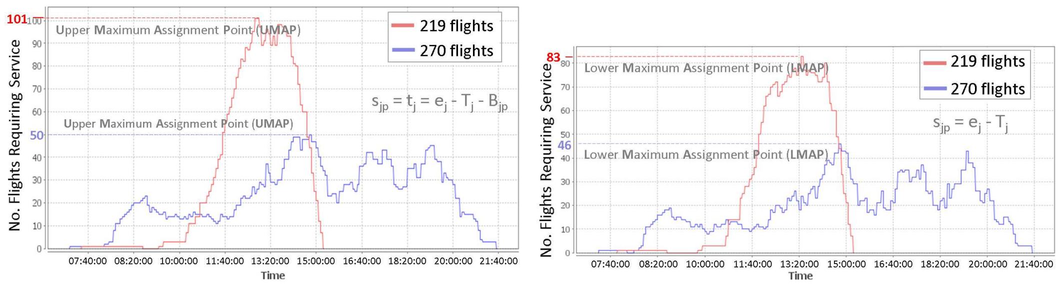


a Terminal 1 overall view. b 4-pier topology with 46 stands.

Figure 4.2: London Heathrow airport Terminal 1.

4.3.1 Baggage Sorting Stations Required

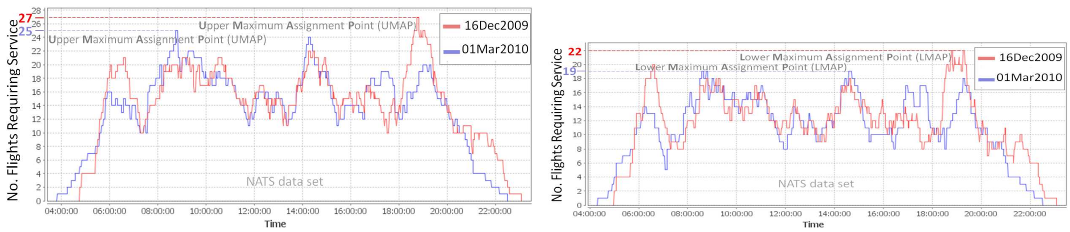
Figures 4.3a, 4.3b, 4.4a and 4.4b show the total number of flights which require service at different times of the day, with and without the buffer times. Figures 4.3a and 4.4a show the number of sorting stations which are required when full buffer times are used (i.e. where there is no service time reduction allowed) and Figures 4.3b and



a Total number of flights requiring service over the day when full buffer times are required, UMAP.

b Total number of flights requiring service over the day when no buffer times are required, LAMP.

Figure 4.3: BAA's website data sets for London Heathrow airport.



a Total number of flights requiring service over the day when full buffer times are required, UMAP.

b Total number of flights requiring service over the day when no buffer times are required, LMAP.

Figure 4.4: NATS data sets for London Heathrow airport.

4.4b represent the number of flights actually requiring service at that time (i.e. no buffer times are included). It is possible to draw the following conclusions:

1. With a limited number of baggage sorting stations, the maximum height of the lines in Figures 4.3 and 4.4 could potentially be an indication of the assignment problem difficulty.

2. Fewer sorting stations are required at the peaks when buffer times are not included, although the absence of the buffer times would result in less robust solutions.

The Lower Maximum Assignment Point (LMAP) here is defined as the minimum number of sorting stations required for the maximum number of assignments to be achieved once maximum reductions in service time have been applied. Similarly, the Upper Maximum Assignment Point (UMAP) is defined to be the minimum number of sorting stations at which the maximum number of assignments can be achieved without the need to reduce service time. The LMAP and UMAP points will be observed to be useful later when interpreting the results.

When no buffer times are considered the maximum assignment (LMAP) is 83, 22, 46 and 19 BSSs for the four data sets (Figures 4.3b and 4.4b), and when no reduction in service is allowed the maximum assignment (UMAP) is 101, 27, 50 and 25 BSSs (Figures 4.3a and 4.4a) for the 219, 194, 270 and 163 flight problems respectively, as shown in Figures 4.3 to 4.4.

4.3.2 Generating Missing Stand Assignments

When assignment information for flights to stands was unavailable, different problems were generated by allocating the flights randomly to the stands avoiding any overlap on a single stand. In order to minimise any bias introduced by these random allocations, a hundred different random allocations were generated. We realise, of course, that real schedules will have some bias, given airline preferences, which is also shown by the data provided by NATS. When assigning flights to BSSs the assignment of flights to stands is only required for those objectives which take account of the position of the BSSs with respect to the stands, Section 3.3.4.

Two examples of flight assignments to 48 stands are illustrated in Figure 3.7 for the data sets from the BAA website. The results of the sorting station assignment algorithms (which are themselves deterministic) across the 100 different stand allocations are presented in the following section using the box-and-whisker diagram. In each diagram, results are shown next to each other for each number of BSSs and are in the same order as listed in the key. In assigning flights to stands, it has been assumed that all of the stands were suitable for any aircraft. The available stands were assumed to be equally distributed over three piers, with 16 stands per pier.

4.4 Results

This section details the experiments which were performed to evaluate the differences between the algorithms described in Section 4.2, and to understand the ways in which these results depend upon the number of BSSs available for assignment to flights. The behaviour is studied in the case where there are too few BSSs as well as when the BSSs are plentiful. Two cases were considered: without allowing reductions in service time (i.e. requiring full buffer time), and allowing reductions in service time (i.e. allowing buffer times to be reduced).

The various experiments were executed for the data which was captured from the BAA website for London Heathrow airport, considering cases where they were assumed to be 14, 16, 18, 20, 22, 24, 26, 28, 34 or 36 BSSs per pier (changing the total number of BSSs by increments of 6). The results were presented in Ascó et al (2011). The data from NATS was also studied, with ranges from 12 to 36 BSSs in increments of 1 BSS. For the purpose of the distance reduction objective, a distance of one unit is assumed between different sides of a pier and a distance of two units was assumed between different piers (as shown in Figure 3.3a), so that it is preferable to use the other side of the same pier before considering BSSs for other piers. It is assumed that reductions in service time can only reduce the buffer time rather than the base service duration (i.e. the base service duration is the minimum which will be available). Service times were set so that $T_j = 1$ hour and $B_j = 15$ minutes for European flights, and $T_j = 1\frac{3}{4}$ hours and $B_j = 30$ minutes for non-European (long-haul) flights, since these are usually larger flights with more baggage and a requirement to check-in earlier.

The larger number of BSSs required to fully assign all of the flights for the data sets from the BAA website compared to those provided by NATS corresponds to a higher flight density for the data sets from the website, as it is shown in Figures 4.4 and 4.3. The reason for this is mainly that flights presented in the website with different flight numbers were considered to be different flights. This is not the case in reality however, as different airlines have mutual agreements whereby they share the aircraft travelling to the same destinations at the same time, so that customers see the expected airline flight reference number. This information was not available to this study at the time. Furthermore, information available on the website was susceptible to change as the day progressed, as some flights were later cancelled but were still considered in the initial study. Additionally, airports only publish information about the gate assigned to a flight near the time the flight is due to arrive or depart, so such information was not available early in the day when the data was collected. This information was randomly generated in the experiments, by applying the random constructive algorithm presented in this chapter multiple times, with different random seeds to reduce any bias.

The absence of information about the number and location of the BSSs directed the study to consider a range of BSSs for each data set which allows investigating the effect and performance of the different algorithms where there are plentiful or few number of BSSs. Within each range both the LMAP, which corresponds to the number of BSSs necessary to be able to assign all of the flights for when no buffer time is considered, and UMAP which refers to the number of BSSs necessary to be able to assign all of the flights without reducing the buffer time (both introduced in Section 3.3.2), were considered, where LMAP $\leq$ UMAP.

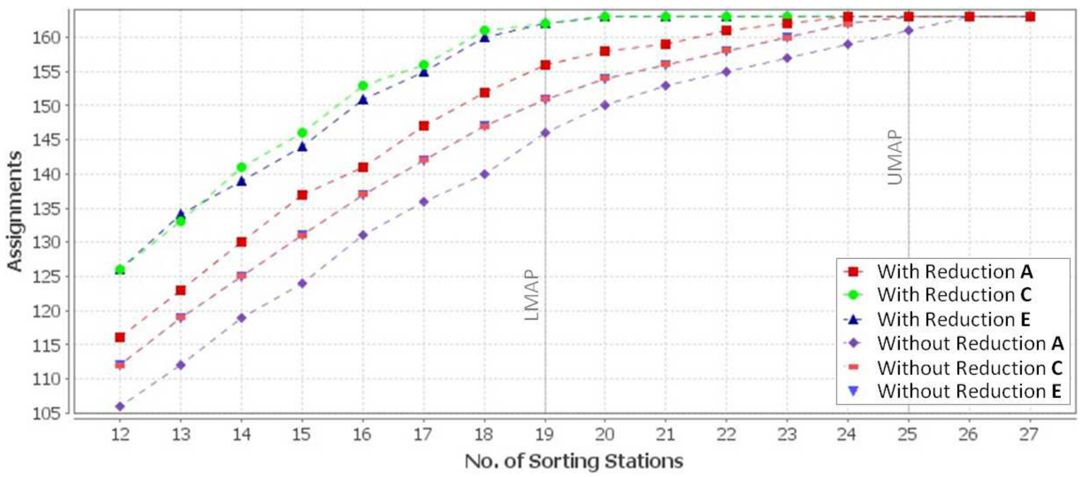


Figure 4.5: Number of sorting stations assigned, OST ordering and LIFO selection method for a 3-pier topology and 163 flights.

Later normality tests were run to identify whether the data could be said to follow a normal distribution, which is a requirement for use of the t-test, otherwise the Mann-Whitney U test is preferable. Razali and Wah (2011) compared some normality tests and concluded that Shapiro-Wilk is the most powerful normality test. Thus the Shapiro-Wilk normality test was run for some of the data to ascertain if the data could be said to be normal, but the data could not be said to follow a normal distribution, the results of which are shown in Appendix B.1. So Mann-Whitney U tests were carried out to ascertain the statistical significance.

The various experiments in this chapter were executed using a single threaded Java application, running on a 3GHz Intel(R) Core(TM)2 Duo CPU, desktop with 2GB RAM under Windows XP (SP3) and in a 2.5Hz 64bit Intel Core i3, laptop with 4GB RAM under Windows 7. Each execution of the constructive algorithms took no more than 9 milliseconds.

A view of the behaviour of the algorithms as the number of BSSs changes is presented in Figure 4.5. This shows the number of sorting stations which could be assigned to flights using the OST flight ordering method using the LIFO sorting station selection method, for various numbers of available sorting stations. This is then compared with the situation when reduction in service is and is not allowed. It was originally planned to use ODT, but it was shown in Ascó et al (2011) that the OST ordering method provided better assignments than ODTLI when reductions in service were allowed and the number of sorting stations was close to, or above, the LMAP. This persuaded me to use OST in my initial observations. A comparison of the results for the different BSS assignment algorithms shows the following:

1. As expected, allowing reduction in service times allows more flights to be serviced, since shorter service times may allow a flight to sit between two other flights where this would otherwise be impossible.

2. Regardless of whether reductions in service time are permitted, sorting station assignment Algorithm 'A' achieves fewer assignments than the other algorithms. This is a consequence of Algorithm 'A' assigning aircraft to their own pier by preference whereas assignment to a different pier may have allowed more flights to be assigned.

3. Where reductions in service are not permitted, the performance of Algorithm 'E' was exactly the same as Algorithm 'C'. The results in Ascó et al (2011) show that in general Algorithm 'E' always performed at least as well as Algorithm 'C', and sometimes better, as we would expect since the pier preference can sometimes conflict with maximising the assignments. Interestingly, this was not always the case when reductions in service time were permitted, and there are instances when the preference for the same pier actually means that more flights can be assigned, as shown in Figure 4.5 for 14 to 18 sorting stations where Algorithm 'C', which takes account of the pier grouping, assigns more flights than Algorithm 'E'. These results were even more pronounced in Ascó et al (2011) which used different data sets with random allocation of stands to flights, and indicate that there are sometimes advantages in assigning flights to sorting stations on their own piers, perhaps requiring a reduction in service time to do so, and thus allowing more flights to be assigned.

   Algorithm 'C' also achieved full assignment at the UMAP, shown in Figure 4.5, when there are 25 BSSs. It is noted however, that this is only guaranteed for no restriction, Algorithm 'E', when ordering flights by departure times using the LIFO selection method and no buffer time.

4. When reduction in service time is permitted, Algorithm 'E' no longer guarantees the maximum assignment of BSSs, as can be seen in Figure 4.5 for between 14 and 18 BSSs for the 163 flights data set, as similarly shown in Figure 4.6 for between 72 and 96 BSSs for 219 flights, and for 48 BSSs shown in Figure 4.7 for 270 flights. Algorithms 'A' and 'C', which consider reduction in service times first and different piers (thus more often reducing service times), sometimes achieve better assignments (e.g. 72 and 78 BSSs Figures 4.6 and 4.7).

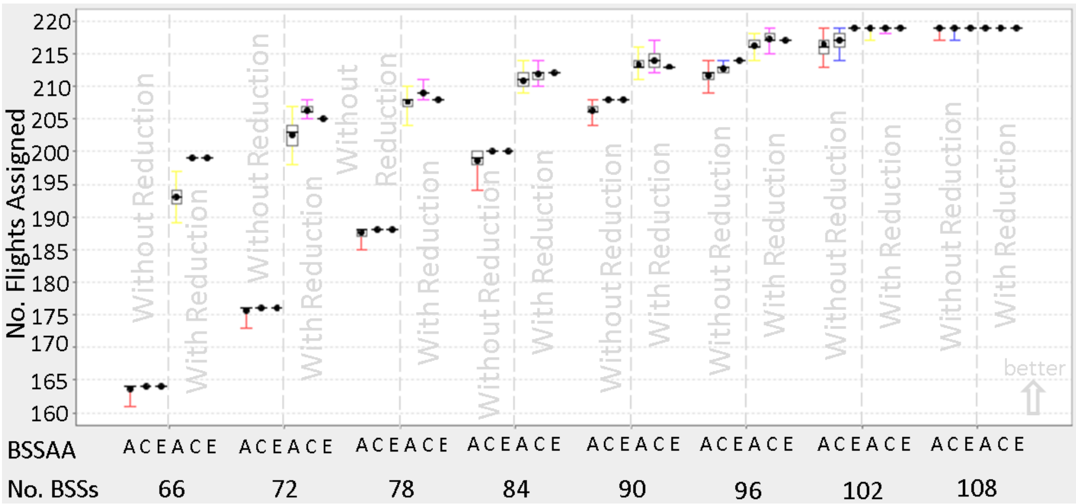


Figure 4.6: Number of assignments for 219 flights (BAA's website), ODTLI and LIFO, with and without permitting reductions in service.

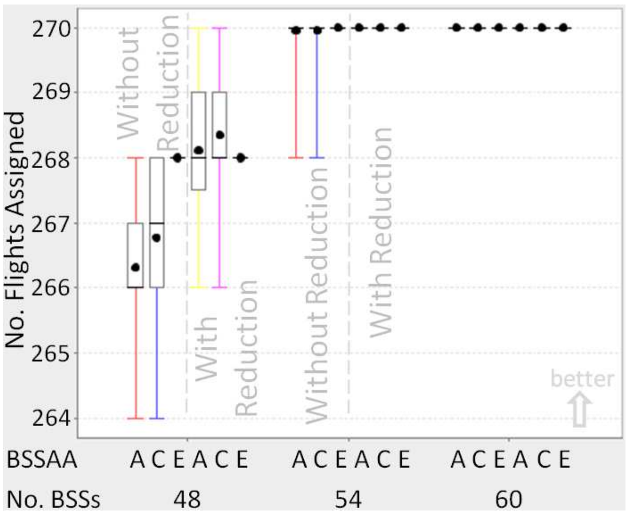


Figure 4.7: Number of assignments for 270 flights (BAA's website), ODTLI and LIFO, with and without permitting reductions in service.

These results corroborate those from Ascó et al (2011), where different data sets were used with a higher number of flights and also higher flight densities, as shown in Figures 4.3 and 4.4. Both sets of results indicate that as to which is the better assignment method will depend upon the ratio of flights to sorting stations. Furthermore, it is noted that counts of the minimum number of sorting stations which are required with and without reductions in service time, shown in Figures 4.4b and 4.4a, provide a simple method of determining whether the available sorting stations are sufficient or not to avoid reductions in service time.

In order to determine the maximum sorting station assignments when reduction in

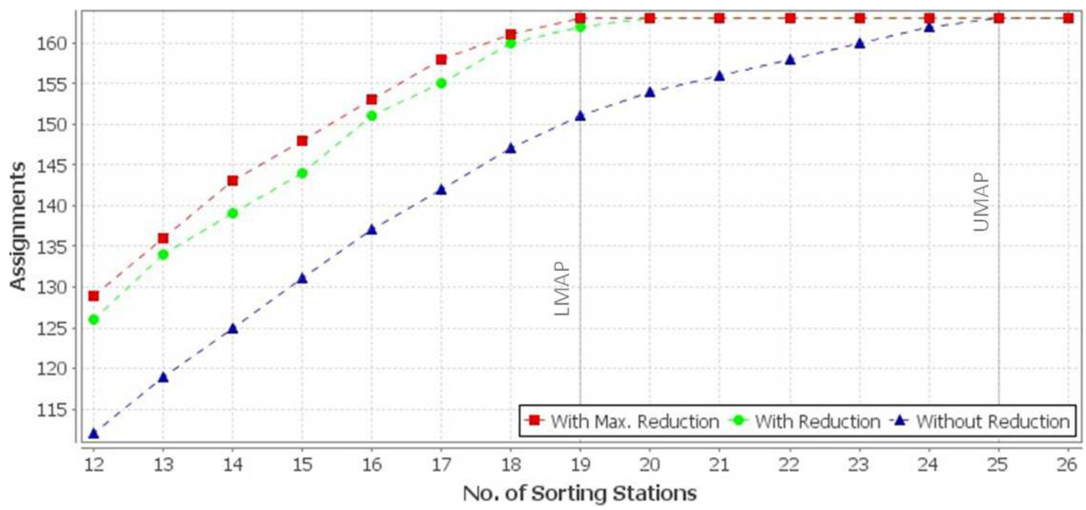


Figure 4.8: Number of assignments, OST ordering method, Algorithm 'E' and LIFO selection method, for a 3-pier topology and 163 flights.

service times is permitted, experiments were executed with the buffer times removed (equivalent to maximal service time reduction), using the OST ordering method, Algorithm 'E' and LIFO selection method. The results are shown in Figure 4.8, for the 163 flights problem. More flights can be assigned when reductions are permitted, as expected, until sufficient sorting stations are available to assign all of the flights (full assignment) even without needing reductions. In most cases, allowing reductions was almost as good as using maximum reductions.

With maximum reductions (i.e. no buffer times), the full assignment occurs when there are 19 BSSs for the 163 flight problem. This value is the same as the theoretical minimum (the lowest maximum assignment point, LMAP) shown in Figure 4.4b. Here, OST (Order by Starting Times) achieves full assignments at the theoretical minimum points (LMAP for maximal reductions and UMAP for no reductions in service times), even though it gives no guarantee of doing so, unlike ODT (Order by Departure Times), as shown in Figure 4.9.

Figure 4.5 can also be used to compare the performance of Algorithms 'A', 'C' and 'E' in terms of the number of assignments which are achieved when reduction in service time is permitted, using the OST ordering method and LIFO selection method. This shows that Algorithm 'A' provides the lowest number of assignments. This was also seen in Ascó et al (2011) for the alternative data sets. Algorithms 'C' and 'E' both provide a similar number of assignments to each other, with Algorithm 'C' providing slightly more than 'E' in some cases.

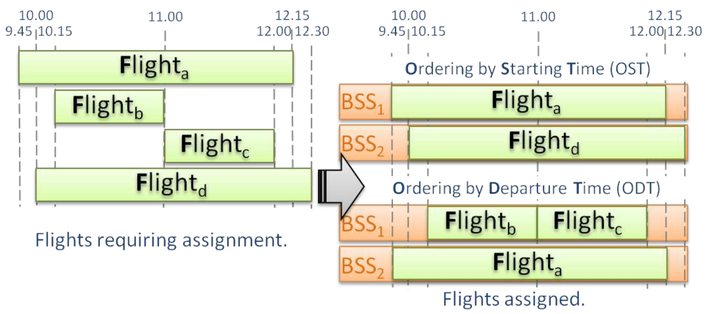


Figure 4.9: Example of assignments achieved by OST and ODT.

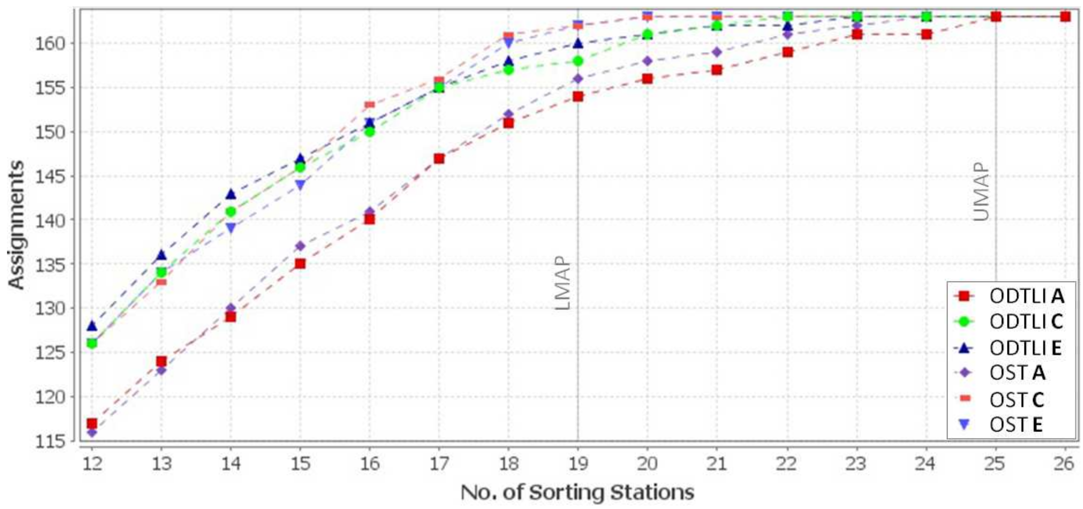


Figure 4.10: Number of assignments for LIFO selection method with different ordering methods and selection methods, for 3-pier topology and 163 flights.

Since reductions in service time have obvious benefits, the remaining experiments consider the cases where these are permitted and evaluate the differences between Algorithms 'A', 'C' and 'E' and between the different flight ordering and BSS selection methods.

4.4.1 Comparison of Assignments with Service Reduction

Figure 4.10 compares the ODTLI and OST flight ordering methods, showing the number of sorting station assignments which were made with the LIFO selection method. This shows that the ODTLI flight ordering method provided a better assignment when there were fewer sorting stations (between 12 and 15 sorting stations), but at some point, as the number of sorting stations increases, the difference decreases. As the number of sorting stations approaches the number necessary for full assignment (the LMAP), the OST flight ordering actually improves upon ODTLI.

Comparison of some resulting assignments showed that, perhaps counter intuitively, not only was ODTLI failing to assign more flights at these times, but the flights which were not assigned had longer service times than those which OST failed to assign. Indeed, there were cases where every aircraft which OST failed to assign was a short-haul flight and every aircraft which ODTLI failed to assign was a long-haul flight. The order of consideration of flights appears to be important in this case.

The key to understanding this behaviour lies in considering the size of the remaining gaps. Since the ODT and ODTLI methods order the flights by their departure times, where flights have similar service starting times, preference will be given to flights with shorter service times (i.e. earlier departure times). On the other hand, the OST choice of flights could be regarded as preferring flights with longer service times (for similar departure/end of service times). By assigning long-haul flights first, the OST algorithm was able to fit short-haul flights into the remaining gaps (with appropriate service time reductions). However, by assigning short-haul flights first, the ODTLI was then unable to schedule the remaining long-haul flights, resulting in fewer assignments. When there are few sorting stations, the ability of the ODTLI choice to minimise the gaps is a useful one and results in more sorting station assignments than the OST ordering method. However, as the number of sorting stations increases, the remaining gaps begin to be large enough to accommodate short-haul aircraft, and OST performs better.

Further experiments showed that this behaviour was not restricted to the LIFO selection method, but also occurred for the FIFO and 'Closest' selection methods, and did so for the same number of sorting stations, Appendix B.1.2.

4.4.2 Reduction in Service

Figure 4.11 shows the total reduction in service time (expressed by Formula 3.9) for all assigned flights, with differing numbers of BSSs, using Algorithm 'E', and comparing the performance of ODTLI and OST flight ordering methods and 'Closest', LIFO and FIFO sorting station selection methods. Where Figure 4.12 shows the mean reduction in service time for the flights which have had a reduction in service time.

Comparing Figures 4.11 and 4.12 it was observed that, using the ODTLI ordering method, as the number of sorting stations is increased the reduction in service time initially increases as more assignments are achieved, although the mean reduction in each case increases more slowly, or not at all, and we know from Figure 4.10 that the number of assignments is increasing at this point. This indicates that the increased

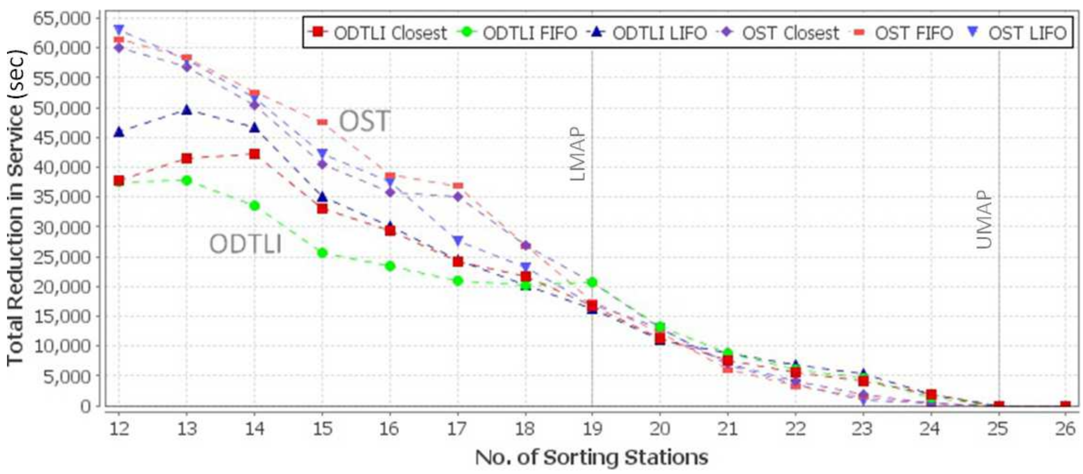


Figure 4.11: Total reduction in service time in seconds for NATS data set 1^{st} March 2010, 3-pier topology and Algorithm 'E'.

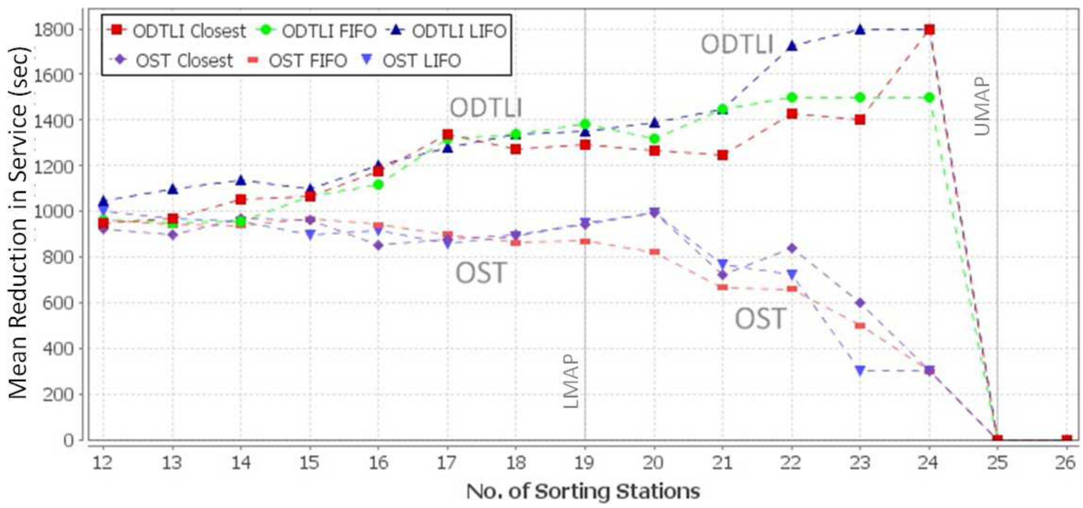


Figure 4.12: Mean reduction in service time in seconds for NATS data set 1^{st} March 2010, 3-pier topology and Algorithm 'E'.

number of available sorting stations and the ability to reduce the service time are both contributing towards the increase in the number of assigned flights at that time.

As the number of sorting stations is increased further, a point is soon reached where the total reduction decreases, but the mean reduction per sorting station goes up. This indicates that more and more of the flights are being assigned with no reduction in service. This continues until the number of sorting stations is sufficient to allow all of the assignments to be made, at which point the total reduction in service decreases, until eventually all flights can be assigned without any reduction in service time. This was also observed for the BAA's website data set in Ascó et al (2011).

For the OST ordering method, the mean reduction is relatively stable, but the total reduction decreases as the number of sorting stations increases, indicating that the number of sorting stations with reduced service decreases over this time. Soon after the LMAP, when there are 20 baggage sorting stations, a point is reached where the number of sorting stations is sufficient to allow all of the assignments to be made (see Figure 4.10), with ever decreasing reductions in the service time. As the number of sorting stations is further increased, the total reduction in service and the mean reduction in service time both decrease, until eventually all flights can be assigned without any reduction in service time.

4.4.3 Comparison of Distances with Service Reduction

Figures 4.13 and 4.14 show the results as far as the distance reduction objective (expressed by Formula (3.10)) is concerned. These show the total distance between the assigned baggage sorting stations and the stands at which the flights are located. Results are shown for the three sorting station selection Algorithms 'A', 'C' and 'E', with the 'Closest' selection methods and the ODTLI and OST flight ordering methods.

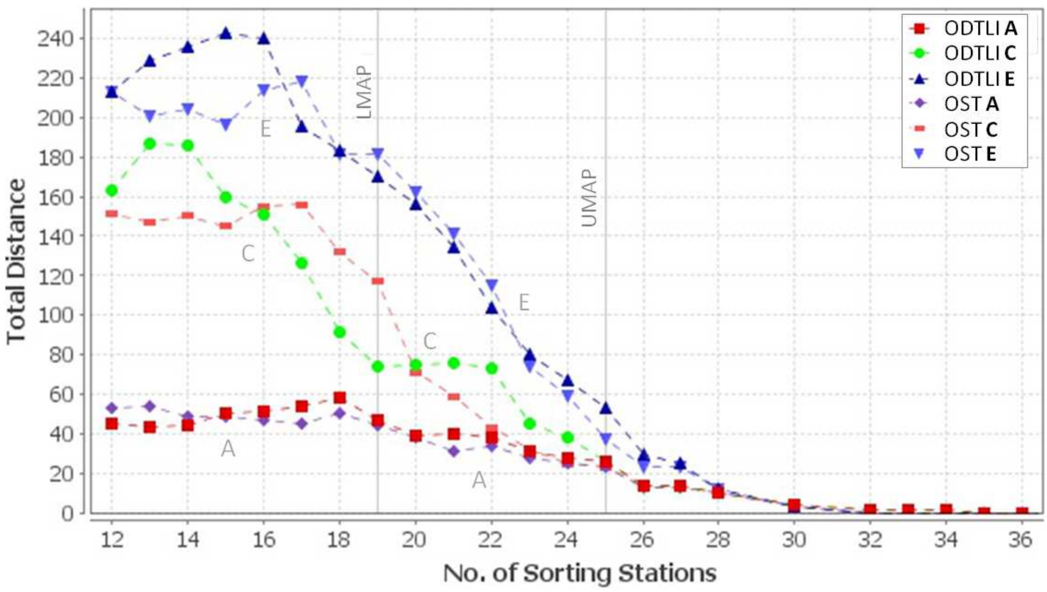


Figure 4.13: Total distance with 'Closest' selection method.

The distance basically measures the number of flights which could not be assigned

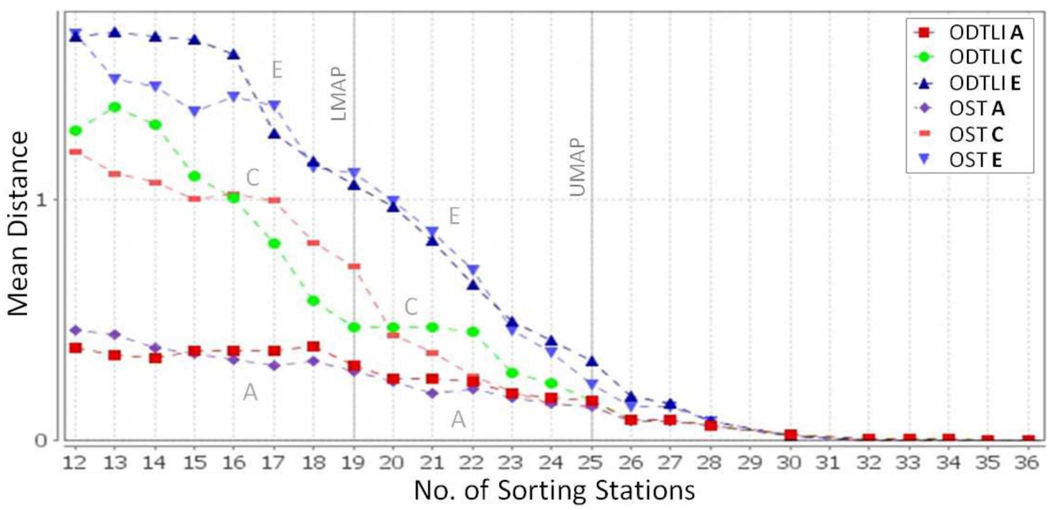


Figure 4.14: Mean distance with 'Closest' selection method.

to sorting stations on their preferred pier. It may be observed that the total distance decreases as the number of sorting stations is increased, since more sorting stations become available on the preferred piers. Even after all flights have been assigned to sorting stations, the distances can be positive, since the availability of a sorting station at the terminal does not imply that it is on the correct pier for the flight.

As expected, since Algorithm '*A*' attempts to assign to the same pier first and considers applying a service time reduction before considering other piers, Algorithm '*A*' performs better than Algorithms '*C*' and '*E*' in terms of distance when there is a shortage of piers. For similar reasons, Algorithm '*C*' performs better than Algorithm '*E*'. However, Algorithm '*E*' assigned more flights to sorting stations, and unassigned flights are here assumed to have no distance, this also needs to be taken into account. Figure 4.14 shows the mean distance per assigned flight necessary to avoid the problem of unassigned flights, and it can clearly be seen that Algorithm '*A*' attained the lower mean distance.

It is possible to conclude from this that for the cases where Algorithm '*C*' achieves at least the same number of assignments as Algorithm '*E*', Algorithm '*C*' would be the preferable choice since the distances would be lower. On the other hand, by the time that Algorithm '*A*' had achieved maximal assignment (which is usually considered to be the primary objective), there would be no distance benefit in using Algorithm '*A*' rather than Algorithm '*C*'.

4.4.4 Fair Workload with Reduction in Service

As a measure of fairness we considered the deviation of the total usage times of the sorting stations from the mean usage time. This corresponds to the fairness objective which was expressed by Formula (3.12).

Figure 4.15 compares the results for the 'Closest', FIFO and LIFO sorting station selection methods and the ODTLI and OST flight ordering methods, showing the total seconds deviation from the mean usage across all baggage sorting stations, using sorting station assignment Algorithm 'E'.

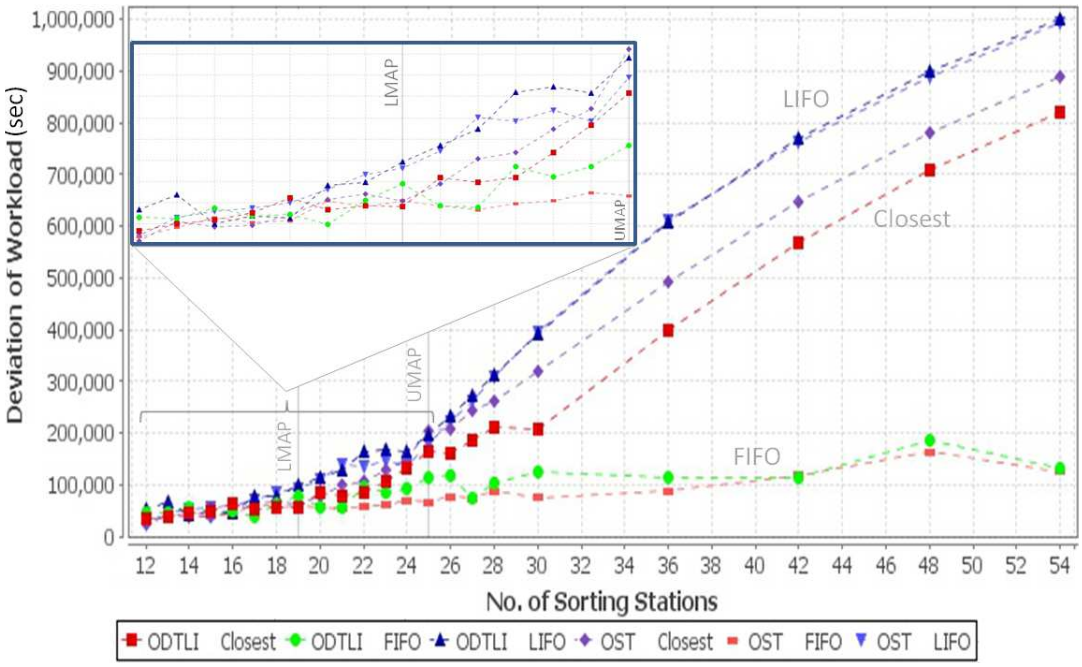


Figure 4.15: Fair workload and Algorithm 'E' (no restriction).

The FIFO selection method may be considered to take fairness into account, only re-using a sorting station once all of the others have been used, and indeed it consistently performs better than LIFO and 'Closest' for both flight ordering methods. However, although the FIFO selection method will cycle through the sorting stations, giving a more equitable number of flights to each sorting station, long-haul and short-haul flights are treated identically. This can result in differences in the total service times. These differences will depend upon how many of the long-haul flight assignments coincide so that they are assigned to the same service stations. As the number of sorting stations is increased a cyclic-type behaviour may be observed.

Conversely, the LIFO selection method will continue to re-use the same sorting stations where possible, thus increasing the number of sorting stations will further increase the inequity, as can be observed in Figure 4.15.

The 'Closest' method takes no explicit account of equity or sorting station re-use frequency, and instead will tend to follow the flight assignment. It was observed that this results in an inequity almost as great as for the LIFO method.

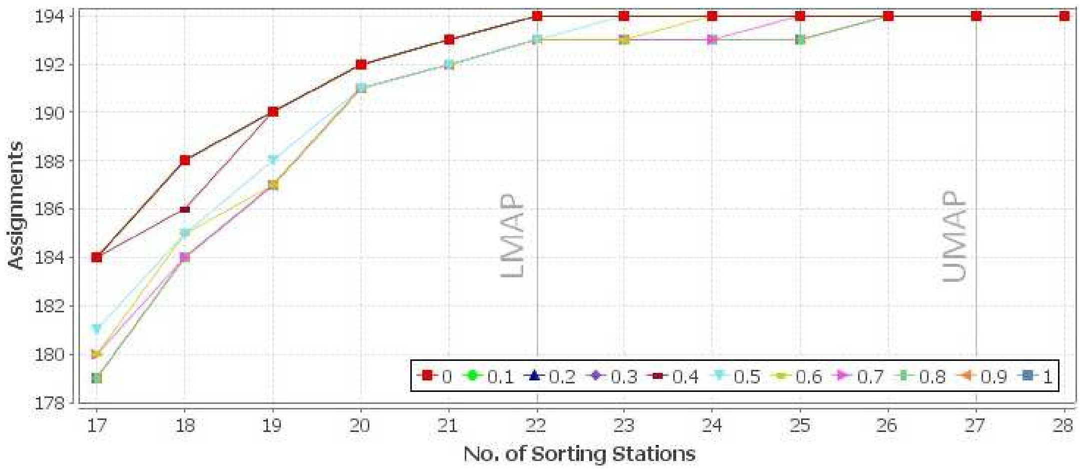
4.4.5 Order Between Times

Previous experiments show that depending on the number of BSSs it is OST or ODTLI which provide better solutions. Some questions arise such as 'Is it possible to quickly generate more solutions which present some differences with respect to each others?' and 'Does an ordering exist which would produce a better solution throughout the range of BSSs under consideration?'. For this propose a general representation of the 'Flight Ordering Methods' was presented in Section 4.2.1 and named OBT, which uses a parameter α to control the behaviour, $0 \leq \alpha \leq 1$, as discussed in Section 4.2.1. As the number of BSSs changes the constructive algorithms' behaviour can be seen in Figure 4.16 for different values of α within the range of BSSs. The empirical results show that higher values of α perform better for a very lower number of BSSs (not shown in figure). This corroborates the results for ODT which performs better than or equal to OST for up to 15 BSSs, as OBT with $\alpha = 1$ is equivalent to ODT. It has been observed that OBT with $\alpha = 0.5$ performs better than OST for up to 15 BSSs and it is overall better than ODT for the region of low number of BSSs, shown in Figure 4.17.

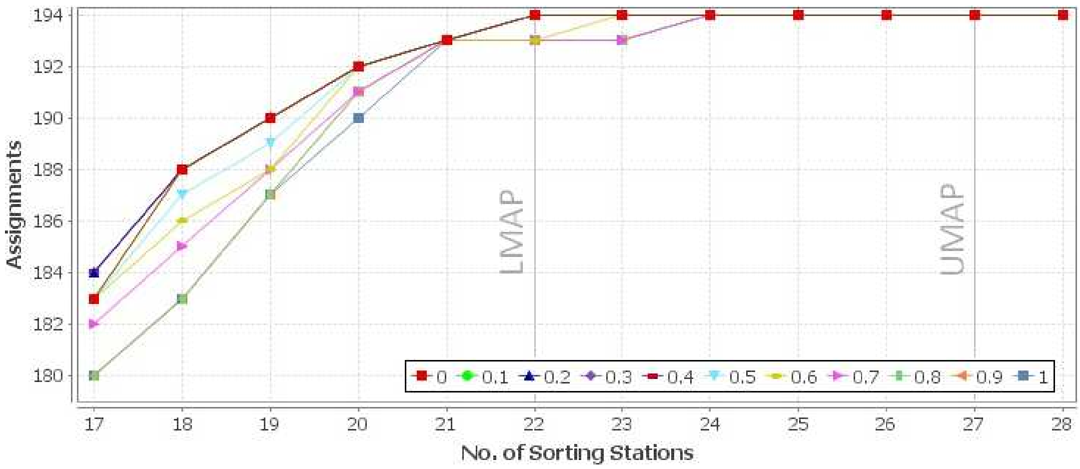
OBT for $\alpha = 0.5$ performs better than ODT (and OST) for very low number of BSSs, Figures 4.17, 4.18 and 4.19, which also applies overall to the range of BSSs considered in respect of ODT.

The Baggage Sorting Station Assignment Algorithm 'A' (highly restrictive) consistently provides the lowest assignments, as has been previously shown and can also be seen in Figures 4.17.

OBT provides many more ways of obtaining different solutions by means of changing the α parameter in comparison to using only OST, ODT and ODTLI. OBT can also be extended to Order Between Times Lookahead and Improvement (OBTLI) in the same way as ODT was extended to create ODTLI. This may increase further the number of solutions which may be useful when many initial solutions are required for a population base optimisation algorithm instead of using random solutions. The solutions generated by OBT and OBTLI may not differ greatly when compared against



a Baggage Sorting Station Assignment Algorithms '*E*'.



b Baggage Sorting Station Assignment Algorithms '*C*'.

Figure 4.16: Assignments for 194 flights with reduction of service, a 3-pier topology, LIFO and OBT.

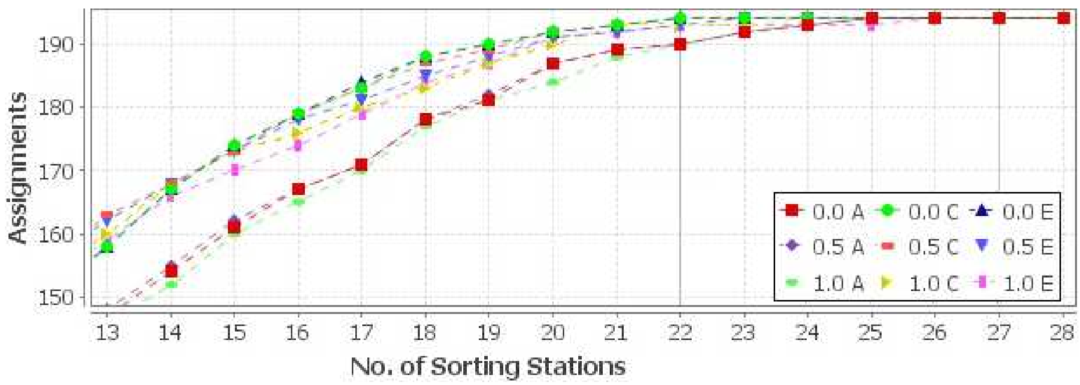


Figure 4.17: Assignments for 194 flights with reduction of service, a 3-pier topology, LIFO, OBT and algorithms '*A*', '*C*' and '*E*'.

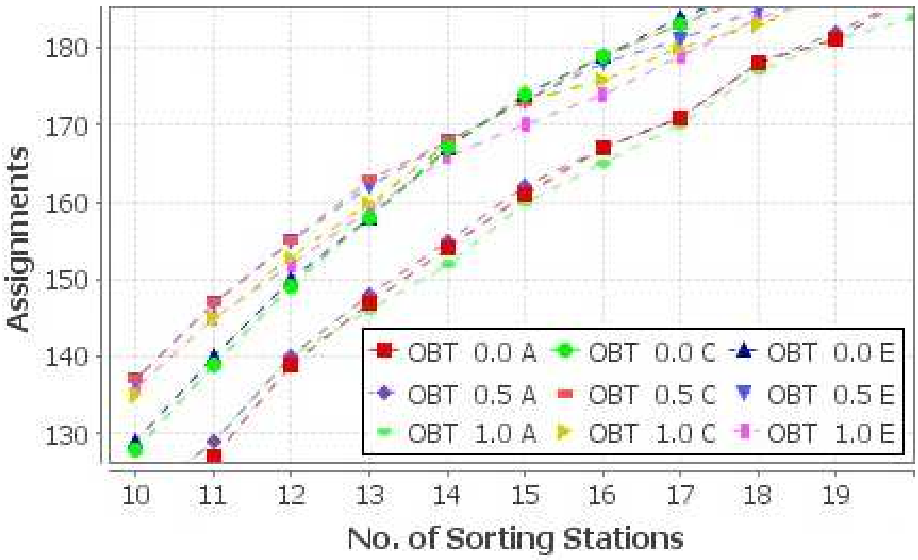


Figure 4.18: OBT assignments for NATS data sets of 194 flights with service reduction, a 3-pier topology and LIFO for a reduced range of number of BSSs.

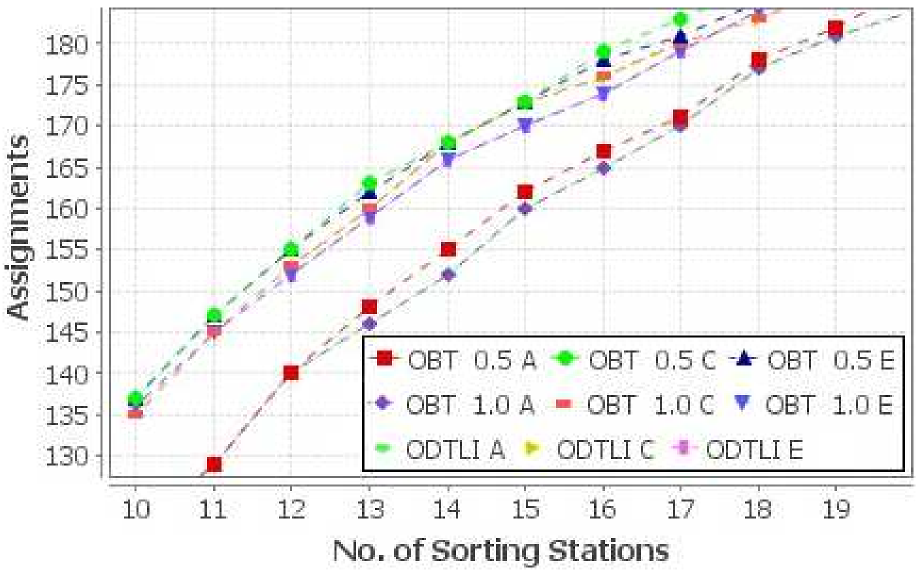


Figure 4.19: OBT and ODTLI assignments for NATS data sets of 194 flights with service reduction of service, a 3-pier topology and LIFO for a reduced range of number of BSSs.

each other so if one of OBT and OBTLI is to be used to generate multiple solutions, either should be used but not both. The fact that the time required to generate a single solution is also very small adds to the advantages already presented.

4.4.6 Combined Objectives

The Branch and Bound (B&B) algorithm belongs to the group of exact algorithms used to find an optimal solution in discrete and combinatorial optimisation, and is composed of two parts. The first splits the domain of solutions into smaller domains, so called branching. The second calculates the upper and lower bounds of the problem, so called bound, which is used to discard large subsets of uninteresting solutions which in turn helps to speed the search for the optimal solution. This is one of the algorithms used in commercial optimisation software packages, such as CPLEX and Gurobi.

Although this is inherently a multi-objective problem, the importance of ensuring maximal assignment of flights to sorting stations (top priority) and the relative importance of keeping reasonable buffer times (second priority) allow these objectives to be combined into a single compound objective (Equation 4.2) with weights W_1, W_2 and W_3 chosen to implement these priorities. So that for a solution quality assessment, the solutions obtained when applying the constructive algorithms can be compared with the Upper Bound and the best solutions obtained from applying CPLEX to the Integer Linear Programming (ILP) representation of the Airport Baggage Sorting Station Problem (ABSSAP) presented in Chapter 3.

$$f = W_1 * \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{p=1}^{P_j} y_{ijp} + W_2 * \sum_{j=1}^{M} \sum_{p=1}^{P_j} r_{jp} + W_3 * \sum_{j=1}^{M} \sum_{p=1}^{P_j} \left(C_{jp} * \sum_{i=1}^{N} (y_{ijp} * d_{ij}) \right) \quad (4.2)$$

Considering that the first objective corresponds to the most important objective, its improvement should be greater than any detrimental effect it may have on the other objectives, which indicates that increasing the assignments at least by 1 should be better than the combined effect of both the maximum reduction in service ($B_{max} =$ 1800 sec for long-haul flights) and being assigned to the most distant sorting station ($D_{max} = 9$ for the 3-pier topology); $|W_1| * 1 > |W_2| * B_{max} + |W_3| * D_{max}$, similarly with the second objective and third which gives $|W_2| * B_{max} > |W_3| * D_{max}$. An objective which increment decreases the quality of a solution needs to have a negative weight (minimisation of the objective) as it happens with the weights for the second and third objectives here. Giving a $W_3 = -1$ then $|W_2| * 1800 > 1 * 9$ so $|W2| > \frac{1*9}{1800} = 0.005$, and a value of $W_2 = -0.008$ was used which gives a $W_1 > 0.008 * 1800 + 1 * 9 = 23.4$. However, an extra assignment may also have a detrimental effect on the rest of the assignments, i.e. reduce the service time of the next flight assigned to the same BSS, which means that the value of W_1 may have to be further increased. The value of W_1 was determined by running initial experiments using different values, from 15 to 100, for the data set of 16^{th} December 2009, 3-pier topology, a fixed W_2 of -0.008

and W_3 of -1. A value of $W_1 = 90$ appeared to give an appropriate balance between the objectives and was adopted. Summarising the weights used are the same as those presented in Ascó et al (2012) with the values of $W_1 = 90$, $W_2 = -0.008$ and $W_3 = -1$ respectively.

Figure 4.20 shows the percentage improvement in fitness of the results for different constructive algorithms and the solution obtained from CPLEX for a one hour run, applied to different numbers of BSSs with respect to the worst constructive solution (f_w) and the Upper Bound obtained by CPLEX (f_{UB}), Equation 4.3. The percentage of the gap to the Upper Bound which is achieved is shown, so 100% improvement corresponds to solutions which reach the upper bound for their specific case, whereas 0% corresponds to no improvement over the worst (constructed) solution. The application of the constructive algorithms to the solutions obtained required no more than 9 milliseconds per solution whereas CPLEX was run for 1 hour and the best solution it found was utilised.

$$\%Fitness = \frac{f - f_w}{f_{UB} - f_w} * 100 \tag{4.3}$$

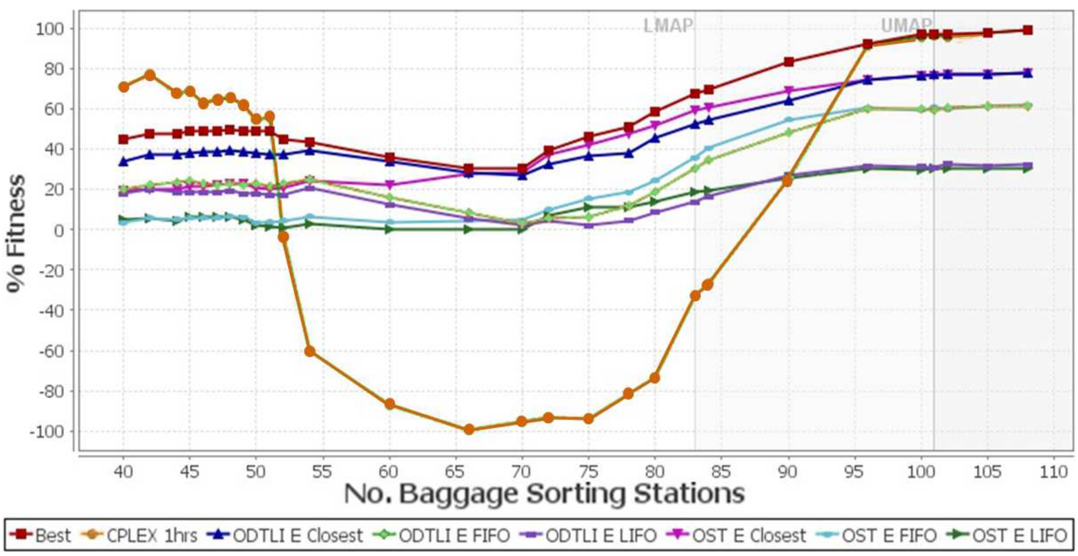


Figure 4.20: Constructive Algorithms and CPLEX percentage improvement in fitness for 219 flights and a 3-pier topology.

Where the situation to the problem is more difficult, which corresponds to the 219 flights obtained from the BAA's website, three areas can be clearly identified with different comparative fitness between the constructive algorithms and CPLEX. For a very low number of BSSs ($N \ll LMAP$) the solutions obtained by CPLEX are better than all of the solutions obtained by applying the different constructive

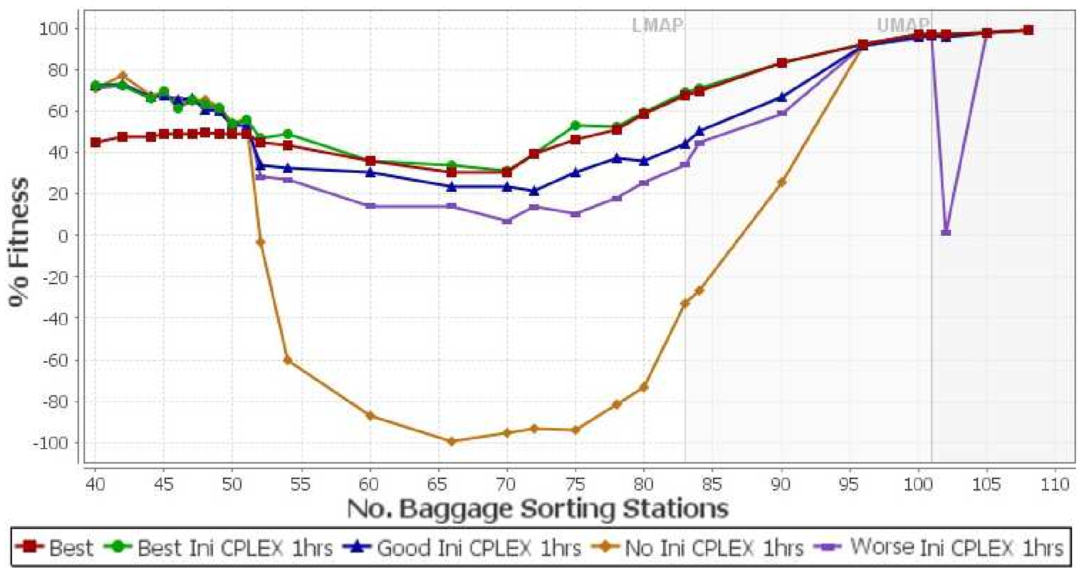


Figure 4.21: CPLEX percentage improvement in fitness for 219 flights and a 3-pier topology.

algorithms which are under consideration, whereas for higher numbers of BSSs up to a point between the LMAP and the UMAP the constructive algorithms provide fitter solutions (for a mere 9 milliseconds run as against 1 hour for CPLEX solutions). Finally for numbers of BSSs near to the UMAP both methods provide solutions with similar or equal fitness. Some factors contributing to the results are firstly that both approaches depend on previous selections, e.g. the selection of a BSS for assignment to a flight may affect possible assignments to the same BSS later, so affecting the solution finally built (an example of this is the improvement of ODT by use of ODTLI), and similarly in CPLEX the solution currently reached will have some bearing on the new solutions, as will be seen next. Secondly, the problem studied has symmetries in regard to the BSSs, for example swapping all assignments between two BSS on the same pier on the same side does not change the fitness. Thirdly, a very low number of BSSs implies that many flights cannot be serviced by a BSS and the number of symmetries are significantly reduced, so the selection of flights for servicing is very important, since this will make a significance contribution to the objective function, so preference to assign flights with smaller service period will imply achieving more assignments than when flights with higher service period are assigned. As the number of BSSs increases the gaps between assignments will also increase, so allowing more assignments, but this will only assign flights with small service periods, since the required gap is smaller, so permitting their assignment. However, a flight with larger service period will require larger gaps, so the number of BSSs will need to increase sufficiently to allow these assignments. Fourthly, once the number of BSSs is sufficient to assign all the flights without the need for a reduction in service time ($N \geq$ UMAP) the effect of the last objective (distance) is not as important as the other two, but there are more opportunities for reduction since more BSSs would be available. In the case of CPLEX the increase of the number of BSSs also implies an increase in the search space, so number of potential solution to investigate for the same time, such that if the initial solution is not sufficiently good much of the search will be spent around that solution before managing to escape to solution of higher quality. In summation, therefore, CPLEX manages to find a good initial solution for a very low number of BSSs, which assigns flights with smaller service times preferentially. As the number of BSSs increases the gaps increase, but are not large enough to permit assignment of flights with large service times (long-haul flights). Thus losing the potential to increase further the assignments but the gaps will eventually become large enough, so long-haul flights can be assigned within those gaps, thus increasing the assignments.

Figure 4.21 shows the effect on the fitness of the final solution obtained by CPLEX when using different constructive solutions as initial solution. The solutions with higher fitness between all of the constructive algorithms considered at each number of BSSs is labelled as 'Best', whereas the less fit constructive solutions are labelled 'Worst' (0%). In Figures 4.21, 4.23 and 4.24 the naming of the results uses the word 'Ini' preceded by a word which specifies the type of constructive solution fed as initial solution to CPLEX. The word 'Best' is used when the best feasible solution is used from all the constructive algorithms run, 'Good' is used when a feasible solution from within the bests but not the best, 'Worse' is used when the worse feasible solution is used from all the constructive algorithms run and 'No' is used when no initial solution was provided to CPLEX. These results show that seeding CPLEX with good solutions obtained from applying the constructive algorithms improves the final solution obtained by CPLEX for when no initial solution is provided ('No Ini CPLEX'), with the exception of when the worst generated solution is fed, which may not always help CPLEX to find better solutions, as can be seen for 102 BSSs in Figure 4.21. For the two areas where CPLEX performs well when compared to the constructive algorithms considered, the use of these constructive solutions does not seem to help CPLEX to find better solutions, indeed the solutions are slightly worse in some cases.

The difference between the solutions provided by the constructive algorithms and CPLEX are not so considerable when the constructive algorithms are used in simpler

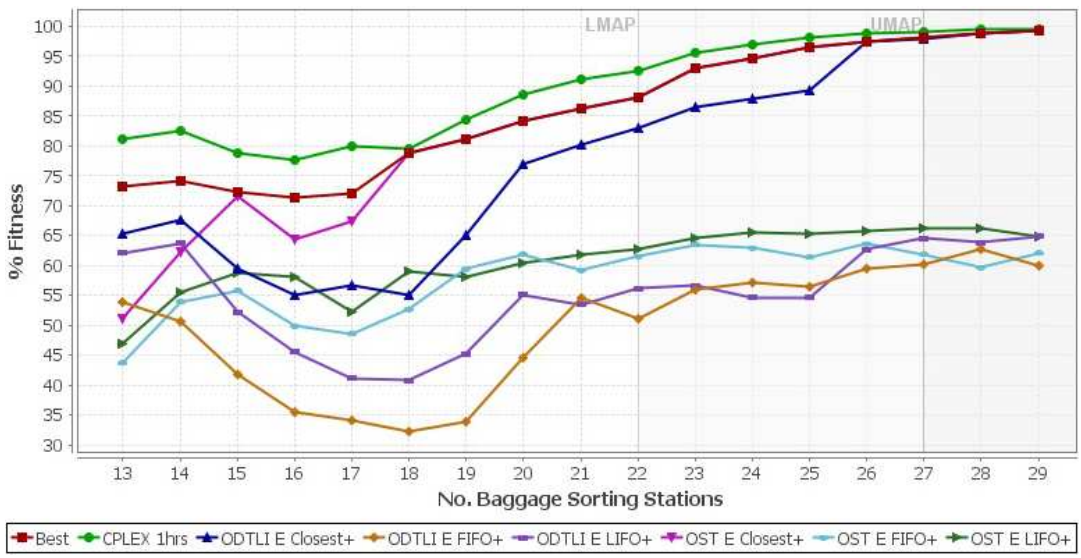


Figure 4.22: Percentage improvement in fitness for 194 flights and a 3-pier topology for Constructive Algorithms and CPLEX solutions.

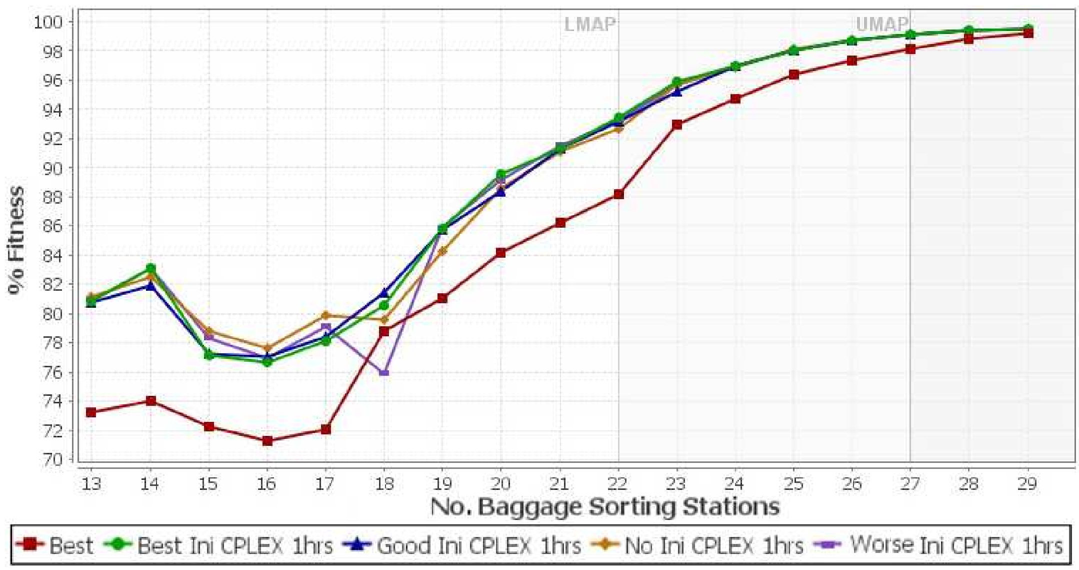


Figure 4.23: Percentage improvement in fitness for 194 flights and a 3-pier topology for CPLEX solutions for different initial solutions.

problems, as may be seen from the BAA's website data for 1^{st} March 2010 and NATS data sets, Figures 4.22 and 4.23. However, it may be seen that the best solution obtained from applying the constructive algorithms has a fitness very close to that of the solution found by CPLEX.

It is also interesting to look at the effect of using the different solutions obtained from the constructive algorithms when reaching the final solution as CPLEX progresses in the search, as can be seen in Figure 4.24. From the empirical results the

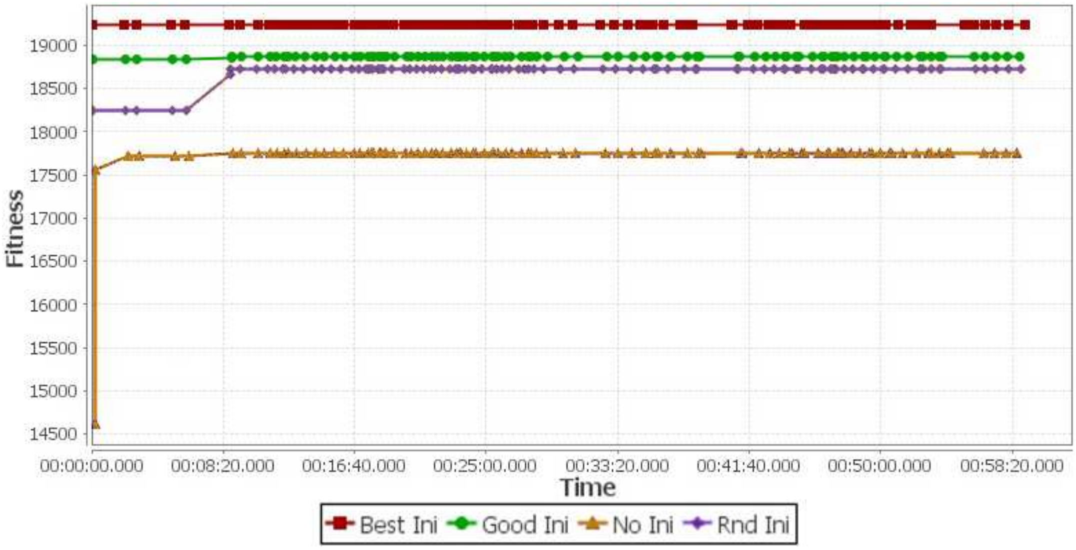


Figure 4.24: Progress fitness for 83 BSSs (LMAP) and 219 flights by CPLEX.

initial solution fed to CPLEX appears to make a difference by allowing CPLEX to reach better solutions earlier. For some numbers of BSSs, the fitness achieved by the final solution is also improved. It can also be seen that some of the constructive solutions are very good, and that CPLEX does not manage to improve on them, as shown in Figures 4.21 and 4.24 when CPLEX is fed with the 'Best' and a good solution obtained by applying the constructive algorithms to the problem.

Evolutionary Algorithms (EAs) are population based algorithms, part of the group of metaheuristics which use the solutions within a population to guide the search to the optimal solution(s). More details about some of these are presented in Chapter 5. For the propose of assessing the quality of the constructive solutions obtained here new experiments were designed and executed for an implementation of the Canonical Genetic Algorithm (CGA) which uses the Evolutionary Computation Java library (ECJ) (Java-based Evolutionary Computation research system, reviewed in Wilson et al (2004)). The operators used are 1-point random crossover and random mutation. An integer encoding of the ABSSAP was also used with randomly generated initial solutions and a population size of 1,000. The average fitness from the solutions obtained by the Genetic Algorithm (GA) implemented together with the constructive algorithm solution fitness is shown in Figures 4.25 and 4.26. The fitness for the GA in such figures corresponds to the average fitness for all of the best solutions found amongst the thirty instances run. These results show that the constructive algorithms used generally provided better solutions than the CGA throughout all of the ranges of numbers of BSSs.

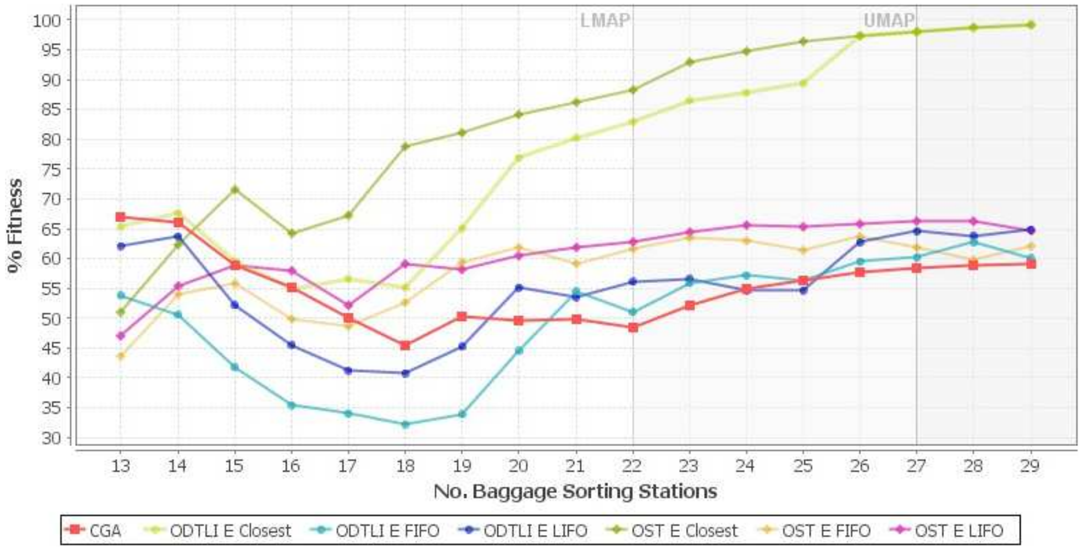


Figure 4.25: Fitness for 194 flights, a 3-pier topology and 48 stands for CGA and some constructive algorithms.

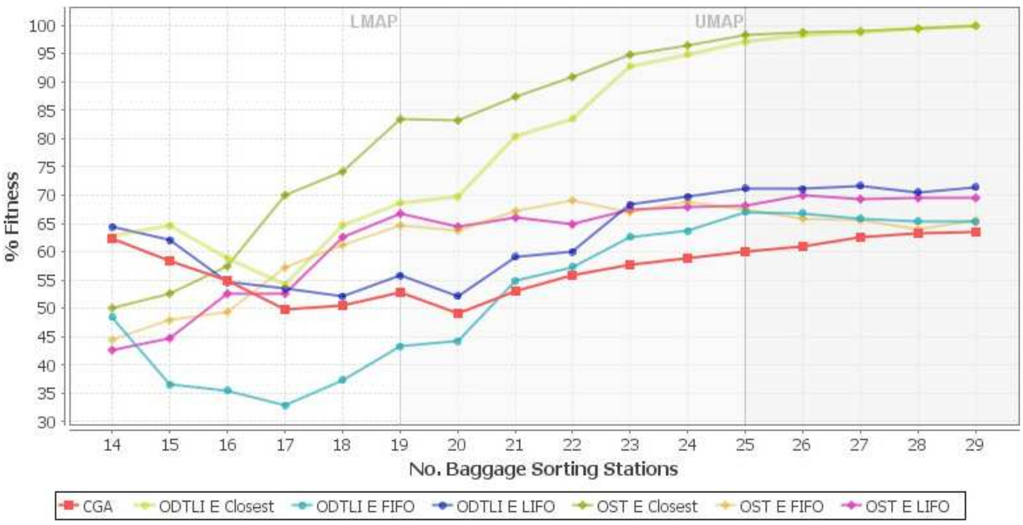


Figure 4.26: Fitness for 163 flights, a 3-pier topology and 48 stands for CGA and some constructive algorithms.

Furthermore, other experiments were run to identify whether the use of these constructive solutions as part of the initial population for the CGA may help the algorithm to reach fitter solutions. The CGA was run thirty times for each quantity of BSSs for a population size of 1,000, using both an initial population of random solutions and the best solutions obtained from applying the constructive algorithm

2,000 times. The 'Best' refers to the best solution amongst all of those generated using the constructive algorithms described in this chapter. The results show that this approach is not detrimental to the algorithm, and in some cases it is even seen to help the CGA to reach fitter solutions, as shown for number of BSSs lower than 17 BSSs in Figures 4.27 and 4.28.

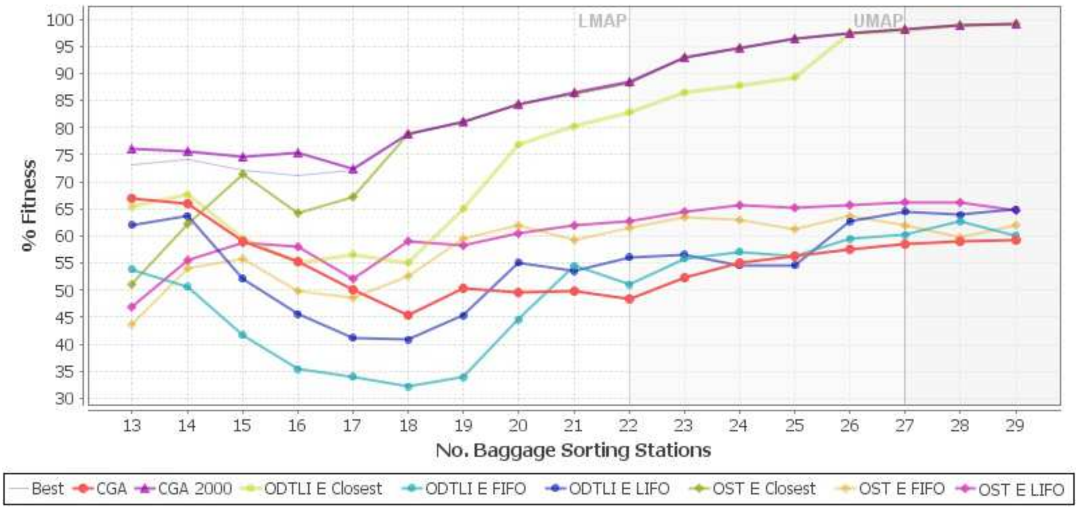


Figure 4.27: Fitness for 194 flights, a 3-pier topology and 48 stands for some constructive algorithms and CGA for different initial population.

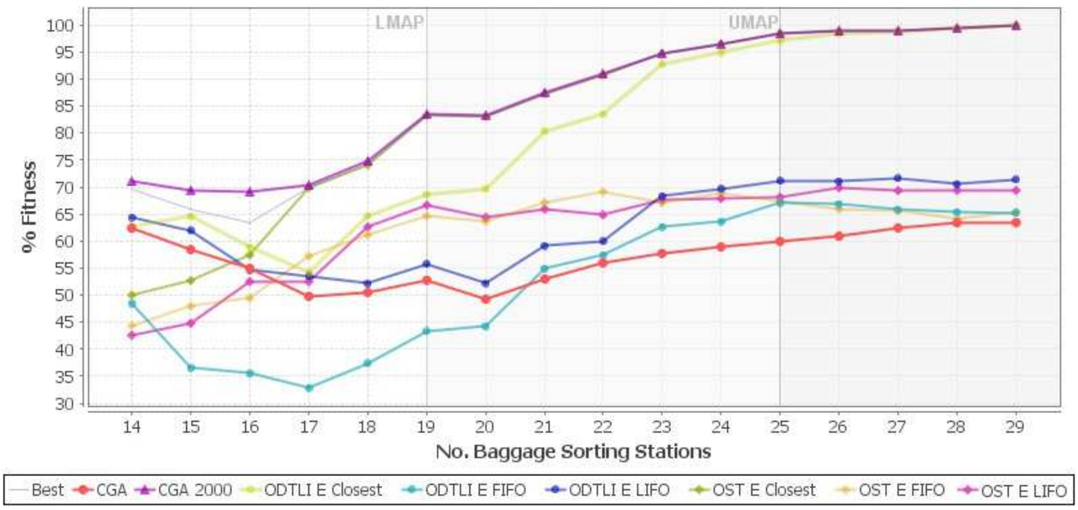


Figure 4.28: Fitness for 163 flights, a 3-pier topology and 48 stands for some constructive algorithms and CGA for different initial populations.

The solutions provided by these constructive algorithms are also used in Chapter 5 to feed initially some metaheuristics which are shown to help finding fitter solutions.

4.5 Conclusions

A framework for constructive algorithms has been presented and has been used to generate some specific constructive algorithms tailored to take account of the airport topology and the position of the assignments. The framework can easily be applied to generate more algorithms where other considerations may be taken into account such as other grouping strategies. For example, the grouping could be based on the type of aircraft (large, medium and small), or the preference of the airline and ground handling contractors, an example of which is presented in Chapter 7 for the AGAP.

When looking at the constructive algorithms, where the grouping considered is by pier, it may be observed that the behaviour of the assignment methods (flight ordering, sorting station assignment algorithm and selection method) depends upon the relationship between the number of flights and the number of sorting stations. The different methods have different effects and can prefer different objectives.

It was observed that a data set with a higher flight density (the number of flights requiring service at any time of day) but fewer flights, was more problematic than one with more flights but a lower density. As expected, the flight density was more important than the total number of flights when determining the number of BSSs required throughout the day. The number of BSSs at which the performance of the algorithms changes was identified in this chapter and it has been noted that these depend upon the distribution of the flights over time.

It has also been noted that the choice of whether or not to allow reductions in service time can affect the relative efficacy of the algorithms. In particular, if reductions in service time are to be permitted, then it may be better to select an algorithm which will not minimise the gap sizes, since these are then less likely to be available to other flights after service time reductions have been applied.

When the above observations are considered together, these effects show that the appropriate algorithm for use depends not merely upon the objective under consideration but also upon the problem characteristics and the relative flight density in relation to the number of sorting stations available.

It has also been seen that the solutions generated are by themselves good when compared to other approaches such as B&B (CPLEX) and the CGA with an overall objective, which does represent a good and realistic preference order between the different objectives. The algorithms have been seen to generate solutions very quickly which, when used as initial solutions in other algorithms, have been seen in many cases to help the search by starting the search from a promising region of the search space.

The aim of this research was not to identify a perfect constructive algorithm which would meet all objectives, but to gain insights into the differing behaviour of the algorithms, particularly when service time reductions are permitted. This research uses these insights in the following chapters to generate better initial solutions for use with perturbative algorithms (particularly Evolutionary Algorithms and Tabu Search, as used in Chapters 5 and 7), improving the quality of the solutions which can be generated within very limited search times. The ability to quickly generate a variety of solutions which have different trade-offs between the objectives has also been particularly useful.

Chapter 5

Evolutionary Algorithms for the Airport Baggage Sorting Station

This chapter investigates metaheuristic approaches to the Airport Baggage Sorting Station Assignment Problem (ABSSAP) (introduced in Chapter 3), defines some of the components of these approaches, and carries out a rigorous analysis of their design and parameters. References to the relevant literature are included for completeness.

This chapter begins with an overview of the Genetic Algorithms (GAs), followed by the description of a new metaheuristic, and a description of several selectors and operators follows there after. Other metaheuristics are then introduced, which will be used in the subsequent section where the proposed approach is rigorously analysed and comparisons made. The final section of this chapter draws some conclusions.

5.1 Overview

A problem is composed of some constraints which must be strictly complied with (known as hard constraints) and other constraints where compliance is desirable (called soft constraints or objectives), see Chapter 3. In order to solve a problem it is necessary to find solutions which comply with both the hard constraints and most or all of the soft constraints. An indication of the compliance with the soft constraints is provided by an evaluation function, sometimes referred to as the fitness function, the results of which give an indication as to the quality or fitness of the solutions.

GAs are one of the methodologies belonging to the population-based model of Evolutionary Algorithms (EAs) presented in Section 2.7.1, based on the Darwin and Wallace (1858) theory of natural selection and Mendelian genetics (Mendel (1865)), which are recognised as the foundation of evolutionary biology. GAs have been used to solve a wide range of airport problems, such as the Airport Gate Assignment Problem (AGAP) in Lim et al (2005), the scheduling of arriving aircraft in Cheng et al (1999), Xiangwei et al (2010) and Hansen (2004), the scheduling of departing aircraft in Bolender (2000) and Caprì and Ignaccolo (2004), the aircraft taxiing in Gotteland and Durand (2003), and the ABSSAP in Ascó et al (2012).

Various different ABSSAP objectives have to be considered, such as maximising assignments, ensuring full service time and allocating preferential positions (Section 3.3.4). Some of these objectives are in obvious conflict (reducing service times in order to service an additional flight for example), thus preventing simultaneous optimisation of each objective.

An encoding of the parameter set for the ABSSAP, presented in Chapter 3, for the Canonical Genetic Algorithm (CGA) was implemented using the Evolutionary Computation Java library (ECJ), used in Section 4.4.6, where a chromosome is composed of the indexes of the baggage sorting station (BSS) assigned to each flight, the flights being ordered by their base service starting time, as shown in Figure 5.1.

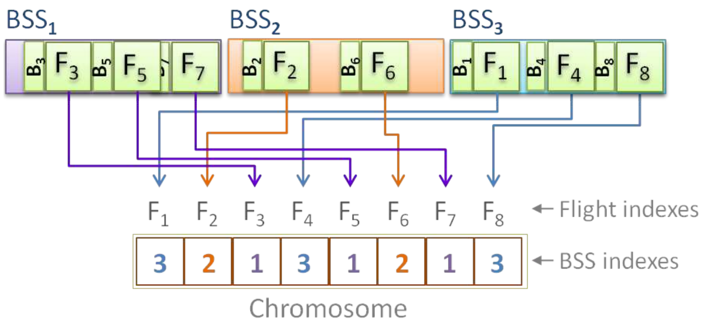


Figure 5.1: An example of encoding for a 3 BSSs and 8 flights.

A different implementation of the CGA, which uses the representation presented in Chapter 3, was also used together with the operators presented in Section 5.4.

Initial studies showed that good initial solutions greatly improve the speed, convergence and quality of the final solutions to the limited time ranges under consideration, as shown in Section 5.7.

The following sections begin by describing the proposed EA with its operators and selectors, followed by a study of the problem, using a fitness function as a single compound objective which represents realistic priorities.

5.2 Steady State Evolutionary Algorithm

A Steady State GA maintains the majority of the population between iterations, only replacing a few individuals at each iteration, a term initially introduced in Syswerda (1989). In the Steady State Evolutionary Algorithm (SSEA) presented here, Algorithm 2, the next population is obtained by applying the population selection operator, some of which are introduced in Section 5.3-1, to the current population. One of the operators is applied to an individual selected from the population by the member selector (Section 5.3-2), this last step being called an iteration and being repeated ℓ times, known as a generation. The newly obtained individuals are added to the population so constituting the current population. This is repeated until the termination condition is reached. In contrast to the CGA, parents and offspring typically coexist such that the parents are also considered for the next generation, which theoretically increases the algorithm's ability to retain information for exploitation in subsequent generations. This creates additional selective pressure towards information already contained in the population. However, keeping the parents does not provide the search with new information since it does not sample new genotypes. The approach may incorporate an aging strategy to ensure that the parents eventually leave the population, thus increasing the chance of offspring contributing to building the next population. Schwefel and Rudolph (1995) incorporated an age by defining a maximum duration of life, so any individual surviving longer than this will be worse than any other which has not reached such limit or has less fitness.

The SSEA is an instance of the Evolutionary Strategies (ESs) which can be described as $(\mu+\lambda)$-ES, $1 \leq \lambda$, where λ may be greater than μ. In the case of the SSEA where the operators used provide only one offspring, when applied, then $\lambda = \ell$. A Steady State GA considering parents in the next generation was presented in Whitley and Kauth (1988); Whitley (1989), which differs from a CGA in that it uses a serial recombination wherein an offspring replaces the lowest ranking individual in the population rather than the parent. Whereas the SSEA may use some or all of the parents in the next generation since the next population in a generation is built by applying the replacement strategy to the current population, which is composed of both the offspring and the parents, so the chance of a parent taking part in the next generation is determined by the replacement strategy used. The SSEA makes use of two selectors, S_p which selects the population which is to take part in the next generation, and S_m which selects the member(s) from within an iteration to which the chosen operator is applied. Likewise, Sokolov and Whitley (2005) follows similar

Algorithm 2: SSEA

Input: Initial population P_0
Input: Number of iterations in a generation $\ell \in \mathbb{Z}^+, \ell > 0$
Input: Operators; $O_j \forall j \in [1 \ldots R]$
Input: Replacement strategies, S_p
Input: Parent(s) selector, S_m

begin
  // Initialise
  $P \leftarrow P_0$; // set initialise population
  repeat
    $P \leftarrow S_p(P)$; // apply replacement strategy to get the new population
    $P_t \leftarrow \emptyset$; // empty population of children
    $i = 0$; // initialise the iterations
    // Run generation
    repeat
      Select an operator, O_k;
      $Q \leftarrow S_m(P, O_k)$; // select parents
      $Q \leftarrow O_k(Q)$; // generate children solutions by applying operator
      $P_t \leftarrow P_t \cup Q$; // add children solutions
      $i = i + 1$; // increment iteration
    until $i = \ell$ *or Termination Condition*;
    $P \leftarrow P \cup P_t$; // merge parents with children solutions
  until *Termination Condition*;
  return P;
end

steps when generating their GA, the main difference to the SSEA is the use of ℓ, two selection processes and the operators being any combination of operators, Figure 5.2. The initial population may also be composed of fewer solutions than the preferred population size. The size should eventually be reached as the new generated solutions are merged with the parent solutions and then the replacement strategy is applied.

For $\ell = \mu$ (the population size) the SSEA algorithm is closer to a CGA but still differs from the CGA in that:

1. The new population to which the replacement strategy is applied is of size $\mu + \lambda$ whereas for the CGA it is λ. Thus not only do parents and offspring coexist in the new population, but also those previous solutions which may not have been selected for the generation of offspring in the current generation.

2. A generation is composed of ℓ iterations in which parents are selected and operators applied to generate the offspring, which together with the previous population, will compose the current population. ℓ does not need to be fixed, and it can be changed as the search progresses, thus providing an additional

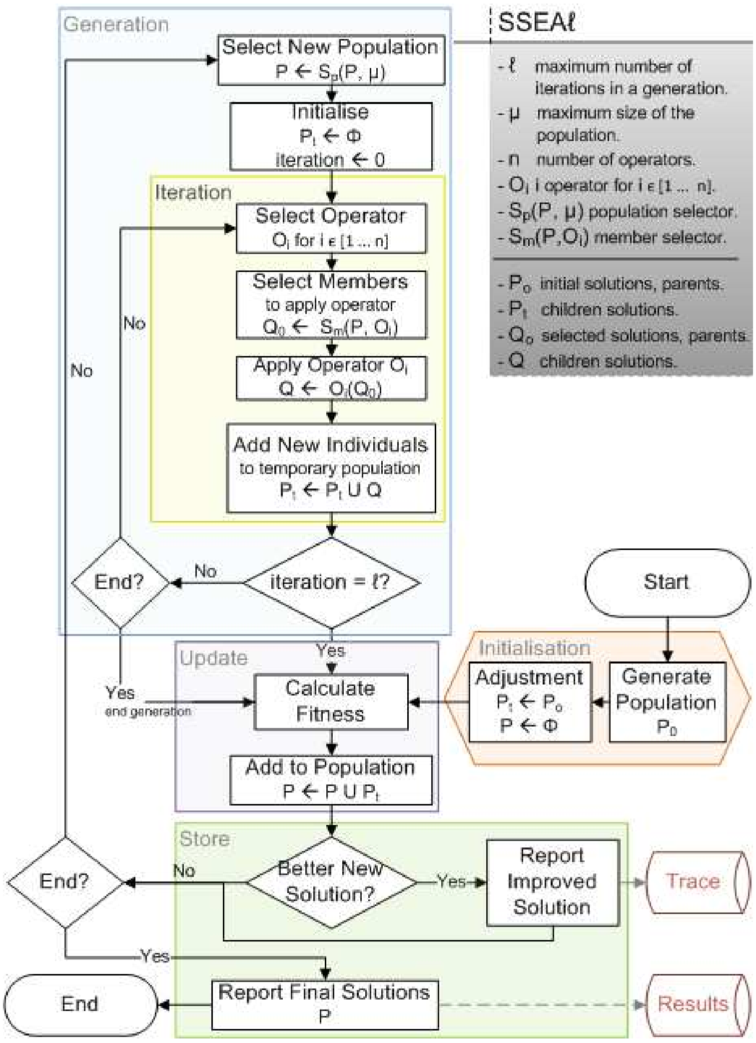


Figure 5.2: SSEA flow chart.

mechanism to control the sampling.

3. Whereas in the CGA reproduction produces two offspring, in the SSEA the reproduction may produce either one or two offspring.

4. In the CGA up to two operators may be applied, namely crossover and mutation. The SSEA does not put any restriction on the operator, so operators may be applied one per iteration or a set of operators in an iteration, as described in the following sections. An operator may be defined which applies a set of sub-operators sequentially to the offspring of the previous operator based on some criterion, such as the probability of a sub-operator being selected. An example of this is where two operators are used one with a probability of 1, so it is always used, and a second operator a probability of being used of 0.1. The

first offspring is always obtained by applying the first operator to the parents from the population, given its probability of 1. This offspring may be further modified by the second operator in order to obtain the final offspring, otherwise where the second operator is not applied the first offspring becomes the final one. If both probabilities are lower than 1 there is a chance of the parent also becomes the final offspring.

5. Each operator has a probability associated with it which represents the chance to be selected, where the overall probability of selecting any of the operators totals 1. In this SSEA any of the operators may be selected at each iteration based on their probabilities.

The implementation of the SSEA algorithm makes use of the problem representation presented in Chapter 3.

5.3 Selectors

The selector methods are responsible for selecting solutions within a population of solutions. Two types of selector are used throughout this thesis which are:

1. **Replacement Strategies**: The replacement strategies generate the new population from the parents and offspring which is used in the following generation. The replacement strategies are used in both CGA and SSEA. They distribute the chance of individuals taking part in the next generation. Normally, the fitter the solution, the more chance it has of being selected for participation in the following generation. A comprehensive analysis of selection schemes used in EAs can be found in Blickle and Thiele (1996).

2. **Parent Selectors**: The member selectors distribute the chance of a given solution within the population taking part in generating new offspring within a generation. Normally, the fitter the solution, the more chance there is of being selected to produce new offspring.

Increase in diversity certainly corresponds to broadening the exploration of the search space, and finding an adjustable balance between exploration and exploitation is the key, March (1991); Levinthal and March (1993). Exploration and exploitation should not be constrained to specific parts of the process, such as only in the early stages of the search, but also be taken into account throughout all the evolutionary process based on the characteristics at each stage.

The selection of solutions for participation in a population is one of the mechanisms for managing diversity, which together with the operators, helps to improve the direction of the search within the domain of solutions into the regions containing solutions with a higher potential.

Some of the terms used are defined below which are based in Baker (1987) and Blickle and Thiele (1996).

- **Selective pressure** is the probability of selecting the best individual compared to the average probability of selection of all the individuals.
- **Bias** is the absolute difference between an individual's normalised fitness and its expected probability of reproduction.
- **Spread** is the range of possible values for the number of offspring of an individual.

Some common selection approaches are presented in Section 2.7.1. There follows an overview of the new approaches proposed.

5.3.1 Stochastic Universal Modified Sampling

The Stochastic Universal Sampling (SUS) may not be appropriate when the order of magnitude of the fitness under study is greater than the difference in the fitness values among individuals, such are the cases studied in this thesis. So Stochastic Universal Modified Sampling (SUMS) is defined in such a way as to provide a greater selection pressure, as shown in Algorithm 3. SUMS provides more selection pressure than SUS and some bias.

A characteristic of the SUMS is that the offsetting of all of the fitness by a constant does not affect those sections of the roulette wheel occupied by each solution as this is not the case for the SUS. For example in the SUS if there are three solutions and their fitness are offset by a very large amount the section occupied by each of the solutions will be close to $\frac{1}{3}$ of the whole roulette wheel. In the case of three solutions (Algorithm 3) with fitness $f_1 = 1003$, $f_2 = 1002$ and $f_3 = 1001$ then the offset $F = f_3 - (f_2 - f_3) = 1001 - (1002 - 1001) = 1000$ and $\sum_{j=1}^{\lambda}(f_j - F) = 6$ so the roulette wheel sections for each solution are $p_1 = \frac{3}{6} = \frac{1}{2}$, $p_2 = p_1 + \frac{2}{6} = \frac{5}{6}$ and $p_3 = p_1 + p_2 + \frac{1}{6} = 1$.

In both versions a single spin of the roulette wheel is made which provides both a starting point and the first individual. The following selections are made by advancing the point in equal step sizes and selecting the individual occupying the section upon

Algorithm 3: Stochastic Universal Modified Sampling

Input: Population P of size λ
Input: Desired population size of μ, $0 < \mu < \lambda$
begin
    // Calculate the two lowest fitness
    $F_{min} = \infty$;
    $F_{min-1} = \infty$;
    for $i = 1 \to \lambda$ **do**
        if $F_{min} > f_i$ **then**
            $F_{min-1} = F_{min}$;
            $F_{min} = f_i$;
        end
        else if $F_{min} > f_i$ **then**
            $F_{min-1} = f_i$;
        end
    end
    $F = F_{min} - (F_{min-1} - F_{min})$;

    // Assign a section to each solution
    $p_0 = 0$;
    for $i = 1 \to \lambda$ **do**
        $p_i = \frac{\sum_{j=1}^{i}(f_j - F)}{\sum_{j=1}^{\lambda}(f_j - F)}$;
    end

    // Initialise
    $P' \leftarrow \emptyset$; // empty next population
    $r_0 = rnd\left[0, \frac{1}{\mu}\right)$; // identify first point
    $i = 1$; // set to first solution in P

    // Select members from the population based on their roulette wheel section
    for $j = 1 \to \mu$ **do**
        $r = \frac{(j-1)}{\mu} + r_0$;
        for $i \to \lambda$ **do**
            if $p_i > r$ **then**
                $P' \leftarrow i$; // add selected solution to next population
                break;
            end
        end
    end

    return P';
end

which the point fell: the process is repeated until all the required individuals have been selected. Some individuals may not be selected where their occupied section is sufficiently small, depending on the starting point.

Both versions of sampling ensure that the observed selection frequencies of each individual are in line with the expected frequencies. So if there is an individual occupying 6.5% of the wheel and it is necessary to select 100 individuals, it is expected, on average, that that individual will be selected between six and seven times. Whereas both SUS and SUMS guarantees this, Roulette Wheel Selection does not make such a guarantee.

5.3.2 Index Selector (ISxy)

This new selector makes sure that no more than a fixed maximum number of fitness duplicates are selected for the next population. This selector requires an integer which corresponds to the maximum number of solutions with the same fitness to keep (x, number of solutions) and a base selector (y, the base selector), one of the selectors presented above, e.g. the Index Selector with the Elitist Selector and a group size of 1 would be represented as $IS1ES$.

The Index Selector is only useful as a replacement strategy, given that as a parent selector it merely selects a very reduced number of solutions.

5.3.3 Range Index Selector (RISxyz)

Empirical results show that when the previous selector ISxy was applied to the AB-SSAP different groups with small differences were generated, which also represented a reduction in diversity, and which diversity may be increased further by changing the ISxy from a unique fitness in each group to a range of fitness per group. This requires a knowledge of group size (x, the maximum number of solutions to be kept within a range), a base selector (y, the base selector), and an indication of the fitness range (z), e.g. the Range Index Selector with Elitist Selector (y = ES), a group size of 1 (x = 1) and fitness range of 50 (z = 50) which may be represented as $RIS1ES50$. For $RIS1ES50$ and a maximisation problem, if the group having a fitness range from 1000 to 1050 already contains a solution with a fitness of 1000, and a new solution is to be added to the population with a fitness of 1010 then the solution of a 1000 is removed and the new solution is introduced into the group in its place, given that x = 1. The selection within a group uses a greedy approach.

Many of the selection approaches presented are not suitable for when only one individual (solution) is required, as is the case for the Index Selection (ISxy) and Range Index Selector (RISxyz), given that in those cases they are equivalent to the underlying selection approach, e.g. the Index Selection with Elitist Selection (ISxES) is the same as the Elitist Selection (ES). Such is the case for the mutation operators (Section 5.4.1) where the Parent Selectors have to select only one parent solution. Similarly, some of the classic selection methods such as SUS, Roulette Wheel Member

Selection (RWMS) and Tournament Member Selection (TMS) are equivalent when just one parent solution has to be selected.

5.4 Operators

Two main groups of operators are reviewed in the following sections: Mutation and Crossover. Both of these are described below.

5.4.1 Mutation

The operators introduced here are local search (guided mutation) operators which generate feasible solutions.

All flights which have not been assigned to a sorting station are assigned to the 'dummy' sorting station. Some operators can switch flights between the real and dummy sorting stations.

When a sorting station is to be selected, the roulette wheel selection method is used where every sorting station has the same probability of being selected.

When a time has to be determined (for instance for the start or end of a time range) a uniform random variable is used so that any time within the time range of the flights under consideration has an equal probability of being chosen.

Dummy Single Exchange Mutation Operator

The Dummy Single Exchange Mutation Operator (DSEMO) is equivalent to the 'Apron Exchange Move' used by Ding et al (2004) and Ding et al (2005). A solution is selected from the population by the member selector (S_m) then a new solution is built by moving a flight from the 'dummy' sorting station from this solution to a randomly selected sorting station, potentially moving another flight back onto the 'dummy' sorting station when it can no longer be fitted in.

This operator may increase the number of assignments where the operation does not move a flight back onto the 'dummy' sorting station.

It is necessary that some flights be unassigned in the parent solution. So when full assignment has been attained for the given number of BSSs this operator clearly will not provide a new solution.

Dummy Single Move Mutation Operator

In the Dummy Single Move Mutation Operator (DSMMO) a random unallocated flight and initial target sorting station are chosen and an attempt is made to assign the flight to the selected sorting station. If the assignment cannot be achieved then the next sorting station is selected and the process is repeated until the flight is assigned or no more sorting stations are available, in which case the flight is returned to the 'dummy' sorting station. When maximum assignments have been attained for the given number of sorting stations, then this operator obviously will not provide a new solution.

Multi Exchange Mutation Operators

A set of sorting stations is randomly selected within a random time period, t_{rs} to t_{re}. All assignments where the base service durations are entirely within the time period are then moved to the next sorting station in the set, as shown in Figure 5.3, provided they fit. This operation is repeated from one sorting station in the set to the next, until they have all been covered. Flights which cannot be moved are added to the set of flights which will be considered for assignment at the end, potentially reducing the number of flights which otherwise would not be assigned. These operators generalise the 'Interval Exchange Move' which was presented by Ding et al (2005), and cannot increase the number of assignments.

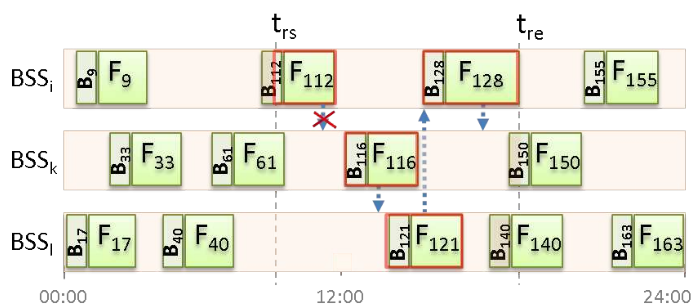


Figure 5.3: Example of multi exchange between 3 BSSs.

Three variants have been developed:

1. Multi Exchange between a Fixed Number of Resources (MEFNRn): The number of sorting stations between which flights are exchanged is fixed at n, where $2 \leq n \leq N$.

2. Multi Exchange between a Random Number of Resources (MERNRn): The number of sorting stations between which flights are exchanged is randomly chosen each time, between 2 and n, where $2 < n \leq N$.

3. Multi Exchange between a Range Random Number of Resources (MERRNRxy):

The number of sorting stations between which flights are exchanged is randomly chosen each time, between x and y, where $2 \leq x < y \leq N$.

Multi Exchange By Pier Mutation Operators

These operators are a specialised case of the Multi Exchange Mutation Operators, where the sorting station selection element ensures that no two consecutive sorting stations in the set are on the same pier. The idea is to improve the distance objective by encouraging the movement of assignments between piers.

Once again, this operator cannot increase the number of assignments. As for the Multi Exchange Mutation Operators, three variants have been created:

1. Multi Exchange By Pier between a Fixed Number of Resources (MEBPFNRn): The number of sorting stations to exchange flights between is fixed at n, where $2 \leq n \leq N$.
2. Multi Exchange By Pier between a Random Number of Resources (MEBPRNRn): The number of sorting stations between which the flights are exchanged is randomly chosen each time, between 2 and n, where $2 < n \leq N$.
3. Multi Exchange By Pier between a Range Random Number of Resources (MEBPRRNRxy): The number of sorting stations between which the flights are exchanged is randomly chosen each time, between x and y, where $2 \leq x < y \leq N$.

Range Multi Exchange Mutation Operators

These are the same as the Multi Exchange Mutation Operators, however they add an additional feasibility recovery step when flights cannot be moved. Flights which cannot be moved are added to the set of flights which will be considered for assignment to the next sorting station, potentially reducing the number of flights which will not be assigned in the end. Finally, flights which have still not been moved are again considered for assignment to the other sorting stations in the set, except the last one, once again potentially reducing the number of flights which otherwise would not be assigned, in the same way as the Multi Exchange Mutation Operators, Figure 5.4.

Once again, this operator cannot increase the number of assignments. Three variants have been developed:

1. Range Multi Exchange between Fixed Number of Resources (RMEFNRn): The number of sorting stations between which to exchange flights is fixed at n, where $2 \leq n \leq N$.

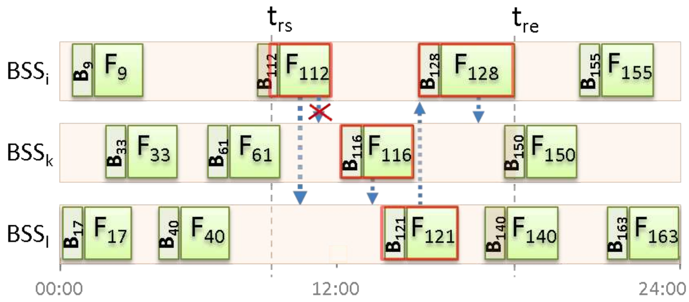


Figure 5.4: Example of range multi exchange between 3 BSSs.

2. Range Multi Exchange between Random Number of Resources (RMERNRn): The number of sorting stations between which to exchange flights is randomly chosen each time, between 2 and n, where $2 < n \leq N$.

3. Range Multi Exchange between Range Random Number of Resources (RMERRNRxy): The number of sorting stations between which to exchange flights is randomly chosen each time, between x and y, where $2 \leq x < y \leq N$.

Range Multi Exchange By Pier Mutation Operators

These are a specialised version of the Range Multi Exchange Mutation Operators, which ensure that consecutive sorting stations in the set are not on the same pier, to encourage the movement of flights between piers, so potentially improving the distance objective. These operators cannot increase the number of assignments. As for the Multi Exchange Mutation Operators, three variants have been created: Range Multi Exchange By Pier between Fixed Number of Resources (RMEBPFNRn) and with Random Number of Resources Range Multi Exchange By Pier between Random Number of Resources (RMEBPRNRn) and Range Multi Exchange By Pier between Range Random Number of Resources (RMEBPRNRxy).

The Multi Exchange Mutation Operators may also be extended by using multiple points in time instead of two points in time (a time range). However, this will also increase the complexity and time required to execute the operations, and equates to several executions of the current implementation and was not therefore investigated.

5.4.2 Crossover

The crossover operators involve the generation of new solutions from multiple parents. Each parent will be chosen using the Parent Selectors (S_m) and multiple child solutions may be generated in each case.

2-point Crossover

In the 2-point crossover (C2P), two points in time are randomly selected within the time range of the flights, to generate a time window. All flight assignments which lie within this time period, for all of the sorting stations in each solution, are exchanged between the parent solutions, as shown in Figure 5.5. The flight timings are identical across all solutions, except that the flights in the exchanged region may overlap flights which are not exchanged in the case of some sorting stations. Such overlapping flights in the exchange region are reassigned to other sorting stations where possible, otherwise they are assigned to the dummy sorting station (i.e. are unassigned).

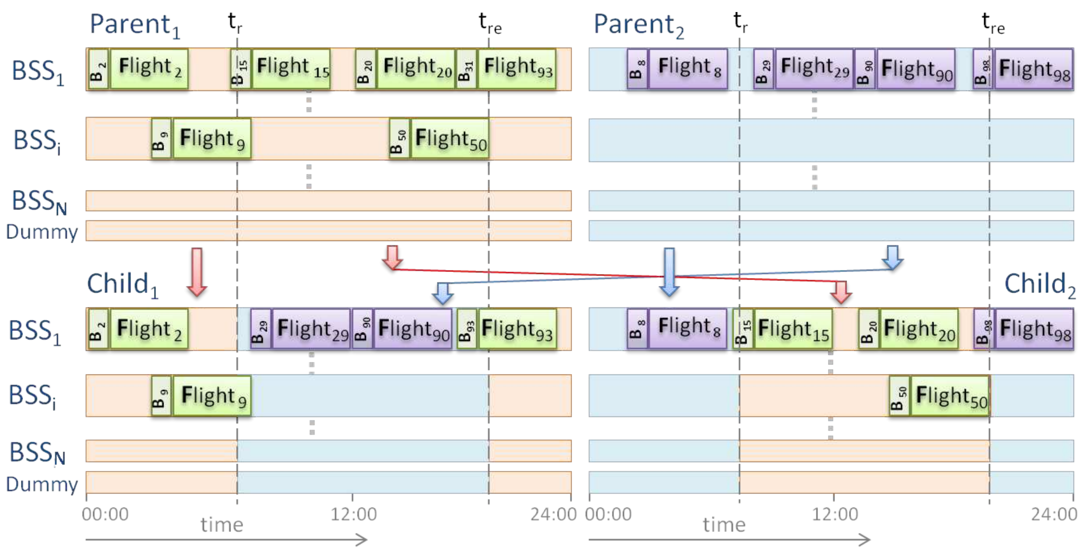


Figure 5.5: 2-point crossover.

Whereas in the classic crossover a chromosome is divided into 3 sections, here the chromosome is divided into $3 * N$ sections which correspond to 3 sections per sorting station.

1-point Crossover

The 1-point crossover (C1P) is a specific case of the above 2-point crossover, where the window extends to the end time of the solution, Figure 5.6.

In my representation, 1-point crossover is a special case of 2-point crossover (n = 2, number of points), where the second point corresponds to the end of the chromosome. This can be better understood if the chromosome is represented as a loop, Figure 5.7.

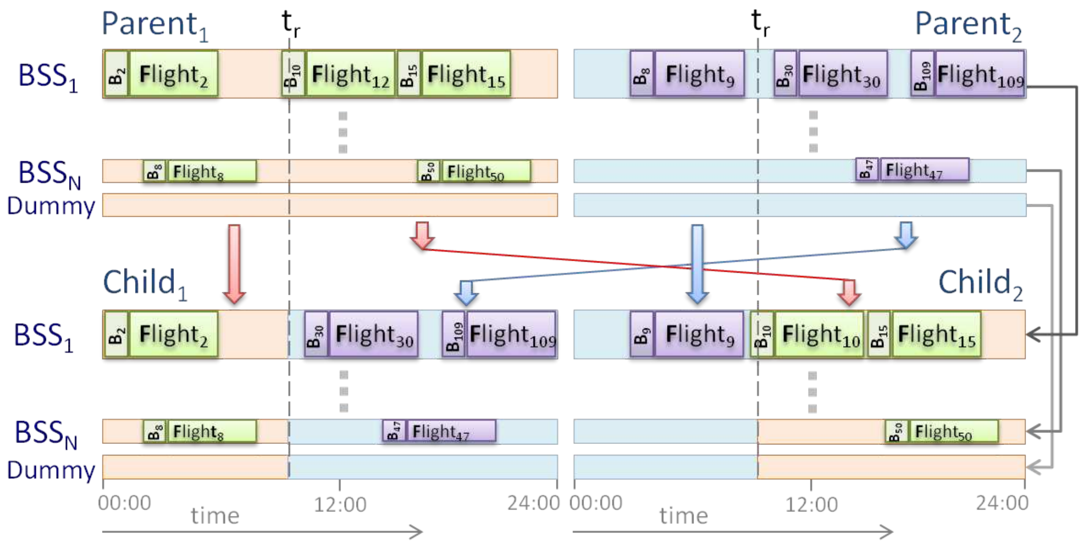
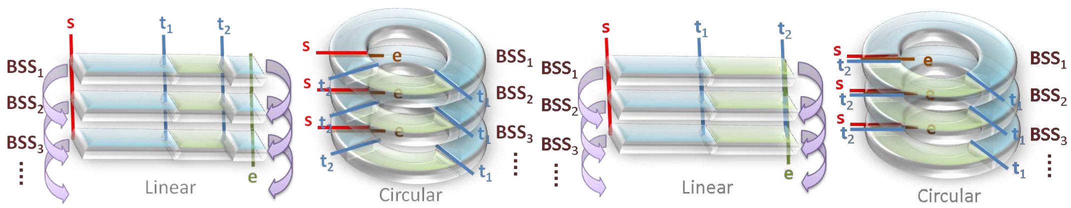


Figure 5.6: 1-point crossover.



a 2-point cross over

b 1-point cross over

Figure 5.7: Crossover representations where 's' refers to the start and 'e' to the end.

n-point Crossover

The n-point crossover (CnP) may use $n+1$ solutions from the population, where n refers to the number of cuts. The full time range is divided into $n+1$ sections and multiple new solutions are obtained by merging the consecutive sections between the different parents, as shown in Figure 5.8. This recombination may leave some flights unassigned, which may be assigned directly to the dummy sorting station (fictitious sorting station) or it could be attempted to repair the solution by assigning them to any available sorting station. An extension to 2-point crossover is the n-point crossover which divides the chromosome into $(n+1)*N$ which equates to $n+1$ sections per sorting station.

Using n-point crossover with $n+1$ parents can provide up to $(n+1)!$ different children. Eiben et al (1994), Eiben et al (1995), Tsutsui and Jain (1998), and Eiben

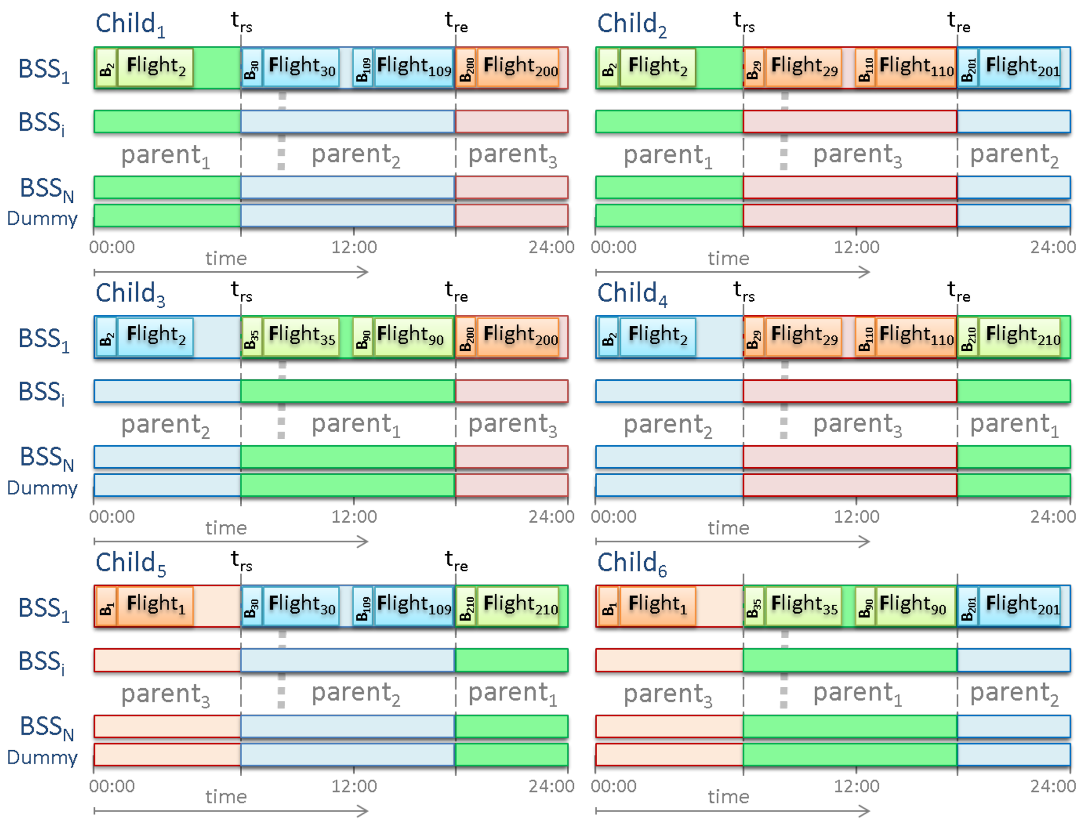


Figure 5.8: All children for 2-point crossover and 3 parents.

(2003) studied the effect of using multiple parents and multiple crossover points and observed that the increase in the success rate is not merely a consequence of using multiple crossover points, leading to the conclusion that using more parents does increase GA performance.

1-point Serial Crossover

The 1-point serial crossover (SC1P) is a different implementation of a crossover operator and may be simpler to understand by representing the problem as a continuous list of BSSs where the crossover cut(s) is in this continuous list, instead of within each BSS as seen in the previously presented crossover operators. The 1-point serial crossover operator is illustrated in Figure 5.9 for the ABSSAP. When the cut(s) has to be determined, a comparison of both parents is made to find the first and last differences in their assignments within the representation, which may be used to restrict the selection of the cut(s). This implementation of a crossover operator is closer to that which is commonly presented in the literature as a 1-point crossover operator, and it is different to that previously introduced in this section.

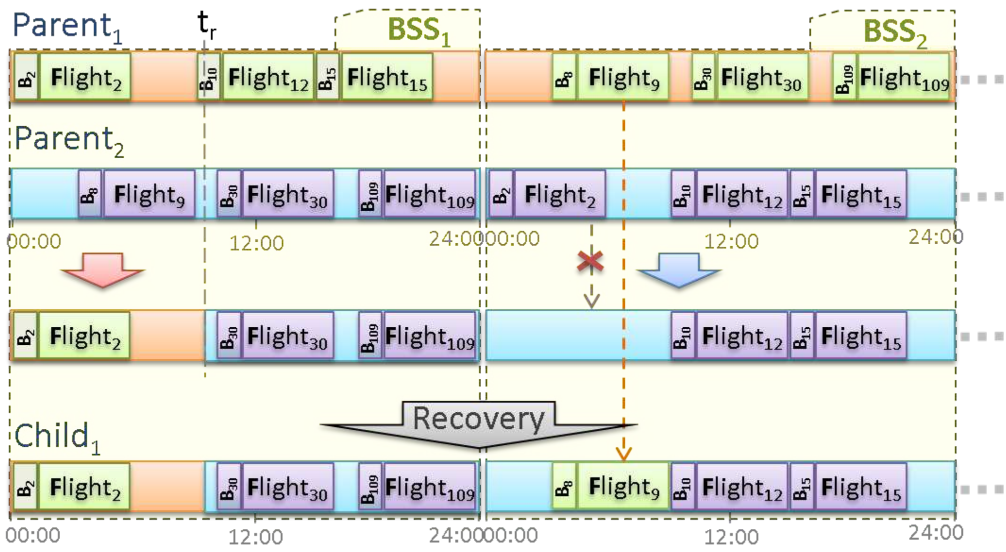


Figure 5.9: 1-point serial crossover.

Figure 5.10 shows a simple example where the same parent solutions are used in a 1-point crossover and a 1-point serial crossover side by side. When considering two parents with full assignment, the cut in time (t_{rs}) in the 1-point crossover (C1P) breaks the assigned flights into two groups, each of which contains the same flights for both parents, whereas this is not the case for 1-point serial crossover (SC1P), as shown in Figure 5.10, where flight '3' is on a different side of the cut in the parents. This means that in the case of SC1P it is required to check the assignments after the cut (t_{rs}) from the second parent to make sure that they have not been already assigned to the first side (from the first parent). Flight '3' was already assigned to the offspring from the first parent and therefore cannot again be assigned from the second parent, as shown in Figure 5.10. So 1-point crossover is simpler to implement than 1-point serial crossover.

Furthermore, this implementation could be easily extended to n points.

Holland (1975) argued that, based on the schema theorem to minimise schema disruption, 2-point serial crossover is better than 1-point serial crossover. Although our results show that in some instances 1-point serial crossover provides better solutions than 2-point serial crossover, in general 2-point serial crossover performs best overall. Nevertheless, the schemata theorem is based on a binary representation of the chromosome and binary operators, which differ from the representation and operators presented here, so its application is of limited interest.

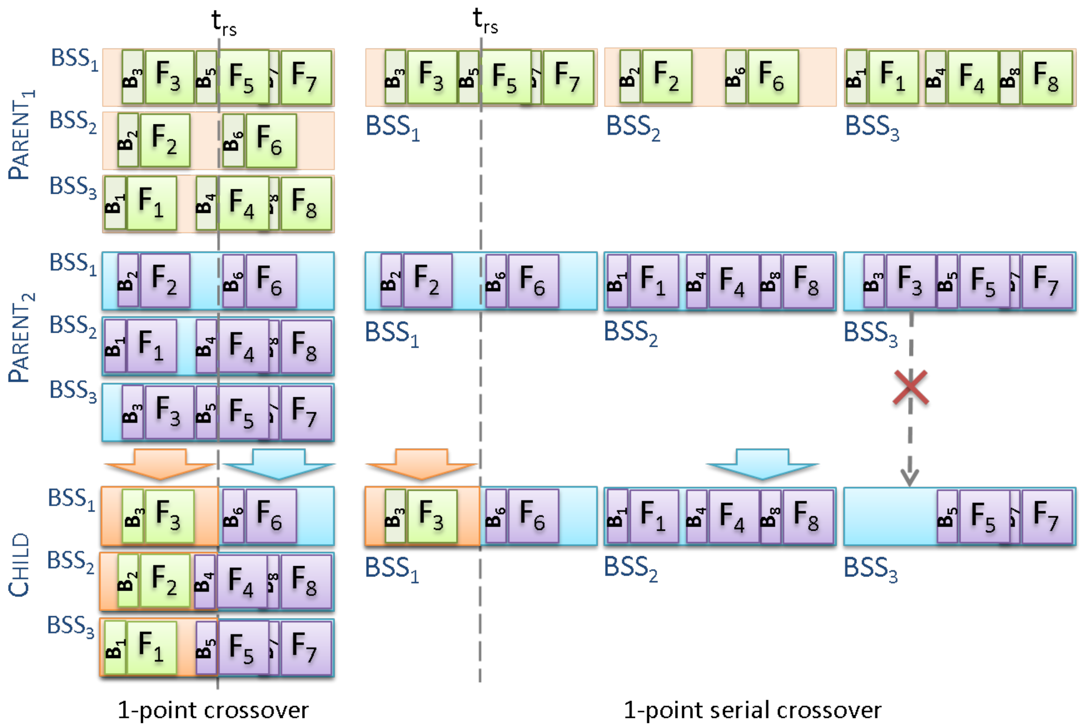


Figure 5.10: Example of 1-point crossover and 1-point serial crossover.

5.4.3 Combination of Operators

Based on how the operator is selected, the types which are of interest are described in the following subsections. It is noted that the operators could be used in complex ways by combining these different types with different parameters.

Probability Single Multi Operator

The Probability Single Multi Operator (PSMO) is composed of several sub-operators (which are described in Section 5.4), each one of which has a specified probability of being used for the creation of new population members, Algorithm 4. The combined probabilities across all operators must add up to 1.

As an example, consider a PSMO operator which uses the operators C1P (with a 0.1 probability of being selected) and Multi Exchange between a Fixed Number of 3 Resources (MEFNR3) (with a 0.90 probability of being selected), which may be represented as PSMO(C1P:10+MEFRN3:90). Given that the total probability must amount to 1, it is not necessary to specify the probability for the last sub-operator, so the representation may also be PSMO(C1P:10+MEFRN3).

Algorithm 4: Probability Single Multi Operator.

Input: Member Selector S_m
Input: Population of solutions P
Input: Operators; $O_k\ \forall\ k \in [1 \dots R]$
Input: Probability for operators p_k, $0 < p_k \leq 1 \forall k \in [1 \dots R]$ and $\sum_{k=1}^{R} p_k = 1$

```
begin
    // Initialise
    P_0 ← ∅; // empty list of children
    r = rnd[0...1);
    k = 1; // initialise sub-operator index to first operation
    p = p_1;

    // Select operator
    while k < R and r > p do
        k = k + 1; // next operator
        p = p + p_k;
    end
    Q ← S_m(P, O_k); // get parent solutions for operator O_k
    P_0 ← O_k(Q); // build children by applying operator to parents

    return P_0; // return the obtained children
end
```

Sequential Operator

Considering the way the CGA operates, where a crossover operator may be applied to the parents with a high probability and its children may be further modified by applying a mutation operator, the operators may be extended by defining a new operator composed of multiple sub-operators, which are applied sequentially with a given probability ($0 < p \leq 1$), Algorithm 5. This new operator is called the Sequential Operator (SO) herein.

As an example, consider the operators C1P with a selection probability of 1 and the MEFNR3 with a probability of selection of 0.01, which may be represented as SO(C1P:100,MEFNR3:1), where a 1-point crossover is always applied to generate the intermediate children for which there is a small probability of 0.01 for application of the MEFNR3 operator in order to generate the final children solutions.

5.5 Tabu Search

In this section the Tabu Search (TS) heuristic is introduced which is later used with the previously introduced operators and the results are compared with those from the SSEA and the CGA in Section 5.7. It uses the problem representation presented

Algorithm 5: Sequential Operator.

```
Input: Member Selector S_m
Input: Population of solutions P
Input: Operators; O_k ∀ k ∈ [1...R]
Input: Probability for each operator p_k, 0 < p_k ≤ 1 ∀ k ∈ [1...R]
begin
    // Initialise
    P_0 ← S_m(P,O); // select parents based on operators

    // Build children
    for k = 1 → R do
        r = rnd[0...1);
        if r < p_k then
            Q ← P_0; // previous children as parent solutions
            P_0 ← O_k(Q); // applying operator to the parent solutions
        end
        i ← k + 1; // next sub-operator
    end

    return P_0; // return the obtained children
end
```

in Chapter 3.

A TS is a metaheuristic which employs a local search which uses a solution to generate a neighbourhood of solutions. The solutions from the neighbourhood are checked in the hope of finding an improved solution. A local search may get stuck within areas of the search space where the neighbourhood is equally fit, so memory structures which describe the neighbourhood visited are incorporated to avoid using that previously-visited solutions/regions again, Glover (1989), Glover (1990) and Burke and Kendall (2005).

The implementation of the TS used in this chapter generated the neighbourhood (also called local walk) using the mutation operators described in Section 5.4.1, which constitute the list of candidate solutions. The fittest non-tabu solution in the candidate list is adopted as new current solution and is also added to the tabu list, as shown in Algorithm 6. Once the tabu list is full, one solution is removed from the tabu list to leave space for the new tabu solution.

5.6 General Experiments Information

A summary of some of the typical values for the different parameters used in the following experiments is shown in Table 5.1.

Algorithm 6: Tabu Search

```
Input: Initial solution Q_ini
Input: Maximum size of the tabu list ℓ_t ∈ ℤ+, ℓ_t > 0
Input: Maximum local walk ℓ_l ∈ ℤ+, ℓ_l > 0
Input: Operators; O_1 : p_1, O_2 : p_2, ··· , O_R : p_R with 0 < p_j ≤ 1∀j ∈ [1 . . . R] and
       ∑_{j=1}^{R} p_j = 1
Input: Fitness function f(Q)
begin
    // Overall initialisation
    Q_c ← Q_ini; // set current solution to initial solution
    Q_b ← Q_ini; // set best solution to initial solution
    P_t ← ∅; // empty tabu list

    // Search space of solutions
    repeat
        // Local search initialise
        Q_next ← ϕ; // set to no next solution
        walk_iteration ← 0; // initialise iteration counter

        // Local search - ℓ_l times
        while walk_iteration < ℓ_l and no Termination Condition do
            walk_iteration ← walk_iteration + 1;
            Select randomly an operator, O_k; // use roulette wheel
            Q ← O_k(Q_c); // apply operator

            // Update if not tabu
            if Q ∉ P_t and Q_next = ϕ or f(Q) > f(Q_next) then
                Q_next ← Q;
            end
        end

        if Q_next ≠ ϕ then
            Q_c ← Q_next; // set next solution as the current solution

            // Update the best
            if f(Q_c) > f(Q_b) then
                Q_b ← Q_c; // update best solution
            end

            // Add to tabu list
            if |P_t| = ℓ_t then
                P_t ← P_t\Q_0; // remove earliest
            end
            P_t ← P_t ∪ Q_c; // add current to tabu list
        end
    until Termination Condition;

    return Q_b;
end
```

Parameter	Value	Comments
Tournament size	2	Tournament selection
Trails / Runs	30	Number of runs per experiment
Significance level	0.05	Mann-Whitney U tests were carried out to ascertain the statistical significance.
Fitness weights	$W_1 = 90$ $W_2 = -0.008$ $W_3 = -1$	For the calculation of the weights see Section 4.4.6

Table 5.1: Default parameter values.

Initial solutions were obtained by running the constructive algorithms presented in Chapter 4.

Unless it is mentioned the parameters presented here refer to all the following experiments for the ABSSAP.

5.7 Results

The described algorithms are applied to the ABSSAP and their results are compared and analysed in this section for both the data sets obtained from British Airports Authority (BAA)'s website and those provided by NATS which are also used in Chapter 4. A fitness function composed of the weighted sum of the different objectives was used to guide the search within the algorithms.

Initial results from experiments executed for BAA's website data sets show that the SSEA presented in this chapter provides better solutions than those obtained by CPLEX and Gurobi for the running times considered. These experiments also highlighted the need to have access to a large quantity of Random Access Memory (RAM) given how memory hungry both commercial solvers CPLEX and Gurobi are, making it necessary to run them on a 64bit machine to be able to use more RAM. Whereas the SSEA was run on both 32bit and 64bit Operating Systems (OSs), as the original results were obtained using a 32bit Windows XP with 1.93GB RAM and 2.99GHz Inter 2 Duo CPU. An initial run of duration 1 hour was executed followed by another one of 24 hours to identify if the exact method could find the optimum and compare the fittest solution obtained with those obtained by the SSEA. Also the best upper bound obtained from each run were used to help to get an idea of the quality of the solutions obtained from the different algorithms used in the following sections. All the Gurobi parameters used were the default ones with the exception of the time, which was limited to 1 hour and 24 hours in the two initial runs, and the parameters values used for CPLEX are presented in Table 5.2. Multiple runs were executed to enable the SSEA to take account of the random characteristics of the algorithm with a PSMO composed of 0.2 MEFNR3, 0.2 Range Multi Exchange between Fixed Number of 2 Resources (RMEFNR2), 0.15 C1P and 0.45 DSEMO (only one of the sub-operators will be used at each iteration) with ES replacement strategy, and the results are shown in Figures 5.11.

The SSEA improves quickly upon the initial solutions used, reaching solutions fitter than those obtained by Gurobi. Further initial experiments were conducted

Parameter	Value	Comments
NodeFileInd	3	Node file on disk and compressed
WorkMem	128	Memory in MB
NodeSel	1	Best-bound search
VarSel	3	Strong branching
TiLim	3600 and 86400	Time in seconds to end the run

Table 5.2: CPLEX none default parameter values used.

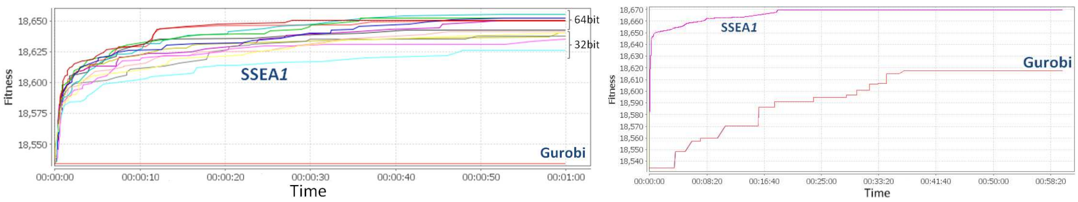


Figure 5.11: Progress for a 3-pier topology, 48 stands, 78 BSSs and 219 flights (H1T091216).

between the SSEA, CGA and TS with the parameters values in Table 5.3. The results for these experiments, which are presented in Figure 5.12, show also that SSEA performs better than the other metaheuristics considered.

Algorithm	Parameters					
	Name	Value	Name	Value	Name	Value
SSEA	Population size	10 and 30	Replacement Strategy	ES	Operator	MEFNR3
	Tournament size	5				
CGA	Population size	10 and 30	Replacement Strategy	ES	Operator	0.99 C1P and 0.1 MEFNR3
TS	Walk size	10	Tabu list size	30	Operator	MEFNR3

Table 5.3: Parameter values used with 30 runs per experiment.

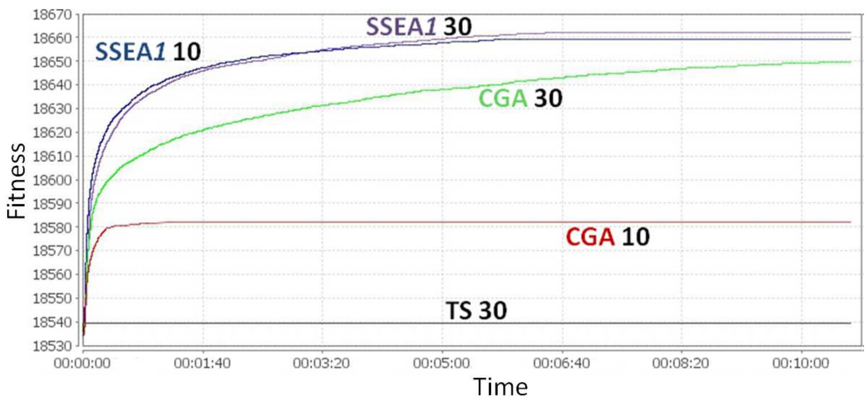


Figure 5.12: Progress for a 3-pier topology, 48 stands, 78 BSSs for 219 flights (H1T091216) and different heuristics.

In general the results obtained show improvements in fitness, as shown in Figure 5.13. Better results were obtained when other Replacement Strategies were used, which are presented in the following sections.

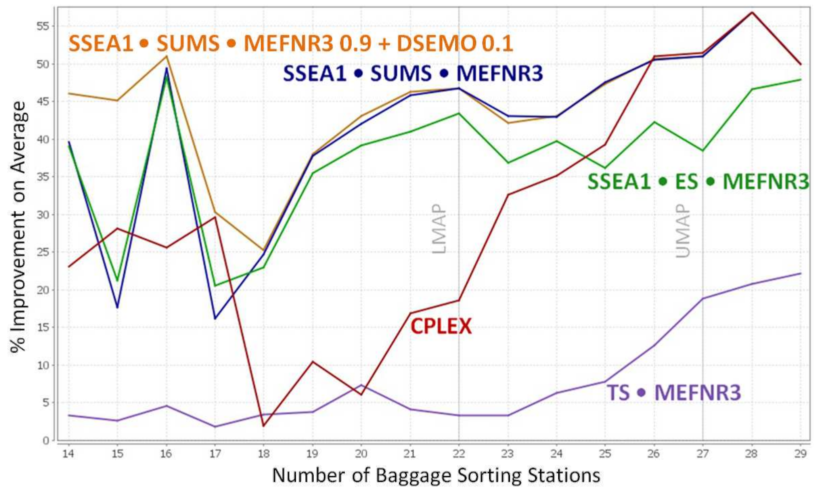


Figure 5.13: Average fitness for a 4-pier topology, 46 stands for 194 flights (H1T091216) and different heuristics.

These results show the potential of the SSEA for obtaining good solutions even on short runs.

In the next sections the experiments and their results are presented which were obtained when studying the different parameters part of the SSEA.

5.7.1 Initial Solutions

Experiments were initially conducted to evaluate the influence of the initial population of solutions in reaching better solutions when using good solutions as initial population. The latter have been obtained by applying the constructive algorithms presented in Chapter 4, to a data set of 219 flights. The operator used is a PSMO composed of the following sub-operators, each with its own probability of being used; 0.2 for RMEFNRn, 0.2 for Dual Exchange Mutation Operator (DEMO), 0.15 for 1-point crossover and 0.45 for DSEMO, for a population size of 10 solutions for population based algorithms and 78 BSSs (lower than the Lower Maximum Assignment Point (LMAP)). Given that for 78 BSSs full assignment is not possible then use of the DSEMO should help reaching other areas of the search space, thereby improving the solutions obtained. The solutions which do not have maximum assignment may further increase the number of assignments by applying the DSEMO.

Maximum assignment is achieved where no buffer time is considered, and no restriction is applied as to where the flights may be assigned when ordering the flights by departure time: this is used to generate some of the constructive solutions. The progress of the search is used here for the different initial solutions being considered, in order to illustrate their contribution in reaching better solutions, as shown in Figure 5.14. This provides a view of the Steady State Evolutionary Algorithm with

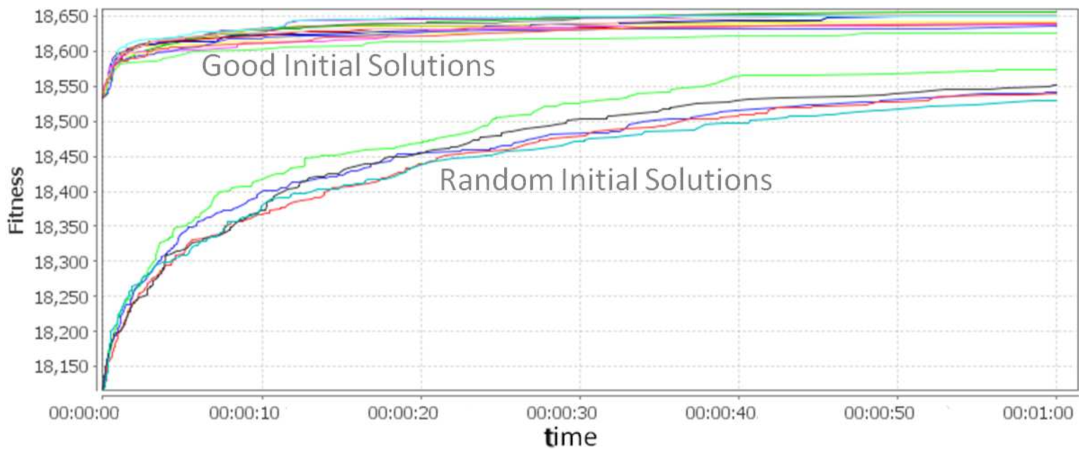


Figure 5.14: Progress in fitness of solutions when run with and without a initial random population for SSEA1, a 3-pier topology, 78 BSSs, 48 stands and 219 flights (H1T091216).

$\ell = 1$ (SSEA1) behaviour, and shows that the algorithm managed to improve on the already good solutions provided as initial solutions, but not as much as when the initial solutions are of lower fitness. This is as expected given that there is more leeway to improve on the solutions, but the final fitness of the best solutions is still less than those obtained when good solutions are used. Furthermore the solver Gurobi was run for one hour, Figure 5.15, when no initial solution was provided and when an initial constructive solution (the best of those used for the SSEA1) was used, which showed Gurobi took over 2 minutes to find a feasible initial solution, when no initial solution is provided. Then quickly improved on this, but still does not manage to reach a fitness such as those reached when a good constructive initial solution is provided, as is the case with the SSEA1 but at a lower rate. The final solution fitness in both Figures shows that SSEA1 provides fitter (better) solutions than those provided by Gurobi, with SSEA1 also improving on Gurobi when no good initial solutions were used.

In summation, the benefits of using good initial solutions in the SSEA are more apparent at short running times, as the differences between fitness decrease as the running time increases, but fitter overall solutions are found when the algorithm uses

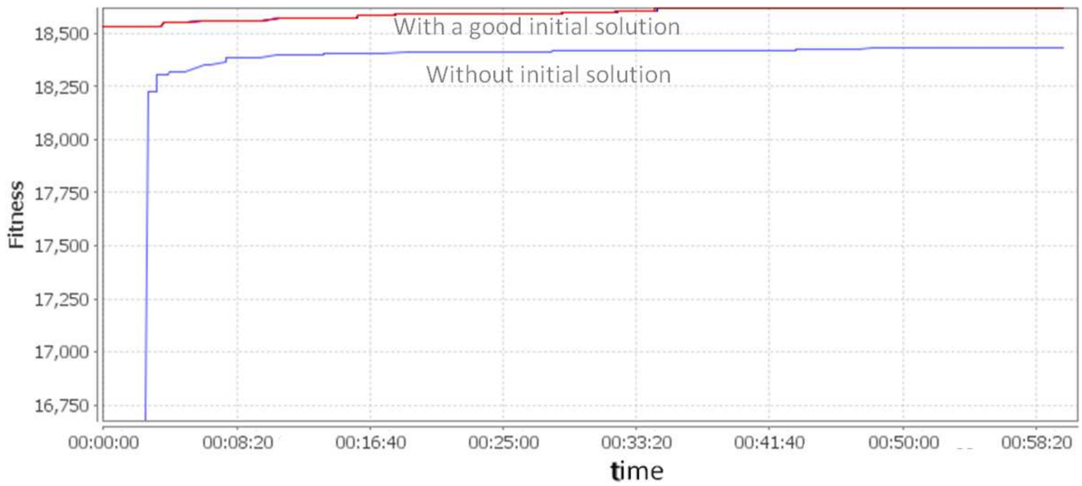


Figure 5.15: Progress in fitness of solutions when run with and without initial random population for Gurobi, a 3-pier topology, 78 BSSs, 48 stands and 219 flights for 1 hour.

fit good initial solutions. This was also noted when using commercial optimisation applications such as Gurobi and CPLEX.

The mutation operators, with the exception of DSEMO, cannot increase the number of assignments, therefore solutions which do not have maximum assignment restrict the search space, and waste iterations which could otherwise be used to widen the search of the space of solutions potentially improving on those already found solutions. This can be particularly detrimental if none of the solutions provided are sufficiently fit, i.e. solutions with at least one unassigned flight which is assigned in the optimal solution, as such flights cannot be assigned by these operators. Therefore, when the initial solutions do not have maximum flight assignment for the given number of BSSs, the search is restricted to flights already assigned which means low fitness. In these cases, the use of another operator which can increase the number of assignments, such as the DSEMO should be used, at least until one or many of the solutions in the population reach maximum flight assignment.

Table 5.4 shows the statistical fitness significance of the best solution obtained by the SSEA1 with a population size of 30 and single operator MEFNR3, when using an initial population composed of good solutions obtained from applying the constructive algorithms studied in Chapter 4. It is compared with those solutions obtained when the initial population is composed of the 30 fittest solutions of 200 randomly generated solutions (random constructive algorithm) for the data set. The empirical results show that the SSEA with good initial solutions provides in most of the instances considered here a superior final best solution (statistical fitness significance < 0.005) than when

Data set	3-pier								
194 flights H1T091216	13	14	15	16	17	18	19	20	21
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	22	23	24	25	26	27	28	29	
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
163 flights H1T100301	13	14	15	16	17	18	19	20	21
	0.0000	0.0686	0.7746	0.0000	0.0000	0.0000	0.1127	0.0000	0.8741
	22	23	24	25	26	27	28	29	
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

Data set	4-pier								
194 flights H1T091216	13	14	15	16	17	18	19	20	21
	0.0000	0.0000	0.0000	0.0000	0.0000	0.1275	0.0000	0.0000	0.0000
	22	23	24	25	26	27	28	29	
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	
163 flights H1T100301	13	14	15	16	17	18	19	20	21
	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	1.0000	0.0000	0.0000
	22	23	24	25	26	27	28	29	
	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

Table 5.4: Statistical fitness significance for a significance level of 0.05, SSEA1 with fit initial solutions and initial random solutions for the data sets provided by NATS.

the initial population is composed of random solutions.

The algorithms in the study in the following sections use the initial solutions obtained by applying the constructive algorithms which were used in this section and introduced in Chapter 4.

5.7.2 Population Size

The effect of the population size (μ) on the results of several of the operators presented in 5.4 was explored. The parameters used in the experiments are:

1. The data sets used relates to those provided by NATS both for 16^{th} December 2009 (H1T091216) and 1^{st} March 2010 (H1T100301), with both a 3-pier and a 4-pier topologies.

2. Number of BSSs of $N \in [13 \dots 29]$.

3. The operators used are: C1P, C2P, DSEMO, Multi Exchange By Pier between a Fixed Number of 3 Resources (MEBPFNR3), MEFNR3 and RMEFNR2. The number of resources (BSSs) considered for the mutation operators used were determined by a comparison of the initial results obtained from runs with a population size of 30 for each of the mutation operators.

4. The number of iterations per generation ℓ was initially set to 1.

5. The replacement strategies used are: ES, SUMS, Index Selection with Elitist Selection and a group size of 1 (IS1ES), and Index Selection with Stochastic Universal Modified Sampling and group size of 1 (IS1SUMS).

6. Population sizes of $\mu \in \{1, 5, 10, 15, 30, 50, 100, 200, 500, 800, 1000, 2000\}$ were considered. The algorithm was initially run for population sizes of 15, 30, 50, 100, 200, 500 and 1000. In some instances the best values appeared at the end of the ranges, which encouraged to extend the range of population sizes studied accordingly to the best population size for each of the operators types and a summary of the results is shown in Table 5.5 (the results are shown in Appendix B.2.1 and B.2.2).

   Regarding the Multi Exchange Operators, only extra population sizes of 1, 5, 10 were studied, given that these operators are guided mutation operators based on chance, and provided better results for the lowest population sizes initially considered. Nevertheless, given the poor results obtained when using the TS, as shown in Figure 5.12, it was anticipated that the size of the population should be higher than 1.

   In the case of the DSEMO, the results indicated that a high population size was preference, such that other appropriate population sizes were then considered. The population sizes of 500 and 1000 gave the best results, which was an indication that population sizes between those sizes may potentially be statistically even better. The population sizes studied were therefore extended to a population size of 800, since a population size of 1000 solutions was statistically significantly fitter in more cases than when using 500 as the population size.

   It was observed that the crossover operators performed better for high population sizes as expected, being consistently better for the largest population sizes evaluated. Given that a higher population size means a higher running time, a further population size of only 2000 was considered for the crossover operators. If there are too few solutions in a population and given that crossover used the information in the parent solutions, then the operator explores only a small part of the search space. On the other hand, if there are too many chromosomes, the algorithm may slow down, as some operations are applied to the full population.

The summary of overall results, when compared using the Mann-Whitney test, are shown in Table 5.5, where light grey is used for the values close to those that provided the overall statistically significantly fitter solutions which are presented in black. For a full list of the summary tables for each result see Appendix B.2.1.

Operator	194 flights (16^{th} December 2009)			
	3-pier topology		4-pier topology	
	Population Size	Selector	Population Size	Selector
C1P	2000	IS1SUMS, IS1ES	2000	IS1SUMS, IS1ES
C2P	2000	IS1ES, IS1SUMS	2000	IS1SUMS, IS1ES
DSEMO	800, 1000	IS1ES	500, 800, 1000	IS1ES
MEBPFNR3	10, 5	IS1ES	5, 15	IS1ES
MEFNR3	1, 5	IS1ES	10	IS1ES
RMEFNR2	15	IS1ES	10, 15	IS1ES

Operator	163 flights (1^{st} March 2010)			
	3-pier topology		4-pier topology	
	Population Size	Selector	Population Size	Selector
C1P	2000	IS1SUMS, IS1ES	2000	IS1ES, IS1SUMS
C2P	2000	IS1ES, IS1SUMS	2000	IS1ES, IS1SUMS
DSEMO	1000, 800	IS1ES	500	IS1ES
MEBPFNR3	5, 10	IS1ES	5, 10	IS1ES
MEFNR3	10	IS1ES	10, 5	IS1ES
RMEFNR2	15	IS1ES	15, 10	IS1ES

Table 5.5: Summary of the results of the Mann-Whitney test for significance level of 0.05, different population sizes and replacement strategies.

With respect to the mutation operators, which are based on a local search, the solutions reached are highly dependent on the individual parent solution, which generally represent small populations. Given that mutation operators require only a parent solution, the population size could range from one solution to many. As the smaller population size would consist of one solution, it may be considered that a population size of one should be the best approach from a mutation operator point of view. This relies strongly on the quality of the solution in reaching either a better or optimal solution, as the fitness does not normally give a clear indication of the solution quality with respect to better solutions in its neighbourhood, which the empirical results corroborate. A solution with lower fitness may be closer to a better or optimal solution for the moves performed by the operators used, thus improving the chances that these latter are reached.

In general crossover operators are expected to benefit from large population sizes, which is corroborated by my results. Given that the crossover operators take advantage of good differences between the parent solutions, then the minimum population size required is two solutions. This is the main factor benefitting crossover operators since a large population size normally results in greater diversity within the population of solutions. Nevertheless, a higher population size also means a slower algorithm execution time, given that some operations are executed for all members of the population, the processing time of which depends on the number of solutions in the population. Additionally, too much diversity may result in a loss of solutions with good building blocks, and have a corresponding detrimental effect on the overall search, the loss of better solutions, or the opportunity to reach these better or optimal solutions.

As observed, the population size and operator have an important impact on the algorithm's performance, but it is not the only factor to consider, as the diversity may also be increased or decreased by changing the selection approaches used, i.e. Replacement Strategies (Section 1) and the Parent Selector (Section 2). Elitist Sampling (ES, Section 2.7.1) reduces the diversity, as it keeps the solutions with higher fitness, which tends in turn to concentrate the solutions around those with fewer differences but increases the pressure, whereas Stochastic Universal Modified Sampling (SUMS, Section 5.3.1) increases the chance of solutions with lower fitness taking part in the population of solutions so increasing the diversity. To reduce the ES potential detrimental effect the Index Selector (ISxy, Section 5.3.2) was designed, implemented and run, the empirical results of which show a better performance than the underling Replacement Strategies used, such as ES and SUMS.

Population Size for when Combined Operators are used

Where different operators have a preference for different population sizes, these results may be taken into account when combining operators in order to improve the performance. So when the operator is selected from a pool of operators, randomly for example, its population size preference should be borne in mind so that the parent(s) may be selected within the solutions in the population, and within that given preferred size. This assumes that the solutions are ordered in some way. This approach allows better solutions obtained by the other operators with larger preferred population sizes to enter the population of the current operator, potentially increasing the diversity, which it could be considered as a type of migration. In this approach only the preferred population size is used to select the parent(s) for a given operator.

Run time Results for the Different Population Sizes

In this section the y-axis of the graphics is the average execution time for each set of 30 experiments with a different number of BSSs (the number of BSSs is shown in the x-axis). Each graph shows the average results for a given operator and data set, taken from those data sets provided by NATS, different replacement strategies (ES and IS1ES) and population sizes. The lines within a graphic identify the set of experiments which were run with the same parameters, i.e. replacement strategy, population size, operator and data set.

The average results for the operator DSEMO and the different data sets are presented in Figure 5.16, which shows that DSEMO requires a more constant running time up to the vicinity of the Upper Maximum Assignment Point (UMAP), where the running time drops to zero. On inspection of the initial population of solutions, it is apparent that the average running time of near zero refers to all the instances where the initial solutions have full assignment of flights to BSSs. So the DSEMO is unable to exchange or increase the flight assignments. As the number of BSSs is reduced up to LMAP more flights are unassigned in the initial solutions, which in turn gives the operator more chance to improve the solutions by increasing the number of assignments, potentially generating solutions with full assignments, so improving on the fitness. Finally, for numbers of BSSs lower than LMAP, not all the flights can be assigned to BSSs, so the operator initially has a chance of increasing the number of assignments for those initial solutions which do not have maximum assignment. This may also improve on the other objectives by exchanging unassigned flights with assigned ones, as will be seen in the following sections. This explains the relatively constant average running time, as the majority of operations are exchanges between assigned and unassigned flights, whereas the small variations in running time are a consequence of the number of solutions without maximum assignments in the initial solution and the speed with which the replacement strategy removes them. The differences between the various lines in Figure 5.16 correspond to different population sizes, so a higher population size results in higher running times as may be expected: this is mainly because other operations are performed on all of the population members, such as applying the Replacement Strategies and the Member Selector. The difference between lines for the same population size and different replacement strategies are an indication of how quickly the replacement strategy manages to remove solutions with low fitness, i.e. those solutions which do not have maximum assignment, such as those introduced as initial solutions. This is corroborated by the fact that ES has smaller average running times than IS1ES as expected, since ES provides a higher search pressure giving less chance for solutions of a lower fitness to generate new solutions. As expected, data sets with a higher number of flights required longer running times. These results also corroborate the findings presented in Ascó et al (2012).

Figure 5.17 shows that the Multi Exchange Mutation Operators have a tendency to increase the running time as N (number of BSSs) increases, which corresponds to an increase in the maximum number of flights assignable and the number of initial solutions which have full assignment. Conversely, RMEFNR2 running time is near

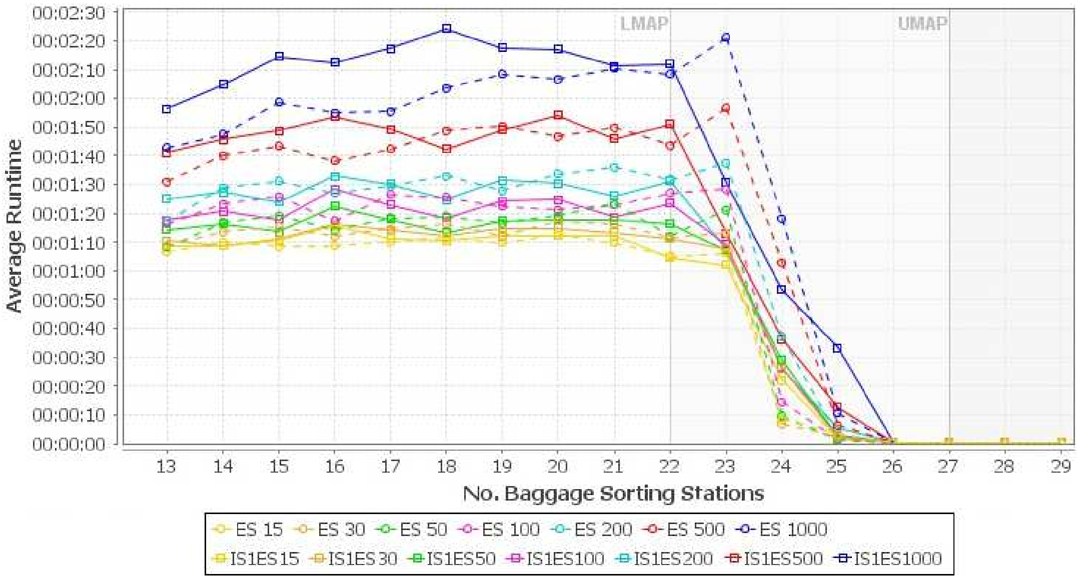


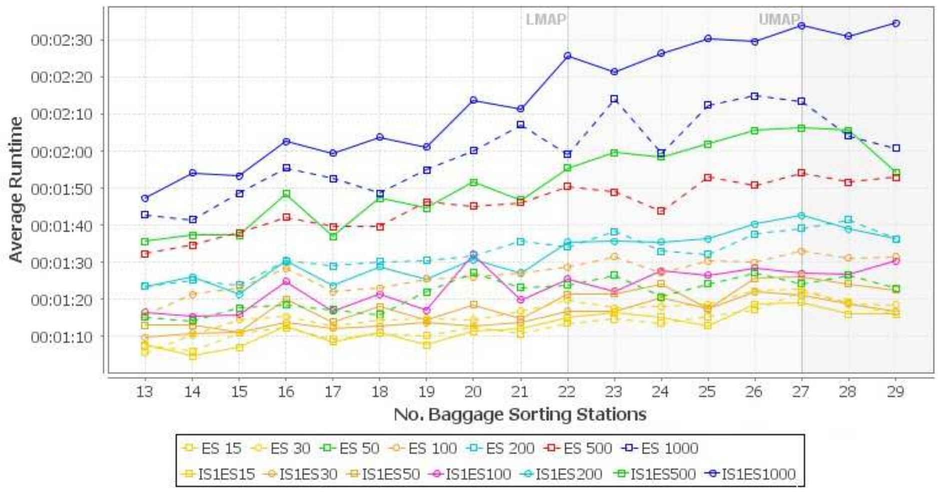
Figure 5.16: Average run-time for a 4-pier topology, 194 flights (H1T091216), DSEMO and different population sizes.

constant in most of the instances. RMEFNR2 running time for IS1ES does not appear to be affected by the number of BSSs, whereas for MEBPFNR3 and MEFNR3 the running time increases as the number of BSSs increases. Similar results were obtained for the data set provided by NATS for 1^{th} March 2010 and both 3-pier and 4-pier topologies.

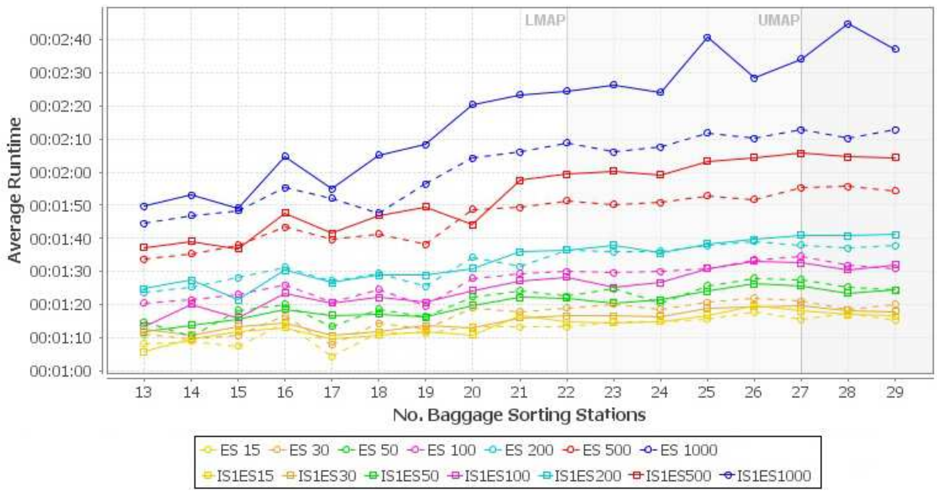
Figure 5.18 shows a considerable difference in behaviour between C1P and C2P as the number of BSSs increases, whereas with C1P the speed fluctuates around an average, and for C2P the speed reduces with minor fluctuations overall according to the number of BSSs.

The mutation operators considered are much faster than the crossover operators as is to be expected. C2P and DSEMO present variations depending on the number of BSSs, whereas C2P expends more time running with very low numbers of BSSs. This is reduced as the number of BSSs increases up to a point just before the LMAP, where the required running time is kept at its lowest and most constant, irrespective of the number of BSSs.

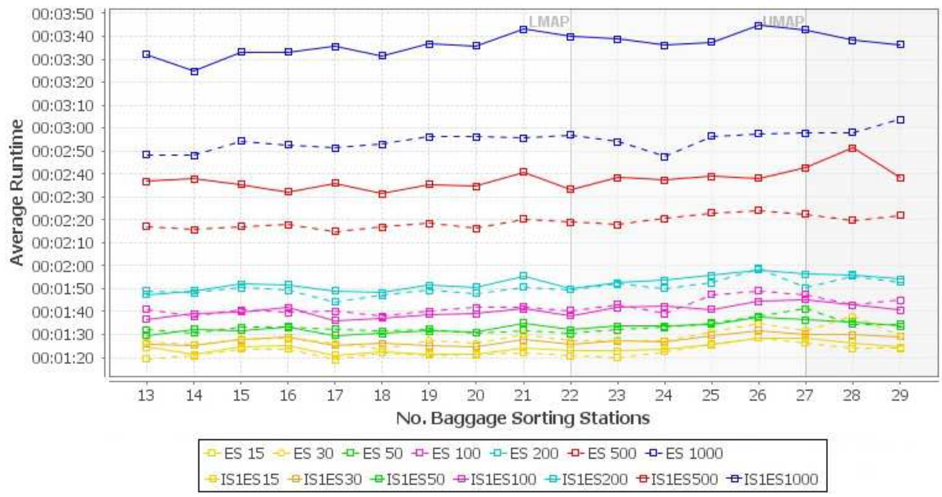
In all of the cases, as the population size increases so the running time also increases as shown in Figures 5.16, 5.17 and 5.18. Similar results were obtained for the data set from London Heathrow airport Terminal 1 for 1^{st} March 2010 as can be seen in Appendix B.2.3.



a MEBPFNR3.

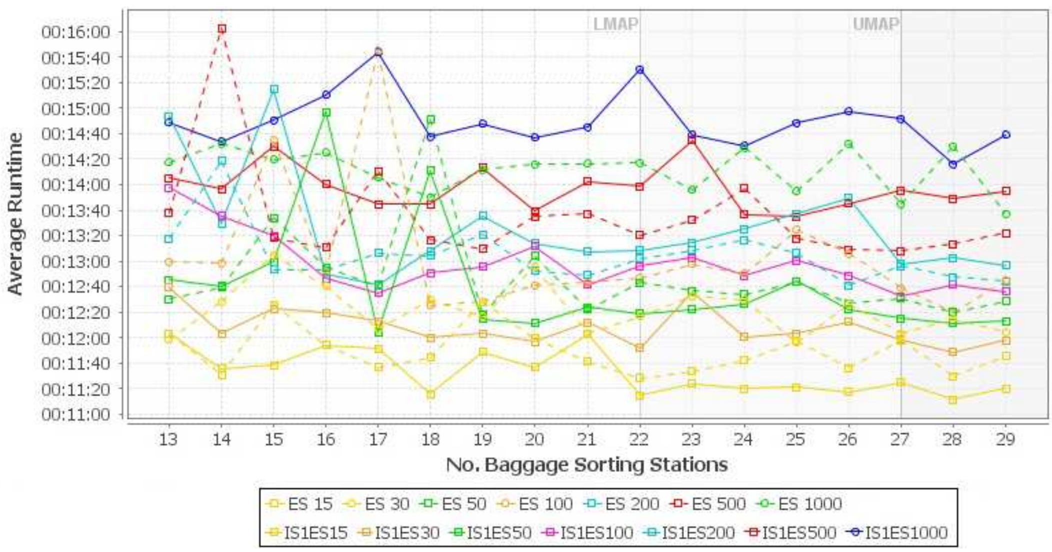


b MEFNR3.

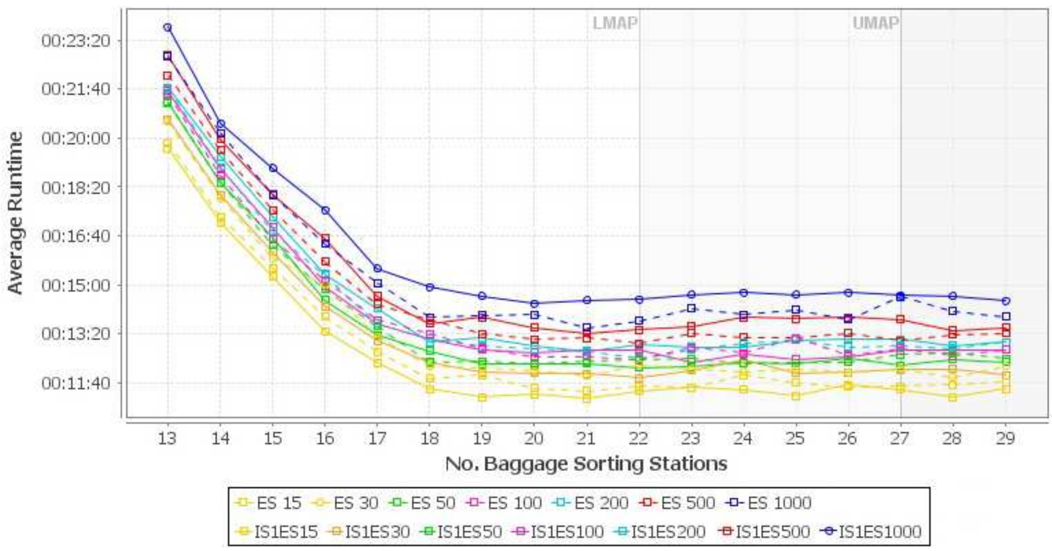


c RMEFNR2.

Figure 5.17: Average run-time for a 4-pier topology, 194 flights (H1T091216) for some mutation operators and different population sizes.



a 1-point crossover (C1P).



b 2-point crossover (C2P).

Figure 5.18: Average run-time for a 4-pier topology, 194 flights (H1T091216), crossover and different population sizes.

5.7.3 Number of Iterations in a Generation

The SSEA is composed of ℓ iterations per generation which contributes to the overall performance of the algorithm. Having an idea of the effects and contributions of this parameter will help in tuning the algorithm. To this end multiple experiments were conducted using different values of ℓ for the different parameters presented below.

1. The operators used: C1P, C2P, DSEMO, MEBPFNR3, MEFNR3 and RMEFNR2.

2. Population sizes used: 1000 for C1P, C2P and DSEMO and 15 for the Multi

Exchange Mutation Operators.

3. Replacement strategies used: ES and IS1ES.

4. Initial solutions were obtained by running the constructive algorithms presented in Chapter 4.

5. Iterations in a generation: $\ell \in \{1, 5, 10, 15, 20, 30, 100\}$.

6. The data sets used correspond to those provided by NATS both for H1T091216 and H1T100301, with both a 3-pier and a 4-pier topologies.

The increase of ℓ equates to a reduction in the search pressure given that the current solutions have more chance of being selected as ℓ increases. Also as the same population exists for longer (ℓ times) then the diversity is kept for longer as ℓ increases, e.g. if SSEA is run with a population size of 1000 solutions, for 1000 overall iterations and $\ell = 1000$ then the initial population will be maintained throughout the whole execution.

These results are similar for the different data sets and topologies considered, an overall summary of which is presented in Table 5.6 and all of the summary results per operator can be seen in Appendix B.3. Table 5.6 summarises the values of ℓ, which provide statistically significantly fitter solutions for the widest range of numbers of BSSs. The values for ℓ between brackets are the next best values of ℓ.

Operator	Selector	194 flights (H1T091216)		163 flights (H1T100301)	
		Topologies			
		3-pier	4-pier	3-pier	4-pier
C1P	IS1ES	1	1	1	1
C2P	IS1ES	1	1	1	1
DSEMO	IS1ES	1	1	1	1
MEBPFNR3	IS1ES	1, 15 (20, 30)	5, 15, 30	10, 15 (30)	5, 30 (1, 20)
MEFNR3	IS1ES	15 (5, 20)	10 (30)	20, 100	10, 100 (20)
RMEFNR2	IS1ES	10 (20, 5, 15)	5 (1)	10 (1, 5, 15)	5 (1)

Table 5.6: Overall summary of the best ℓ of each operator, data set and topology considered and significance level of 0.05.

Table 5.6 shows that C1P, C2P and DSEMO provide statistically significantly fitter solutions for all data sets considered, topologies and number of BSSs for $\ell = 1$, whereas the remaining operators considered provide statistically significantly fitter solutions in the range of ℓ from 5 to 30.

Increasing ℓ gives more chance for other solutions to be selected to generate new solutions, which equates to a reduction in pressure (but not an increase in diversity).

Given that C1P, C2P and DSEMO have a large population size of 800, 1000 and 2000 solutions respectively, which provides diversity, the same cannot be said about search pressure, which may be said to explain the preference for low values of ℓ. This also seems to be corroborated by the results for MEBPFNR3, MEFNR3 and RMEFNR2, which prefer higher values of ℓ.

5.7.4 Index for ISxES

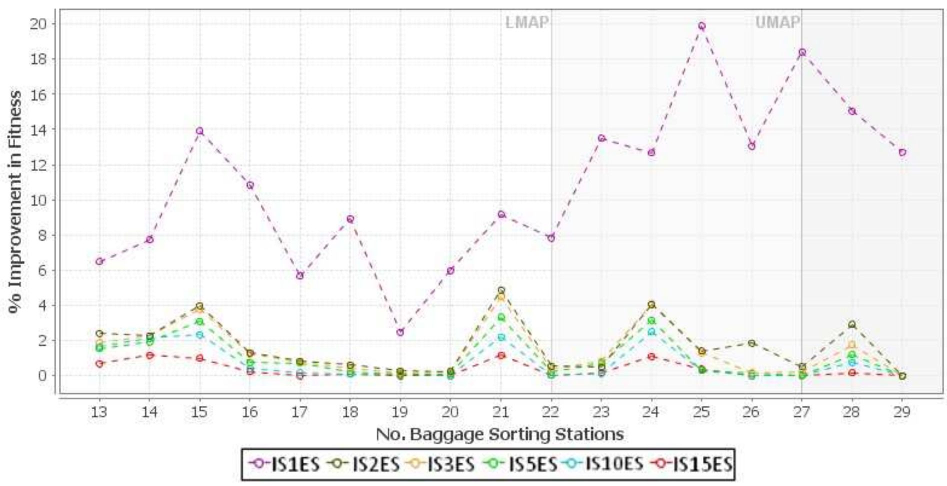
The initial results obtained from the Index Selector were for a group size of $x = 1$ for the Elitist Selector (ISxES, Section 5.3.2) and provided solutions with good fitness. Other experiments were conducted to see what other values of x could achieve. The characteristics of the selector indicate that any index must be greater than zero as a maximum group size of zero does not have any meaning. Moreover there is no significance in having an index higher than the population size, since the maximum size of a group cannot be larger than the population size. Taking these factors into account together with the previous results in which the Multi Exchange operators, provides statistically significantly fitter solutions for population sizes of 5, 10 and 15. Some experiments were then designed to examine the effect of changing the index $x \in \{1, 2, 3, 5, 10, 15\}$ for a population size of 15 for the Multi Exchange Operators. Given that the preferred population sizes for the crossover operators and DSEMO are high (around 1000 solutions), a population size of 1000 was used for these operators.

The figures used in this section show the experiment results for different group maximum sizes when using some of the operators previously presented. The results are presented as an average percentage improvement on fitness (y-axis), with 0% referring to the best initial solutions used and 100% referring to the upper bound obtained when running CPLEX solver with the Integer Linear Programming (ILP) presented in Chapter 3, for different number of BSSs (x-axis), Equation 5.1. Negative percentages refer to the best final solutions which have a worse fitness than the best initial solution.

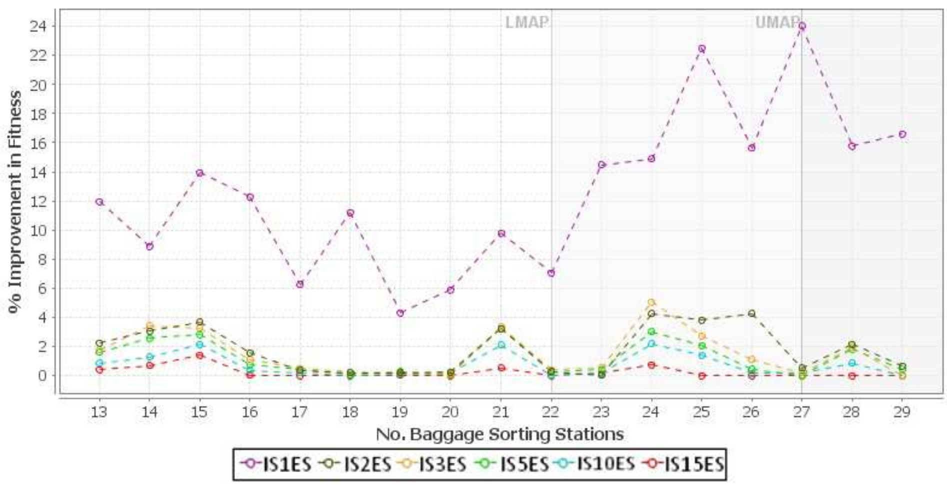
$$\%Improvement\ Fitness = \frac{f - f_{Best}^{Ini}}{f_{UB}^{CPLEX} - f_{Best}^{Ini}} * 100 \qquad (5.1)$$

Figures 5.19 and 5.20 show the results for the data set H1T091216, different operators and a 4-pier topology. Similar results were obtained for a 3-pier topology and the data set H1T100301, the results of which can be seen in Appendix B.4.

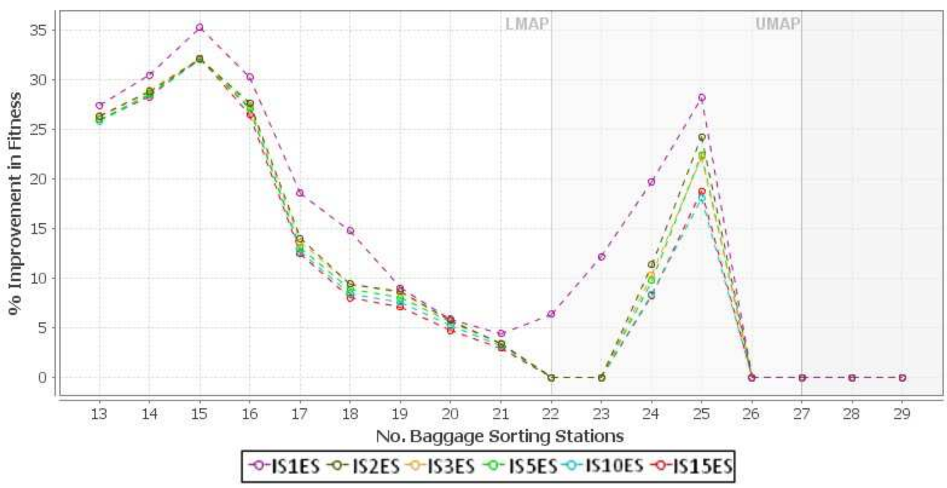
Initial inspection of the way the operator works suggests that an increase in the



a 1-point crossover (C1P) with a 1000 population size.

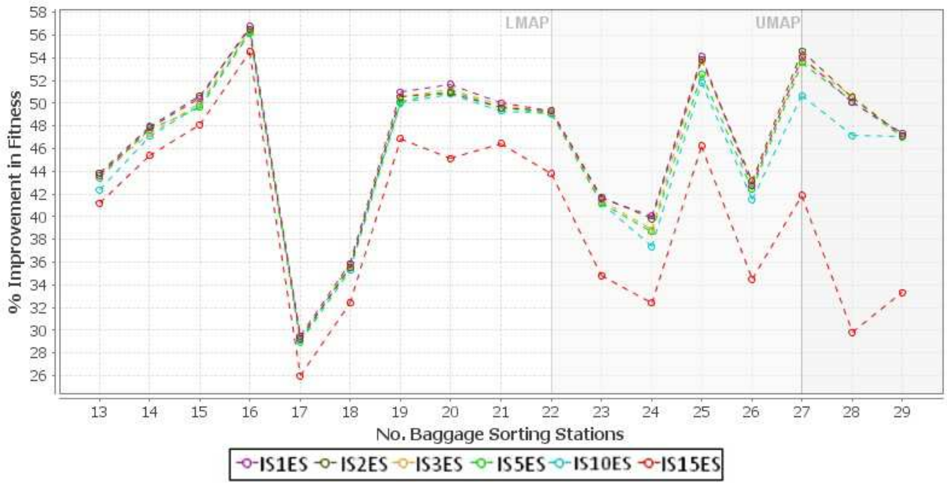


b 2-point crossover (C2P) with a 1000 population size.

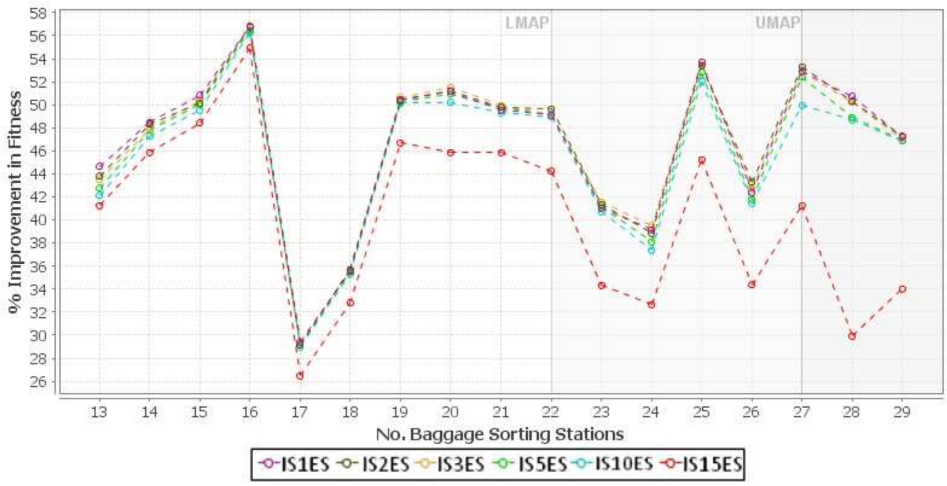


c DSEMO with 1000 population size.

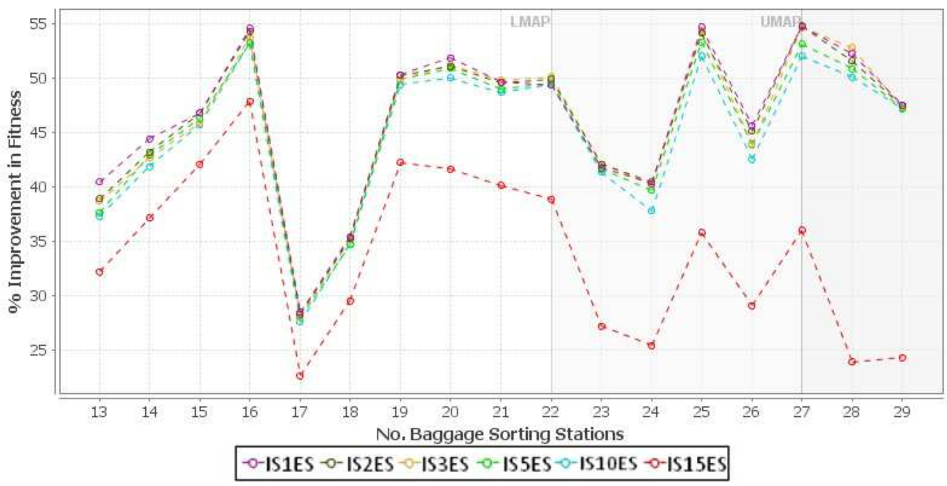
Figure 5.19: ISxES $x \in \{1, 2, 3, 5, 10, 15\}$ for H1T091216 and a 4-pier topology.



a MEBPFNR3 with a 15 population size.



b MEFNR3 with a 15 population size.



c RMEFNR2 with a 15 population size.

Figure 5.20: ISxES $x \in \{1, 2, 3, 5, 10, 15\}$, mutation operators for H1T091216 and a 4-pier topology.

index should correspond to a reduction in the diversity as the overall number of different solutions will be reduced since many solutions with the same fitness are included in each group. As an illustration of this, the case of an operator with a population size of 10 and index of 10 is explored. As the execution progresses it could at some time finish with 10 solutions having the same fitness, which corresponds to a behaviour similar to ES.

The normality test showed that it was not possible to assume that the distributions are normal, thus it was appropriate to use the Mann-Whitney statistical significance test for each of the number of BSSs considered and between the different operators and indexes. A summary of the results for these experiments is shown in Table 5.7, which shows the maximum group sizes (x) only, which provided statistically significantly fitter solutions.

Operator	194 flights (H1T091216)		163 flights (H1T100301)	
	Topologies			
	3-pier	4-pier	3-pier	4-pier
C1P	IS1ES	IS1ES	IS1ES	IS1ES
C2P	IS1ES	IS1ES	IS1ES	IS1ES
DSEMO	IS1ES	IS1ES	IS1ES	IS1ES
MEBPFNR3	IS1ES and IS2ES	IS1ES	IS1ES	IS1ES and IS2ES
MEFNR3	IS1ES	IS2ES	IS2ES	IS2ES
RMEFNR2	IS1ES	IS1ES	IS1ES	IS1ES

Table 5.7: Overall summary of the Mann-Whitney statistical significance tests for index in ISxES and significance level of 0.05.

In general IS1ES provided more instances with statistically significantly fitter solutions than IS2ES, IS3ES, IS5ES, IS10ES and IS15ES. In cases where both IS2ES and IS1ES perform well, IS2ES was considered better because in the cases where it provided statistical significantly fitter solutions these corresponded to a high number of BSSs, which incidentally also corresponds to the range of numbers of BSSs normally operating at an airport.

5.7.5 Single Operators

Several experiments were run to establish the performance of each of the operators considered individually when used with the proposed SSEA. Following the previous results, new experiments were designed to establish an appropriate combination for use of an operator and replacement strategy. The parameters used in the experiments are:

1. The data sets used correspond to those provided by NATS both for H1T091216

and H1T100301, with both a 3-pier and a 4-pier topologies.

2. Number of BSSs of $N \in [13 \ldots 29]$.

3. Initial solutions were obtained by running the constructive algorithms presented in Chapter 4, as in previous sections.

4. Operators used: MEBPFNRn, MEFNRn and RMEFNRn with $n \in [2 \ldots 10]$. Also MEBPRNRn, MERNRn and RMERNRn with $n = 10$ were studied.

5. Population sizes used: 30.

6. Iterations in a generation used: $\ell = 1$.

7. Replacement strategies used: ES, IS1ES, SUMS and IS1SUMS.

Once again given that the data cannot be said to follow a normal distribution the Mann-Whitney test was used to establish the statistical significance of the solutions' fitness. Tables 5.8, 5.9, 5.10, 5.11 and 5.12 show a summary of the replacement strategies for different operators which cannot be said to provide statistically significant solutions with a lower fitness than the others for a significance level of 0.05. Those operators providing statistically significantly less fit solutions than any other are not shown for simplicity and clarity. The selection operators with the highest number of statistically significantly fitter solutions than other selection operators have been underlined.

Looking at the results obtained by the operator MEBPFNRn, a pattern can be seen where the best solution obtained throughout the studied range of BSSs is obtained for a parameter $n \in [3 \ldots 6]$. This behaviour, together with the results obtained for the operator Multi Exchange By Pier between a Random Number of 10 Resources (MEBPRNR10), which provides similar results on average to MEBPFNRn, prompted me to consider an extension of the MEBPRNRn for a range of numbers of BSSs, instead of a maximum value only as in MEBPRNRn, known as MEBPRRNRxy presented in Section 5.4.1.

On examining the results for the DSEMO it is apparent that for a number of BSSs greater or equal to the LMAP ($N \geq$ LMAP), in some instances the DSEMO still manages to improve the initial solutions, even where the fittest initial solutions have assigned all the flights, as can be observed for the 25 BSSs in Figure 5.19c. Simply

Operator	Number of BSSs			
	13	14	15	16
MEBPFNR3			IS1ES and IS1SUMS	IS1ES and IS1SUMS
MEBPFNR10	IS1ES			
MEBPRNR10	IS1ES	IS1SUMS	IS1SUMS	IS1ES and IS1SUMS
MEFNR4		IS1SUMS	IS1ES and IS1SUMS	IS1ES and IS1SUMS
MEFNR5	IS1ES	IS1ES	IS1ES and IS1SUMS	IS1ES and IS1SUMS
MEFNR6	IS1ES	IS1ES and IS1SUMS	IS1ES	IS1ES and IS1SUMS
MEFNR7	IS1ES	IS1ES	IS1ES and IS1SUMS	IS1EA and IS1SUMS
MEFNR8	IS1ES and IS1SUMS	IS1ES and IS1SUMS	IS1ES and IS1SUMS	IS1ES
MEFNR9	IS1SUMS	IS1ES and IS1SUMS	IS1ES	IS1ES
MEFNR10	IS1SUMS	IS1SUMS	IS1ES and IS1SUMS	

Operator	Number of BSSs			
	17	18	19	20
MEBPFNR3		IS1ES	IS1ES and IS1SUMS	IS1ES
MEBPFNR4				SUMS
MEBPFNR10				IS1ES
MEBPRNR10	IS1SUMS		IS1ES	
MEFNR2				IS1ES and IS1SUMS
MEFNR3	IS1ES and IS1SUMS			IS1ES, IS1SUMS and SUMS
MEFNR4	IS1SUMS		IS1ES and IS1SUMS	IS1ES and IS1SUMS
MEFNR5	IS1ES	IS1ES and IS1SUMS	IS1ES and IS1SUMS	IS1ES and IS1SUMS
MEFNR6	IS1ES	IS1ES	IS1ES	IS1ES and IS1SUMS
MEFNR7		IS1ES and IS1SUMS	IS1ES and IS1SUMS	IS1ES and IS1SUMS
MEFNR8	IS1ES and IS1SUMS	IS1ES and IS1SUMS		IS1SUMS
MERNR10	IS1ES			
RMEFNR2				IS1ES

Operator	Number of BSSs			
	21	22 (LMAP)	23	24
MEBPFNR3	IS1ES and IS1SUMS	SUMS	IS1ES	
MEBPRNR10	IS1ES	SUMS	SUMS	
MEFNR3	IS1ES and IS1SUMS	SUMS	SUMS	
MEFNR4	IS1ES and IS1SUMS		SUMS	
MEFNR5	IS1ES and IS1SUMS and SUMS	SUMS		
MEFNR6	IS1ES and IS1SUMS			
MEFNR7	IS1ES			
MEFNR8	IS1ES			
RMEFNR2	IS1ES	IS1ES, IS1SUMS and SUMS	IS1ES and SUMS	IS1ES and IS1SUMS

Operator	Number of BSSs			
	25	26	27 (UMAP)	28
RMEFNR2	IS1ES	IS1ES and IS1SUMS	IS1ES	IS1ES and SUMS

Operator	Number of BSSs
	29
RMEFNR2	IS1ES and IS1SUMS

Table 5.8: Summary for SSEA1 with a single operator, 30 population size, 800000 iterations, a 4-pier topology for 194 flights (H1T091216) and a significance level of 0.05.

Operator	Number of BSSs			
	13	14	15	16
DSEMO	IS1ES and IS1SUMS	IS1SUMS		
MEBPFNR10			IS1ES	IS1ES
MEBPRNR10			IS1ES and IS1SUMS	IS1ES and IS1SUMS
MEFNR4			IS1ES and IS1SUMS	
MEFNR5			IS1ES and IS1SUMS	IS1ES and IS1SUMS
MEFNR6			IS1ES and IS1SUMS	IS1ES and IS1SUMS
MEFNR7			IS1ES and IS1SUMS	IS1SUMS
MEFNR8			IS1ES	IS1ES and IS1SUMS
MEFNR9				IS1SUMS
MEFNR10			IS1ES	IS1ES and IS1SUMS
Operator	Number of BSSs			
	17	18	19 (LMAP)	20
DSEMO	IS1SUMS			
MEBPFNR3			IS1SUMS	IS1SUMS
MEBPFNR6				SUMS
MEBPFNR8				SUMS
MEBPRNR10			IS1ES	IS1SUMS and SUMS
MEFNR3			IS1SUMS	IS1ES and IS1SUMS
MEFNR4			IS1ES and IS1SUMS	IS1ES, IS1SUMS and SUMS
MEFNR5		IS1SUMS	IS1ES, IS1SUMS and SUMS	IS1ES, IS1SUMS and SUMS
MEFNR6			IS1ES, IS1SUMS and SUMS	IS1SUMS
MEFNR7			IS1SUMS	IS1ES and IS1SUMS
MEFNR8			IS1SUMS	IS1ES
MEFNR9				IS1SUMS
MERNR10				IS1ES
Operator	Number of BSSs			
	21	22	23	24
MEBPFNR2				IS1ES
MEBPFNR3	IS1ES	IS1ES	IS1ES	IS1ES and SUMS
MEBPFNR4				SUMS
MEBPFNR5	SUMS			SUMS
MEBPRNR10		IS1ES		
MEFNR2				SUMS
MEFNR3		IS1ES and IS1SUNS	IS1ES	IS1ES and SUMS
MEFNR4	IS1ES, IS1SUMS and SUMS	IS1ES and IS1SUMS	IS1SUMS	IS1ES, IS1SUMS and SUMS
MEFNR5	IS1ES and IS1SUMS	IS1ES and IS1SUMS	IS1ES	IS1ES, IS1SUMS and SUMS
MEFNR6	IS1ES and IS1SUMS			
MEFNR7	IS1SUMS			
RMEFNR2	IS1ES and IS1SUMS			
RMEFNR3				IS1ES, IS1SUMS and SUMS

Table 5.9: Summary for SSEA1 with a single operator 30 population size, 800000 iterations, a 4-pier topology for 163 flights (H1T100301) and a significance level of 0.05.

applying the DSEMO alone provides an improvements up to 25% for a 4-pier topology. Examining the initial solution provided, some of these solutions do not contain full assignments, so when the DSEMO operator is applied improvement can be achieved by means of an increase in assignments, which may in future guide the search in a different direction, in order to reach better solutions than those initially provided as

Operator	Number of BSSs				
	25 (UMAP)	26	27	28	29
MEBPFNR2			SUMS	SUMS	
MEBPFNR3	IS1ES	IS1ES and SUMS	IS1ES, IS1SUMS and SUMS	IS1ES, and SUMS	
MEBPFNR4		SUMS	SUMS	SUMS	
MEBPFNR5				SUMS	
MEFNR2		IS1SUMS and SUMS		SUMS	
MEFNR3		IS1ES	IS1ES	SUMS	
MEFNR4	IS1ES	IS1SUMS		SUMS	SUMS
MEFNR5				SUMS	
RMEFNR2	IS1ES	IS1ES	IS1ES and IS1SUMS	IS1ES, IS1SUMS and SUMS	IS1ES, IS1SUMS and SUMS
RMEFNR3	IS1ES	IS1ES and IS1SUMS	IS1ES	SUMS	

Table 5.10: Summary for SSEA1 with a single operator 30 population size, 800000 iterations, a 4-pier topology for 163 flights (H1T100301) and a significance level of 0.05.

initial solutions. This behaviour could be advantageous where this operator is used in conjunction with others, since it could move the search into other areas of the solution space which might otherwise not be investigated if this operator were not used. To evaluate whether this is the case it is necessary to design some experiments where the capabilities of the DSEMO operator can be seen working together with other operators which do not depend on the full assignment of flights to BSSs for a solution, which is explored in Section 5.7.6.

The search is said to be stagnated when the search is confined to a part of the solution space where there are no fitter solutions than those which have already been found. Figure 5.21 may also give an indication of this situation, as it presents the average time at which the last fitter solution was found for both the C1P and C2P operators and the different replacement strategies considered in the experiments conducted. The time between the last fitter solution found and that taken to complete all of the generations gives an idea as to whether the algorithm for a given operator and replacement strategy has become stagnated. In the case of 1-point and 2-point crossovers IS1ES preserves the search pressure and diversity better than the other replacement strategies, as shown in Figure 5.21. It does not merely continue to find solutions for a longer time, but these solutions are better, as shown previously.

RMEFNR2 on its own also provides fitter solutions than any of the other operators considered on their own for the normal operational range of BSSs at an airport, i.e.

Operator	Number of BSSs			
	13	14	15	16
MEBPFNR3		IS1ES		
MEBPFNR10	IS1ES	IS1ES		IS1ES and IS1SUMS
MEFNR4	IS1ES and IS1SUMS			IS1ES and IS1SUMS
MEFNR5	IS1ES and IS1SUMS	IS1SUMS		IS1ES
MEFNR6	IS1ES and Is1SUMS	IS1ES and IS1SUMS		IS1ES
MEFNR7	IS1ES and Is1SUMS	IS1ES and IS1SUMS		
MEFNR8	IS1ES and IS1SUMS	IS1ES and IS1SUMS		IS1ES and IS1SUMS
MEFNR9	IS1ES and IS1SUMS			IS1ES
MEFNR10	IS1ES	IS1ES and IS1SUMS		IS1ES
RMEFNR2			ES, IS1ES and IS1SUMS	
RMEFNR3			ES and IS1ES	
RMERNR10			ES	

Operator	Number of BSSs			
	17	18	19	20
MEBPFNR3		IS1ES		
MEBPFNR5	ES	SUMS		
MEBPFNR6	ES	SUMS		
MEBPFNR8	ES			
MEBPFNR9	ES			
MEBPFNR10	ES and IS1ES			
MEBPFNR10		IS1ES		
MEFNR4		IS1ES and IS1SUMS		
MEFNR5		IS1SUMS		
MEFNR6		IS1ES and IS1SUMS		
MEFNR7	ES	IS1ES and IS1SUMS		
MEFNR8		IS1ES and IS1SUMS		
MERNR10		SUMS		
MEFNR9	ES			
RMEFNR2	ES and IS1ES	IS1ES	IS1ES	IS1ES
RMEFNR3	ES and IS1ES			
RMEFNR4	ES			
RMEFNR5	ES			
RMERNR10	ES and IS1ES			

Operator	Number of BSSs			
	21	22 (LMAP)	23	24
RMEFNR2	IS1ES and SUMS	IS1ES	IS1ES	IS1ES and SUMS

Operator	Number of BSSs			
	25	26	27 (UMAP)	28
MEBPFNR2				IS1ES and IS1SUMS
MEBPFNR3				IS1ES, IS1SUMS and SUMS
MEBPFNR4				SUMS
MEFNR2				IS1ES, IS1SUMS and SUMS
MEFNR3				IS1ES, IS1SUMS and SUMS
MEFNR4				IS1ES, IS1SUMS and SUMS
MEFNR5				IS1ES, IS1SUMS and SUMS
RMEFNR2	IS1ES, IS1SUMS and SUMS	IS1ES and SUMS	IS1ES and SUMS	IS1ES, IS1SUMS and SUMS
RMEFNR3			IS1ES	IS1ES and SUMS

Table 5.11: Summary for SSEA1 with a single operator 30 population size, 800000 iterations, a 3-pier topology for 194 flights (H1T091216) and a significance level of 0.05.

Operator	Number of BSSs
	29
MEBPFNR2	IS1ES, IS1SUMS and SUMS
MEBPFNR3	IS1ES, IS1SUMS and SUMS
MEBPFNR4	SUMS
MEBPFNR5	SUMS
MEBPRNR10	IS1ES and SUMS
MEFNR2	IS1ES, IS1SUMS and SUMS
MEFNR3	IS1SUMS and SUMS
MEFNR4	IS1ES, IS1SUMS and SUMS
MEFNR5	IS1SUMS and SUMS
MEFNR6	SUMS
MERNR10	SUMS
RMEFNR2	IS1ES, IS1SUMS and SUMS
RMEFNR3	IS1ES and SUMS
RMERNR10	SUMS

Table 5.12: Summary for SSEA1 with a single operator 30 population size, 800000 iterations, a 3-pier topology for 194 flights (H1T091216) and a significance level of 0.05.

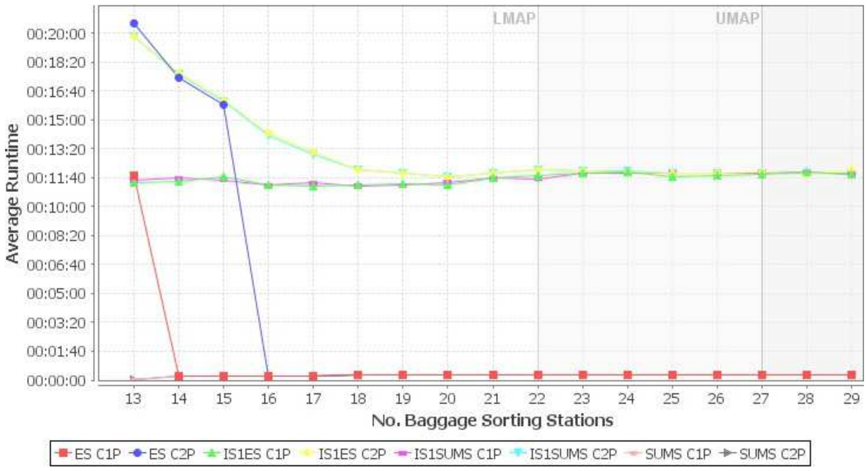


Figure 5.21: Last solution found for SSEA1, 1-point and 2-point crossovers for 194 flights (H1T091216) and a 3-pier topology for 800,000 total iterations.

$N \geq LMAP$, and the data set of H1T091216, as shown in Table 5.8. On the other hand, for a less dense schedule represented by the data set of H1T100301 this range is reduced to $N \geq UMAP$, as shown in Tables 5.9 and 5.10.

5.7.6 Multiple Operators

In this section, the combination of multiple operators (C1P, C2P, DSEMO, MEBPFNR3, MEFNR3 and RMEFNR2) with different percentages is studied. A full summary of the Mann-Whitney statistical significance tests can be seen in Appendix B.5.

The tables used in the following sections are a summary of the statistical significance tests which show the number of instances between parenthesis and separated by a comma, where the combined operator cannot be said to be statistically significantly less fit than any of the other operators studied for each of the number of BSSs grouped into ranges of $N <$ LMAP, LMAP $\leq N <$ UMAP and UMAP $\leq N$, ranges which are separated by a comma e.g. 7 (1, 3, **3**) means that there is 1 instance for $N <$ LMAP, 3 for LMAP $\leq N <$ UMAP and 3 for UMAP $\leq N$ where the operators cannot be said to be worse than any of the other operators and 7 being the sum of the values between parenthesis. Furthermore, in the table headers starting with 'Max.' between parenthesis the count of numbers of BSSs part of the group is specified. These groups only depend on the data set, e.g. for the data set of H1T091216 there are 9 instances of numbers of BSSs (13, 14, 15, 16, 17, 18, 19, 20 and 21 BSSs) where $N < 22$ (LMAP), 5 with $22 \leq N < 27$ (UMAP), and 3 with $27 \geq N$. These values help to give an idea of how often an operator performs well in each group of numbers of BSSs, where full coverage occurs when the number in the 'Max.' for the group is the same as for the operator the number is shown in bold and underlined, which in the example currently considered only happens in the last group, where $27 \leq N$.

The following parameters apply to all the experiments conducted in the sections and its subsections:

1. Data sets used: those provided by NATS both for H1T091216 (194 flights) and H1T100301 (163 flights), with both a 3-pier and a 4-pier topologies.

2. Initial solutions were obtained by running the constructive algorithms presented in Chapter 4 as in previous sections.

3. Population sizes used: 30.

4. Iterations in a generation: $\ell = 1$ with 800,000 iterations overall.

5. Replacement strategies used: ES, IS1ES, SUMS and IS1SUMS.

Probability Single Multi Operator Composed of Two Operators

The 'Probability Single Multi Operator' described in Section 5.4.3 is used. This uses two of the following operators: MEBPFNR3, MEFNR3, RMEFNR2, DSEMO, C1P and C2P, with probabilities of 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 and 0.9, eg. (0.9, 0.1) and (0.7, 0.3). A summary of the statistical significance is shown in Table 5.13, for the full statistical results see Appendix B.5.1.

It is apparent that a higher use of multiple exchange mutation operators with a preference for a crossover or the DSEMO perform better for combined operators as shown in Table 5.13. The results are better in a combination of two operators than

Operators	H1T091216			
	Max. (9, 5, 3)			
	3-pier		4-pier	
	Selector	Significance	Selector	Significance
RMEFNR2 0.8 + DSEMO 0.2	IS1ES	11 (6, 3, 2)		
RMEFNR2 0.9 + C2P 0.1			IS1ES	9 (1, *5*, *3*)
Operators	H1T100301			
	Max. (6, 6, 5)			
	3-pier		4-pier	
	Selector	Significance	Selector	Significance
MEFNR3 0.7 + DSEMO 0.3	IS1ES	7 (*6*, 0, 1)		
MEFNR3 0.9 + MEBPFNR3 0.1	IS1ES	7 (0, 5, 2)		
MEBPFNR3 0.9 + C1P 0.1			IS1ES	9 (0, *6*, 3)

Table 5.13: Number of occurrences which cannot be said to be statistically significantly less fit than the other for the single multi operator with two base operators and a significance level of 0.05.

with a single operator for $N <$ LMAP with preference for DSEMO for a very low N, which could be regarded as expected since this range of numbers of BSSs is where the DSEMO performs better. A higher N in both DSEMO and crossover operators combined with other operators performs better up to LMAP. As N (number of BSSs) increases, $N \geq$ LMAP, some single operators perform as well as two combined operators and as $N >$ UMAP and the number of combinations of operators doing well also increases. This has been seen in the different data sets and topologies studied.

Probability Single Multi Operator Composed of Three Operators

The 'Probability Single Multi Operator' described in Section 5.4.3 is used, which is composed of the base operators MEBPFNR3, MEFNR3, RMEFNR2, DSEMO, C1P and C2P in the following combinations:

1. MEBPFNR3 0.8 + DSEMO 0.1 + C1P 0.1
2. MEBPFNR3 0.8 + DSEMO 0.1 + C2P 0.1
3. MEFNR3 0.7 + DSEMO 0.2 + C1P 0.1
4. MEFNR3 0.7 + DSEMO 0.2 + C2P 0.1
5. MEFNR3 0.8 + DSEMO 0.1 + C1P 0.1
6. MEFNR3 0.8 + DSEMO 0.1 + C2P 0.1
7. RMEFNR2 0.8 + DSEMO 0.1 + C1P 0.1
8. RMEFNR2 0.8 + DSEMO 0.1 + C2P 0.1

The main reason for using these combinations is based on their individual performance which has been noted in previous sections. The mutation operators have been seen to perform very well, mainly MEBPFNR3, MEFNR3 and RMEFNR2, where DSEMO does not perform so well for a high number of BSSs but exhibited some potential for extending the search further, which may help to find fitter solutions. As the Multi Exchange Operators are the operators most similar to each other they were not considered as a third operator, so the crossover operators were used. As previously noted it may be possible to improve on the performance of combined operators if the selection of an operator is also based on the search point at the time and it may be further improved if the solutions considered by the member selector are based on the operator to be applied, i.e. its preferred population size.

A summary of the statistical significance of the experiments is shown in Table 5.14, and for the full statistical results see Appendix B.5.2.

Operators	H1T091216			
	Max. (9, 5, 3)			
	3-pier		4-pier	
	Selector	Significance	Selector	Significance
RMEFNR2 0.8 + DSEMO 0.1 + C1P 0.1	IS1ES	14 (6, **5**, **3**)		
RMEFNR2 0.8 + DSEMO 0.1 + C2P 0.1			IS1ES	11 (3, **5**, **3**)

Operators	H1T100301			
	Max. (6, 6, 5)			
	3-pier		4-pier	
	Selector	Significance	Selector	Significance
MEFNR3 0.7 + DSEMO 0.2 + C2P 0.1	IS1ES	13 (5, 4, 4)		
MEBPFNR3 0.8 + DSEMO 0.1 + C1P 0.1			IS1ES	14 (4, **6**, 4)
MEBPFNR3 0.8 + DSEMO 0.1 + C2P 0.1			IS1ES	14 (**6**, **6**, 2)
MEFNR3 0.7 + DSEMO 0.2 + C2P 0.1			IS1ES	14 (**6**, 5, 3)

Table 5.14: Number of occurrences which cannot be said to be statistically significantly less fit than the other probability single multi operator with three operators for a significance level of 0.05.

For the 3-pier topology: With reference to the Replacement Strategy, the IS1ES consistently provides better overall results. On the other hand for $N \geq$ UMAP both MEFNR3 0.8 + DSEMO 0.1 + C1P 0.1 and RMEFNR2 0.8 + DSEMO 0.1 + C2P 0.1 provide statistically significantly fitter solutions, whereas for lower N ($N <$ UMAP) then MEFNR3 0.8 + DSEMO 0.1 + C1P 0.1 provides statistically significantly fitter solutions but not in so many cases as MEFNR3 0.7 + DSEMO 0.2 + C2P 0.1. For LMAP $\leq N <$ UMAP MEBPFNR3 0.8 + DSEMO 0.1 + C1P 0.1 provides statistically significantly fitter solutions, covering the middle range of the numbers of BSSs better.

For the 4-pier topology: It can be seen that for the real range of BSSs, $N \geq$ LMAP, MEBPFNR3 0.8 + DSEMO 0.1 + C1P 0.1 provides more cases where it cannot be said that it is worse than the others, and it covers the whole range from

LMAP up to UMAP. This will be the preferred combination for the normal running of a terminal. However, if only $N \geq$ UMAP is considered then RMEFNR2 would be the preferred operator as it covers all that range, which no other one does. So for the static problem where the number of BSSs would normally be within the range of $N \geq$ UMAP the RMEFNR2 would be the preferred operator.

A summary of the results for those single operators, and for the 3 and 2 combined operators which perform well for all of the instances with $N \geq$ UMAP, and the combinations which cover a wider range of number of BSSs, are shown in Tables 5.15. The full summary tables for both data sets and topologies can be seen in Appendix B.5.3. These combinations perform better overall for the IS1ES. The results show that the 'Probability Single Multi Operator' (Section 5.4.3) performs better than a single operator in general with the appropriate combination of operators depending upon the data set and the number of Baggage Sorting Station Selections (BSSSs).

Operators	H1T091216	
	Max. (9, 5, 3)	
	3-pier	4-pier
	IS1ES	IS1ES
RMEFNR2	8 (0, **5**, **3**)	
RMEFNR2 0.8 + DSEMO 0.1 + C2P 0.1	7 (1, 3, **3**)	8 (3, 2, **3**)
RMEFNR2 0.8 + DSEMO 0.2	11 (6, 3, 2)	
RMEFNR2 0.9 + C1P 0.1	8 (1, 4, **3**)	9 (1, **5**, **3**)
RMEFNR2 0.9 + C2P 0.1	9 (2, 4, **3**)	
RMEFNR2 0.9 + DSEMO 0.1	9 (5, 1, **3**)	

Operators	H1T100301	
	Max. (6, 6, 5)	
	3-pier	4-pier
	IS1ES	IS1ES
MEBPFNR3 0.8 + C2P 0.2	5 (0, 0, **5**)	
MEFNR3 0.8 + C1P 0.2	6 (0, 1, **5**)	
MEFNR3 0.8 + DSEMO 0.1 + C1P 0.1	9 (2, 3, 4)	
MEBPFNR3 0.8 + DSEMO 0.1 + C2P 0.1		10 (4, 5, 1)
MEFNR3 0.9 + C1P 0.1	6 (0, 1, **5**)	9 (0, **6**, 3)

Table 5.15: Summary of the number of occurrences which cannot be said to be statistically significantly less fit than the others for a significance level of 0.05.

5.7.7 Trade-off Between Objectives

Figure 5.22 shows the non-dominated solutions obtained by different runs with single operators for 27 BSSs (UMAP) for the data set H1T091216, which illustrates the trade-off between distance and reduction in service. It shows that the improvement in one objective corresponds to a deterioration in the other. Given that the number of BSSs is the UMAP, then full assignment of all of the flights is achievable without needing to reduce the service time, which removes the need to plot the first and most important objective, the maximisation of the assignment. It should be noted that the first solution plotted corresponds to the situation where there is no reduction in service, which is possible given that 27 BSSs correspond to the UMAP.

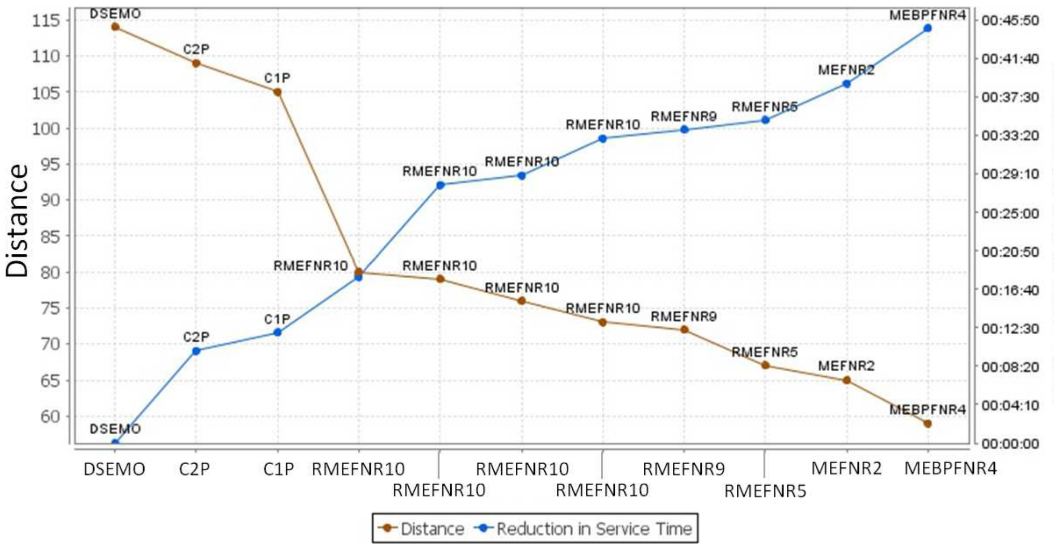


Figure 5.22: Trade-offs between objectives for 4-pier topology, 194 flights, 27 BSSs and SUMS for the operators MEBPFNRn, MEBPRNR10, MEFNRn, MERNR10, RMEFNRn, RMERNR10, C1P, C2P and DSEMO with $n \in [2 \ldots 10]$.

5.8 Conclusions

The aim of this chapter was to see how well the SSEA performs and to gain more general insights into the appropriate operator choices for the SSEA, especially since some operators (such as crossover) are slower to apply than others, and the appropriate operator percentages may differ depending upon the situation.

The SSEA, operators and selectors were presented. The empirical results for the SSEA show that this algorithm performs better than the other algorithms considered, which suggests a potential application to the problem under consideration as well as other resource assignment problems, such as the AGAP which is studied in Chapters 7 and 8.

The DSEMO extends the search to other areas of the search space which may help to improve the solutions, but it is only useful when there are unassigned flights, e.g. for $N < LMAP$. In the case of $N \geq LMAP$, the DSEMO should only be used when the solution selected from the population has unassigned flights, most commonly closer to the start of the search.

The different Multi Exchange Mutation Operators presented here do not have the ability to increase the number of assignments so for solutions which do not have maximum assignment and when maximum assignment is one of the most important objectives then these operators should not be used on their own. Given that each of the operators presented has particularities then these could be used to guide the search by deciding which operators should be considered, based on the stage the search has reached at each time, e.g. if the population at a specific point in the search contains only solutions with full assignment then the DSEMO operator should not be used.

The results presented in this chapter corroborate the importance of choosing a population size which is not only determined by the problem under consideration but also by the operator used. The best population sizes for different operators have been shown to be very different so there is potential for improving the performance of the algorithm when multiple operators are used by considering, for each particular operator, a sub-population of the size best suited to the operator.

Given the diverse ways in which the operators work, it is expected that their combination will further improve the solutions even when a significant difference was not shown as against use of only a single operator. Furthermore, the combination of different operators together with an adaptive method of selecting operators seems to be the most promising approach for future work. This approach could be extended further to consider the number of iterations in a generation (ℓ), the value of which could be adjusted as the search progresses, to take account of the particular situation at each time.

Future work should consider extending the model to examine the capacity of each BSS, so that a more realistic number of BSSs required to service each flight can be established. The number of BSSs for each flight may initially be obtained from historical data giving the number of passengers and baggage. Furthermore, better results and robustness may be obtained if the number of BSSs required for each flight is not fixed, but depends on the capacity of the BSSs assigned to each flight and the expected checked-in baggage load each time. This means that the model not only evaluates the BSSs assigned to each flight but also when each assignment should commence, since they may not start at the same time, thus increasing the availability of the BSSs for use in servicing other flights or absorbing disruption on the day of operation. It has been assumed that the end of the service time for all the BSS assignments to the same flight will also be the same, as it is anticipated that the volume of checked-in baggage increases as it nears the check-in desk closing time and the time for flight departures.

Chapter 6

Robustness in Assignment of Airport Baggage Sorting Stations

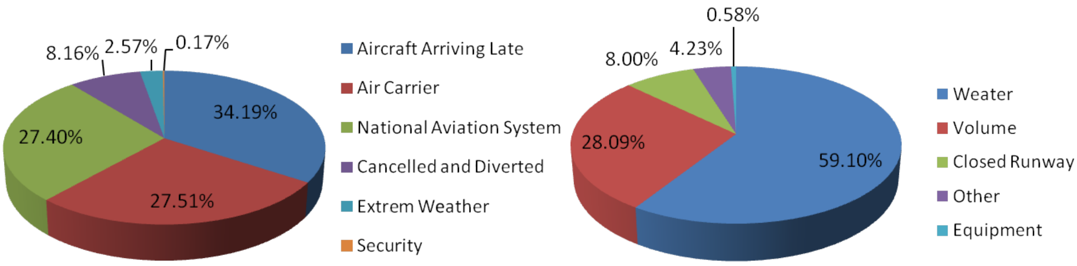
A conflict occurs when two flights originally assigned to the same baggage sorting station (BSS) have overlapping service times. Conflicts depend on the original assignments for the real arrival and departure flight times on the day of operation. It is therefore desirable to consider potential delays on the day of operation when generating the original flight assignments to BSSs, such that the final flight assignments differ little or not at all from the original assignments on the day of operation. The term robustness is here used to give an indication of the degree to which this has been achieved.

This chapter investigates some existing approaches and suggests others for generating assignments which take account of potential perturbations on the day of operation.

This chapter begins with an overview of the problem, followed by an examination of one of the most common approaches considered in the literature. A number of approaches are then introduced which take account of potential conflicts in the overall assignment of flights to BSSs (the Airport Baggage Sorting Station Assignment Problem (ABSSAP)) in Section 6.3. The main objective is to reduce the number of reassignments which may require to be performed once the schedule is put into practice on the day of operation. Some experiments are then conducted to determine the efficiency of these approaches to the ABSSAP in Section 6.4. The final section of this chapter provides some conclusions.

6.1 Overview

Flight delays are caused by many factors such as airport security, weather conditions, unavailability of required resources (mechanical breakdown), delayed propagation, airport congestion, etc. Figure 6.1 shows the delays caused, reported by the Research and Innovative Technology Administration (2012).



a Flight Delays by Cause.   b Causes of National Aviation System Delays.

Figure 6.1: Flight delays for December 2011 to November 2012, Source: Research and Innovative Technology Administration (RITA)

Congestion also plays an important part in flight delays, being of two main types, firstly airspace congestion and secondly airport congestion. Both types of congestion have a direct or indirect impact on ground delays.

It is currently believed to be more advantageous to move potential airspace congestion back to the gate of the departure airport, given that this makes a significant reduction in cost and a potentially favourable contribution to safety. The airlines also incur an extra cost when extending a flight's air time, such as in the extra fuel consumed, increase in staff working hours, and potential penalties imposed by the destination airport for arriving late. There are nevertheless indirect costs, such as customer satisfaction and impact on other resources, gates and BSSs. Extending the time in which a flight uses the gate facilities may have a repercussion on the following flight assigned to the same gate, and that delay may spread to other gates and resources such as BSSs. Once the baggage has been loaded into the aircraft hold, the BSSs assigned to that flight are ready for use on the next flights scheduled, and any delay on a flight at the gate will not necessarily have repercussions on these. However, such delays may affect the BSSs assigned to the flight scheduled for that gate next if such a flight is held waiting for the gate to become free. BSSs will then need to hold the baggage longer, thus potentially affecting the following assignment in turn. Holding flights on gates for longer than originally scheduled is a situation discussed in Sections 7.5.5 and 8.4.

A scheduled assignment is said to be in conflict if the completion of its service time is greater than the commencement of the service time of the next assignment at the same BSS. When a delayed flight affects the assignment of subsequent flights to the BSS, then there are two ways it can be corrected: either to reassign the conflicting flight or reassign the flight subsequent to the conflicting one. When reassigning a conflicting flight or subsequent flights to another BSS, a situation may arise where the reassigned flight is in conflict with the subsequent flight at the new BSS. Some reassignment may therefore have a downstream effect on the overall schedule, producing further conflicting flights requiring further reassignments, thus potentially increasing the problem difficulty later on.

In the ABSSAP, those flights which are late in arriving at their assigned stand are considered to be a perturbation, since baggage cannot be loaded into the aircraft at the scheduled time, and needs to be held longer at the BSS. Any extension of an aircraft's stay on its assigned gate should not have an effect on its assigned BSS, as the baggage should have already been loaded into the aircraft already, such that the BSS is free for use in its next assignment. This means that not all aircraft delays will affect their assigned BSS.

The main objective of this is to reduce the number of BSSs which have to be reassigned on the day the schedule is put into practice. In the ABSSAP, e_j is the end of the service time of flight j, τ_j is the base starting service time of flight j, and the variable y_{ij} has a value of 1 if flight j is assigned to BSS i, or y_{ij} is zero otherwise, as described in Section 3.3.2. A new decision variable x_{jk} is introduced with a value of 1 if flights j and k are assigned to the same BSS (ie. if $y_{ij} = y_{ik} = 1$ for $i \in [1 \ldots N]$ and $j, k \in [1 \ldots M]$), or x_{jk} is 0 otherwise. On the day the schedule is implemented the real times for the flights j and k correspond to $e_j^{'}$, $\tau_j^{'}$, $e_k^{'}$ and $\tau_k^{'}$ for flights j and k respectively, where $e_j^{'} \geq e_j$ and $\tau_k^{'} \geq \tau_k$, and the flights ordered by their base starting times, for $j < k$ and $j,\ k \in [1 \ldots M]$ then $\tau_j \leq \tau_k$. A conflict occurs when two flights j and k with $j < k$ and $j,\ k \in [1 \ldots M]$ originally assigned to the same BSS (ie. $x_{jk} = 1$ with $i \in [1 \ldots N]$) have overlapping service times, i.e. $e'_j > \tau_k^{'}$ and $\tau_j^{'} < e_k^{'}$, as shown in Figure 6.2. Conflicts depend on the original assignments for the real arrival and departure flight times on the day of operation. It is therefore desirable to consider potential delays on the day of operation when generating the original flight assignments to BSSs, such that the final flight assignments differ little or not at all from the original assignments on the day of operation. The term robustness is normally used to give an indication of the degree to which this has been achieved.

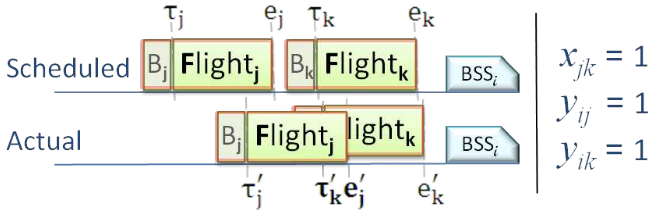


Figure 6.2: Example of conflict between two flights originally assigned to the same BSS.

6.2 Buffer Time

There are different ways of increasing robustness depending on the intended effect. One of the simplest and most frequently used methods involves the introduction of a buffer time between assignments which permits absorption of small disturbances. Mangoubi and Mathaisel (1985) proposed the use of 'buffer times' between two flights which are assigned consecutively to the same gate in order to obtain robust assignments. Wu and Caves (2000) showed the significance of an appropriate use of schedule buffer time in maintaining schedule punctuality performance. Yan and Huo (2001) applied buffer time to the Airport Gate Assignment Problem (AGAP), and concluded that the length of the buffer time significantly influences the gate assignment process. Thus an appropriate value should be used, which is discussed in Section 6.3.1. Other approaches for improving the robustness relate to the distribution of idle time, which is presented in Section 6.3.2, and the reduction of reassignment on disruption, which is presented in Section 6.3.3.

The amount of buffer time may take different contributory factors into account, which could perturb the schedules of handlers, airline, airport origin, destination and flight. The size of the flight is normally related to the travel distance, longer distances presenting a higher probability of disturbances, which may accumulate generating a higher level of delays. Shorter distances present less chance of disturbances. The location of the originating and destination airports has a direct effect on potential disruptions, given that they place a constraint on the permitted routes possible and certain circumstances applicable to them, such as weather patterns during summer and winter, or on the equator.

The buffer time may be fully located at the beginning of the base service starting time, at the end, or apportioned between either extremity. The position of the buffer time in respect to the base service duration only matters when the buffer times do not have the same duration, as in the case considered here. An example where the buffer times are located at the beginning of the base service duration is shown in Figure

6.2. If a departing flight arrives at its assigned gate earlier than scheduled, and there is no change in the departure time, this change does not affect the assigned BSS. Similarly this would happen where the departure time changes, but the time during which the flight is at the gate is sufficient for all of the baggage to be transported from the BSS to the aircraft. It only affects the assigned BSS when the available BSS service time is no longer sufficient to complete the service, which will normally occur when there are delays due to the aircraft arriving late at its assigned stand. The flight will consequently also leave late, given that there is a minimum time required to complete all the necessary operations before the aircraft is ready to depart. This means that placing the buffer time at the end of the service time would be preferable. Conversely in the case of a flight arrival it would be preferable to begin processing the passengers' baggage as early as possible, which may well mean commencing use of the assigned BSS earlier, consequently requiring a change in the BSS assignment where this is not possible.

A non-linear cost for service time reduction may also help to reduce the number of conflicts on the day of operation, so that fewer and larger reductions are more heavily penalised than many smaller ones, since large reductions in the buffer time are far less favourable than smaller reductions, as presented in Sections 6.3.4 and 6.3.5. Following this, some stochastic approaches are presented, Sections 6.3.6, 6.3.7 and 6.3.8.

6.3 Robustness Approaches

In this section some robustness approaches are presented and others are suggested. These are then studied and compared in the following section.

6.3.1 Minimise Reduction in Service Time

It may be possible to gain robustness by re-ordering assignments between BSSs so that 'idle time' between flights consecutively assigned to the same BSS is greater, as shown in Figure 6.3.

Figure 6.3 shows two potential solutions with different robustness. While any delay to flight 'a' in the 'less robust' solution will certainly affect flight 'b', which will in turn have to be reassigned to another BSS, in the 'more robust' solution a delay in flight 'a' may not affect flight 'b'. So the 'more robust' solution is preferable to that of the 'less robust' solution.

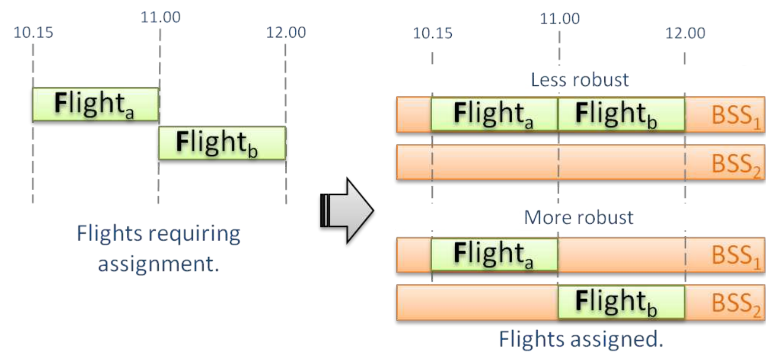


Figure 6.3: Simple example of two schedules, with the same flights, where one is obviously more robust in respect of perturbations than the other.

Mangoubi and Mathaisel (1985) proposed the use of 'buffer times' between two flights which are consecutively assigned to the same gate in order to obtain robust assignments, defining the reduction in service as that part of buffer time which overlaps with the previous assignment to the same gate. Given the detrimental effects that the reduction in service time has on the robustness of the assignment as against real-life delays, it is advisable to minimise the total reduction in service time, thus maximising buffer times. This objective can be expressed by Formula (6.1).

$$\min \sum_{j=1}^{M} \sum_{p=1}^{P_j} r_{jp} \tag{6.1}$$

The reduction in service could be extended to cover unassigned activities, by giving each unassigned activity a cost equal to a factor times the maximum reduction in service time, as the main purpose is to ensure that solutions with fewer assignments are never better than those with a higher number of assignments, Equation 6.2. Consideration of a factor of two ($\beta = 2$) represents inserting a new assignment exactly between two flights so that the reduction in service will be the full buffer time for both the new assignment and the flight assigned next. Therefore unassigned flights with higher buffer times are not penalised more than those with lower buffer times, i.e. those long-haul and short-haul flights considered here.

$$\min \left(\sum_{j=1}^{M} \sum_{p=1}^{P_j} r_{jp} + \beta * \max_{k=1}^{M} (B_k) * \underbrace{\sum_{j=1}^{M} \sum_{p=1}^{P_j} \left(1 - \sum_{i=1}^{N} y_{ijp} \right)}_{\text{number of unassigned activities}} \right) \tag{6.2}$$

If a fitness function is defined as a weighted sum of the different objectives as used both here and in Chapter 5, and the robustness uses the reduction in service time only for assigned flights, then particular weights may compromise the importance of the main objective (maximisation of the number of assignments), where solutions with lower numbers of assignments are favoured over those with higher numbers of assignments because of the robustness objective. When the robustness objective also takes account of the unassigned flights, i.e. Equation 6.2, the selection of weights for these two objectives will be decoupled, so it will be easier to assign a value to them.

The use of Formula 6.1 treats any reduction in service equally, so it does not make a distinction between reducing all an assignment's buffer time and allocating it to another flight from another solution where both flights share the available 'idle time', as shown in Figure 6.4. However, the flight 3 with maximum reduction in service in solution 'b' (Figure 6.4) will be unable to absorb any delay on the day of operation, although the same flight in the alternative solution 'a' (Figure 6.4) will be able to do so, making it the preferable choice.

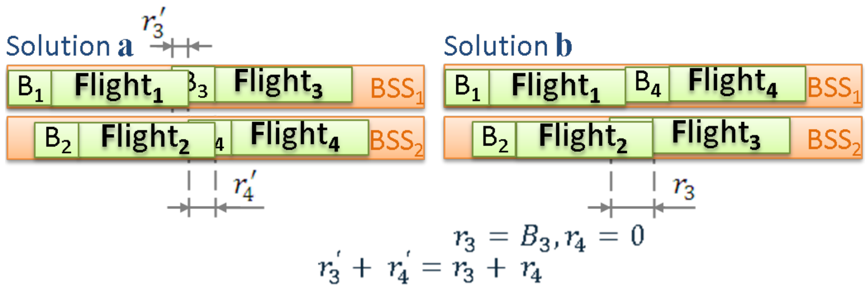


Figure 6.4: Simple example of a more robust schedule of four flights and two sorting stations using the reduction in buffer time.

A non-linear penalty function is required to take account of this, such as Formula 6.4.

$$u_{ijp} = \begin{cases} \arctan\left(\frac{\tau_j - e_k}{B_j}\right) - \frac{\Pi}{2} & \text{if } k < j,\ y_{ijp} = y_{ikq} = 1 \text{ and} \\ & \sum_{l=k+1}^{j} \sum_{p=1}^{P_l} y_{ilp} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

$$\max \sum_{i=1}^{N} \sum_{j=1}^{M-1} \sum_{p=1}^{P_j} u_{ijp} \tag{6.4}$$

Formula 6.3 defines the penalty u_{ijp} for assigning activity p of flight j to sorting station i, with the total penalty being represented by Formula 6.4. There is only a penalty for consecutive assignments to the same sorting station, between the assignment and the previous assignment to the same sorting station. The reason for using arctangent concerns the properties of this function which imposes stronger penalties around the point at which the flights are assigned service time without any buffer time, to the point at which all the buffer time is retained (by dividing the member of arctangent by B_k). The effect is reduced as the time separation between the assignments increases, but the contribution of any time separation is always considered negative. The constant $\frac{\pi}{2}$ is used so that there is always a cost associated with assigning two flights to the same sorting station, otherwise it would be a benefit, Figure 6.5. Considering the arctangent for each flight increases the individual effect in the overall objective, in contrast to using the arctangent of the sum of all of the flight contributions.

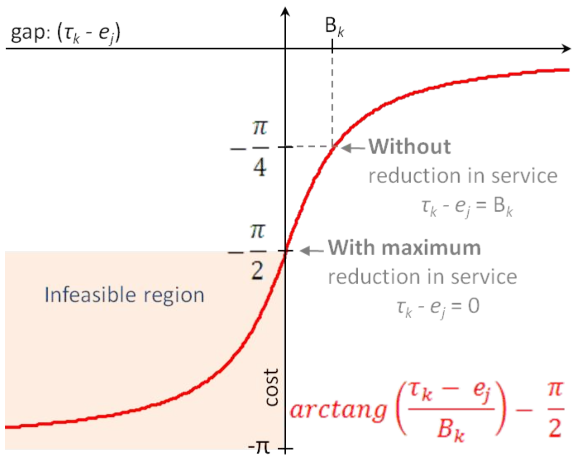


Figure 6.5: Representation of u_{ijp}, $j < l$, $y_{ijp} = y_{ilq} = 1$, $j, l \in [1 \ldots M]$, $p \in [1 \ldots P_j]$ and $q \in [1 \ldots P_l]$.

The robustness objective represented by Formula 6.4 also takes account of the objective of 'Minimising the Service Reduction', as being the sum of the reduction in buffer time of each flight assigned.

The main issue with the arctangent function is the time it takes to obtain the values. This could be speeded up while still maintaining the desirable characteristics. It can be reduced by using a piecewise linear cost function as shown in Figure 6.6, composed of penalisation segments.

The idea of extending the penalisation beyond a reduction in service is introduced here to increase the robustness of assignments where possible, as a wider separation between assignments to the same BSS reduces the chance of a delay or early arrival affecting the assignments.

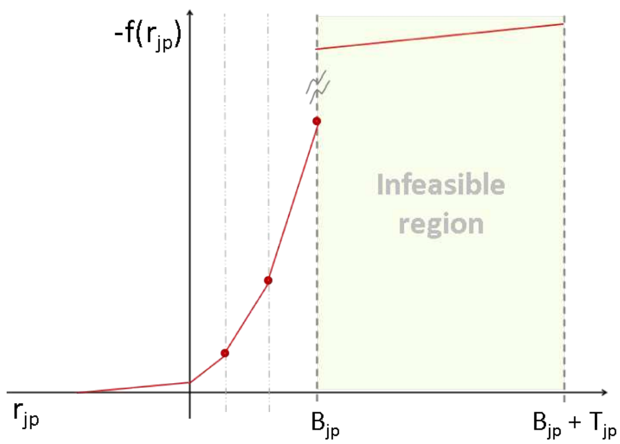


Figure 6.6: Cost of the Reduction in Service.

6.3.2 Distribute Idle Time

Bolat (1999) proposed the distribution of 'idle time' uniformly amongst gates. In the case of the ABSSAP the distribution of the 'idle time' uniformly amongst BSSs can be considered, where 'idle time' is calculated as the time between the start of service time on a flight's assignment to a BSS and the end of service time of the flight assigned immediately previous to the same baggage sorting station, Formula 6.5. Whereas the buffer time implies preference for a particular gap size between consecutive assignments to the same sorting station, an 'idle time' does not. When using the 'idle time' it is normally intended that the gap size should be as large as possible. The reason for this is to increase the probability that even with a delay, completion of service time for flight presently assigned will still be earlier than the start of service time for the next flight assigned to the same BSS.

$$\Delta t_j = (s_j - e_k) \text{ for } \begin{array}{l} j < k, y_{ijp} = y_{ikq} = 1 \text{ where } \nexists\, l \in [k \ldots M] \text{ with } \tau_l < e_k \\ j, k \in [1 \ldots M], p \in [1 \ldots P_j], q \in [1 \ldots Pk] \text{ and } i \in [1 \ldots N] \end{array} \tag{6.5}$$

$$v^+ = \max\{\Delta t_j | j \in [1 \ldots M]) \tag{6.6}$$

$$v^- = min\{\Delta t_j | j \in [1 \ldots M]) \tag{6.7}$$

$$\min(v^+ - v^-) \tag{6.8}$$

Formulas 6.6 and 6.7 refer to the maximum and minimum 'idle time' for a solution respectively, while Formula 6.8 represents the objective as the difference between both the maximum (v^+) and minimum (v^-) 'idle times' (Δt_j, Equation 6.5) for the same solution. Figure 6.7 shows a simple example of the robustness of two solutions, where solution 'a' is more robust than solution 'b'. In solution 'a' the delay of flight 3 will not affect flight 4 and needs to be considerably larger to affect flight 5, whereas in solution 'b' small delays in flight 3 will affect flight 4.

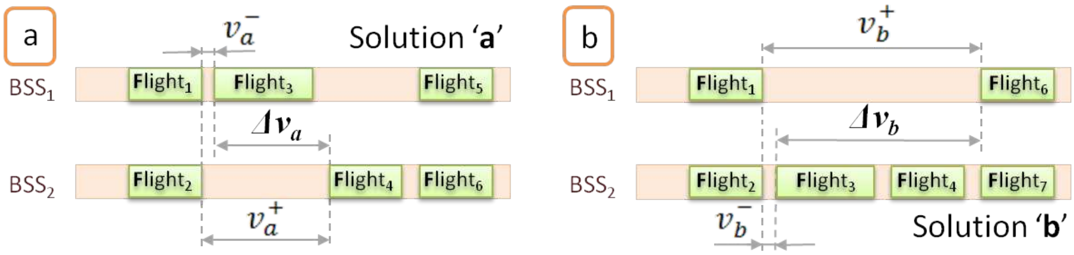


Figure 6.7: Examples of a range of idle times for different solutions, where solution 'a' is clearly better than solution 'b' for Formula 6.8.

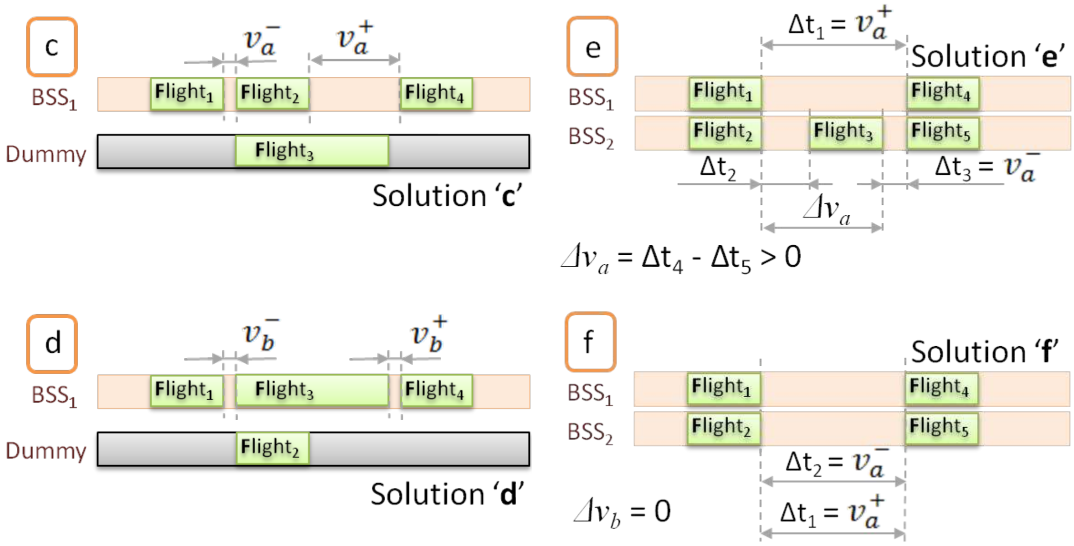
Formula 6.8 assumes that all flights within a solution have been assigned, which may not be the case in certain circumstances, where the maximum possible assignment is lower than a full assignment (simulation) or where the initial solution(s) has some flights remaining unassigned, and an example of this is shown in Figure 6.8a. It should also be observed that this objective may conflict with the maximum assignment objective (3.3.4), as shown in Figure 6.8b, where solution 'f' has a smaller $\triangle v$ than solution 'e', which is based on Formula 6.8. This means that solution 'f' is considered more robust, but solution 'e' would be preferable because it achieves more assignments. Thus it will be necessary to select the objective weights appropriately, where the fitness is a weighted sum of the different objectives, in order to ensure the correct selection of the solution.

Bolat (2000) and Bolat (2001) extended the model by minimising the variance of the idle times, Formula 6.9.

$$\min \sqrt{\sum_{j=1}^{M} \left(\triangle t_j - \underbrace{\frac{\sum_{j=1}^{M} \triangle t_j}{M}}_{mean\ `idle\ time'} \right)^2} \tag{6.9}$$

6.3.3 Reduce Reassignment on Disruption

The ability to reassign all flights directly affected by a disruption is desirable, without the need to reassign other flights. The intention here is to generate schedules which take account of this objective, allowing such reassignment to be performed more frequently.



a Solutions without full assignment. b Solution 'f' is more robust than solution 'e'.

Figure 6.8: Examples of a range of idle times for different solutions using Formula 6.8.

One way to achieve this objective would be to count the number of assignments between which a reassignment could be placed when necessary. Whether the reassigned flights are on the same pier/side, and how many reassignments could be absorbed by a pair of assigned flights, must all be taken into account. Figure 6.9 shows the ability of the 'idle time' between the two flights 1 and 8 to accommodate flights 3, 4 and 5, should one of them be delayed. Its reassignment to BSS 1 may be sufficient, thus avoiding transfer of any delay to other assignments.

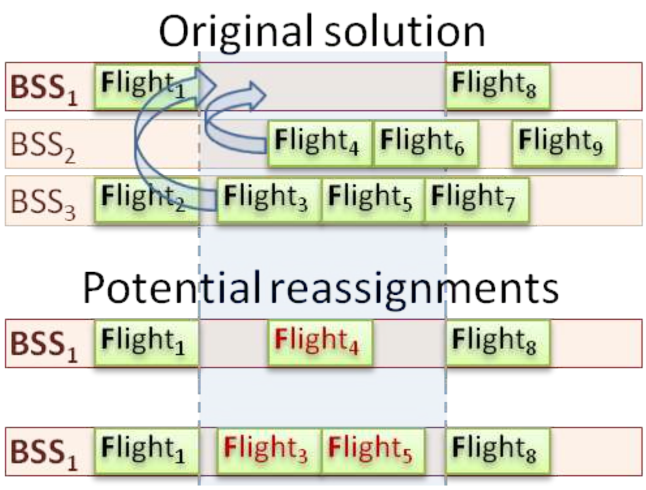


Figure 6.9: An example of the capacity to absorb reassignments.

The following model is proposed whereby the capacity to absorb reassignments may be achieved by weighting each reassignment by the inverse of 1 plus the distance between the BSSs (d'_{in}, presented in Section 3.3.2), given that such distance (d'_{in}) may be zero, where all flights are ordered by their base start time (τ_j), Equations 6.10 and 6.11.

The intention is to use the number of flights which could be reassigned between two flights already consecutively assigned to the same BSS, without the need to reassign either of these flights in order to achieve this. Equation 6.10 states that φ_{jkl} is equal to 1 if flight k's base service duration does not overlap with the base service duration of both flights j and l ($j < l$), both of which are assigned to BSS i, and there is no other flight r between these ($j < r < l$) already assigned to the same BSS as flight j, otherwise φ_{jkl} is zero, Equation 6.10.

$$\varphi_{jkl} = \begin{cases} 1 & \text{if } j < k < l,\ e_j \leq \tau_k \text{ and } e_k \leq \tau_l\ \forall j,k,l \in [1 \ldots M] \\ & \quad \nexists\, r \in (j \ldots l),\ \text{ with } y_{ijp} = y_{irq} = 1 \text{ for any } p \in [1 \ldots P_j] \\ & \quad \text{and } q \in [1 \ldots P_r],\ \text{ and } y_{ikz} = 0 \forall z \in [1 \ldots P_k] \\ 0 & \text{otherwise} \end{cases} \tag{6.10}$$

The objective is to maximise Formula 6.11, which weights the contribution of each potential reassignment, based on to which BSS the reassigned flight was originally assigned. The underlying idea is that closer reassignments are preferred to more distant ones, but the ability to reassign without affecting other assignments is preferable.

$$\sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{p=1}^{P_j} \sum_{l=j+1}^{M} \sum_{q=1}^{P_l} \sum_{k=j+1}^{l-1} y_{ijp} * y_{ilq} * \varphi_{ikl} * \underbrace{\sum_{n \neq i}^{N} \sum_{z=1}^{P_k} \frac{y_{nkz}}{d'_{in} + 1}}_{effect\ of\ the\ BSS\ assigned} \tag{6.11}$$

The representation in Formula 6.11 also needs to include the border cases relating to the first and last assignments in a BSS as shown in Figure 6.10.

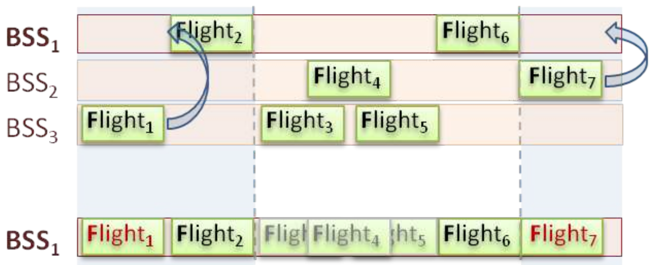


Figure 6.10: An example of border assignments in baggage sorting stations.

The previous formula could be extended to cover the border cases by assigning two extra dummy flights to all available BSSs; first, $j = 0$, with the end time being the start of the time period studied, and the second flight, $j = M + 1$, with the start time being the completion of the time period studied, which for our time period would be $e_0 = 0$ and $(e_{M+1} - T_{M+1}) = 24$hr, with $y_{i0} = y_{i(M+1)} = 1 \; \forall i \in [1 \dots N]$, Formula 6.12.

$$\sum_{i=1}^{N} \sum_{n \neq i}^{N} \left(\frac{\sum_{j=0}^{M} \sum_{p=1}^{P_j} \sum_{l=j+1}^{M+1} \sum_{q=1}^{P_l} \sum_{k=j+1}^{l-1} y_{ijp} * y_{ilq} * \varphi_{ikl}}{d_{in}^{BSSs} + 1} \right) \tag{6.12}$$

6.3.4 Area Reduction in Service

Bolat (1999) examined minimisation of the range of idle time, and the difference between the maximum and minimum idle times for the AGAP, which was later extended by Bolat (2000) to consider both minimisation of the idle time range and minimisation of the idle time variance (Section 6.3.2). However these do not take account of the influence or effect which the disruptions have on the schedule due to the time of their occurrence, as all reductions in service time are treated as being the same, irrespective of the time period considered in the whole 'planned schedule' for the given set of departures. It is anticipated that the more congested time periods in the 'planned schedule' will also represent periods where disruptions are more likely to occur and propagate, extending their effect and further increasing costs.

As discussed in Section 3.3.2 the Lower Maximum Assignment Point (LMAP) is the minimum number of sorting stations needed for assignment of all flights without using buffer times. Similarly, the Upper Maximum Assignment Point (UMAP) is the minimum number of sorting stations needed to assign all flights without reducing the target service time (base service duration plus buffer time) as presented in Section 3.3.2. Their values are an indication of the difficulty of the problem, and these may be obtained from the distribution of the number of flights requiring service over time, as shown in Figures 6.11 and 6.12.

The effect of service reduction is not the same throughout the day, but depends on the time of day. It is more likely that disruptions will occur during periods when the flight density is higher than when fewer flights require servicing, i.e. delay during high flight density is more likely to propagate given that less resources will be available to absorb any reassignment without repercussions on other flights. At the same time, it is these cases where it is most difficult to keep a sufficiently large gap between

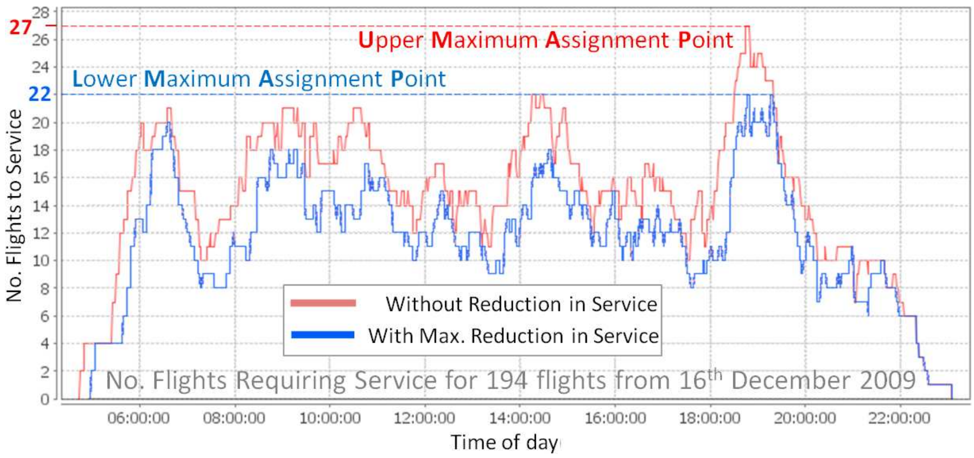


Figure 6.11: Flight distributions with LMAPs and UMAPs for 194 flights on 16^{th} December 2009 at Terminal 1 of London Heathrow airport.

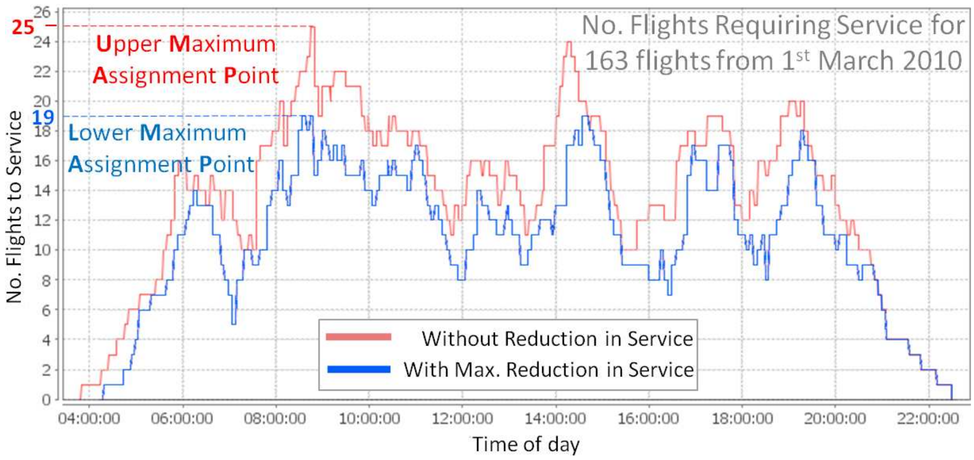


Figure 6.12: Flight distributions with LMAPs and UMAPs for 163 flights on 1^{st} March 2010 at Terminal 1 London Heathrow airport.

assignments to the same BSS. In assessing the importance of the time of day when service reduction is performed, it is suggested that it would be advisable to increase penalisation of flight assignments with reduced service time when there is a higher flight density. This can be accomplished by calculating the required number of BSSs at different times of the day, as shown in Figures 6.11 and 6.12.

Average Assignments Point: The function $f_u(t)$ refers to the distribution of flights over time with t_s being the schedule starting time and t_e the schedule end time, as shown in Figure 6.13. The Average Assignment Point (AAP) is here defined as the number of BSSs for which the distribution of flights would be uniform, which can be calculated by Equation 6.13.

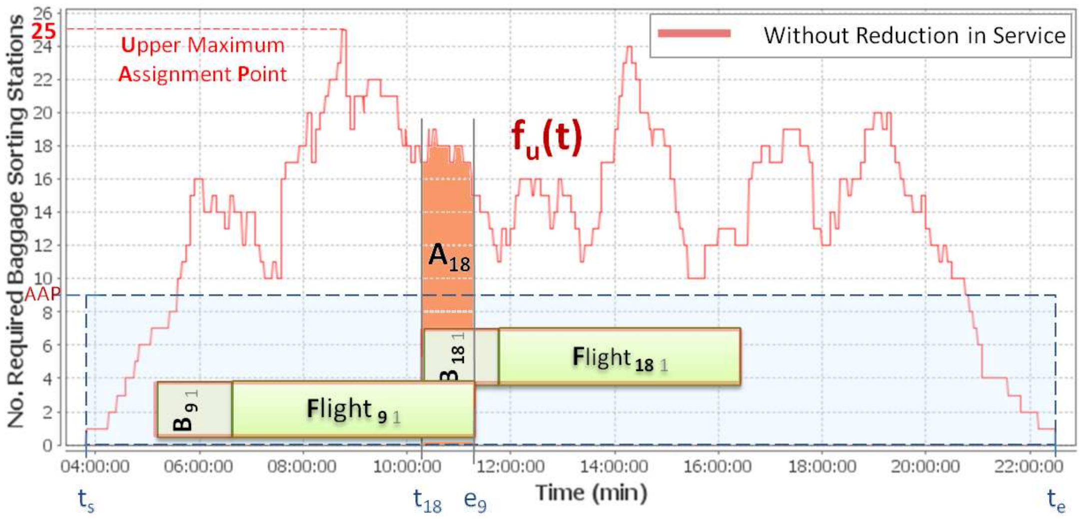


Figure 6.13: Distribution of flights over time and Area Reduction in Service Objective.

$$AAP = \frac{\int_{t_s}^{t_e} f_u(t)\ dt}{t_e - t_s} \tag{6.13}$$

A_j is the density distribution area for the time period from the target service time of flight j (t_j) to the end of service time for the previous flight assigned to the same BSS, for example in Figure 6.13 $A_{18} = \int_{t_{18}}^{e_9} f_u(t)\ dt$. AAP_j is here defined as the mean number of flights over the target start time for flight j and the end of service time for the previous flight assigned to the same BSS as flight j, e.g. $AAP_{18} = \frac{A_{18}}{e_9 - t_{18}}$. So the contribution to the objective for assignment j is $\frac{AAP_j}{AAP}$ times the reduction in service previously considered in Section 6.3.1. This corresponds to values greater than one for dense flight regions of the schedule, and less than one for under-used regions.

The approach proposed intends to increase penalties for those reductions in flight service time during time periods where more flights require servicing than those periods with less service load by means of the AAP and the distribution of flights over time, Equation 6.14. The idea is that flights which require servicing during congested periods are more likely to have a knock-on effect than those in less congested periods. It is therefore preferable not to reduce the service time of flights at more congested times so much, in order to limit the effect of potential delays.

$$f_2 = -\frac{1}{AAP} * \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{p=1}^{P_j} y_{ijp} * A_j \tag{6.14}$$

This approach is compared with some of the other approaches introduced in this chapter in Section 6.4.1.

6.3.5 Sub-Area Reduction in Service

In this case, the objective is based on the area between both flight densities 'Without Reduction in Service Time' and 'With Reduction in Service Time' shown in Figure 6.14, and called a sub-area. The flight density sub-area corresponds to the area from the flight target start time (described in Section 3.3.2) to the previously assigned flight's end service time, which lies between both flight density distributions (f_u and f_l), where a reduction in service time is not permitted and when all the buffer time has been reduced (considering only the base service duration, T), Equation 6.15.

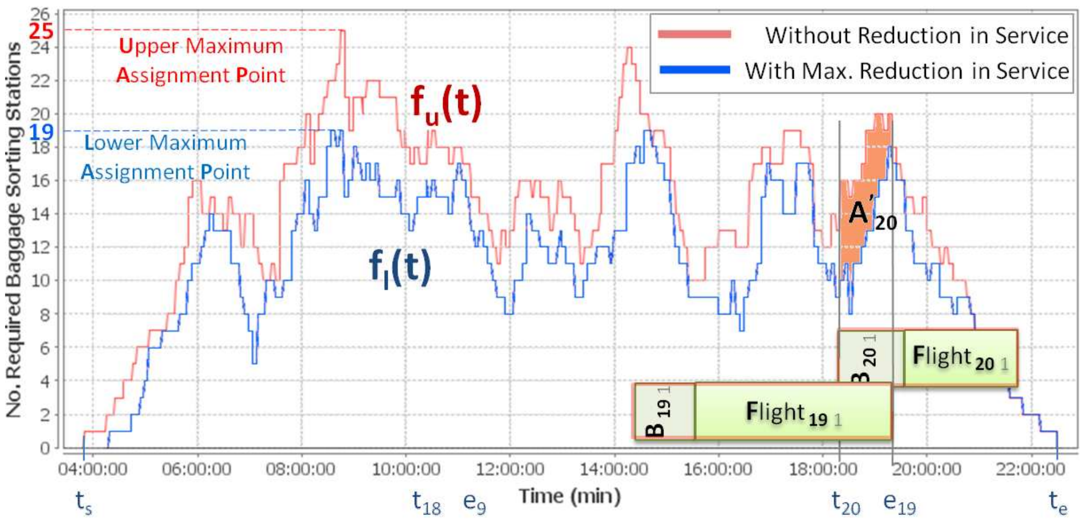


Figure 6.14: Sub-Area Reduction in Service Objective.

$$A'_j = \int_{t_j}^{e} (f_u(t) - f_l(t)) \ dt \tag{6.15}$$

The approach now proposed has a fitness which covers the sub-area divided by the difference between the UMAP and the LMAP for all the assigned flights, Equation 6.16

$$f_2 = -\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{p=1}^{P_j} y_{ijp} * \frac{A'_j}{UMAP - LMAP} \tag{6.16}$$

Similarly to the Area Reduction in Service (Section 6.3.4), the main idea is to penalise more heavily those reductions in flight service time which occur in regions with high flight density, as these are more likely to further disrupt the schedule in case of delays. Adding both approaches together with the Total Reduction in Service Time (TRS) will be seen in Section 6.4 to increase the robustness as compared to using each approach individually.

Nevertheless, in cases where the LMAP is equal to the UMAP, the Sub-Area of Reduction in Service (SARS) approach cannot be used, and an example is shown in Figure 6.15. In these cases, the area between both flight densities could be used, an approach that is here named Base Sub-Area Reduction in Service (BSARS). As will be seen in Section 6.4.1, this approach significantly increases the robustness and widens the range of the quantity of BSSs, in which it performs better when compared to the Area of Reduction in Service (ARS), TRS and SARS.

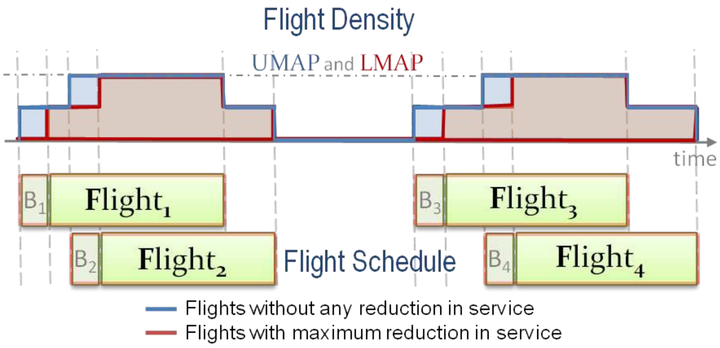


Figure 6.15: Example of distributions with the same LMAP and UMAP.

These approaches are compared with some of the other approaches introduced in this chapter in Section 6.4.1.

6.3.6 Unsupervised Estimated Stochastic Reduction in Service

Lim and Wang (2005) proposed a stochastic programming model for the AGAP with a robustness cost of conflicts, which is estimated by a function, $v(j, k)$. Flights are ordered by their base starting service time, so the gap between two flights j and k, $l(j, j)$, assigned to the same BSS, where $j < k$ and $j,\ k \in [1 \dots M]$, is the difference between flight k's target service time and the prior assigned flight j's end service time, Equation 6.17, where $l(j, k) = -r_j$ for $l(j, k) < 0$, as shown in Figure 6.16. $v(j, k)$ is used to estimate the mean conflict probability between flights j and k assigned to the same BSS, which is a function of the gap $l(j, l)$, where larger gaps between assignments to the same BSSs result in lesser probability of real flight conflicts. $v(j, k)$ is normalised in Equation 6.18.

$$l(j, k) = t_k - e_j \qquad (t_k = \tau_k - B_k) \tag{6.17}$$

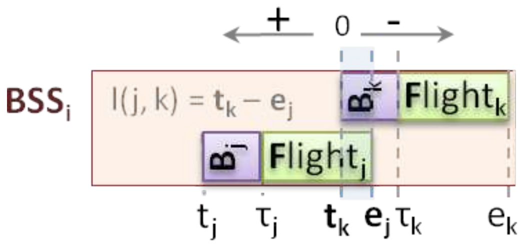


Figure 6.16: Overlap between two flights j and k assigned to the same BSS.

$$E(p(j,k)) = \sum_{i=1}^{N} \left(y_{ij} * y_{ik} * \frac{v(j,k) - v_{min}(j,k)}{v_{max}(j,k) - v_{min}(j,k)} \right) \tag{6.18}$$

$$f_2 = -\sum_{j=1}^{M-1} \sum_{k=j+1}^{M} E(p(j,k)) \tag{6.19}$$

The definition of $v(j,k)$ comes from the application domain, in the absence of historical data; some unsupervised estimation functions were introduced in Lim and Wang (2005). Figure 6.17 shows the penalty (y-axis) incurred for different unsupervised estimation functions as a function of the gap (x-axis). Negative values refer to reductions in service time between two assignments to the same BSS, which are heavily penalised as they may require reassignment should delays occur, whereas positive gaps are penalised less. Wider gaps between two assignments reduce the need to reassign delayed flights, given that the delay has to be larger than the gap in order to affect the following assignment to the same BSS. Similarly, to start the service earlier may not require the flight to be reassigned because the duration of earliness has to be lower than the gap in order to affect the previous assignment. Both earliness and delay probabilities decrease as the gap increases. Sufficiently large gaps may also be used on the day of operation by disrupted flights which need to be reassigned, such that the detrimental effect of disruptions on that day is reduced.

The unsupervised estimation functions introduced in Lim and Wang (2005) are presented below and are shown in Figure 6.17:

1. Linear estimation:
$$v(j,k) = -l(j,k) \tag{6.20}$$

2. Exponential estimation:
$$v(j,k) = e^{-\beta * l(j,k)} \tag{6.21}$$

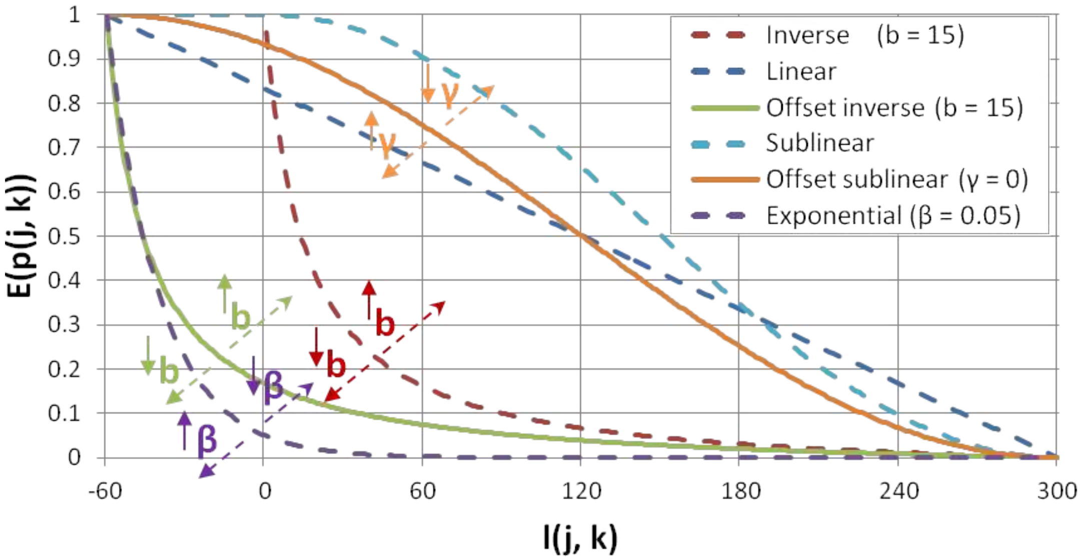


Figure 6.17: Penalty for different unsupervised estimation functions based on the gap between assignments.

3. Inverse estimation:

$$v(j,k) = \begin{cases} \frac{b}{l(j,k)+b} & \text{if } l(j,k) > 0 \\ 1 & \text{otherwise} \end{cases} \tag{6.22}$$

The value of the constant 'b' changes the penalisation as shown in Figure 6.17, so a higher 'b' increases the penalisation and a lower 'b' decreases it. An appropriate value should be selected to properly weight the influence of the potential conflicts. Lim and Wang (2005) used $b = 15$ minutes, which proved to provide rather poor results when compared with the exponential estimation function, which may partly be caused by the fixed cost when $l(j,k) < 0$ (dark red dash line, Figure 6.17), whereas in the exponential estimation function (purple dash line, Figure 6.17) this is not the case. The value used for 'b' may be too great, and a lower value would make this estimation function provide values closer to those provided by the exponential estimation function which provided fitter solutions in the results presented by Lim and Wang (2005). Consequently, a value $b = 6$ was seen in the experiments studied in Section 6.4.2 to provide better results than when $b = 15$. In general an even lower value did appear to perform better in some instances but not as well as $b = 6$, as shown in Section 6.4.2.

The inverse estimation function as considered by Lim and Wang (2005) treats all gaps smaller than the buffer time equally, which does not represent a real case since smaller gaps between flights are more likely to result in conflicts than larger ones on the day of operation. Given this, and that the exponential estimation function performs best and treats all gaps differently, it is proposed that all of the gaps be treated differently, as shown by the modified version which is herein named 'Offset inverse', Equation 6.23, which is shown in Figure 6.17 for $b = 15$ (green line).

$$v(j,k) = \frac{b}{l(j,k) - \min\{l(j,k)\} + b} \tag{6.23}$$

4. Sublinear estimation:

$$v(j,k) = \begin{cases} \cos\left(\frac{\pi * l(j,k)}{l_{max}}\right) & \text{if } l(j,k) > 0 \\ 1 & \text{otherwise} \end{cases} \tag{6.24}$$

This estimation also suffers from the same problem as the Inverse estimation, and may be improved by offsetting its value so that the maximum penalisation corresponds to l_{max} and the minimum to l_{min}, Equation 6.25, which is shown in Figure 6.17 for $\gamma = 0$.

$$v(j,k) = \cos\left(\frac{\pi * (l(j,k) - l_{min} + \gamma)}{l_{max} - l_{min} + \gamma}\right) \tag{6.25}$$

The gap definition used takes account of the buffer time, as the target service duration is the base service duration (T_k) plus the buffer time (B_k) for the flight. This makes the estimation functions dependent on the buffer time of each flight, as shown in Figure 6.18 for two buffer times of 30 min and 15 min each. When the buffer time is the same irrespective of the flights, $B_k = B \ \forall \ k \in [1 \dots M]$, as considered in Lim and Wang (2005), the cost is the same irrespective of the flight, depending only on the separation between consecutive flight assignments, but this is not the situation when the buffer time depends on the flight, namely long, medium or short distance flights, which are the cases studied here.

6.3.7 Reduction in the Number of Conflicts

A stochastic approach for improving schedule performance is described here, when disruptions occur on the day of operation. A similar approach was used in Yan and Tang (2007) where random delay scenarios are generated in the 'Planning Stage' which are used to account for the potential disruptions in the schedule on the day

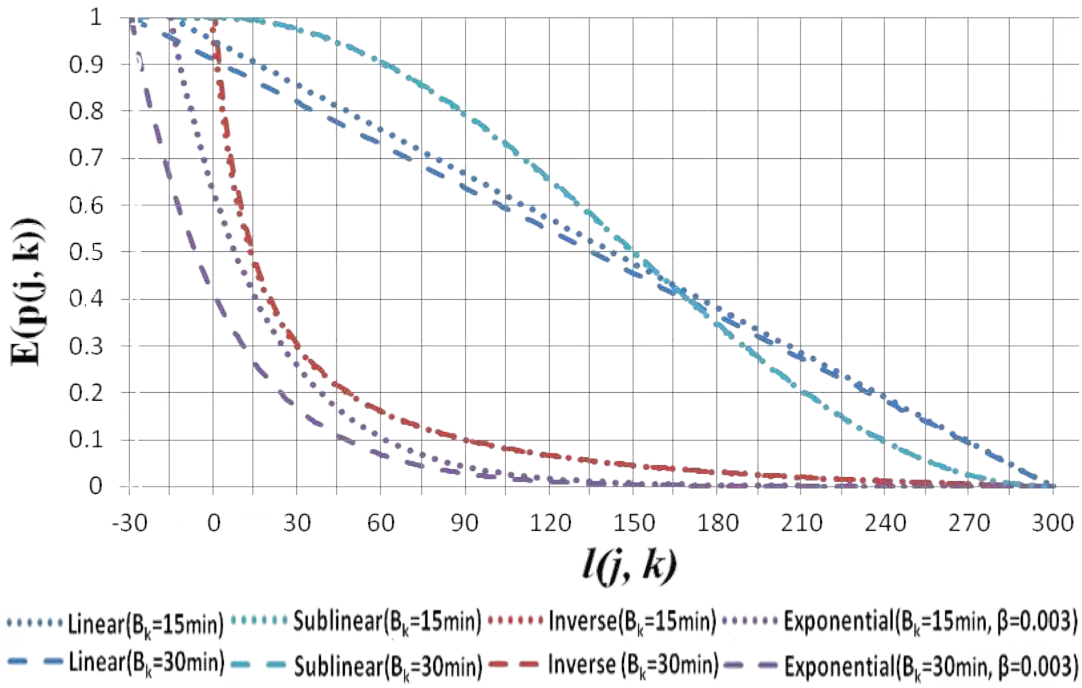


Figure 6.18: Penalty for different unsupervised estimation functions based on the gap between assignments for different buffer times.

of implementation by means of calculating the expected semi-deviation risk measure (Ruszczynski and Shapiro (2003)) for all those delay scenarios.

This approach is based on reducing the number of conflicts on the day of operation. Given that the real perturbed conditions will not be available until the day the schedule is implemented, these perturbed conditions are simulated by examining a set of perturbed base cases, S, which may be obtained in different ways, such as randomly, e.g. from historical data or calculated using known distribution(s) from information available at the time of generating the assignments.

Considering a set of perturbed schedules S, which simulate the perturbations on the day of operation. A new variable is introduced c_{js}, which for a given solution of assignments has the value 1 if flight j is in conflict with another flight in the perturbed schedule $s \in S$, or zero otherwise. The average number of conflicts in the set of perturbed schedules S is calculated by Equation 6.26, which is a measure of the solution robustness.

$$f_2 = \frac{-1}{|S|} * \sum_{s \in S} \sum_{j=1}^{m} c_{js} \tag{6.26}$$

When reassigning conflicting flights or subsequent flights to other BSSs, a situation can arise where the reassigned flight interferes with the subsequent flight at the new BSS, a so-called secondary conflict. Some reassignment may therefore have a downstream effect on the overall schedule, producing more conflicting flights, in turn requiring further reassignments, thus potentially increasing the difficulty of the problem later on.

The above version considers all the conflicts to be of the same importance, but it is preferable to have conflicts which do not have repercussions later, that is, can be reassigned to another BSS without affecting any of the assignments already in existence. To account for this situation a new variable $c^{'}_{js}$ is defined which takes the value 1 if the reassignment of conflicting flight j in a perturbed schedule 's' affects other assignments already in existence, or zero otherwise. The objective is presented as Equation 6.27 where the constant, α, $0 \leq \alpha \leq 1$, denotes the importance of the conflicting flight repercussions on other assignments; $\alpha = 0$ corresponds to the case where no account is taken of any repercussion on other assignments, which corresponds in turn to Equation 6.26, and $\alpha = 1$ corresponds to the cases in which both the conflicting flights and their repercussions on other assignments are considered to be of the same importance. An $\alpha > 1$ refers to the cases where more importance is given to the repercussions of a conflict on assignments other than the conflict itself.

$$f_2 = \frac{-1}{|S|} * \left(\sum_{s \in S} \sum_{j=1}^{m} (c_{js} + \alpha * c^{'}_{js}) \right) \tag{6.27}$$

Calculation of the conflicts is time consuming, and even more so if the effect of the conflict repercussions is also calculated, which is further aggravated by the need to use a large number of schedules in the perturbed set S in order to achieve a good representation of all the potential situations. The execution time is one of the disadvantages of using this approach as was experienced when executing the experiments presented in Section 6.4.

6.3.8 Probability of Conflicts Based on the Gap

The previous approach would normally require a large number of perturbed data sets, which makes its application very slow. Given that we are still interested in reducing the number of conflicts, but without the heavy cost in speed imposed by the Reduction in the Number of Conflicts (RNC) approach, then the probability of having a conflict in a given 'idle time' is used for each flight. This can be easily obtained if the delay distribution is known. In the ABSSAP the early arrival of a flight does not normally affect the assignment to the Baggage Sorting Station Selection (BSSS) as this does not extend the time in which the BSS is required for servicing the flight, but transportation of the baggage already in the BSS to the flight may start earlier, so no earlier arrival is considered here. In the case of a normal folded distribution (no negative numbers and with zero mean) and independent delays for flights, the probability of a conflict for different standard deviations is shown in Figure 6.19. This could also be extended to other distributions and to non-zero means.

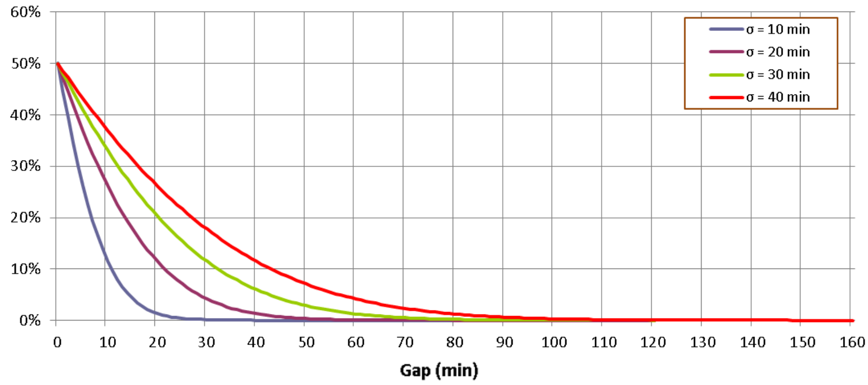


Figure 6.19: The probability of a conflict for two consecutive flights assigned to the same BSS based on the intervening gap modelled with Folded Normal distributions of a zero mean and standard deviation σ.

Independent delays are considered initially, where a conflict between two consecutive flights is independent of other flights assigned to the same BSS. The assignments to different BSSs are independent from the point of view of conflicts. The probability of two consecutive flights having overlapping service times (a conflict), corresponds to the sum of the product between the probability of a sufficiently large delay on the part of the previous flight assigned to that BSS, and the probability of the next consecutive assignment to the same BSS not being sufficiently delayed, as shown in Figure 6.20, where $\eta_j(t)$ is the probability density function for flight j and $t_0 = \tau_k - e_j$.

The probability of a conflict is equal to the probability of exceeding the gap between both assignments, multiplied by the probability of the following assignment not being sufficiently delayed to avoid conflicts, as expressed by Equation 6.28. $p(j,k)$ is the probability a conflict between two flights j and k assigned consecutively to the same BSS with a gap between them of $t_0 = \tau_k - e_j$ given their respective probability density distribution of delay $\eta_j(t)$ and $\eta_k(t)$.

$$p(j,k) = \int_{\tau_k - e_j}^{\infty} \eta_j(t) * \left(1 - \int_0^{t-(\tau_k - e_j)} \eta_k(x)\ dx\right)\ dt \qquad (6.28)$$

A simplification of the conflict probability is represented in Equation 6.29, which

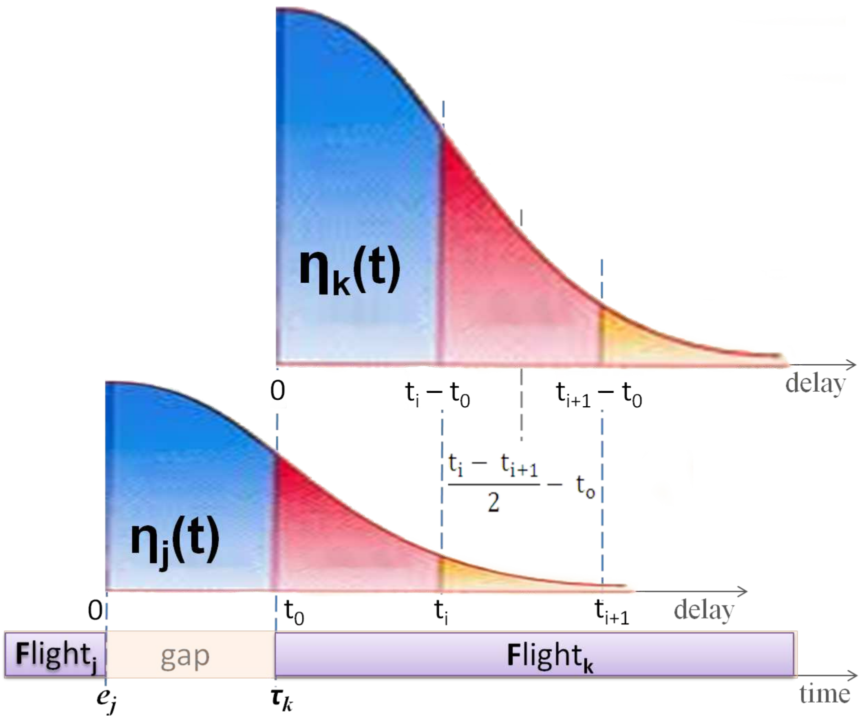


Figure 6.20: Probability of a conflict between two consecutive flights based on the intervening gap.

uses the 'Riemann integral' approach for the range of delays between $(\tau_k - e_j)$ and four times the delay distribution standard deviation of flight j, σ_j, and a time increment of Δt.

$$p(j,k) = \sum_{i=0}^{\frac{4*\sigma_j-(\tau_k-e_j)}{\Delta t}} \eta_j(\tau_k - e_j + (i+\frac{1}{2}) * \Delta t) * \left(1 - \sum_{l=0}^{i} \eta_k((l+\frac{1}{2}) * \Delta t) * \Delta t\right) * \Delta t \tag{6.29}$$

6.4 Results

This section and subsections look at the performance from the point of view of the robustness by measuring the number of conflicts for a given set of perturbed schedules, for all the approaches presented in Section 6.3; firstly when they are used alone and secondly when combined with the TRS approach. The comparison made between the results obtained, when applying the different approaches, uses the Mann-Whitney test to establish the statistical significance of the different approaches, and were presented in the results table summary as the number of instances in a range of the number of BSSs which can be said to have no statistically significantly higher numbers of conflicts than any of the other approaches compared. Regarding an airport, where

N refers to the number of BSSs available, three ranges of the number of BSSs have been defined, based on the LMAP and UMAP, the first being for $N <$ LMAP, the second for LMAP $\leq N <$ UMAP, and the third for UMAP $\leq N$. These are shown within brackets and separated by a comma in the following tables. Any approach achieving full coverage of a range of the number of BSSs is presented in bold font and the approaches with higher numbers for a range, covering the most number of BSSs in the range compared, are presented in underlined font to assist in the interpretation of the summary tables.

The robustness approaches described are applied to the ABSSAP using the Steady State Evolutionary Algorithm (SSEA) from Chapter 5 and their results are compared and analysed in this section using the data sets obtained from NATS for London Heathrow airport Terminal 1, which were also used in Chapters 4 and 5.

To compare the performance of each of the robustness approaches introduced in the previous sections, three sets of perturbed schedules were generated using a folded normal distribution with a zero mean and 10, 20 and 30 minute standard deviations. These sets are used to calculate the average number of conflicts for each robustness approach, where a lower value represents a more robust solution than those with higher values. Each experiment is repeated at least 30 times. To calculate the number of conflicts within a solution in the perturbed schedule, each flight in the original solution is assigned to the same BSS as in the original solution where possible, otherwise it is assigned to the dummy. This is repeated until all of the flights are assigned to a BSS or the dummy. The number of flights assigned to the dummy represents the number of conflicts. If a solution does not achieve the maximum assignment possible, then the number of extra unassigned flights may be accounted for with a higher contribution, given that they are less desirable solutions. Nevertheless, all of the solutions having the highest fitness in the following experiments achieve maximum assignments, which simplifies comparison of the different robustness approaches, based on the described measure. This is possible since the measure only depends on assignments to the BSSs and not on any other objective, such as those upon which the fitness depends.

The number of possible perturbed schedules depends on the number of flights, and this accounts for an extremely large number of possible combinations, making it impossible to consider them all. The number of combinations for 194 flights corresponds to $194! \approx 1.3291 * 10^{361}$ which is far greater than the 10^{82} atoms estimated to exist in the observable universe. The number of perturbed schedules necessary to calculate the quality of a solution should therefore be as large as possible to account for as many potential combinations as possible. However, as the number of perturbed schedules increases, so does the time required to perform the calculations, and the memory requirements also increase likewise: moreover, this is further multiplied by the number of solutions which will be used in the comparison. However, not all combinations are likely to represent a valid schedule. So, 10,000 perturbed schedules were used based on the number of solutions to be processed, the time available and the memory required.

The stochastic robustness approach RNC is time consuming when compared with the other approaches reviewed. It would be desirable to use an approach which provides solutions closer to, or better than, those provided by the stochastic robustness approach, without the heavy cost of the time required. With the aim of assessing the difference in performance when the number of perturbed schedules is reduced, two sets of 1,000 and 25 perturbed schedules used with the RNC were also considered. Initial experiments were conducted using SSEA with $\ell = 1$, the RNC and Multi Exchange between a Fixed Number of 3 Resources (MEFNR3) for 25 perturbed schedules and 800,000 total iterations which required an average execution time of 52 min per instance. These, when extrapolated to 1,000 perturbed schedules, provide an execution time of around 34 hours per instance, whereas the other approaches require no more than two minutes to complete the full set of iterations. Both applications of RNC required too long an execution time for the number of iterations specified, so they were run with a time limit of 30 min, to execute the total number of 800,000 iterations, as the other approaches required less than 2 min.

The objective importance presented in Section 4.4.6 is used in the following experiment where the most important objective is to achieve maximum assignment, the second in importance being to maximise robustness and the third objective being minimisation of the distance between flights and their assigned BSS being the last objective considered. The fitness function used to guide the search in the SSEA is a weighted sum of the different objectives evaluated, introduced previously in Section 4.4.6, and which weights were also calculated in the same section. Both approaches Unsupervised Estimated Stochastic Reduction in Service (UESRS) and Probability of Conflict Based on the Gap (PCBG) need their robustness weight (W_2) to be recalculated based on those values obtained in Section 5.7. Thus given a maximum distance between a flight and its assigned BSS, D_{max}, which depends on the airport topology, which for the topologies studied here is $D_{max} = 9$ (a distance of one unit is assumed between different sides of a pier and a distance of two units was assumed between different piers, as shown in Sections 4.4 and 4.4.6). A new assignment between two previously assigned flights may incur a service reduction for the new assignment and next flight, which is used to obtain the decrease in robustness (the second objective), which for UESRS and PCBG cannot be greater than 1 for each flight, so totalling 2 in this case. Finally, using the objective priorities the following relations can be established: $W_1 * 1 > |W_2| * 2 + |W_3| * D_{max}$ and $|W_2| * 2 > |W_3| * D_{max}$. The original conditions for the weights, when the 'Minimise Reduction in Service Time' objective was used (Section 4.4.6), are $W_3 = -1$ and $W_1 > 23.4$, which together with the objective priorities obtain a $|W_2| > \frac{1*9}{2} = 4.5$ and $|W_2|*2+|W_3|*9 < 23.4$ giving $|W_2| < \frac{14.4}{2} = 7.2$. The value used for W_2 is -7.2 as $W_1 = 90 > 23.4$. The value used for W_1 is greater than the value originally used to calculate W_2, so a value of -10 was also used. The fitness function used for UESRS and PCBG are respectively:

Number of Assigned Flights * 90 - UESRS(*unsupervised estimator*) * $|W_2|$
- Distance between Flights and their assigned BSS * 1

Number of Assigned Flights * 90 - PCBG(*standard deviation*) * $|W_2|$
- Distance between Flights and their Assigned BSS * 1

The sets of 10,000 perturbed solutions generated from normal folded distributions of *standard deviations* of 10, 20, 30, and zero means, are used to calculate the average number of conflicts in each solution when using the different robustness approaches presented in Section 6.3. The unsupervised estimation functions introduced in Section 6.3.6 were used for the same parameter values as those used in Lim and Wang (2005), and with Offset Inverse $b = 6$, Offset Inverse $b = 15$, Offset Sublinear $\gamma = 0$ and Offset Sublinear $\gamma = 1000$. A summary of the robustness approaches studied and their parameter values is shown in Table 6.1. In Table 6.1 the first column contains the name of the robustness approach considered, all of which were introduced in Section 6.3; the second column shows the weights for each approach, and the subsequent columns show the name and value of the parameters for the corresponding robustness approach. The SSEA with $\ell = 1$, operator MEFNR3, a population size of 10, the replacement strategy being Index Selection with Elitist Selection and a group size of 1 (IS1ES) and a maximum of 800,000 iterations were used to obtain the solutions for comparison. The fitness is the sum of the total number of assignments with weight of 90, less the distance between assigned flights and their assigned BSS with a weight of 1, less the robustness approach with the appropriate weight, all of which are shown in Table 6.1.

The results presented in the following sections were summarised for simplicity and

Approach	Weight	Parameters	
		Name	Values
TRS	0.008	Buffer Time	30 min long-haul and 15 min others
ATRS	0.008	Buffer Time	30 min long-haul and 15 min others
ARS	0.008	Buffer Time	30 min long-haul and 15 min others
PCBG	7.2 and 10	Std. deviation	10, 20 and 30 min
RNC	10 and 14	Std. deviation	10, 20 and 30 min
		Num. schedules	25 and 1000
		Max. execution time	30 min
SARS	0.008	Buffer Time	30 min long-haul and 15 min others
UESRS	7.2 and 10	Estimation function	Exp 0.03, Exp 0.05, Inverse 6, Inverse 15, Linear, Offset Inverse 6, Offset Inverse 15, Offset Sublinear 0, Offset Sublinear 1000 and Sublinear
		Buffer Time	30 min long-haul and 15 min others

Table 6.1: Robustness approaches used with their parameter values.

clarity by considering the average number of times an approach achieves statistical significantly lower conflicts, or at least no worse, than the other approaches, in the different regions of numbers of BSSs, based on the LMAP and UMAP, which divide the range of BSSs studied into three areas corresponding to $N < LMAP$, $LMAP \leq N < UMAP$ and $UMAP \leq N$. The values between brackets correspond to the number of times the approach provides significantly statistically no worse solutions than the other approaches used, for each of the BSS ranges. Table 6.2 shows that the ARS approach for the data set of H1T091216 and a 4-pier topology has the values (2, 3, **<u>3</u>**) showing that it achieves a statistically significant number of conflicts no higher in two instances for the range of $N <$ LMAP, three for the range LMAP $\leq N <$ UMAP, and three for $N \geq$ UMAP. Thus, the larger the number between parenthesis the better the performances in respect of robustness.

Bold font is used to identify those cases where the robustness approach achieves good results for all numbers of BSSs in a range, e.g. in the previous example the approach performs well for all numbers of BSSs in the range of $N \geq 27$ (UMAP). Underlining is used to identify those cases where the robustness approach performs well for greater numbers of BSS in a range. The maximum quantity of instances of number of BSSs in a range is presented between brackets at the top of the table for each of the ranges discussed, preceded by the word 'Max.' e.g. in Table 6.2 the cell in the second column and second row of the header shows that the first range contains nine instances of numbers of BSSs for $N <$ LMAP, five for LMAP $\leq N <$

UMAP, and three for $N \geq$ UMAP. For simplicity and clarity, those approaches which do achieve statistically significantly higher conflicts on average than any of the other approaches in all three BSSs regions are not shown in the tables which followed.

Approach	H1T091216		H1T100301	
	Max. (9, 5, 3)		Max. (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
ARS	(1, 2, 2)	$(2, 3, \underline{\mathbf{3}})$	(3, 3, 4)	(1, 2, 1)
BSARS	$(\underline{\mathbf{9}}, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(\underline{8}, 2, \underline{\mathbf{3}})$	$(\underline{4}, \underline{\mathbf{6}}, \underline{\mathbf{5}})$	$(\underline{4}, \underline{5}, \underline{\mathbf{5}})$
SARS + TRS	(3, 4, 2)	$(3, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	(3, 3, 4)	(0, 2, 1)
TRS	(0, 1, 1)	(1, 0, 0)	(2, 3, 4)	(1, 2, 0)

Table 6.2: Number of instances with significantly statistically lower conflicts in each range of numbers of BSSs for disruptions of standard deviation delays of $\sigma = 10$min.

The next section presents the results of the robust approaches which make use of the reduction in service time, with the following section discussing and analysing the UESRS results for different estimation functions and objective weights. Finally, all of the results for the robust approaches considered are presented and analysed before closing this chapter with some conclusions.

6.4.1 Results of the Approaches Using Buffer Time

In this section, only experiments and results relating to robustness approaches which only use the buffer time are assessed, in order to establish how well they perform when compared with each other. It is also suggested that a combination of these approaches with the TRS approach might improve the robustness. Other approaches introduced in Section 6.3 take account of the buffer time: however this is mainly based on the 'idle time' between assignments to the same BSS, i.e. Act Tangent Reduction in Service (ATRS) and UESRS, so they are studied in the following Sections 6.4.2 and 6.4.3.

The ARS, BSARS, SARS and TRS approaches are based on the reduction of the buffer time. The BSARS approach performs much better in respect of the number of conflicts than the other approaches for all the ranges of N (the number of BSSs), delay standard deviations and data sets considered in this section, as shown in the statistical significance summaries in Tables 6.2, 6.3 and 6.4. Furthermore, the quality of its results is maintained at approximately the same steady rate as the delay standard deviation increases, which cannot be said of the other approaches.

The BSARS and SARS differ in a constant factor equal to the UMAP less the LMAP, which only depends on the data set under consideration, i.e. $f_2^{BSARS} = f_2^{SARS} * (UMAP - LMAP)$. Thus the performance of SARS should be the same as that of BSARS if the weight is appropriately increased, given that BSARS= $-\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{p=1}^{P_j} y_{ijp} * A_j$ and SARS is expressed by Equation 6.16. Therefore the weight should be $W_2^{SARS} = W_2^{BSARS} * (UMAP - LMAP)$.

Approach	H1T091216		H1T100301	
	Max. (9, 5, 3)		Max. (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
ARS	$(1, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(2, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(\underline{3}, 2, 4)$	$(1, 3, 1)$
BSARS	$(\underline{\mathbf{9}}, 4, \underline{\mathbf{3}})$	$(\underline{8}, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(\underline{3}, \underline{\mathbf{6}}, \underline{\mathbf{5}})$	$(\underline{4}, \underline{\mathbf{6}}, \underline{\mathbf{5}})$
SARS + TRS	$(5, \underline{\mathbf{5}}, 2)$	$(4, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(\underline{3}, 3, \underline{\mathbf{5}})$	$(0, 4, 2)$
TRS	$(0, 2, \underline{\mathbf{3}})$	$(2, 0, 0)$	$(1, 3, \underline{\mathbf{5}})$	$(1, 2, 0)$

Table 6.3: Number of instances with significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with standard deviation delays of $\sigma = 20$min.

Approach	H1T091216		H1T100301	
	Max. (9, 5, 3)		Max. (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
ARS	$(3, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(2, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(\underline{3}, 4, 4)$	$(1, 4, 2)$
BSARS	$(\underline{\mathbf{9}}, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(\underline{8}, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(\underline{3}, \underline{\mathbf{6}}, \underline{\mathbf{5}})$	$(4, \underline{\mathbf{6}}, \underline{\mathbf{5}})$
SARS + TRS	$(7, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(4, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(\underline{3}, 3, 4)$	$(0, 2, 1)$
TRS	$(2, 3, \underline{\mathbf{3}})$	$(2, 0, 0)$	$(1, 4, \underline{\mathbf{5}})$	$(1, 1, 0)$

Table 6.4: Number of instances with significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with standard deviation delays of $\sigma = 30$min.

In the case of $N \geq UMAP$ it is possible to assign all flights without the need to reduce the service time, which indicates that any approach depending only on the reduction in service, such as ARS, BSARS, SARS and TRS, would not eventually contribute to the fitness, as the final solution should have achieved maximum fitness for this objective. Thus all these approaches may be expected to perform similarly for this range of BSSs, but this is not the case, as is shown in Tables 6.2, 6.3 and 6.4. The behaviour of these robustness approaches is affected by, amongst others things, the stochastic nature of the SSEA, and also by the effect of the initial solutions, not all of which may have full assignment without reduction in service. This is thus achieved as the search progresses, reaching better and more promising areas of the search space than the other approaches, and also by the last objective which will direct the search to solutions with a lower distance between BSSs and flights, which may not necessarily correspond to assignments with lower reduction in service.

The approaches ARS, BSARS and SARS take note of the amount of reduction in buffer time and the time of the day, but both influences are heavily interlaced, such that greater emphasis on the influence of the reduction in service (reduction in the buffer time) may further improve the robustness, given that it increases the search pressure. Other experiments were therefore conducted to determine whether an increase in the importance of reducing service time provides an improvement in robustness. The results, which are summarised in Appendix B.6.1, show that ARS improves when it is used in conjunction with TRS, and ARS with TRS performed overwhelmingly better than ARS, SARS and SARS with TRS. However, the improvement of BSARS with TRS was not so significant when compared with BSARS, where there are ranges of numbers of BSSs for some data sets in which the BSARS performs better than BSARS with TRS. BSARS with TRS performs better overall than the other robustness approaches compared in this section, and was similarly seen with BSARS, its performance is steadily maintained at the same level at which the delay standard deviation increases.

Future study should try to establish whether the same results could be achieved by means of changing the weight of the robustness objective.

6.4.2 Results of Unsupervised Estimated Stochastic Reduction in Service

The UESRS with an exponential estimation function, $\beta = 0.03$ and a weight of 10 provides solutions with a statistically significantly lower number of conflicts throughout all of the ranges of numbers of BSSs, as shown in Tables 6.5, 6.6 and 6.7. This corroborates the results reported in Lim and Wang (2005) for the AGAP. Not only does UESRS with $\beta = 0.03$ perform well over all of the ranges of numbers of BSSs, but it fully covers many of the ranges of numbers of BSSs.

In some instances when the number of BSSs is very low, i.e. 13 or 14, the robustness weight (W_2) may need to be reduced in order to improve the performance.

Unsupervised Estimation Function	Weight	H1T091216		H1T100301	
		Max. (9, 5, 3)		Max. (6, 6, 5)	
		3-pier	4-pier	3-pier	4-pier
Exp 0.03	10	(**9**, **5**, **3**)	(7, **5**, **3**)	(**6**, **6**, **5**)	(5, **6**, **5**)
	7.2	(0, 0, 0)	(2, 0, 0)	(0, 0, 0)	(1, 0, 0)
Exp 0.05	7.2	(1, 0, 0)	(2, 0, 0)	(0, 0, 0)	(3, 0, 0)
Inverse 6	7.2	(0, 0, 2)	(0, 0, 0)	(0, 0, 0)	(2, 0, 0)
Inverse 15	7.2	(0, 0, 0)	(2, 0, 0)	(0, 0, 0)	(2, 0, 0)

Table 6.5: Number of instances with significantly statistically lower number of conflicts in each range of number of BSSs for disruptions with standard deviation delays of $\sigma = 10$min and UESRS approach for a significance level of 0.05.

Unsupervised Estimation Function	Weight	H1T091216 Max. (9, 5, 3)		H1T100301 Max. (6, 6, 5)	
		3-pier	4-pier	3-pier	4-pier
Exp 0.03	10	(**9**, **5**, 1)	(8, **5**, **3**)	(**6**, **6**, **5**)	(5, **6**, **5**)
	7.2	(0, 0, 0)	(2, 0, 0)	(1, 0, 0)	(1, 0, 0)
Exp 0.05	7.2	(0, 0, 0)	(2, 0, 0)	(0, 0, 0)	(2, 0, 0)
Inverse 6	7.2	(0, 3, **3**)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
Offset Inverse 15	7.2	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(1, 0, 0)

Table 6.6: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with standard deviation delays of $\sigma = 20$min and UESRS approach for a significance level of 0.05.

Unsupervised Estimation Function	Weight	H1T091216 Max. (9, 5, 3)		H1T100301 Max. (6, 6, 5)	
		3-pier	4-pier	3-pier	4-pier
Exp 0.03	10	(**9**, 4, 2)	(7, **5**, **3**)	(**6**, **6**, **5**)	(5, **6**, **5**)
	7.2	(0, 0, 0)	(2, 0, 0)	(0, 0, 0)	(1, 0, 0)
Exp 0.05	7.2	(0, 0, 0)	(2, 0, 0)	(1, 0, 0)	(2, 0, 0)
Inverse 6	7.2	(0, 4, **3**)	(0, 0, 0)	(1, 1, 0)	(0, 0, 0)
Inverse 15	7.2	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 0, 1)
Offset Inverse 15	7.2	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(0, 0, 0)

Table 6.7: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with a standard deviation delays $\sigma = 30$min and UESRS approach for a significance level of 0.05.

Clearly the introduction of some penalty, even where the service time has not been reduced (the gap between two assignments to the same BSS is larger or equal to the buffer time), seems to be advantageous. This will be seen in Section 6.4.3, where the approaches which also penalised 'ideal times' greater than the buffer times are contrasted with those which only penalise reduction in buffer times.

As seen from the empirical results presented in Section 6.4.1 for ARS, BSARS and SARS, increasing the buffer time contribution to the fitness by combining them with TRS may assist in reaching promising areas of the search space. Therefore a combination of the UESRS approach with the TRS was also studied, and a summary is shown in Tables 6.8, 6.9 and 6.10.

The results in Tables 6.8, 6.9 and 6.10 show that combining UESRS with TRS, using the exponential estimation function, with $\beta = 0.03$ and a robustness objective weight of 10 ($|W_2|$) continues to provide solutions with statistically significantly lower conflicts than all the other estimation functions, achieving a good performance with

Approach		Weight	H1T091216 Max. (9, 5, 3)		H1T100301 Max. (6, 6, 5)	
			3-pier	4-pier	3-pier	4-pier
UESRS	Exp 0.03	7.2	(0, 0, 0)	(1, 0, 0)	(0, 0, 2)	(0, 0, 0)
		10	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(0, 0, 0)
TRS + UESRS	Exp 0.03	7.2	(7, 1, 0)	(4, 4, 0)	(4, 4, 1)	(4, 3, 0)
		10	$(\underline{\mathbf{9}}, \underline{\mathbf{5}}, 1)$	$(\underline{6}, \underline{\mathbf{5}}, \underline{\mathbf{3}})$	$(\underline{\mathbf{6}}, \underline{\mathbf{5}}, 3)$	$(\underline{\mathbf{6}}, \underline{4}, 1)$
	Inverse 6	7.2	$(0, 0, \underline{2})$	(0, 1, 0)	(0, 1, 1)	(0, 0, 0)
		10	$(0, 1, \underline{2})$	(0, 2, 2)	$(0, 1, \underline{4})$	$(0, 2, \underline{4})$
TRS		0.008	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(0, 0, 0)

Table 6.8: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with standard deviation delays of $\sigma = 10$min and TRS + UESRS approaches with a significance level of 0.05.

Approach		Weight	H1T091216 Max. (9, 5, 3)		H1T100301 Max. (6, 6, 5)	
			3-pier	4-pier	3-pier	4-pier
UESRS	Exp 0.03	7.2	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(0, 0, 0)
		10	(0, 0, 0)	(1, 0, 0)	(0, 0, 1)	(0, 0, 0)
	Exp 0.05	7.2	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(0, 0, 0)
TRS + UESRS	Exp 0.03	7.2	(3, 1, 0)	(3, 0, 0)	(3, 1, 0)	(2, 2, 0)
		10	$(\underline{\mathbf{9}}, \underline{3}, 0)$	$(\underline{6}, 3, \underline{\mathbf{3}})$	$(\underline{\mathbf{6}}, \underline{4}, \underline{4})$	$(\underline{5}, 3, 1)$
	Inverse 6	7.2	(0, 0, 0)	(0, 0, 0)	(0, 1, 0)	(0, 0, 0)
		10	$(0, 2, \underline{\mathbf{3}})$	$(3, \underline{\mathbf{5}}, 2)$	(0, 1, 3)	$(2, \underline{4}, \underline{\mathbf{5}})$
TRS		0.008	(0, 0, 0)	(0, 0, 1)	(0, 0, 0)	(0, 0, 0)

Table 6.9: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with standard deviation delays of $\sigma = 20$min and TRS + UESRS approaches with a significance level of 0.05.

Approach		Weight	H1T091216 Max. (9, 5, 3)		H1T100301 Max. (6, 6, 5)	
			3-pier	4-pier	3-pier	4-pier
UESRS	Exp 0.03	7.2	(0, 0, 0)	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)
		10	(0, 0, 0)	(0, 0, 0)	(1, 0, 1)	(0, 0, 0)
	Exp 0.05	7.2	(0, 0, 0)	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)
TRS + UESRS	Exp 0.03	7.2	(3, 0, 0)	(2, 0, 0)	(3, 0, 0)	(1, 1, 0)
		10	$(\underline{\mathbf{9}}, \underline{3}, 1)$	$(\underline{6}, 3, \underline{\mathbf{3}})$	$(\underline{\mathbf{6}}, \underline{5}, \underline{4})$	$(\underline{3}, 2, 1)$
	Inverse 6	7.2	(1, 0, 0)	(2, 0, 0)	(2, 1, 0)	(2, 0, 0)
		10	$(2, \underline{3}, \underline{\mathbf{3}})$	$(5, \underline{\mathbf{5}}, 1)$	$(2, \underline{5}, 3)$	$(\underline{3}, \underline{4}, \underline{\mathbf{5}})$

Table 6.10: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with standard deviation delays of $\sigma = 30$min and TRS + UESRS approaches with a significance level of 0.05.

54, 47 and 46 instances of a total of 68 for 10, 20 and 30 delay standard deviations respectively. However, the overall performance of the estimation function reduces as the disruption increases in favour of the inverse estimation function, specifically for $b = 6$, achieving a good performance in 16, 25 and 41 instances of a total of 68 for 10, 20 and 30 delay standard deviations respectively. Given that when $N \geq UMAP$ it is possible to assign all flights without reducing the service time, and as these approaches perform better when combined with TRS, which only has an effect where there is a reduction in the assignments service time, so the improvement will arise when solutions have a reduction in service for some of their assignments. This corresponds to the beginning of the search, thus allowing TRS to direct the search into more promising areas of the search space, as is similarly found with ARS, BSARS and SARS.

The experiments were extended to cover the inverse estimation function with values of $b = 2$ and $b = 4$, but the robustness measure showed that the results were not as good as the results achieved when using the same estimation function for $b = 6$. Thus these values are omitted from the above tables for simplicity and clarity.

One characteristic of the TRS, ATRS, ARS, BSARS and SARS is that they only require the buffer time as a parameter. The UESRS requires the buffer time and an estimation function. A comparison of the results from the different approaches considered up to now shows that the exponential estimation function generally provides solutions with fewer statistically significant conflicts than the other approaches considered up to this point, as shown in Tables 6.11, 6.12 and 6.13. These tables do not show approaches which provide statistically significantly higher numbers of conflicts throughout the range of numbers of BSSs. The UESRS performs overwhelmingly better in respect of robustness than ARS, BSARS, SARS and TRS alone or when combined with TRS. The approach performing best in this group was the BSARS with TRS but only for very low numbers of BSSs ($N <$ LMAP). In Table 6.11 for a 4-pier topology and the data set from H1T091216, this approach achieved a good performance in 4 instances for $N <$ LMAP but in none for $N \geq$ LMAP.

All of the approaches which make use of the buffer time when combined with TRS provide overall solutions with a statistically significantly lower number of conflicts overall for the different standard deviation delays of 10, 20 and 30 minutes than when used alone. The performance of UESRS with the Inverse estimation function and $b = 6$ also improves as the delay increases (for delay distributions with higher standard deviations) when compared with the exponential estimation function.

6.4.3 Results for All Approaches

This section looks at the performance from the point of view of robustness by measuring the number of conflicts in a given set of perturbed schedules, for all the approaches

Approach		Weight	H1T091216		H1T100301	
			Max. (9, 5, 3)		Max. (6, 6, 5)	
			3-pier	4-pier	3-pier	4-pier
TRS + ARS		0.008	(4, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
BSARS		0.008	(5, 0, 0)	(1, 0, 0)	(0, 0, 0)	(3, 0, 0)
TRS + BSARS		0.008	(5, 0, 0)	(4, 0, 0)	(0, 0, 0)	(4, 0, 0)
TRS + SARS		0.008	(2, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
UESRS	Exp 0.03	10	(0, 0, 0)	(0, 0, 0)	(0, 0, 2)	(0, 0, 0)
TRS + UESRS	Exp 0.03	7.2	(4, 1, 0)	(3, 1, 0)	(4, 4, 1)	(3, 4, 0)
		10	(8, 4, 1)	(4, **5**, **3**)	(**6**, 5, 4)	(3, 4, 1)
	Inverse 6	7.2	(0, 0, 2)	(0, 1, 0)	(0, 1, 1)	(0, 0, 0)
		10	(0, 1, 2)	(0, 2, 2)	(0, 1, 3)	(0, 2, 4)

Table 6.11: Number of instances significantly statistically lower number of conflicts in each range of number of BSSs for disruptions of standard deviation delays of $\sigma = 10$min and all approaches which make use of the buffer time for significance level of 0.05.

Approach		Weight	H1T091216		H1T100301	
			Max. (9, 5, 3)		Max. (6, 6, 5)	
			3-pier	4-pier	3-pier	4-pier
BSARS		0.008	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(2, 0, 0)
TRS + BSARS		0.008	(0, 0, 0)	(4, 0, 0)	(0, 0, 0)	(3, 0, 0)
UESRS	Exp 0.03	10	(0, 0, 0)	(1, 0, 0)	(0, 0, 1)	(0, 0, 0)
TRS + UESRS	Exp 0.03	7.2	(3, 1, 0)	(3, 0, 0)	(3, 1, 0)	(2, 2, 0)
		10	(**9**, 3, 0)	(6, 3, **3**)	(**6**, 5, 4)	(5, 3, 1)
	Inverse 6	7.2	(0, 0, 0)	(0, 0, 0)	(0, 1, 0)	(0, 1, 0)
		10	(0, 2, **3**)	(2, **5**, 2)	(0, 1, 3)	(3, 4, **5**)

Table 6.12: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with standard deviation delays of $\sigma = 20$min and all approaches which make use of the buffer time with a significance level of 0.05.

Approach		weight	H1T091216		H1T100301	
			Max. (9, 5, 3)		Max. (6, 6, 5)	
			3-pier	4-pier	3-pier	4-pier
BSARS		0.008	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(1, 0, 0)
TRS + BSARS		0.008	(0, 0, 0)	(2, 0, 0)	(0, 0, 0)	(2, 0, 0)
UESRS	Exp 0.03	7.2	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(0, 0, 0)
	Exp 0.03	10	(0, 0, 0)	(0, 0, 0)	(1, 0, 1)	(0, 0, 0)
	Exp 0.05	7.2	(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(0, 0, 0)
TRS + UESRS	Exp 0.03	7.2	(3, 0, 0)	(1, 0, 0)	(3, 0, 0)	(1, 1, 0)
		10	(**9**, 3, 1)	(6, 3, **3**)	(**6**, 5, 4)	(3, 2, 1)
	Inverse 6	7.2	(1, 0, 0)	(2, 0, 0)	(1, 1, 0)	(2, 0, 0)
		10	(3, 3, **3**)	(4, **5**, 1)	(2, 5, 3)	(3, 4, **5**)

Table 6.13: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with standard deviation delays of $\sigma = 30$min and all approaches which make use of the buffer time with a significance level of 0.05.

presented in Section 6.3; first when they are used alone and then when combined with the TRS approach. A comparison of the results obtained when applying the different approaches uses the Mann-Whitney test to establish the statistical significance of the different approaches. They are then presented as the number of instances in a range of numbers of BSSs which can be said not to have a higher number of conflicts than any of the other approaches in the comparison. The ranges of numbers of BSSs are those for $N <$ LMAP, LMAP $\leq N <$ UMAP, and UMAP $\leq N$, which are presented within brackets and separated by a comma in the following tables. N is the number of BSSs in an instance, as previously introduced in Section 3.3.2.

The results which are summarised in Tables 6.14, 6.15 and 6.16 show that the PCBG does not gain any advantage when combined with the TRS, but it appears to be detrimental, as PCBG alone performs better throughout whole the ranges of the numbers of BSSs than when used combined with TRS. Even with the RNC for 25 and 1,000 instances and taking account of the distribution of delays, the other approaches provide a statistically significantly lower number of collisions, but with a much lower running time. This could be due to the fact that the number of iterations which it is possible to execute in the 30 minutes is too low to find promising solutions with a lower number of collisions than those obtained by the other approaches. Around 33,000 to 70,000 iterations were executed for the experiments conducted here, the numbers depending mainly on the data set and the number of BSSs). Additionally, where the perturbed set of schedules used in RNC does not accurately represent the real perturbation on the day of operation, then the search will be wrongly guided, so achieving less robust solutions. Similar results are obtained when using a set of 1,000 disrupted schedules with RNC 1,000.

The PCBG provides statistically significantly lower number of conflicts through a wider range of BSSs, and such a range also includes the range of BSSs used in the real problems, i.e. $N \geq UMAP$. Nevertheless, this result could be regarded as biased, given that the PCBG considers a normal folded distribution of the same standard deviation as that from which the perturbed schedules were generated. The PCBG could consider different standard deviations and distributions depending on the aircraft, season, route, destination and time of the day which should further improve the results in real situations. The running time is also of the same magnitude as that for the other approaches, with the exception of the RNC, which has a much higher running time for the same number of iterations.

Future studies could look at the dependency of the robustness objective weight

Approach			H1T091216 Max. (9, 5, 3)		H1T100301 Max. (6, 6, 5)	
			3-pier	4-pier	3-pier	4-pier
TRS + ARS			(2, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
BSARS			(2, 0, 0)	(1, 0, 0)	(0, 0, 0)	(3, 0, 0)
TRS + BSARS			(3, 0, 0)	(3, 0, 0)	(0, 0, 0)	(4, 0, 0)
PCBG		7.2	(5, 2, 1)	(3, 1, 0)	(1, 1, 0)	$(\underline{5}, 2, 0)$
		10	$(\underline{7}, \underline{4}, 0)$	$(\underline{8}, 1, 0)$	$(2, \underline{4}, 0)$	$(\underline{5}, \underline{4}, 0)$
TRS + SARS			(1, 0, 0)	(1, 0, 0)	(0, 0, 0)	(0, 0, 0)
TRS + UESRS	Exp 0.03	7.2	(2, 1, 0)	(0, 1, 0)	(2, 1, 1)	(1, 2, 0)
		10	(2, 1, 0)	$(1, \underline{4}, \mathbf{\underline{3}})$	$(\underline{4}, 2, \underline{4})$	(2, 1, 1)
	Inverse 6	7.2	$(0, 0, \underline{2})$	(0, 1, 0)	(0, 1, 1)	(0, 0, 0)
		10	$(0, 0, \underline{2})$	(0, 1, 2)	$(0, 1, \underline{4})$	$(0, 2, \underline{4})$

Table 6.14: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with a delay standard deviation of $\sigma = 10$min) and all the approaches presented in Section 6.3 with a significance level of 0.05.

Approach			H1T091216 Max. (9, 5, 3)		H1T103010 Max. (6, 6, 5)	
			3-pier	4-pier	3-pier	4-pier
BSARS			(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(2, 0, 0)
TRS + BSARS			(0, 0, 0)	(4, 0, 0)	(0, 0, 0)	(3, 0, 0)
PCBG		7.2	(5, 1, 0)	(2, 0, 0)	(3, 2, 0)	(3, 0, 0)
		10	$(\underline{7}, 2, 1)$	$(\underline{8}, \mathbf{\underline{5}}, 1)$	$(1, \underline{5}, 1)$	$(\underline{5}, \mathbf{\underline{6}}, 1)$
TRS + UESRS	Exp 0.03	7.2	(2, 1, 0)	(1, 0, 0)	(2, 1, 0)	(1, 0, 0)
		10	$(5, \underline{3}, 1)$	$(2, 0, \mathbf{\underline{3}})$	$(\underline{5}, 1, \underline{4})$	(4, 0, 1)
	Inverse 6	7.2	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)	(0, 0, 0)
		10	$(1, 1, \mathbf{\underline{3}})$	(0, 2, 2)	(0, 1, 2)	$(2, 2, \mathbf{\underline{5}})$

Table 6.15: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with a delay standard deviation of $\sigma = 20$min) and all the approaches presented in Section 6.3 with a significance level of 0.05.

Approach			H1T091216 Max. (9, 5, 3)		H1T100301 Max. (6, 6, 5)	
			3-pier	4-pier	3-pier	4-pier
BSARS			(0, 0, 0)	(1, 0, 0)	(0, 0, 0)	(1, 0, 0)
TRS + BSARS			(0, 0, 0)	(2, 0, 0)	(0, 0, 0)	(2, 0, 0)
PCBG		7.2	(4, 2, 0)	(0, 0, 0)	(1, 4, 0)	(1, 0, 0)
		10	$(1, 4, \mathbf{\underline{3}})$	$(\underline{5}, \underline{4}, \mathbf{\underline{3}})$	$(2, \mathbf{\underline{6}}, \mathbf{\underline{5}})$	$(\underline{3}, \mathbf{\underline{6}}, \mathbf{\underline{5}})$
TRS + UESRS	Exp 0.03	7.2	(3, 0, 0)	(1, 0, 0)	(3, 0, 0)	(1, 0, 0)
		10	$(\mathbf{\underline{9}}, 1, 0)$	(4, 1, 0)	$(\underline{5}, 1, 0)$	(2, 0, 0)
	Inverse 6	7.2	(1, 0, 0)	(1, 0, 0)	(1, 0, 0)	(2, 0, 0)
		10	(3, 3, 1)	(2, 3, 0)	(0, 0, 0)	(3, 0, 0)

Table 6.16: Number of instances with a significantly statistically lower number of conflicts in each range of numbers of BSSs for disruptions with a delay standard deviation of $\sigma = 30$min) and all the approaches presented in Section 6.3 with a significance level of 0.05.

on performance. If other robustness weights are to be considered, all of the weights taking part in the fitness function should be modified accordingly to maintain the order of importance of each objective. When the unassigned flights are not taken into account by the robustness measure used in the fitness function, then particular care has to be taken when selecting the appropriate weights for maximisation of the number of assignments (the first objective), since incorrect selection of this objective weight may sometimes deem solutions with a lower number of assignments to be fitter. This interdependency, mainly between the maximum number of assignments objective and the robustness could also be decoupled by penalising the unassigned flights in the robustness.

6.5 Conclusions

Several approaches to taking account of solution robustness by applying the SSEA, presented in Chapter 5, are presented in this chapter. It has been shown that the TRS approach provides solutions with a statistically significantly higher number of conflicts than those obtained by some of the other approaches considered. TRS does not consider the extra increase in conflict either, as the service time is reduced between assignments, thus further penalising those assignments. Where the service reduction is higher the ATRS was also used, but the results were not very encouraging as some of the other approaches improved on it.

When looking at the overall schedule it is evident that the number of flights requiring assignment at each time is not uniformly distributed, as shown in Figure 4.4, such that when the number of flights to be serviced at any one time increases, the potential for conflict also increases, when perturbed. There is potential for further improving the results when this is taken into account, so other approaches were proposed and studied. The first approaches examined were the ARS, BSARS and SARS. These approaches performed better for a very low number of BSSs, but in general provided solutions inferior in respect of conflicts than the PCBG and the UESRS with exponential and inverse estimation functions. The advantage of the TRS approach in conjunction with some of the other approaches was anticipated, as was corroborated by results from the experiments conducted for the combination of TRS with the ARS, BSARS, SARS and UESRS.

Several stochastic approaches were also considered, namely: UESRS was evaluated for different estimation functions and the exponential function with $\beta = 0.03$ provided the best results in respect of robustness for the measure considered. PCBG provides the best overall performance regarding robustness but did not seem to improve when combined with TRS, rather showing a deterioration. RNC appeared to provide solutions with a higher number of conflicts than the other approaches, but this could be a consequence of the low number of iterations which it was possible to execute given both the low speed which is a characteristic of this approach and the higher memory requirement. The UESRS approach also improved when combined with TRS, showing the Inverse function for $b = 6$ with TRS providing good results for high numbers of BSSs ($N \geq LMAP$)

The study in this chapter was based on results which used the MEFNR3 operator. In the results shown in Chapter 5, this was seen to perform better for a number of BSSs lower than the UMAP, so there is potential for improvement by using different operators or a combination of these based on the number of BSSs.

It would be interesting to apply these approaches to some data sets where the original flight schedules and final real schedules are known, to see if the solutions obtained by these approaches did cope well with the changes, but unfortunately this was impossible at the time, given the unavailability of such data. Furthermore, if the data sets contain the real flight assignments to BSSs on the day of implementation, then it would be possible to quantify the actual improvement which could have been achieved by each of the approaches presented in this chapter if they were implemented.

There is a possibility of combining the different approaches, in particular either ARS or BSARS with UESRS. There is also a question about the preference as to how these approaches are combined, either as a sum of each individual with the TRS, as studied in this chapter, or as a product of their individual contributions.

Future work should consider the use of multiple distributions, in general one per flight, based on the particular characteristics of each flight, such as aircraft type, airline, destination, route, season etc. This could be applied similarly to generation of the buffer times. This approach was not used, since such information was unavailable at the time this study was conducted. It is envisaged that the use of this information in the PCBG and RNC will improve their performance, and may also be used to generate the disrupted schedules used to measure the solution quality provided by all of the approaches, so assisting in the identification of the best approach for use in the specific problem.

Chapter 7

Airport Gate Assignment Problem

The constructive and search algorithms in previous chapters are potentially important for a wider variety of problems other than the Airport Baggage Sorting Station Assignment Problem (ABSSAP). By way of illustration, this chapter looks at applying the same techniques to the widely studied Airport Gate Assignment Problem (AGAP).

The chapter introduces the AGAP, defines its scope, and presents a mathematical model to represent it, which is used throughout this and the subsequent chapter. References to the relevant literature are included and some constructive algorithms are presented, which are extensions of those provided for the ABSSAP in Chapter 4. Their performance is studied in regard to the different objectives currently used at London Heathrow airport. The final section of this chapter provides some conclusions. Chapter 8 then considers the application of the evolutionary algorithms to the problem.

7.1 Overview

Aircraft depart from an airport and arrive at their destination airport, from which the aircraft may again depart to yet another airport, and this may be repeated many times a day for each aircraft. During the time between arrival and departure, while the aircraft is still at the airport, it needs to have a space allocated at a stand on the airport air-side, where some operations may need to be performed before it is ready to continue its cycle of departure and arrival. The stands next to the airport gates are scarce and expensive resources which must be used efficiently and be assigned to aircraft effectively. The gate assigned to an aircraft arrival may not be the same as that assigned to the same aircraft for departure, and the intermediate parking operation if any is required, between arrival and departure assignments may also be at a different stand. This may either be a remote stand (not a gate) or another gate depending on the availability of these resources at the time. Aircraft only directly compete for resources if their stay at the airport overlaps in time.

Given that a gate may be required for up to three different operations, namely arrival, parking and departure, the number of assignments required may have increased significantly in comparison with those for the ABSSAP. These also increase the problem complexity and provide more reasons for investigating some metaheuristic approaches, such as the Steady State Evolutionary Algorithm (SSEA) presented in Chapter 5, which will be adapted to the AGAP in Section 8.2.

With the increase in passenger traffic volumes and number of flights, the complexity of this task and the number of factors to be considered have increased significantly, and efficient gate utilization has received considerable attention in past years, e.g. Hu and Di Paolo (2007), Li (2009), Jaehn (2010), Seker and Noyan (2012) and Kim and Feron (2012).

The AGAP relates to the assignment of stands to flights already scheduled. The flights have a scheduled arrival and departure time, between which the aircraft is located on either one or several stands sequentially. The movement of an aircraft between stands normally requires the use of tugs (towing trucks) which add extra cost to the aircraft's operations, which airlines would prefer to avoid whenever possible. When an aircraft is assigned to different stands in the same arrival/departure period, to disembark passengers from one stand and embark them on another, then the aircraft must be moved from its assigned stand to the stand assigned to it next until it is located on the stand assigned for its departure. This operation may be executed either by using the aircraft's engines or by tugs. Given the disadvantages of using the aircraft's engines, which have already been discussed in Section 2.3.2, the use of towing trucks is the preferred alternative to pushback when moving from stands to the departure gate, or away from the gate ready to join the departure sequencing in the departure holding areas.

Stands are one of the most important resources available to airlines at an airport. The effective use of these stands is extremely important when considering operational costs and passenger satisfaction. However given the different constraints, objectives and the large number of flights involved in the problem, optimisation of this is exceptionally challenging. An introduction to this problem was given in Section 2.4. The model is introduced in Section 7.2.

Furthermore, the assignment of flights to gates must consider the interests of the airport, airlines, handlers, and also take account of passenger satisfaction, at the same time as the constraints, such as safety, are complied with. These interests are taken into account by means of objectives which are introduced in Section 7.5. Five objectives are considered in the study presented in this Chapter: 'Maximise Number of Assignments' (Section 7.5.1), 'Maximise Airline Preferences' (Section 7.5.2), 'Minimise Reduction in Service' (Section 7.5.5), 'Minimise Number of Towing Operations' (Section 7.5.3) and 'Maximise Handler Preferences' (Section 7.5.4).

The process currently followed at London Heathrow airport incorporates the preferences of airlines and handling agents when assigning flights to stands. The assignment of all flights to available stands is the main objective, given that failure to assign a flight results in a cancelation, which would mean lost revenue for many of the companies which provide airport services, directly affecting the airport authority, airline and handling agent. Such failure would also have an indirect effect on the companies providing services within the terminal building, whose trade is based on passengers visiting the terminal. The airlines have contracts with handling agents which usually clearly state the level of service required, which reflect the specific circumstances of the airline. Airlines normally have a preference for gates located in proximity to the airline offices and/or resources, so that, for example, staff can attend to their duties quickly at the stand allocated to their airline flight. Certain areas of the terminal are frequently reserved for specific airlines, and international and national flights are usually kept separate, due to differing customs and security regulations. The handling agents would normally prefer all the flights which they have to service to be physically close to each other, so reducing the time and cost which would otherwise be incurred if they were far apart. Nevertheless, given the contractual obligations between the handling agents and the airlines, the airline preferences often prevail, whereas handling agent preferences may be used to break any tie in consequence of the airline preference. Both the airline stand preferences and the towing objectives represent the main airline preferences.

The AGAP model presented in Section 7.2 is different to the ABSSAP model presented in Chapter 3.3. Some modifications are required in order to use constructive algorithms similar to those presented in Chapter 4, and which are presented in Section 7.6. The problem data is described in Section 7.7. The results of the experiments conducted for the constructive algorithms are presented next in Section 7.9, where a study is also conducted to establish their contribution to each AGAP objective. The final section of this chapter draws some conclusions.

7.2 Model

The model used for the AGAP is based on that proposed in Dorndorf (2002), which considers the problem as a resource constrained project scheduling problem, originally presented in Dorndorf et al (2000). Flights serviced by the same aircraft may not generally be assigned to the same stand, in which case they may need to be moved to another assigned stand by using tugs, and known as towing, but the use of towing should be kept to a minimum given the extra cost involved, such as hiring towing trucks and the increase of ground traffic. If the time for servicing an aircraft between its arrival and departure is sufficiently long then the aircraft may be assigned to a parking stand in order to release the stand originally assigned to its arrival. It is assumed that even if there are insufficient stands at the pier, there are always sufficient remote stands where the aircraft awaiting departure can be parked. The aircraft will then be towed to its assigned departure stand, which may not be the same as that assigned to it on arrival, Figure 7.1. The movement of aircraft around the airside of the airport terminal potentially poses difficulties, e.g. increasing traffic on the airport ground side, and always incurs extra costs, such as hiring the towing machine. To consider these, an objective is introduced into the model with the aim of reducing the level of unnecessary remote parking, Section 7.5.3. This also penalises the impact of towing an aircraft from the assigned arrival stand to the remote stand and finally to the departure stand. The problem is an Activity Assignment Problem

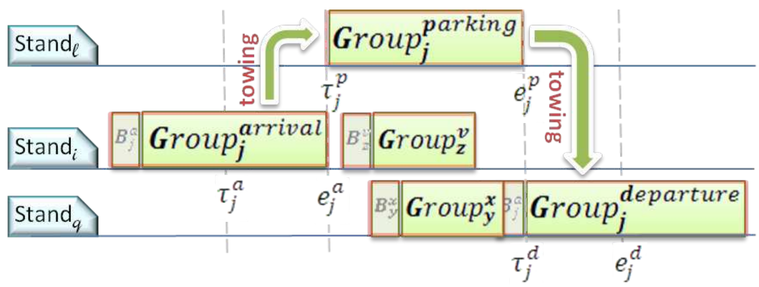


Figure 7.1: Assignment of flights to stands when towing is required.

where the arrival, departure and parking periods of an aircraft at a stand are the activities and the stands are the resources.

It is anticipated that there will always be sufficient remote parking stands, so there is unlikely to be a problem in their use, nor as to which stand is used. Furthermore, given that arrival and departure flights require the extra facilities provided by a gate, whereas the parking activity does not, it is preferable to assign the parking activity to a remote stand if this increases the assignment of more arrival and departure flights to gates. This may be modelled either by introducing a new objective which takes this preference into account, or by building it into the model as a hard constraint. In this thesis the latter approach is used, where parking assignments are restricted to gates already assigned to arrival or departure flights of the same aircraft or a remote stand. This prompted modelling the use of a remote stand in a similar fashion to the dummy, which is equivalent to a stand with unlimited capacity, where assignments can overlap, but is restricted to intermediate operations such as parking. The service time must be of at least a specified minimal duration otherwise the remote stand will not be required, when arrival and departure flights will be treated as one activity. The current procedure in London Heathrow airport usually involves the assignment of arrival, parking and departure of the same aircraft to the same gate when the time between arrival and departure is less than 3 hours.

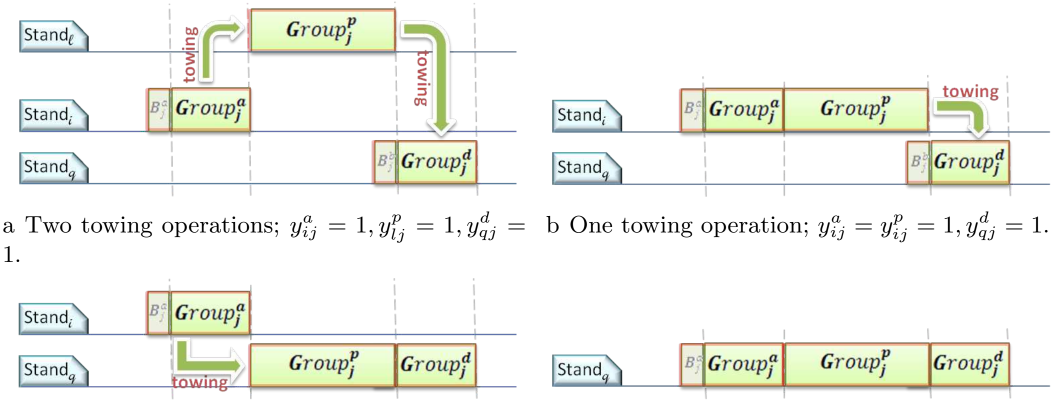
The model used in the ABSSAP is modified and extended to represent the remote stand, where $i = 0$ equates to the dummy stand as used in the ABSSAP, and $i = N+1$ represents the remote stand, and where N represents the number of real stands at gates. Where the term remote stand is used, it refers to the dummy remote stand. The dummy remote stand may also be used solely for parking operations where arrivals and departures are not permitted by this resource. This means that when a solution is obtained and the dummy remote stand has been assigned to certain aircraft, then these aircraft must be assigned to real physical stands whether remote or otherwise.

The main reason for this procedure is to speed up the generation of solutions, as the algorithm should already be endeavouring to reduce the number of tows (objective 'Minimise Number of Towing Operations' presented in Section 7.5.3), which includes the reduction of the number of remote assignments. In this representation, those which are commonly regarded as two different flights, namely the arrival and departure flights, are considered here to be only one group composed of three operations: arrival, parking and departure, as shown in Figure 7.1. The relationship between these operations relates to the same aircraft, but may differ in the use of other resources such as the crew.

7.3 Problem Representation

The problem is presented as an Integer Linear Programming (ILP) with y_{ij}^k being a Boolean variable with a value of one if activity k ($k \in \{a, p, d\}$ where 'a' corresponds to the arrival flight, 'p' to the parking and 'd' to the departure flight activities) of group j is assigned to gate i or zero otherwise. The degree of reduction in service time for the assignment of activity k of flight j is represented by r_j^k which is deemed to be calculated in seconds (as an integer). These constants and variables are listed in Tables 7.1 and 7.2, the full model being presented in the following sections.

In the following sections and Chapter 8 the only case under consideration is that where the parking of an arriving and/or departing flight may be assigned to the same stand as the arrival or departure activity, or to the remote dummy stand, Figure 7.2.



a Two towing operations; $y_{ij}^a = 1, y_{lj}^p = 1, y_{qj}^d = 1$.

b One towing operation; $y_{ij}^a = y_{ij}^p = 1, y_{qj}^d = 1$.

c One towing operation; $y_{ij}^a = 1, y_{qj}^p = y_{qj}^d = 1$.

d No towing operations; $y_{qj}^a = y_{qj}^p = y_{qj}^d = 1$.

Figure 7.2: Different stand assignments for group j.

It is assumed that aircraft j may be used for three assignments, such that the model may be expressed as follows: the assignment of aircraft j to stands i, l, and q is expressed as $y_{ij}^a = 1$, $y_{lj}^p = 1$ and $y_{qj}^d = 1$ respectively. There are now two new base service duration constants, one for the flight arriving, T_j^a, and the other which corresponds to the parking base service duration, T_j^p, if the corresponding aircraft were to be assigned to a remote stand. The commencement and completion times of a parking operation are fixed, based on the values of the departure time e_j^d and arrival time τ_j^a, given that $e_j^a = \tau_j^a + T_j^a$ and $\tau_j^d = e_j^d - T_j^d$ then $\tau_j^p = e_j^a$, $e_j^p = \tau_j^d$ and $T_j^p = \tau_j^d - e_j^a$. The list of constants for this model is shown in Table 7.1 and the list of decision variables is shown in Table 7.2.

The representation of the AGAP is similar to the ABSSAP presented in Section 3.3. In the AGAP the activity k of group j may take the values a, p or d ($P_j = 3$) whereas in the ABSSAP activity p was represented as an integer ($1 \leq p \leq P_j$). However in the AGAP y refers to aircraft, and there are extra constraints which do not

Name	Description
N	The total number of gates under consideration.
M	The total number of aircraft to which gates should be allocated.
k	The type of operation, **a**rrival, **p**arking or **d**eparture, $k \in \{a, p, d\}$.
P_j	The total number of activities associated with aircraft j in a full cycle, $1 \leq P_j \leq 3$.
T_j^k	The base service duration for aircraft j and activity k.
B_j^k	The desired buffer time for aircraft j and activity k. The parking operation does not have any buffer time associated with it, i.e. $B_j^p = 0$.
e_j^k	The end service time for aircraft j and activity k.
τ_j^k	The base starting service time for aircraft j and activity k, $\tau_j^k = e_j^k - T_j^k$.
t_j^k	The target starting service time for aircraft j and activity k, $t_j^k = e_j^k - T_j^k - B_j^k$, assuming the full buffer time is available. Whereas $T_j^p = t_j^d - e_j^a$ and $e_j^p = t_j^d$.
x_{ij}	Expresses to which stand (i) aircraft j can be assigned, $i \in (1 \ldots N)$. $x_{ij} = 1$ if aircraft j can be assigned to stand i, otherwise $x_{ij} = 0$.

Table 7.1: List of the constants and input values for the AGAP model.

Name	Description
y_{ij}^k	Specifies the assignment of aircraft to stands. $y_{ij}^k = 1$ if gate $i \in [1 \ldots N]$ is allocated to aircraft $j \in [1 \ldots M]$ for activity $k \in \{a, p, b\}$, and 0 otherwise.
r_j^k	Specifies the necessary reduction in service time for activity k of aircraft $j \in [1 \ldots M]$, given the allocated starting service time, s_j^k.
s_j^k	The allocated starting service time for activity $k\{a, p, d\}$ of aircraft $j \in [1 \ldots M]$ and given that a gate can only service one aircraft at a time then s_j^k can be determined from r_j^k since $s_j^k = t_j^k - r_j^k$ and $t_j^p = e_j^a$.

Table 7.2: List of the decision variables which are used in this AGAP model.

allow parking activities to be assigned to stands other than the dummy parking stand (also called the dummy remote stand), or their corresponding arrival and departure stands. The operations represented by 'a', 'p' and 'd' refer to the arrival, parking and departure operations respectively of the same aircraft.

7.4 Constraints

7.4.1 Assignment Limits

Each stand can only be used by one aircraft at a time, with the exception of the dummy stand and the remote dummy stand, Equation 7.1.

$$\sum_{i=1}^{N+1} \sum_{k \in \{a,p,d\}} y_{ij}^k = P_j \quad \forall\ j \in [1 \ldots M] \tag{7.1}$$

7.4.2 Assignment Restrictions

Each activity may be only assigned to one stand, Equality 7.2.

$$\sum_{i=1}^{N+1} y_{ij}^{k} = 1 \quad \forall\ j \in [1 \ldots M] \text{ and } k \in \{a, p, d\} \tag{7.2}$$

The remote dummy stand ($i = N+1$) is only suitable for parking and not for any other activity, i.e. arrivals or departures, Equation 7.3.

$$y_{(N+1)j}^{a} = y_{(N+1)j}^{d} = 0 \quad \forall\ j \in [1 \ldots M] \tag{7.3}$$

The dummy stand ($i = 0$) cannot be assigned to parking activities. The remote dummy ($i = N+1$) allows overlapping activities and always has the capacity to be assigned to a parking activity, Equation 7.4.

$$y_{0j}^{p} = 0 \quad \forall\ j \in (1 \ldots M) \tag{7.4}$$

The parking activity may be assigned to the same stand as its associated arrival or departure activities, Inequality 7.5.

$$y_{ij}^{p} \leq (y_{ij}^{a} + y_{ij}^{d}) \quad \forall\ i \in [1 \ldots N],\ j \in [1 \ldots M] \tag{7.5}$$

7.4.3 Stand and Aircraft Size Restriction

Each stand has a size code assigned to it which identifies the range of aircraft assignable to it. The aircraft sizes match those specified by the International Civil Aviation Organization (ICAO). This constraint is modelled by the constant x_{ij} shown in Table 7.1 (example in Figure 7.3), Inequality 7.6 .

$$y_{ij}^{k} \leq x_{ij}\ \forall\ i \in [1 \ldots N],\ j \in [1 \ldots M] \text{ and } k \in \{a, p, d\} \tag{7.6}$$

7.4.4 Combining Stands

This section discusses the situation where certain gate assignments may cause blocking of neighbouring gates, sometimes described as ‘Shadowing Assignments’. For example, in some cases two adjacent stands can jointly host a larger aircraft where they would be unable to do it individually. To represent the combined stand a fic-

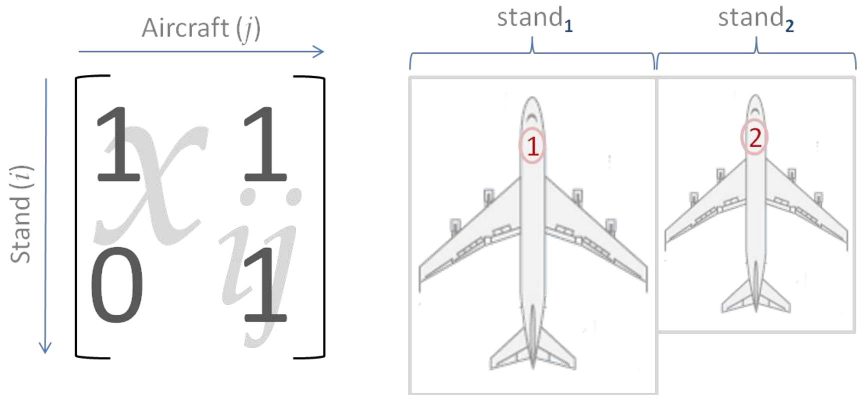


Figure 7.3: Example of two stands ($i \in \{1, 2\}$) of different sizes and two aircraft ($j \in \{1, 2\}$) where aircraft 1 is too large to fit in stand 2 ($x_{21} = 0$).

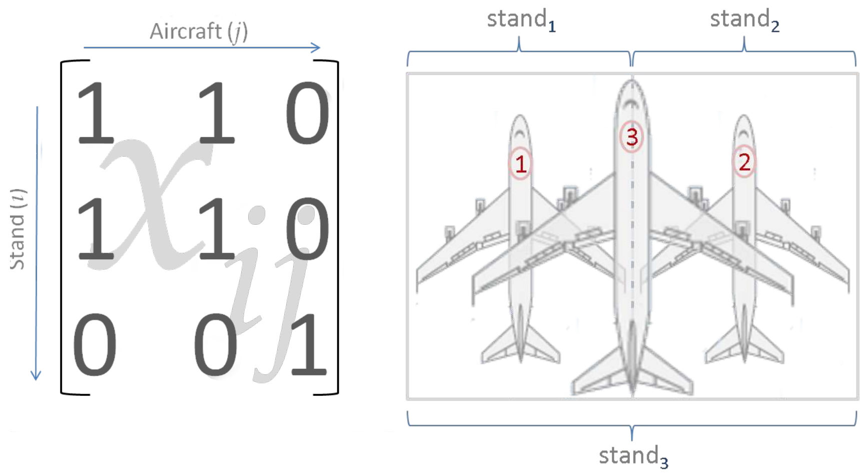
titious stand is postulated which can only be assigned to aircraft larger than the largest capable of assignment to any of its component stands, Figure 7.4. It does not make any sense for the combined stand to be assigned to an aircraft which could also be assigned to any of its component stands. In this case the smaller stands can accommodate an aircraft each, or the combined fictitious stand may host a larger aircraft. So for a stand i composed of stands l and r, and an aircraft j if $x_{lj} = 1$ or $x_{rj} = 1$ then $x_{ij} = 0$, whereas if $x_{ij} = 1$ then $x_{lj} = x_{rj} = 0$. If O_j contains all of the flights overlapping this flight j then all assignments to stands r, l and i ($r \cup l$), which overlap, must comply with Inequalities 7.6 and 7.7.

$$\sum_{q,u,v \in O_j \cup j} \sum_{k \in \{a,p,d\}} \left(y_{rq}^k + y_{lu}^k + (2 * y_{iv}^k) \right) \leq 2 \tag{7.7}$$

Based on Inequality 7.6, example in Figure 7.4 provides the following restrictions. This means that stand 3 occupies the combined space of stands 1 and 2, and aircraft 3 cannot be assigned to either stands 1 or 2 ($x_{13} = x_{23} = 0$), whereas aircraft 1 and 2 cannot be assigned to stand 3 ($x_{31} = x_{32} = 0$), as shown in Table 7.3.

$y_{11}^k \leq x_{11} = 1$	$y_{12}^k \leq x_{12} = 1$	$y_{13}^k \leq x_{13} = 0$
$y_{21}^k \leq x_{21} = 1$	$y_{22}^k \leq x_{22} = 1$	$y_{23}^k \leq x_{23} = 0$
$y_{31}^k \leq x_{31} = 0$	$y_{32}^k \leq x_{3\ 2} = 0$	$y_{33}^k \leq x_{33} = 1$
$k \in \{a, p, d\}$		

Table 7.3: Valid values for x_{ij} and y_{ij}^k for the example in Fugure 7.4.



a Stand size restriction matrix. b View of potential stands assignment.

Figure 7.4: Example of two stands ($i \in \{1, 2\}$) and one combined fictitious stand ($i = 3$) for three aircraft ($j \in \{1, 2, 3\}$) where aircraft 3 is too large to fit in either stands 1 and 2 but both aircraft 1 and 2 fit any of the stands.

7.5 Objectives

7.5.1 Maximise Number of Assignments

This objective aims to maximise the number of assignments, Formula 7.8, and it is equivalent to the same objective as defined in the ABSSAP. In airport practice, this objective would probably be a hard constraint at most times, since all flights would normally have to be serviced, but it is desirable to observe the performance of the algorithms when the algorithms are only guided to achieve it and allow for a study where there are too few gates, as well as when these are sufficient or plentiful.

$$\max \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k \in \{a,p,d\}} y_{ij}^{k} \tag{7.8}$$

Given that parking is always assured since the parking activity can always be assigned to the dummy remote stand, then Formula 7.8 is equivalent to Formula 7.9.

$$\max \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k \in \{a,d\}} y_{ij}^{k} \tag{7.9}$$

The preference for assigning the parking activities to the same stand as one of the associated arrival or departure flights, is taken account of in the objective ‘Minimise Number of Towing Operations’ presented in Section 7.5.3.

7.5.2 Maximise Airline Preferences

Airlines may have some preference as to the gates for assignment to their flight. These could be based on their position in relation to some of the airline facilities such as offices or other resources used, for example buses.

To take account of the different airline preferences a list of gates and a weight, representing the level of preference for the gate, is used. This list could be compiled based on past historical data of gates assigned to the airline, such that the constant $\theta_{\alpha j}$ is 1 when aircraft j belongs to airline α and zero otherwise, and there is a set of historical flights represented by H. The preference of airline α for stand i may then be expressed by Equation 7.10, and the objective by Formula 7.11.

$$\delta_{\alpha i} = \frac{\sum_{j}^{H} \sum_{k \in \{a,d\}} \theta_{\alpha j} * y_{ij}^{k}}{\sum_{i=1}^{N} \sum_{j}^{H} \sum_{k \in \{a,d\}} \theta_{\alpha j} * y_{ij}^{k}} \tag{7.10}$$

$$\max \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{k \in \{a,d\}} \sum_{\alpha} \delta_{\alpha i} * y_{ij}^{k} \tag{7.11}$$

7.5.3 Minimise Number of Towing Operations

This objective aims to minimise the number of towing operations required, Formula 7.12. A towing operation is required every time an aircraft changes its location. The number of towing operations is shown in Figure 7.2.

$$\min \sum_{i=1}^{N} \sum_{j=1}^{M} \left(|y_{ij}^{a} - y_{ij}^{p}| + |y_{ij}^{d} - y_{ij}^{p}| \right) \tag{7.12}$$

7.5.4 Maximise Handler Preferences

Handlers normally provide their services to multiple customers, so one of their preferences may be to concentrate their operations within the minimum number of piers, considering gates within the same pier to be closer to each other than those in other piers.

To take account of the preferences of the different handling agents, it is assumed that fitness increases as the number of assignments to a handler at the same pier increases.

If n_{gj} is the number of assignments to stands in pier $j \in [1 \dots J]$ for agent $g \in [1 \dots G]$, and n_g corresponds to the total number of flights serviced by agent g the fitness may be calculated by Formula 7.13.

$$\max \sum_{g=1}^{G} \sum_{j=1}^{J} \frac{n_{gj}}{n_g} \tag{7.13}$$

7.5.5 Robustness

Different methods for increasing robustness exist, similar to the ABSSAP (Section 3.3.4), where the existence of a gap is taken into account. Given that the model always takes account of the possibility of assigning the parking operation of an aircraft to a remote stand, which is discouraged, then no buffer time is ever associated with a remote operation. Moreover, if the parking activity is assigned to the same stand as the departure activity, then the departure activity will have no reduction in service where the duration of the parking activity is at least as long as the buffer time associated with the departure activity.

If an aircraft j does not have an arrival activity ($T_j^a = B_j^a = 0$) then no parking is considered in the model, which means the base and target service times for the parking operation are both zero: $T_j^p = 0$, (note that B_j^p is already zero). The approach already considers the case where a flight arriving does not have a departing flight associated with it, i.e. after the aircraft has completed the arrival procedure the aircraft is required to follow maintenance procedures for which it will be taken to the appropriate installation to perform any of these necessary maintenance operations.

The robustness approaches used for the AGAP in this thesis are the same as those introduced in Section 3.3.4 and further extended in Chapter 6 for the ABSSAP. The robustness approaches are: 'Distribute Idle Time' (Section 6.3.2), 'Reduce Reassignment on Disruption' (Section 6.3.3), 'Area Reduction in Service' (Section 6.3.4), 'Sub-area Reduction in Service' (Section 6.3.5), 'Unsupervised Estimated Stochastic Reduction in Service' (Section 6.3.6), 'Reduction in the Number of Conflicts' (Section 6.3.7) and 'Probability of Conflicts Based on the Gap' (Section 6.3.8). The 'Minimise Reduction in Service' requires some changes in order to be applicable to the AGAP which are described below.

Minimise Reduction in Service

This objective may be expressed by Formula 7.14.

$$\min \sum_{j=1}^{M} \sum_{k \in \{a,d\}} r_j^k \tag{7.14}$$

The unassigned flights may also be penalised by a cost proportionally higher than the buffer time. In order to treat all unassigned flights equally, and not discriminate between them, this extra cost may involve a multiple of the maximum buffer time for all flights, Formula 7.15 (similar to Section 6.3.1). In this case, given that the number of assignments is the most important objective, the factor used to multiply the maximum buffer time should be higher than two, given that this would represent the buffer time taken by the reduced service incurred when a new assignment is placed between two consecutive flights. In this case, the buffer times at each end are removed, Figure 7.5.

$$\min \left(\sum_{j=1}^{M} \sum_{k\in\{a,d\}} r_j^k + \beta * \max_{j=1,k\in\{a,d\}}^{M} (B_j^k) * \underbrace{\sum_{j=1}^{M} \sum_{k\in\{a,d\}} \left(1 - \sum_{i=1}^{N} y_{ij}^k\right)}_{\text{number of unassigned flights}} \right) \tag{7.15}$$

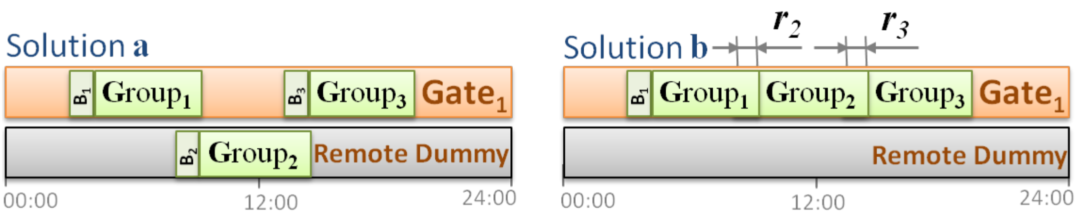


Figure 7.5: Example: two solutions for the same problem with different assignments.

7.6 Constructive Algorithms

The constructive algorithms described here are based on those initially presented in Chapter 4 for the ABSSAP, and are modified here for application to the AGAP. The constructive algorithm performances are then studied in Section 7.9 when applied to some real data sets from London Heathrow airport summarised in Section 7.7.

Algorithm 7 assigns gates to service activities one at a time until no further assignments are possible. Flights are first grouped, based on a cycle of consecutive flight arrival, parking and flight departure activities which use the same aircraft. A parking activity between two consecutive flight arrivals and departures which use the same aircraft is introduced if the gap between the servicing time of the flight arriving and the departure time of the departing flight is sufficiently large, London Heathrow airport considers this should be more than 3 hours. These activities are then ordered according to one of the activity ordering methods under assessment.

These are variants of the ordering methods presented in Section 4.2.1 for the ABSSAP, where the flights refer to the activities requiring service. A set of gates is then selected according to certain restrictions, which are represented here as Algorithms 'A', 'C', and 'E'. Algorithm 'A' corresponds to the most restricted and Algorithm 'E' to one having no restrictions at all, with the remaining approaches lying between these two. These were all previously presented as 'Baggage Sorting Station Assignment Algorithms' in Section 4.2.2 and the activity preference for a pier is replaced by the airline preferences, if any exist. A gate is then selected for each activity in turn from within the set of gates, based on a specific criterion, i.e. Last In First Out (LIFO), First In First Out (FIFO), 'Closest' and random selection. Again these were initially presented as the 'Baggage Sorting Station Selections' in Section 4.2.2 for the ABSSAP.

Algorithm 7: Constructive Algorithms Overview for the AGAP

```
Order activities for assignment (Section 4.2.1 for the ABSSAP);
Determine the sets of feasible gates to consider ('Baggage Sorting Station Assignment
Algorithms' in Section 4.2.2 for the ABSSAP);
foreach activity do
    Select a set;
    repeat
        if the set of feasible gates is not empty then
            Select a gate from the current set based on certain criterion ('Baggage
            Sorting Station Selections' in Section 4.2.2 for the ABSSAP);
            Assign activity to gate;
        end
    until activity has been assigned OR there are no more sets to choose from;
    if activity was not assigned then
        if is a parking activity then
            Assign to the remote dummy gate;
        else
            Assign to the dummy;
        end
    end
end
```

Thus the assignment process follows these two stages:

1. Generating assignments to gates, Algorithm 7.

2. Each aircraft assigned to the dummy remote stand (parking activity) is then assigned to a real stand by

   (a) First attempting to assign it to an existing gate, although not to a remote one. Note that this may still be possible depending on the type of assignment method used.

(b) Otherwise assigning it to a real remote stand.

The procedure currently followed at London Heathrow airport takes no account of the distance passengers have to walk in order to board their flight and leave the airport, and only includes the type of flight, i.e. international or domestic, and the preferences of airlines and handlers. To account for these preferences it is necessary to implement a different 'Baggage Sorting Station Selections' method based on the 'Closest' method previously presented for the ABSSAP in Section 4.2.2 which is described in Section 7.6.1.

Table 7.4 summarises the different elements in the constructive algorithms (Algorithm 7) studied here for the AGAP.

Component	Approaches
Activity Ordering Methods (based on the 'Flight Ordering Methods' for the ABSSAP, Section 4.2.1) * departure time is the activity service end time for the AGAP.	Order by Starting Time (OST), Order by Departure Time (ODT), Order by Departure Time Lookahead and Improvement (ODTLI), Order Between Times (OBT) and Order Between Times Lookahead and Improvement (OBTLI)
Activity Assignment Algorithms (based on the Assignment Algorithms' for the ABSSAP, Section 4.2.2)	'E' (no restriction), 'C', 'A' (most restrictive)
Activity Selections (based on the 'Baggage Sorting Station Selections' for the ABSSAP, Section 4.2.2)	FIFO, LIFO, 'Airline Preference' (Section 7.6.1) and Random

Table 7.4: Summary of the components of the constructive algorithms.

7.6.1 Activity Selections

The Activity Selections is an extension of the 'Baggage Sorting Station Selections' presented in Section 4.2.2 for the ABSSAP, where the resources are gates instead of baggage sorting stations (BSSs). The different selection approaches are FIFO, LIFO and Random which need no modification for use in the AGAP, and 'Closest' which needs to be modified to account for the airline preferences used in this problem. The modifications necessary to include the airline preferences are described in this section, which has been named 'Airline Preference', and it could similarly be extended to consider the handler agent preferences.

In the 'Airline Preference' method the gates available for assignment are organised in decreasing order of preference, which constitutes the set of gates from which to select the gate for assignment. The gate most strongly preferred is selected from those in the set. This preference method is useful for meeting the preference objectives presented in Sections 7.5.2 and 7.5.4. When some gates are equally preferred, a LIFO or FIFO method is used to break the ties. When LIFO is used it equates to minimising the number of open gates, whereas FIFO corresponds to maximising the number of open gates which equates to increasing the fairness, as has also been all seen in Section 4.2.2 for the ABSSAP. This approach may be used to implement selection methods which account for different preferences, i.e. 'Airline Preference' and the 'Handling Agent Preference'.

7.7 Problem Data

Long-haul flights have a buffer time of 900 seconds (15 min) and a service time of 2400 seconds (40 min), whereas the other flights have 600 seconds (10 min) and a service time of 1500 seconds (25 min). The buffer time may only be reduced for pre-scheduled flights, i.e. no buffer time is considered for the parking activities.

Not every aircraft can be parked at all stands, and a stand code identifies the type of aircraft which can be parked at a stand. Each code identifies the largest aircraft which may be parked at the stand. The stand codes at London Heathrow airport are shown in Table 7.5 and stand codes per gate are presented in Appendix A.1.

Code	Comment
F	
E3	744 & 773/A346
E2	744 but not 773/A346
E1	772
D (767-300)	
D (757)	
C (A321)	
C (A319)	

Table 7.5: Stand codes for London Heathrow airport (provided by BAA).

The representation presented here provides two values for each Lower Maximum Assignment Point (LMAP) and Upper Maximum Assignment Point (UMAP) for those occasions when the parking activity is not taken into account and when it is taken into account respectively. They are named according to occasions when parking is not taken into account LMAP, UMAP, and when parking is taken into account Lower Maximum Assignment Point with Parking (LMAP$_p$) and Upper Maximum Assignment Point with Parking (UMAP$_p$). These maximum assignment points comply with the Inequalities $LMAP \leq LMAP_p$ and $UMAP \leq UMAP_p$, given that the LMAP and UMAP only consider flight arrivals and departures, whereas LMAP$_p$ and UMAP$_p$ cover those activities already considered by the LMAP and UMAP together with the parking activities, potentially requiring extra resources to service them all.

A week's record of flight assignments to stands was provided by London Heathrow airport for terminal four, composed of schedules from the 6^{th} to the 12^{th} September 2010 (H4T1009dd). Some details are shown in Table 7.6 which were generated from the data supplied. Using this data summarised in Table 7.6, tables were generated showing the preferences of each airline, and these were used in the 'Maximise Airline Preferences' objective, which is described in Section 7.5.2, and shown in Appendix A.2. Also a table was generated showing the preferences of each handler, which is used in the 'Maximise Handler Preferences' objective described in Section 7.5.4, and shown in Appendix A.3.

ID	Date	LMAP	UMAP	LMAP$_p$	UMAP$_p$	No. Activities	No. Parking Activities
H4T100906	6 Sept 2010	8	10	17	19	118	15
H4T100907	7 Sept 2010	11	14	18	20	120	15
H4T100908	8 Sept 2010	7	10	16	18	119	16
H4T100909	9 Sept 2010	8	10	18	20	119	15
H4T100910	10 Sept 2010	9	12	15	18	120	15
H4T100911	11 Sept 2010	9	10	16	16	110	11
H4T100912	12 Sept 2010	11	11	18	19	117	15

Table 7.6: Data set information provided by British Airports Authority (BAA) for London Heathrow airport Terminal 4.

The topology used for Terminal 4 at London Heathrow airport is shown in Figure 7.6, which is composed of 23 gates and three piers.

The consecutive arrival and departure flights which make use of the same aircraft are considered a group, and if the times between the consecutive flight arrival and flight departure for the same aircraft are less than 3 hours then they are considered as one activity, stretching from the arrival to the departure times, otherwise the service of each individual flight is considered to be an activity, and a parking activity links them both (for the time between the completion of the arrival activity and the commencement of the departure activity).

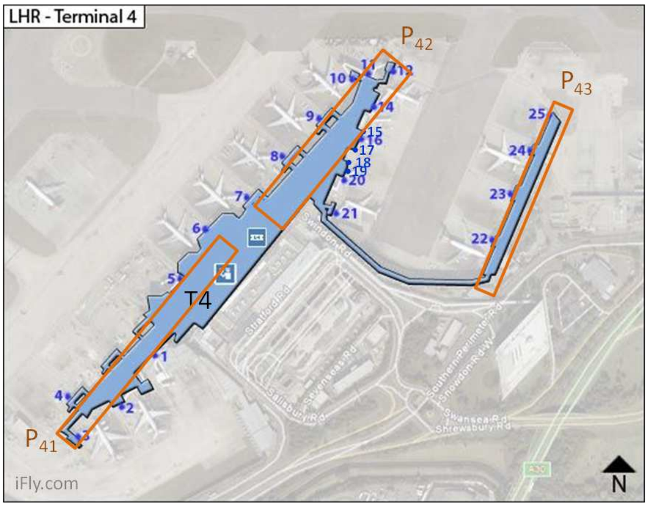


Figure 7.6: General view of London Heathrow airport Terminal 4 composed of three piers.

7.7.1 Generate New Base Schedules

Based on the London Heathrow airport Terminal 4 schedules for 6^{th} to 12^{th} September 2010 and Algorithm 8 new schedules were generated with 37 extra groups, a summary of which is shown in Table 7.7.

Real schedules for London Heathrow airport Terminal 4 were used, which were obtained from BAA. The number of flights are lower than for Terminal 1, however, data for terminals 1, 3 and 5, was not available. Schedules were generated based on the density of the schedules provided for Terminal 4, to investigate the effect on busier terminals as described below.

The consecutive flights serviced by the same aircraft are herein called a group, such that any flight always belongs to a group, although it may be the only flight in its group. For an original schedule of G_o groups, ordered in ascending base starting time (τ_j), a set of group lists is generated, each containing all of the groups where the base starting time is within the same time range (n_i). Algorithm 8 was then used to generate the new schedule.

Experiments were conducted with these new data sets which are studied in Section 8.3.5.

Algorithm 8: Generation of new scheduled from distribution of groups.

```
Input: G_o groups in the base schedule
Input: n_i list of groups with a given base starting time for the original schedule
Input: N number of extra groups, N > 0
Input: rnd random number generator
begin
    // Initialise
    G ← ∅; // empty new group
    j = N/|G_o|; // number of full copies of original
    n = 0;

    // Copy all groups from original schedule
    while j ≥ 0 do
        // Copy groups from original schedule
        forall the g_o ∈ G_o do
            g = clone(g_o); // build copy of original group
            G ← g ∪ G; // add new group to new schedule
        end
        j = j − 1;
        n = n + 1;
    end
    N = N − (n − 1) ∗ |G_o|; // subtract the number of full copies

    // Partial copies
    while N > 0 do
        // Use roulette wheel to select new group
        n = 0; // no copies yet
        p_o = rnd; // probability of being selected
        forall the n_i do
            n = n + |n_i|; // add number of groups in n_i
            p = n/∑|n_i|; // probability of selecting a group from n_i
            if p_o < p then
                // Select randomly a group
                j = rnd(|n_i|); // select randomly a group from n_i
                g_o ← G_o(n_i, j); // get group in n_i at j

                // Build and add group to new schedule
                g = new(g_o); // build a new group with different aircraft
                G ← g ∪ G; // add new group to new solution
                n_i ← g_o\n_i; // remove used group from n_i

                break;
            end
        end
        N = N − 1; // decrease number of groups left to generate
    end

    return G; // returns the new schedule
end
```

ID	Date	LMAP	UMAP	$LMAP_p$	$UMAP_p$	No. Activities	No. Parking Activities
N4T100906	6 Sept 2010	17	20	23	26	164	21
N4T100907	7 Sept 2010	21	23	25	28	160	19
N4T100908	8 Sept 2010	18	20	23	25	169	24
N4T100909	9 Sept 2010	21	21	28	28	168	22
N4T100910	10 Sept 2010	19	20	20	21	164	21
N4T100911	11 Sept 2010	19	21	21	21	154	15
N4T100912	12 Sept 2010	19	21	23	24	167	22

Table 7.7: Generated data sets information with an extra 37 groups.

7.8 Fitness

The problem is one of maximisation, where the fitness function is represented as a weighted sum of the objectives, Formula 7.16, similar to that which was considered for the ABSSAP in Section 5.7. The Reduction in Service (3^{rd} objective, f_3) and Towing (4^{th} objective, f_4) should be minimised (incorporated into the overall weighted fitness function with negative weights), whereas the other objectives are to be maximised. Higher values of Reduction in Service and Towing are undesirable.

$$\sum_{i=1}^{4} f_i * W_i \tag{7.16}$$

The importance of the objectives considered here, based on London Heathrow airport procedure, is as follows: The highest priority is the 'Maximise Number of Assignments' (Section 7.5.1, first objective is f_1), followed by the 'Maximise Airline Preferences' (Section 7.5.2, the second objective arrival f_2), the 'Minimise Reduction in Service' (Section 7.5.5, the third f_3), the 'Minimise Number of Towing Operations' (Section 7.5.3, forth objective f_4) and then 'Maximise Handling Agent Preferences' (Section 7.5.4, fifth and the final objective f_5). The fitness weights for each of the objectives were deduced based on the maximum and minimum values for the different objectives of an extra assignment (Table 7.8), which are summarised in Table 7.9.

The weights W_1, W_2, W_3, W_4 and W_5 are normalised, Equation 7.17. The objective priorities provide the Inequalities 7.18 and 7.19, where Inequality 7.18 states the order of importance of all the objectives, and Inequality 7.19 states that the first objective is more important than the combined contributions of the reduction in service

Objective	Increment	Limit values	
		Min.	Max.
'Maximise Number of Assignments'	Δf_1	0	1
'Maximise Airline Preferences'	Δf_2	0	1
'Minimise Reduction in Service'	Δf_3	$-2 * \max_{j=1}^{N} B_j$	0
'Minimise Number of Towing Operations'	Δf_4	-3	0
'Maximise Handling Agent Preferences'	Δf_5	0	1

Table 7.8: Changes in objectives ('fitness') for an extra assignment.

and towing objectives for a change in the assignments.

$$\sqrt{W_1^2 + W_2^2 + W_3^2 + W_4^2 + W_5^2} = 1 \tag{7.17}$$

$$f_2 * W_1 > f_2 * W_2 > -f_3 * W_3 > -f_4 * W_4 > f_5 * W_5 \tag{7.18}$$

$$f_1 * W_1 > -f_3 * W_3 - f_4 * W_4 \tag{7.19}$$

These inequalities are used to assist in finding appropriate weights. Thus in the case of an extra assignment, the contribution to the first objective is equal to 1; for the second objective the minimum contribution to the airline preference is zero and the maximum is no greater than one; for the third objective the greater effect is equivalent to twice the maximum buffer time as a consequence of adding the new assignment tied between two previously assigned flights, as shown in Figure 7.5. Regarding the next objective, the maximum effect would be to increase towing by a multiplier of three, as a consequence of the flight being assigned to a stand to which a parking activity is already assigned, and which parking activity will then be moved to the remote dummy stand. Finally, the last objective makes a minimum contribution to the handling agent preference of zero and a maximum contribution no greater than one.

Thus when an extra assignment is achieved, the maximum change in the first objective (Δf_1) corresponds to one, and for the second objective (Δf_2) is 1 with 0 as a minimum, and for the third objective (Δf_3) is $-2 * \max_{j=1}^{N}(B_j)$ (maximum reduction in service between two flights assigned to the same gate). The change in the forth objective (Δf_4) is -3 which corresponds to two when the new assignment is between two flights using the same aircraft, and their intermediate assignment must be assigned to the parking dummy stand. One unit more, making it three, is for the flight's own parking assignment to the remote dummy stand. Finally the variation in maximisation of the handler agent preference, the fifth objective (Δf_5), is 1 with 0 as a minimum.

If it is assumed that the reduction in service should also be applied to unassigned activities, Formula 7.15, then Inequality 7.19 converts to $f_1 * W_1 > -f_4 * W_4$. If it is assumed all the contributions are the same as the maximum for one assignment of the reduction in service, approximating all the weights to that for W_3 and using Inequality 7.18 then $|w_3| < 3.149 * 10^{-4}$. So taking a $W_3 = -0.00025$, provides a $|W_4| < \frac{2*900}{3} * 2.5\ 10^{-4} = 0.15 \Longrightarrow W_4 = -0.11$, with $W_5 < 1800 * 0.00025 = 0.45$ and $W_5 < -3 * W_4 = 0.33 \Longrightarrow W_5 = 0.25$ and $W_2 > 1800 * 2.5e^{-4} = 0.4 \Longrightarrow W_2 = 0.6$. By using the normalisation Equation 7.17 then $W_1 = 0.75193$. The choice of a high value for W_2 reflects the importance of airline preferences. The weight values are summarised in Table 7.9.

Weight	W_1	W_2	W_3	W_4	W_5
Value	0.75193	0.6	-0.00025	-0.11	0.25

Table 7.9: List of weights.

7.9 Results of the Constructive Algorithms

Given that the number of gates, N = 23 from Table 7.6, is greater than the UMAP$_p$, it follows that there are sufficient gates to accommodate all of the arrivals, parking and departures without the need to reduce their service time, which meets the third objective. This may, however, be detrimental to the second objective 'Airline Preference', mainly when the selection method used does not take account of this objective.

The results when reduction in service is permitted for the different objectives and Algorithms ('*A*', '*C*' and '*E*') with selection methods (FIFO, LIFO, 'Airline Preference' and Random) for OST and the data sets in Table 7.6 are shown in Figures 7.7, 7.8 and C.1. In the figures a grey arrow with the word 'Better' shows the direction of better values.

Figure 7.7 shows the 'Maximise Number of Assignments' objective (x-axis) for the different data sets (y-axis, Table 7.6), where a dashed grey line represents both the total number of flights which require a gate (lower line) and the total number of activities (which includes the parking activities, upper line). It is shown that only Algorithm '*E*' achieved full assignment of all activities for the selection methods LIFO and 'Airline Preference', where lines overlap with those for the total number of activities. The 'Airline Preference' selection method, which was described in Section 7.6.1, will then try to concentrate the assignments within a group of gates, as the

LIFO selection method does. The selection method LIFO only assigns an activity to a gate which does not already have an assignment whereby it cannot be assigned to the gates which do already have activities assigned to them. As was seen in Chapter 4, such a characteristic helps to achieve a greater number of assignments. However, the restrictions provided by the other algorithms, '*C*' and '*A*', are shown to have a detrimental effect, as they tend to spread the assignments. Given that maximising the number of assignments is the most important objective, then when a fast solution is required either of the selection methods 'Airline Preferences' or LIFO without restriction should be used when there is a plentiful number of gates.

The first column in Figure 7.8 shows the results for the 'Maximise Airline Preferences' objective (Section 7.5.2) for the different selection methods considered. As expected, the selection method 'Airline Preferences' together with the Algorithm '*E*' performs best, but deteriorates when extra restrictions are introduced, such as those provided by the Algorithms '*C*' and '*A*'. Neither selection method LIFO or FIFO show any particular characteristic which makes one better than the other for Algorithm '*E*'.

The second column in Figure 7.8 shows the 'Minimise Reduction in Service Time' objective (Section 7.5.5) which shows that all the selection methods, with the exception of random selection, perform well, especially 'Airline Preferences' and LIFO for Algorithm '*E*'. Given that the number of gates is higher than the UMAP_p, assignment of all the activities without reducing the buffer time is achievable, such as is shown for both the 'Airline Preferences' and LIFO selection methods for Algorithm '*E*'. Once again the introduction of restrictions (Algorithms '*A*' and '*C*') is detrimental to the 'Minimise Reduction in Service Time' objective.

Regarding the 'Minimise Number of Towing Operations' objective (Section 7.5.3), results which are shown in the first column in Figure C.1, the increase in the selection restrictions, represented by the selection methods '*C*' and '*A*', reduce the number of towing operation required. This is a consequence of reducing the number of gates between which activities are assigned, which in turn is based on the airline preferences implemented within the selection methods '*C*' and '*A*', so increasing the chance of assigning a parking activity to the same gate as both its flight arrival or departure. Nevertheless, the 'Airline Preference' selection method with no restrictions achieves the lowest towing as expected.

FIFO does not manage to assign all the activities to gates and archives no towing for all the data sets but H4T100906 (6^{th} Sept 2010). This means that all the parking

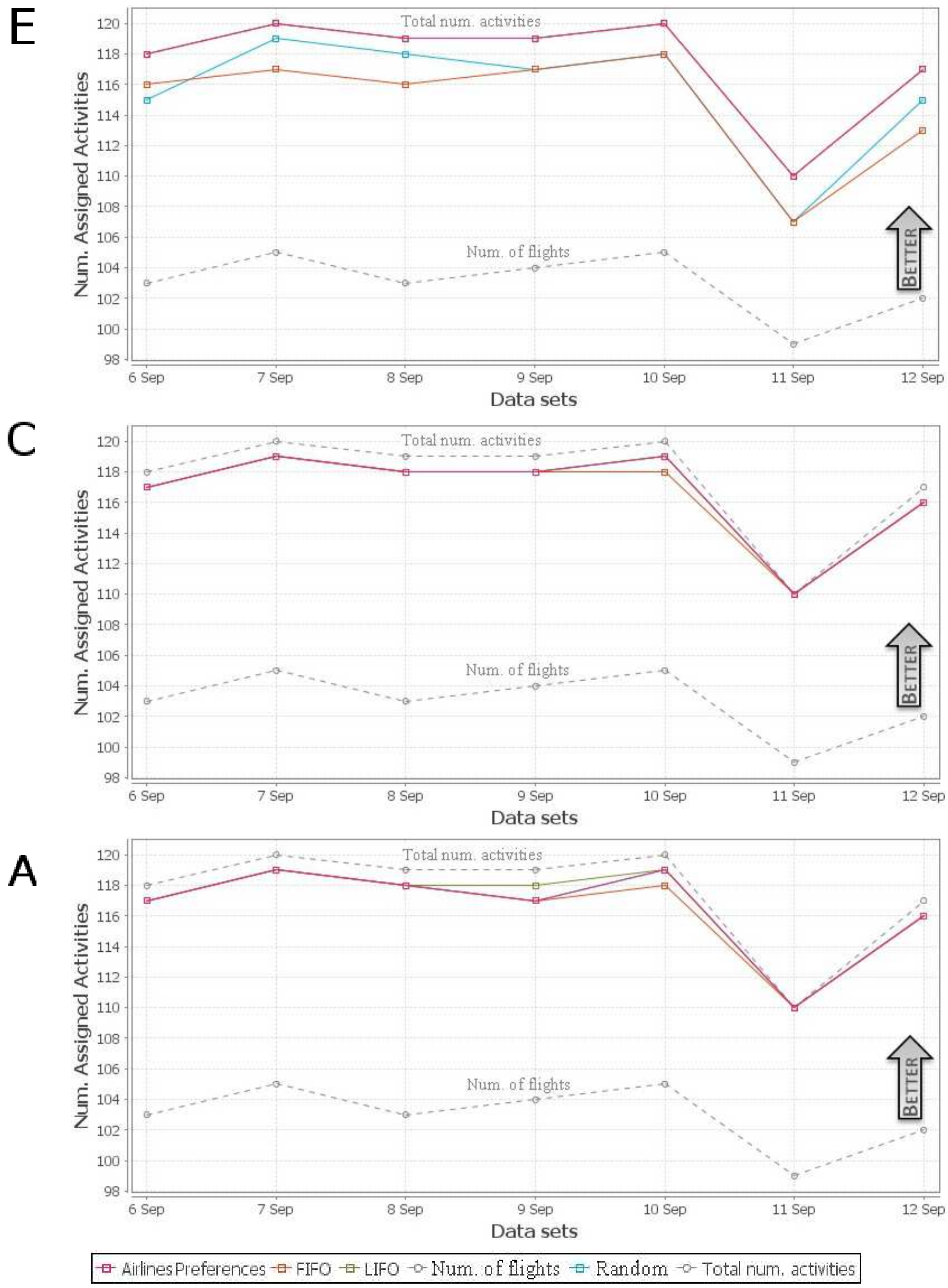


Figure 7.7: Comparison of results for the first objective for the OST ordering method, the four selection methods and three assignment algorithms, for 3-pier and 23 stands.

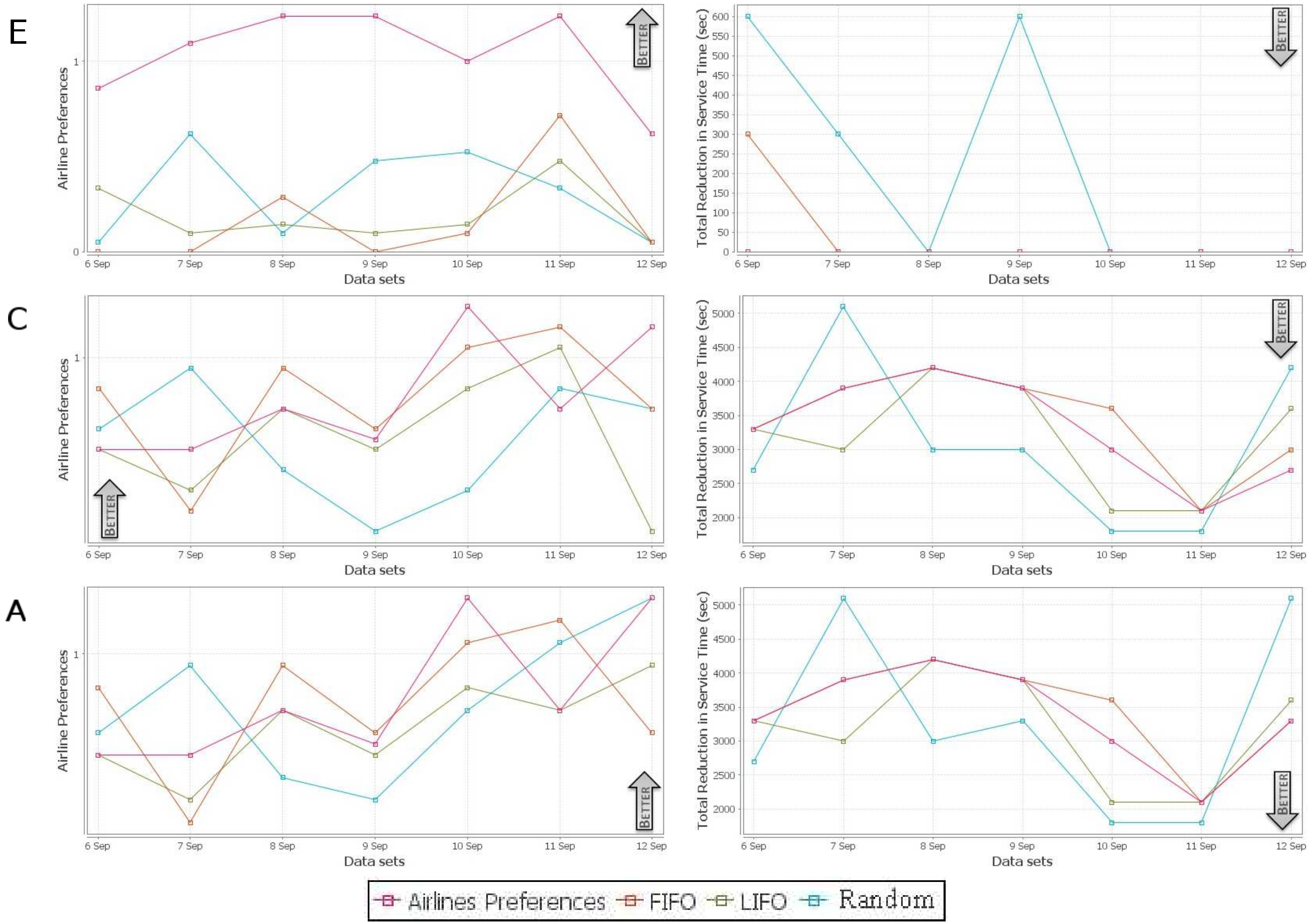


Figure 7.8: Comparison of results for the second and third objectives for the OST ordering method, the four selection methods and three assignment algorithms, for 3-pier and 23 stands.

activities are assigned to the same gate as their arrival and departure flights. So the activities unassigned correspond to groups without parking activity (full service time no longer than 3 hours). This means that the FIFO ordering method does not achieve full flight assignment, the first and most important objective. This solution is not desirable, since those unassigned flights will have to be cancelled.

The ODTLI does not perform as well as OST for those data sets considered where there are sufficient gates to which all the activities may be assigned (including parking activities, $\text{UMAP}_p \leq N$), as shown in Appendix C.1.

As has been shown previously in Section 4.4.1 in the ABSSAP, it appears that OST's preference for assigning first those activities having long service times allows more assignments to be achieved. It therefore follows that activities requiring a shorter service time, when assigned at a later stage, will be more likely to find gates with gaps between assignments large enough to allow for another assignment. OST would be preferable to ODT and ODTLI, where a better solution is required quickly, but they may all be used to provide diverse initial solutions to a population based algorithm in which diversity enhances the algorithm performance as happens in the Evolutionary Algorithm (EA) originally introduced in Chapter 5 for the ABSSAP and adapted to the AGAP in Chapter 8.

7.10 Conclusions

This chapter provided a view of the AGAP, and presents both the model used together with some constructive algorithms. Both model and constructive algorithms are produced by modifying those presented in previous chapters for the ABSSAP, and are based on the specific characteristics of the problem. It has been shown that the potential of these constructive algorithms, presented in Chapter 4, may have their use extended by modifying them so that they can be applied to another problem, such as the AGAP.

The different constructive algorithms and their parameters were studied to find characteristics which may be used to identify the algorithm and the parameters most appropriate to the AGAP for real data. The 'Airline Preference' selection method has been seen to perform better overall than the other selection methods when taking account of the objective priorities. On the other hand, LIFO also performs well in respect of the first and most important objective, but it is not as good as the 'Airline Preference'. The introduction of restrictions represented by the Activity Assignment Algorithms 'A' and 'C' appears to affect the different selection methods in different ways, with the three first main objectives deteriorating when the restrictions are increased, and improving for the minimisation of the towing and handling agent preference.

These constructive algorithms are used to generate a population of solutions for use as initial solutions in the population based algorithms studied in Chapter 8.

Chapter 8

Evolutionary Algorithms for the Airport Gate Assignment Problem

The Steady State Evolutionary Algorithm (SSEA) and robustness approaches in previous chapters are potentially important for a wider variety of problems other than the Airport Baggage Sorting Station Assignment Problem (ABSSAP). By way of illustration, this chapter looks at applying the SSEA and robustness approaches to the more widely studied Airport Gate Assignment Problem (AGAP).

The chapter begins with an overall view of the AGAP, Section 8.1, followed by a description of the modifications required for using some of the metaheuristics previously presented in Chapter 5, and which are now presented in Section 8.2. The SSEA is studied next, and compared with the Canonical Genetic Algorithm (CGA) and Tabu Search (TS) in Section 8.3. Given the importance of assignment performance on the day of operation, some robustness approaches, previously presented in Chapter 6 for the ABSSAP, are modified for use in the AGAP, and are presented in Section 8.4. These robustness approaches are then studied in Section 8.5 and the chapter concludes with a summary and some suggestions in Section 8.6.

8.1 Overview

The assignment of gates already scheduled to flights is known as the AGAP and is one of the most important operations in an airport, having repercussions on many other resources, such as baggage sorting stations (BSSs) (ABSSAP). A description of the problem and some constructive algorithms are presented in Chapter 7. The similarities between the AGAP and the ABSSAP makes particularly interesting to determine whether the SSEA, previously presented for the ABSSAP and studied in Chapter 5, is also appropriate for the AGAP. However, the SSEA needs first to be adapted to the AGAP, Section 8.2, and it is studied in Section 8.3.

When delays, cancellations or early arrivals may cause substantial changes in current assignments, it may no longer be feasible to modify the relevant part of the assignments, since aircraft may already be parked, or a gate assignment may have been announced to passengers. Some of the changes may incur additional costs, for example an increase in the towing required and in the inconvenience to passengers and staff, which has to be balanced against maintaining smooth operations without causing further flight delays. The different approaches, which take account of potential disruption on the day of operation, are presented in Section 7.5.5, and studied in Sections 8.4 and 8.5.

8.2 Steady State Evolutionary Algorithm

In this section the AGAP model presented in Section 7.2 is used in the SSEA originally presented in Chapter 5 for the ABSSAP. The operators and selectors are those introduced in Chapter 5 for the ABSSAP, but the resources are now gates instead of BSSs and only the corresponding constraints and objectives presented in Section 7.2 are applicable. Some modifications are necessary before the SSEA can be applied to the AGAP, and these are the only ones described in this section. The main intention is to establish the suitability of the SSEA for the AGAP. Some experiments were conducted using metaheuristics, the results of which are compared and studied in Section 8.3, showing that the SSEA also provides good results for the AGAP.

New operators are required to allow the reassignment of parking activities from the dummy remote stand to a gate. This is not required if the Multi Exchange Mutation Operators described in Section 5.4 for the ABSSAP are modified, such that their recovery stage also considers the parking activities assigned to the dummy remote stand. This removes the need to use tailored mutation operators to assign parking activities from the dummy remote stand to gates. Some of these tailored operators are presented in Appendix C.2.

The Dummy Single Move Mutation Operator (DSMMO) originally presented in Section 5.4.1 for the BSSs, which here moves assignments from the dummy stand to a gate, is preferred to the Dummy Single Exchange Mutation Operator (DSEMO) (presented in Section 5.4.1 for the BSSs), which exchanges assignments between the dummy stand and a gate, where N is greater than or equal to the Lower Maximum

Assignment Point (LMAP), given that full assignment is achievable, and this operator helps to achieve it. The DSEMO may also need to execute an exchange, which will not improve the number of assignments. Nevertheless, in the early iterations, where there may be many solutions lacking full assignment, these may be forced into assignment by the use of the DSMMO ($N \geq$ LMAP). When $N <$ LMAP it is impossible to reach full assignment, so the DSEMO is preferable to the DSMMO.

Another version of the Multi Exchange Mutation Operators presented in Section 5.4 was therefore generated, whereby the recovery step is extended to cover not only the unassigned activities from one gate to another, but also all of the parking activities assigned to the dummy remote stand. To dispense with the need to use any of the Remote Mutation Operators presented above, the recovery step will always include the parking activities assigned to the remote dummy, by trying to assign all of the parking activities assigned to the remote dummy to gates.

The value of the LMAP and Upper Maximum Assignment Point (UMAP) were calculated on the assumption that all the gates can accommodate any aircraft, but this is not correct in reality. These values are therefore only used here as a reference point, since in the model each flight has its own restrictions regarding the gates to which they can be assigned.

8.3 Results for the Steady State Evolutionary Algorithm

Several experiments were conducted to evaluate the performance of the different operators (Section 8.2) for the problem data presented in Section 7.7. Many results obtained from the different experiments cannot be said to follow a normal distribution for a significance level of 0.05, so the t-test for statistical significance cannot be used and the Mann-Whitney statistical significance test was used instead.

The data sets used in the experiments conducted are presented in Section 7.7, and a summary is presented in Table 8.1 for convenience. Multiple combinations of

Name	Values
Terminal	4
Topology	3-pier
Data sets	London Heathrow airport schedules for Terminal 4 shown in Table 7.6
Fitness weights	$W_1 = 0.75193$, $W_2 = 0.6$, $W_3 = 0.00025$, $W_4 = 0.11$ and $W_5 = 0.25$ (Section 7.8)

Table 8.1: Summary of the general data used in the experiments conducted.

the parameters used in both the SSEA and the CGA were used, the values for which are summarised in Table 8.2.

Name	Values
Total number of iterations	800,000
Population sizes	5, 10, 15, 30, 50, 100, 200, 500, 1000 and 2000
Iterations in a generation (ℓ, only for SSEA)	1, 5, 10 and 20
Replacement Strategies	ES, SUMS, IS1ES and IS1SUMS (later IS1fES and IS1fSUMS are used too)
Member Selectors	Tournament Selection (described in Section 2.7.1)
Operators	C1P, C2P, MEFNR2, MEFNR3, MEFNR4, MEFNR5, IMEFNR2, IMEFNR3, IMEFNR4, IMEFNR5, RMEFNR2, RMEFNR3, RMEFNR4, RMEFNR5, IRMEFNR2, IRMEFNR3, IRMEFNR4, IRMEFNR5

Table 8.2: Summary of the algorithm parameters used in the experiments conducted.

The following section looks at the performance of the SSEA for all combinations of the parameters presented in Tables 8.1 and 8.2, when only one operator is used in each execution. Experiments were next conducted with various combinations of the same operators and parameters, which are compared and studied, and a single operator is used to establish their performance, Section 8.3.2. The results from the experiments conducted for the SSEA are compared to those obtained from the application of other metaheuristics, which are then studied in Section 8.3.3. Some experiments were next conducted to determine the effects of the parking restriction, where a parking activity may only be assigned to a gate if such a gate is already assigned to either the flight arrival or departure associated with it, the results of which are presented in Section 8.3.4. Finally in Section 8.3.5 experiments are conducted to study the effect on the performance of the different operators when the number of flights increases.

8.3.1 Single Operators

In this section the results obtained from the experiments conducted when only one operator is used per run are presented and studied. The parameters are those summarised in Tables 8.1 and 8.2.

Given that the Multi Exchange Mutation Operators lack the ability to assign activities from the dummies, thus keeping fitness low, removal of this disadvantage improves the Multi Exchange Mutation Operator by introducing a recovery stage after the child solution has been generated whereby unassigned activities (those assigned to one of either dummies) are randomly assigned to a gate where possible. This is very important as it is a constraint on the static problem achieving full assignment, $N \geq$ LMAP, as presented in Section 7.5.1. As a solution reaches full assignment this extra step may not be needed, therefore having no detrimental effect on the speed of the operator. The word 'Improved' is added at the beginning of the name of the original base operators to identify the new operators. This explains the poor fitness results obtained when the Multi Exchange Mutation Operators were used when compared with those obtained by their 'Improved' version, as shown in Table 8.3. The Improved Multi Exchange Mutation Operators do not need to be combined with any of the Dummy operators to allow them to increase the number of assignments to gates.

When studying all of the operators for the SSEA it was found that the population sizes which provide the best overall performance correspond to small values of between 5 to 15 for the 'Improved Multiple Exchange Operators' (Table 8.3) similar as was seen in the ABSSAP for the 'Multiple Exchange Operators', Section 5.7.2. Furthermore, this also applies to all of the Multi Exchange Mutation Operators examined, as shown in Tables 8.3 and 8.5. Tables 8.3 and 8.5 only show those replacement strategies and population sizes that cannot be said to be statistically significantly less fit (Mann-Whitney test) than any of the other replacement strategies and population sizes for each of the operators and data sets studied. The population sizes in bold text are those which can be said to provide significantly statistically fitter solutions in many more cases than the other combinations considered.

A summary of all of the experiments conducted for the SSEA using different combinations of ℓ, operators, selectors and population sizes (Section 7.7) is presented in Table 8.3, which only shows those combinations which best solution obtained cannot be said to be significantly statistically less fit than the best solutions obtained for any of the other combinations. Table 8.3 clearly shows that the Index Selection with Elitist Selection and a group size of 1 (IS1ES) replacement strategy provides significantly statistically better solutions overall, which is similar to the result obtained for the ABSSAP shown in Section 5.7.4. The only operators covering all of the data sets studied, where a fitness solution cannot be said to be significantly statistically worse than the solutions obtained when using any of the other combinations, are:

1. $\ell = 1$, IRMEFNR2 and IS1ES and population size of 5

2. $\ell = 5$, IRMEFNR2 and IS1ES

Similarly, the good performance of the operator IRMEFRN2 was also seen for the ABSSAP in Chapter 5 for operator RMEFRN2, particularly where the number

ℓ	Operator	Selector	Population sizes			
			H4T0100906	H4T100907	H4T100908	H4T100909
1	IMEFNR2	IS1ES	5, 10, 15, 30	**5**	5	5, 10, **15**, 30
		IS1SUMS	5			
		SUMS	5, 10, 15, 50, 500		15, 50, 200, 1000	
	IRMEFNR2	IS1ES	5, 15	5	5	5, 10
	RMEFNR2	IS1ES				10
5	IMEFNR2	IS1ES	10	5	5, 15	5, 10
	IRMEFNR2	IS1ES	**5**, 10, 15	5	5, 15	5, 10
	MEFNR2	IS1ES				5
		SUMS			200	
10	IMEFNR2	IS1ES	5, 10, 15	**5**	5	5, 10
	IRMEFNR2	IS1ES	5, 10, 30	10	**5**, 10	5, 10
	MEFNR2	IS1ES	5			
20	IMEFNR2	IS1ES	5, 10, 15		10	5, 10
	IRMEFNR2	IS1ES	5, 10	10	5, 10	5
	MEFNR2	IS1ES	15			10

ℓ	Operator	Selector	Population sizes		
			H4T100910	H4T100911	H4T100912
1	IMEFNR2	IS1ES	5	5, 10	
		SUMS	**15**, 30, 100, 2000	10, 50, 100, 500, 2000	
	IRMEFNR2	IS1ES	5, 15	5, 15	5, 10
5	IMEFNR2	IS1ES		5	**5**
	IRMEFNR2	IS1ES	15	**5**	10
10	IRMEFNR2	IS1ES	5	5, 10	
	MEFNR2	IS1ES		5	
20	IMEFNR2	IS1ES	15		
	IRMEFNR2	IS1ES	5	5	
	MEFNR2	IS1ES		10	

Table 8.3: SSEAℓ single operators which provide statistically significantly fitter solutions for the data sets from 6^{th} to 12^{th} September 2010, where the 'Index Selectors' remove all duplicates.

of BSSs is greater than the UMAP, which also equates here to the data set cases considered for Terminal 4 at London Heathrow airport.

The 'Index Selector' operators remove all solutions having the same fitness, but which do not have interesting parts useful in future generations. Furthermore, this also reduces the fitness pressure, since there are fewer solutions with duplicates. This approach may be too strong so a new version of the Index Selection with Elitist Selection (ISxES) and Index Selection with Stochastic Universal Modified Sampling (ISxSUMS) is proposed, whereby duplicates are removed only if the population is greater than expected. The results, which include the new 'Index Selector', are shown in Table 8.4, where an 'f' is inserted in the selector's name to represent the new 'Index Selector'. The performance of both versions of the 'Index Selector', with and without full removal of duplicates, generally appears to be close. The overall approaches achieving solutions where fitness is no worse than in the other operators in all the data sets considered are:

1. $\ell = 1$, IMEFNR2 and IS1fES

ℓ	Operator	Selector	Population sizes			
			H4T100906	H4T100907	H4T100908	H4T100909
1	IMEFNR2	IS1ES	5	5, 10, 15	5	5, 10, 15
		IS1fES	10	**10**	10	10
		SUMS		15, 50	15, 50, 200, 1000	
	IRMEFNR2	IS1ES	5	5, 10	5	5, 15
		IS1fES			10	10, 15
5	IMEFNR2	IS1ES	5		5, 15	
		IS1fES	**5**	5, 10	5	5
	IRMEFNR2	IS1ES	5	5	5, 15	5
		IS1fES	5	5		5
	MEFNR2	SUMS			200	
10	IMEFNR2	IS1ES	5	10	5	
		IS1fES	5	10	5, 10	5
	IRMEFNR2	IS1ES			**5**, 10	5, 10
		IS1fES	5		5, 10	5
20	IMEFNR2	IS1ES		10, 15	10	5, 10
		IS1fES	5		5, 10	**5**, 10, 15
	IRMEFNR2	IS1ES	10		5, 10	5
		IS1fES		5		
	MEFNR2	IS1ES		15		

ℓ	Operator	Selector	Population sizes		
			H4T100910	H4T100911	H4T100912
1	IMEFNR2	IS1ES	5	5, 10	
		IS1fES	5	5, 15, 30	5, 10
		SUMS	**15**, 30, 100, 2000	10, 50, 100, 500, 2000	
	IRMEFNR2	IS1ES		5, 15	5, 10
		IS1fES		5	5
		IS1fSUMS		5	
5	IMEFNR2	IS1ES	5, 10	5	**5**
		IS1fES	5, 10, 15	5, 10, 15	
	IRMEFNR2	IS1ES	15	**5**	10
		IS1fES	5, 10	5, 15	5
10	IMEFNR2	IS1ES			
		IS1fES	5	5, 10, 15	
	IRMEFNR2	IS1ES	5	5, 10	
	MEFNR2	IS1ES		5	
20	IMEFNR2	IS1ES	15		
		IS1fES		5	
	IRMEFNR2	IS1ES	5	5	
	MEFNR2	IS1ES		10	

Table 8.4: SSEAℓ single operators which provide statistically significantly fitter solutions for data sets from 6^{th} to 12^{th} September 2010.

2. $\ell = 5$, IRMEFNR2, IS1ES and population size of 5

When the new 'Index Selector' is used these empirical results show an improvement by the IMEFNR2 over previous results using a lower number of iterations per generation (ℓ), which also corresponds to a small increase in search pressure as less fit solutions in the population have less chance of being selected (the diversity is retained for longer at a higher ℓ). Whereas, the removal of duplicate solutions by the replacement strategy equates to an increase in diversity. The new 'Index Selectors' were not applied to the ABSSAP. These results show that the solutions obtained may potentially be improved when using this selection enhancement in other resource assignment problems, such as the ABSSAP, especially for data sets where a slight extra increase in the search pressure may be advantageous. Furthermore, the ability to activate or deactivate this characteristic may be beneficial as the search advances, based on the particular circumstances at each time, such as switching it off later in the search when diversity has sufficiently decreased in order to slightly reduce the search pressure.

8.3.2 Multiple Operators

In order to establish which operator combinations provide solutions which are statistically significantly fitter than the solutions obtained by single operators, or which cannot at least be said to be worse, two operator approaches, composed of a crossover and mutation, are analysed and compared with the single operators for the AGAP. It emerges that multiple operators do not perform well when compared to single operators. This may be because the crossover operators cannot provide different solutions, where the parent solutions selected are identical, which may occur more often at a later time in the search when the population has lost diversity, such that many more duplicates and solutions with less differences may exist. To alleviate or remove this disadvantage, the solution selector should be modified to take account of the operator characteristics which will be used when selecting solutions to generate a new solution. Alternatively the crossover operators may be used only early in the search, where more diversity exists and there are therefore fewer duplicates. Furthermore, cases may exist where even when the parent solutions are different, the new solutions are the same as some of the solutions already present in the population. This may, however, be reduced by increasing the population size and using a replacement strategy which is able to maintain the population diversity for longer, namely Stochastic Universal Modified Sampling (SUMS) and ISxSUMS.

Furthermore, given the good performance of the mutation operators when used in the SSEA on their own, some experiments were executed using a 'Probability Single Multi Operator' (Section 5.4.3) with both a crossover operator and a mutation operator with a probability from 0.1 to 0.9. The solutions obtained by the 0.1 crossover + 0.9 mutation were statistically significantly fitter than those obtained by the other combinations of operators and either of the crossover operators on their own. The 1-point crossover did perform better than the 2-point crossover, which may be attributed in part to the fact that some service periods are very long and the extra hard constraints, i.e. parking, can only be assigned to a gate which has previously been assigned to the arrival and/or departure flights of the same aircraft. This may in turn reduce efficiency when using time regions, as it is more probable that the time limits of the region fall within those long time services, so reducing the number of activities with which to exchange. It also appears that IS1ES and SUMS assist in reaching fitter solutions.

8.3.3 Other Metaheuristics

Following the results for the SSEA, two other metaheuristics are studied and compared with the proposed SSEA. Firstly, the TS, described in Section 5.5, is considered, which adds the best solution in each local walk to the tabu list. Table 8.5 shows a summary of the statistical significance with a significance level of 0.05 for the TS, where operators provide solutions which cannot be said to be less fit than any of the solutions obtained by the other operators, and which also cover all of the data sets for different local walk sizes and tabu list sizes. These empirical results show that the TS performs better for multi exchange mutation operators with a higher number of gates between which to exchange assignments, i.e. $n = 3$, than those seen for the SSEA. Higher n in the Multi Exchange Mutation operators corresponds to children with a potentially greater number of differences than their parents, which should mean more diversity over a longer period.

Algorithm	Walk Size	Operator	Tabu List Sizes			
			H4T1009dd			
			6	7	8	Thur 9
TS	10	IMEFNR3	5	5, 10	5, 15	15
		IRMEFNR3	5, 30	15	30	15
	30	IRMEFNR3	5, 15	10	5, 15	5, 10, 15, 30

Algorithm	Walk Size	Operator	Tabu List Sizes		
			H4T1009dd		
			10	11	12
TS	10	IMEFNR3	10, 30	10	5, 30
		IRMEFNR3	5, 10, 15, 30	10, 15	5, 10, 15, 30
	30	IRMEFNR3	5, 10, 15, 30	30	5, 10, 15

Table 8.5: TS summary of statistical significance of fitness with a significance level of 0.005 and different 'tabu list' sizes for the data sets of September 2010.

Several experiments were designed and executed to examine the performance of the CGA described in Section 2.7 when using the same operators as previously described, so this implementation of the CGA does not correspond to the standard definition of the CGA since it does not make use of a binary representation, and does not use binary or random mutation operators as classically presented in Holland (1975). Both 1-point and 2-point crossover operators together with one of the previously described mutation operators were used, with a probability of 0.99 for the crossover operators and 0.01 for the mutation operators with population sizes of 500, 1000 and 2000. Also based on the good performance of the mutation operators, some experiments with a probability of 0.9 crossover and 0.1 mutation were executed to establish whether a higher percentage of mutation operators was desirable for the data sets used. These were later extended to include a population size of 400, given that a number of the experiments provided good results for a population size of 500.

Table 8.6 provides a summary of the results, only showing those operators which provide solutions which cannot be said to be statistically significantly less fit than any of the other operators studied for the CGA. A reduction in the preferred population size can be seen for the combined operators when compared with the single operators used, as was also previously seen when using SSEA. The solutions obtained by the probability of 0.9 crossover + 0.1 mutation were significantly statistically fitter than those obtained by the other combination of operators for the CGA, and cannot be said to be less fit than those solutions obtained by the alternative operators evaluated. These results show that a higher participation by mutation operators is advantageous in this problem.

The 1-point crossover did perform better than the 2-point crossover, which may partly be attributable to the fact that some service periods are very long and the extra hard constraints, i.e. assignment of parking activities , may reduce the efficiency when using time regions. In these cases, it is more probable that the time limits of the region fall within those long services, thus reducing the number of assigned activities for use in the children. Table 8.6 only shows combinations of operators with the replacement

Operator		Selector	Population Sizes H4T1009xx						
Crossover 0.9	Mutation 0.1		06	07	08	09	10	11	12
C1P	IMEFNR3	ES	400 and 500	400, 500 and 2000	400 and 500	400, 1000 and 2000	400 and 500	400, 500 and 1000	400, 500, 1000 and 2000
	IRMEFNR2	ES	400 and 500	500 and 1000	400	400, 500 and 1000	400, 500 and 2000	400, 2000	2000
	MEFNR2	ES	500	400, 500, 1000 and 2000	400, 500 and 2000	400, 500, 1000 and 2000	400, 1000 and 2000	500 and 1000	2000
	RMEFNR2	ES	400	400 and 500	400	1000 and 2000	400, 1000 and 2000	1000	400 and 2000

Table 8.6: CGA summary of statistically significant fitness with a significance level of 0.005, 1-point (C1P) and 2-point (C2P) crossover, mutation operators and different population sizes.

strategy Elitist Selection (ES) which indicates that the tendency of the ES in reducing the diversity, which is normally maintained for longer by the larger population sizes typically used by crossover operators when used alone, is a beneficial one for the data sets considered.

8.3.4 No Restrictions on Assigning Parking Activities

Some experiments were also executed when no hard constraint was applied to the assignment of parking activities, such that they could be assigned to any gate. The algorithms have been identified by appending an extra character, '+', to the name. The summary Tables 8.7 and 8.8 only show those combinations providing solutions no less fit than any other solution considered using all of the data sets. Given that both models with and without the extra parking constraint appear in Table 8.7, then both models cannot be said to provide statistically significantly worse solutions than the other. Nevertheless the preferred population sizes for the SSEA and the IRMEFNR2 operator is slightly lower than when the parking hard constraint is in place, whereas for the crossover the preferred population sizes are also slightly lower.

8.3.5 Generate New Base Schedules

In this section some experiments were designed and executed to evaluate the performance of the operators when used in the SSEA for a lower number of available gates for the number of flights. A set of new schedules was generated based on those already available from London Heathrow airport Terminal 4 from 6^{th} to 12^{th} September 2010 as described in Section 7.7.1 and summarised in Table 7.7.

The new data sets all have a number of gates (N = 23) larger or equal to the UMAP, so it should be possible to assign all the flights to gates. However on reviewing the parking activities there was found to be an insufficient number of gates to which to assign all of the activities ($N <$ Lower Maximum Assignment Point with Parking (LMAP_p)), with the exception of the first and last data sets. The most difficult data sets to solve were for H4T100907 with UMAP equal to 23 BSSs, which is also the number of gates, followed by the H4T100909, which has the highest LMAP_p and Upper Maximum Assignment Point with Parking (UMAP_p) as well. Thus all flights from the sets generated can be assigned to a gate as $N \leq$ UMAP, which meets one of the conditions for a real static airport problem.

Experiments using the SSEA were executed for the new base schedules, and a summary of the results is shown in Table 8.9. The results show an improvement on

Algorithm	Operator		Selector	Population Sizes						
				H4T1009xx						
				06	07	08	09	10	11	12
CGA	C1P 0.9	IRMEFNR2 0.1	ES	500, 1000	400, 500, 1000	400	400, 500	400, 500, 2000	400, 2000	400, 500, 1000, 2000
		MEFNR2 0.1		400, 500, 1000, 2000	400, 500, 1000, 2000	400, 500, 2000	500	1000, 2000	500, 1000	1000, 2000
		RMEFNR2 0.1		400, 500	400, 1000, 2000	400	400	400, 1000, 2000	1000	400, 500, 2000
CGA+	C1P 0.9	IMEFNR3 0.1	SUMS	400, 500	400	400	400, 500	400, 500	400	400, 500
		IMEFNR3 0.1		400	400	400	400	400, 500	400	400, 500
SSEA	1	IMEFNR2	IS1fES	10	10	10	10	5	5, 15, 30	5, 10
	5	IRMEFNR2	IS1ES	5	5	5, 15	5	15	5	10
SSEA+	1	IMEFNR2	IS1ES	5, 10, 15	5, 10, 15, 30	5, 10, 15, 30	10, 15	10, 15	5, 10	5, 10, 15, 30
		IRMEFNR2		5, 10, 15, 30	5, 10, 15, 30	5, 10, 15, 30	30	30	5, 10	5, 10, 15, 30
	5	IMEFNR2		5, 10, 15	5, 10, 15, 30	5, 10, 15, 30	30	30	5, 10, 30	5, 10, 15, 30
		IRMEFNR2		5, 10, 15, 30	5, 10, 15	5, 10, 15, 30	5, 10, 15, 30	10, 15, 30	30	5, 10, 15, 30
	10	IMEFNR2		10, 15, 30	5, 10, 15, 30	5, 10, 15, 30	15, 30	15, 30	5, 10, 15, 30	5, 10, 15, 30
		IRMEFNR2		5, 10, 15, 30	5, 10, 15	5, 10, 15	10, 15, 30	10, 15, 30	15	5, 10, 15, 30
	20	IMEFNR2		5, 10, 15, 30	5, 10, 15, 30	5, 10, 15, 30	10, 15	10, 15	5, 15	5, 10, 15, 30
		IRMEFNR2		5, 10, 15	5, 10, 15	5, 10, 15	30	30	5, 10	5, 10, 15, 30

Table 8.7: Summary of algorithms which provide statistically significantly fitness solutions for the data sets from 6^{th} to 12^{th} September 2010 and both models.

the solution obtained by the IMEFRN2, IMEFRN3 and IRMEFRN3 as is similarly seen in Chapter 5 for MEFRN2, MEFRN3 and RMEFRN3 when fewer resources (BSSs and gates) are available.

Results of the experiments executed for the CGA, different operators and the data sets generated, are summarised in Table 8.10, which shows a preference for lower population sizes than when used alone in the SSEA, which may be taken as an indication of the mutation operator's influence. This also reveals a preference for a lower use of the crossover operators and a higher use of the mutation operators, indicating

Algorithm	Walk Size	Operator	Tabu List Size H4T1009xx						
			06	07	08	09	10	11	12
TS	10	IMEFNR3	5, 10	15	5, 15	5	10, 30	10	15
		IRMEFNR3	15	15	30	5, 30	5, 10, 15, 30	10, 15	5
	30	IRMEFNR3	10	5, 10, 15, 30	5, 15	5, 15	5, 10, 15, 30	30	5, 15
	50	IRMEFNR3	5, 30	5, 15, 30	10	5, 30	5, 10, 15, 30	10, 15	5, 10, 30
TS+	10	IRMEFNR3	5, 10, 15, 30	5, 10	5, 30	5, 10, 15	10, 30	5, 10, 15, 30	5, 10, 15, 30
		IRMEFNR4	5, 10, 15, 30	5, 30	10, 15	5, 15	5, 10, 15	10	5, 15, 30
	30	IMEFNR3	5, 10, 15, 30	5, 10, 15, 30	10, 15	30	15, 30	5, 10, 15, 30	5, 10, 15, 30
		IRMEFNR4	10, 15, 30	10, 15	5, 30	5, 10, 15, 30	5, 15, 30	10, 15	5, 15, 30
	50	IMEFNR3	10, 15, 30	10, 15, 30	5, 10, 15, 30	10, 15	15	10, 15	5, 10, 15, 30
		IRMEFNR4	5, 30	5, 10, 15, 30	10, 30	5, 15	5, 15	5, 15, 30	5, 10, 15, 30

Table 8.8: TS with a single operator which provides statistically significantly fitness solutions for the data sets from 6^{th} to 12^{th} September 2010 and both models.

ℓ	Operator	Population Sizes N4T100906	N4T100907	N4T100908	N4T100909
1	IMEFNR2	5		**10**, 15, 30	**5**, 10
	IRMEFNR2			3	5, 10
	IMEFNR3		5, 10, 30		
	IRMEFNR3		10, 30		
5	IMEFNR2	5, 10, 15		15, 30	5
	IRMEFNR2	10, 15		5, 10, 30	**5**, 10
	IMEFNR3		15		
	IRMEFNR3	5	5, 10		
10	IMEFNR2	10, 15		5, 10, 30	
	IRMEFNR2			10, 15, 30	
	IMEFNR3		**5**, 15		
	IRMEFNR3	5			
20	IMEFNR2	5, 10, 15		5, 10	5, 10
	IRMEFNR2	5, 10		15	
	IMEFNR3		5, 15		
	IRMEFNR3		5		

ℓ	Operator	Population Sizes N4T100910	N4T100911	N4T100912
1	IMEFNR2	5	5, 10	5, 15
	IRMEFNR2	**5**	**5**	5
5	IMEFNR2			10
	IRMEFNR2	5	10	5
10	IMEFNR2		5, 10	**10**, 15
	IRMEFNR2		5	
20	IMEFNR2	5		
	IRMEFNR2		5, 10	5, 15

Table 8.9: SSEAℓ single operator with significantly statistically fitter solutions for data sets generated from original date sets from 6^{th} to 12^{th} Sept 2010 and 23 gates.

a departure from the general view as to what extent mutation operators should be used in a CGA, as may also be noted in the results of the real data sets for London

Operator		Selector	N4T1009xx						
Crossover 0.9	Mutation 0.1		06	07	08	09	10	11	12
C1P	IMEFNR2	IS1ES	400, 500, 1000	**400**, 500	400, 500	400, **500**	400, 500	**400**, 500	**400**, **500**
		SUMS		400					
	IRMEFNR2	IS1ES	400, 500	400, 500, 1000	400		**400**, 500	**400**, **500**	**400**, **500**
		SUMS		400, 500				400, 500	400
	IMEFNR3	IS1ES			400		400	400, 500	
		SUMS						500	
	IRMEFNR3	IS1ES	400		500	400, 500		**400**, 500	
C2P	IMEFNR2	IS1ES			400, 500	400			
	IMEFNR3	IS1ES			500				
	IRMEFNR3	IS1ES			400				

Table 8.10: CGA significantly statistically fitter solutions for data sets generated from original date sets for 6^{th} to 12^{th} September 2010 and 23 gates.

Heathrow airport Terminal 4. The deterioration in the performance of the 2-point crossover operator may in part be a consequence of the need to identify two cutting points in time which delimit the time region within which the assignments are copied from each parent. This, plus the long service time postulated, may excessively reduce the effective range of assignments from which to copy, thus reducing the operator's efficiency. The crossover operator does not consider assignments where the base service duration lies between two different time sections from which to copy, and this situation is more likely to occur in cases of longer base service duration.

8.4 Robustness

The provision of solutions which reduce the potential detrimental effect of perturbations in the resources already assigned on the day of operation is desirable and was previously studied for the ABSSAP in Chapter 6.

Some approaches attempt to consider potential disruptions at an early stage, so as to reduce their effect on the day of operation, but at the expense of the optimality, although this is far from easy, as the perturbations are not known in advance. It would also be advantageous if the disrupted assignments have no knock-on effect or if any, only a minor one.

A flight is said to be in conflict if the departure time of the flight is greater than the arrival time of the next flight at a gate. A situation may arise when reassigning conflicting flights or the subsequent flight to another gate, where the reassigned flight is interfering with the subsequent flight at the new gate. Thus some reassignments may therefore have a downstream effect on the overall schedule, producing more conflicting flights requiring further reassignments, and potentially increasing the difficulty of the problem at a later stage.

The approaches considered here take account of the potential disruptions on the day of operation, and are those presented in Chapter 6. The approaches are: Total Reduction in Service Time (TRS) originally described in Section 6.3.1, Area of Reduction in Service (ARS) described in Section 6.3.4, Sub-Area of Reduction in Service (SARS) described in Section 6.3.5, Unsupervised Estimated Stochastic Reduction in Service (UESRS) described in Section 6.3.6, Reduction in the Number of Conflicts (RNC) described in Section 6.3.7 and Probability of Conflict Based on the Gap (PCBG) described in section 6.3.8. Only the TRS have some differences which are described in Section 8.4.1.

8.4.1 Total Reduction in Service

The arrival and departure flights correspond to the arrival and departure activities respectively. The time between the scheduled arrival time at the stand and the time at which the flight is scheduled to leave is called the base service duration. A predefined period of time, called buffer time, the value of which depends on the flight, is pre-appended to the flight base starting service time, so that such buffer time may be reduced to allow other assignments to be placed before this flight, but the base service duration must not be affected. The use of buffer service time implies a preference for a greater predetermined service time for each flight, and this buffer time may be obtained from historical information. A reduction in the buffer time for the arrival and departure of aircraft j has been named r_j^a and r_j^d respectively, and the sum of these constitutes the reduction in service cost, so this objective can be expressed as $-\sum_{j=1}^{M}\sum_{x}^{a,d} r_j^x * \sum_{i=1}^{N} y_{ij}^x$, which is described in more detail in Section 6.3.1.

If the remote parking activity is assigned to the same stand as the departure activity, then the reduction in service for the departure flight is zero. This is a consequence of both activities referring to the same aircraft.

8.5 Robustness Results

In this section some experiments are conducted using the robustness approaches summarised in Section 8.4 for the respective weights shown in Table 8.11.

These weights are smaller than the weights used in the ABSSAP because they have been normalised, Equation 7.17. The results are summarised in tables which only show the robustness approaches which, at least in one instance of the disruptions

Approach	Weight	Parameters	
		Name	Values
ARS	0.00025	Buffer Time	15 min long-haul and 10 min others
ATRS	0.00025	Buffer Time	15 min long-haul and 10 min others
BSARS	0.00025	Buffer Time	15 min long-haul and 10 min others
PCBG	0.225 and 0.3125	Std. deviation	10, 20 and 30 min
TRS	0.00025	Buffer Time	15 min long-haul and 10 min others
UESRS	0.225 and 0.3125	Estimator	Exp 0.03, Exp 0.05, Inverse 6, Inverse 15, Linear, OffsetInverse 6, OffsetInverse 15, OffsetSublinear 0, OffsetSublinear 1000 and Sublinear
		Buffer Time	15 min long-haul and 10 min others

Table 8.11: Weights for the different robustness approaches considered with SSEA1.

for a given standard deviation, provide statistically significantly less collisions than other approaches evaluated, and cannot be said to be statistically worse than any of the approaches considered. The tables show for each standard deviation the number of times an approach cannot be said to be statistically significantly worse than any of the other approaches. The last column provides the sum of each result for each of the standard deviations. The case where all instances in a given standard deviation cannot be said to be statistically worse than any other are shown in bold text, and in underlined text for those cases which provide the highest number of all the approaches considered.

The results for the different robustness approaches, when applied to the data sets in Table 7.6 (data sets from British Airports Authority (BAA) for London Heathrow airport Terminal 4) are summarised in Table 8.12. There is no apparent statistical difference between them for short disruptions (10 min standard deviation). For longer disruptions it is the UESRS with exponential unsupervised estimation function with $\beta = 0.03$, weight of 0.3125, with and without TRS approach which performs best for each of the similar disruptions considered. These results correspond to data sets where there is a sufficient number of gates for assignment to all of the activities ($N <$ UMAP$_p$). No general gain is shown by combining the base approach with TRS. Nevertheless, there seems to be no detriment in combining with TRS either. The approaches ARS and Base Sub-Area Reduction in Service (BSARS) do not perform well in any instance when either used alone or combined with TRS, which has also been observed when the rate of activities per gate increases (Table 8.13). These results also corroborate those presented in Lim and Wang (2005), namely, when the number of gates is greater than the UMAP, the exponential unsupervised estimation function performs better, but only when compared with the other unsupervised estimation functions.

Approach	Standard Deviation (x) in min			Total
	10	20	30	
0.225PCBG(x)+0.00025TRS(2)	**7**	1	1	9
0.3125PCBG(x)+0.00025TRS(2)	**7**	1	4	12
0.225PCBG(x)	**7**	0	2	9
0.3125PCBG(x)	**7**	1	4	12
0.225UESRS(E0.03)+0.00025TRS(2)	**7**	4	5	16
0.225UESRS(E0.05)+0.00025TRS(2)	**7**	1	0	8
0.225UESRS(I4)+0.00025TRS(2)	**7**	1	0	8
0.225UESRS(I6)+0.00025TRS(2)	**7**	3	0	10
0.3125UESRS(E0.03)+0.00025TRS(2)	**7**	6	6	19
0.3125UESRS(E0.05)+0.00025TRS(2)	**7**	2	2	11
0.3125UESRS(I4)+0.00025TRS(2)	**7**	3	1	11
0.3125UESRS(I6)+0.00025TRS(2)	**7**	2	1	10
0.225UESRS(E0.03)	**7**	5	6	15
0.225UESRS(E0.05)	**7**	3	1	11
0.225UESRS(I4)	**7**	1	0	8
0.225UESRS(I6)	**7**	3	0	10
0.225UESRS(I15)	**7**	3	3	12
0.3125UESRS(E0.03)	**7**	6	6	19
0.3125UESRS(I4)	**7**	1	0	8
0.3125UESRS(I6)	**7**	3	2	12
0.3125UESRS(I15)	**7**	6	5	18

Table 8.12: Summary of statistical significance of AGAP robustness (significance level 0.05) using perturbed schedules generated from normal distributions of 10, 20 and 30 min standard deviations (x), data sets H4T1009dd and SSEA1 (Appendix C.2.3).

Table 8.13 shows the summary results for the new data sets with an extra 37 groups for the same number of gates (a summary of data sets is shown in Table 7.7). These data sets are equivalent to a reduction in the number of gates available per group, representing more activities for the same number of resources. The UESRS approaches alone or in combination with TRS still perform well for low disruptions (particularly with the exponential estimation function with $\beta = 0.05$), and is even better than the PCBG(x), but PCBG(x) subsequently performed better for longer disruptions. The ARS and BSARS also achieved solutions with statistically significantly less collisions when they were used together with the TRS (see Chapter 6) than when used alone, but not when compared to UESRS and PCBG(x).

The empirical results show, when comparing the results of Tables 8.11 and 8.12, that combining the approaches with TRS helps to reduce the number of collisions where there is a lower number of gates per activity. These results suggest that when fewer resources (gates) are available the increase in the influence of the buffer time is advantageous, given that there is more chance of future disruptions as there is less 'idle time' available for the overall problem. It is therefore anticipated that combining both UESRS and PCBG(x) with other approaches using the buffer time, such as ARS and

Approach	Standard Deviation (x) in min			Total
	10	20	30	
0.225PCBG(x)+0.00025TRS(2)	3	3	5	11
0.3125PCBG(x)+0.00025TRS(2)	2	6	6	14
0.225PCBG(x)	2	4	6	12
0.3125PCBG(x)	3	5	5	13
0.225UESRS(E0.03)+0.00025TRS(2)	5	0	0	5
0.225UESRS(E0.05)+0.00025TRS(2)	6	1	0	7
0.225UESRS(I4)+0.00025TRS(2)	3	0	0	3
0.225UESRS(I6)+0.00025TRS(2)	5	0	5	5
0.3125UESRS(E0.03)+0.00025TRS(2)	6	2	1	9
0.3125UESRS(E0.05)+0.00025TRS(2)	6	4	2	12
0.3125UESRS(I4)+0.00025TRS(2)	5	0	0	5
0.3125UESRS(I6)+0.00025TRS(2)	4	1	0	5
0.225UESRS(E0.03)	3	0	0	3
0.225UESRS(E0.05)	4	1	0	5
0.225UESRS(I4)	4	0	0	4
0.225UESRS(I6)	3	0	0	3
0.225UESRS(I15)	4	1	0	5
0.3125UESRS(E0.03)	5	2	0	7
0.3125UESRS(I4)	5	0	0	5
0.3125UESRS(I6)	4	1	0	5
0.3125UESRS(I15)	3	1	0	4

Table 8.13: Summary of statistical significance of AGAP robustness (significance level = 0.05) using perturbed schedules generated from normal distributions of 10, 20 and 30 min standard deviations (x) and SSEA1 for new data sets N4T1009dd with 37 extra groups each (Appendix C.3).

BSARS, should also further improve the results. The ARS and BSARS are tailored to take account of the influence of the flights distribution over time, so increasing the penalty in periods where there is a higher demand for gates, which the experiments indicate improves results.

8.6 Conclusions

Different algorithms and their parameters were studied to find characteristics which could be used to identify the algorithm and parameters most appropriate to the AGAP. Both the model and algorithms are derived by modifying those presented in previous chapters, and are based on the specific characteristics of the problem.

These approaches were tested on real data from London Heathrow airport, using a fitness function composed of the weighted sum of the different real world objectives currently used in London Heathrow airport. When there were plentiful gates to which the flights could be assigned, there was little difference between the algorithms studied (SSEA, CGA and TS) when the same operators were used. Nevertheless, the mutation operators used are potentially faster than the crossover operators and have been able to cover more search space. There is also potential for combining these algorithms to generate new ones, which may improve the solutions still further. This potential may also be better fulfilled by combining algorithms with significant differences in their underlying approach, e.g. SSEA which is a population based approach, and TS which is an individual approach (local search).

The SSEA has been shown to provide fitter solutions for the Improved Range Multi Exchange with Fixed Number of Resources with two gates between which to exchange assignments (IRMEFNR2), and a sufficient number of gates to which all flights and parking activities can be assigned ($N \geq \mathrm{UMAP}_p$, Upper Maximum Assignment Point with Parking) with a preference for the IS1ES, and in some cases also its modified version (IS1fES). As the number of gates in the problem decreases, both the original and improved Multi Exchange between Fixed Number of Resources (MEFNRn and IMEFNRn) with a higher 'n' (number of gates between which to exchange assignments), are preferable, as was similarly seen in Chapter 5 for the ABSSAP.

The TS with Multiple Exchange Mutation operators has been seen to perform better for a higher number of gates between which to exchange assignments than the same operators for the SSEA. This is believed to be a consequence of the higher number of gates between which to exchange assignments extending the search to a wider area of the search space. This would help to find a better solution, but only where the extra search space covered is not too wide, since this may also reduce the effectiveness of the iterations as there is more danger of straying into disinterested areas of the search space. The SSEA achieves the same, however, partly as a result of the differences within its population of solutions, so it does not necessarily need to extend the search further as this may well increase the number of iterations used to investigate uninteresting areas of the search space. This effect depends on both the problem under study and the model used in the two algorithms, which have a direct impact on the shape of the search space, as can be seen when comparing these results with those obtained for the ABSSAP in Chapter 5. The good performance of the Multiple Exchange Mutation operators was also seen to extend to the CGA, where a higher probability of mutation was preferred, which I attribute to the good results provided by these mutation operators. Nevertheless too high an intervention by the mutation operators has been seen to be detrimental, perhaps in part due to the potential disruptive effect of the mutation operators.

It is envisaged that use of the serial crossover presented in Chapter 5 might be better suited to this problem than the 1-point and 2-point crossover operators studied here, given the longer service time required for the combined arrival/departure flights and parking. In the case of the serial crossover, since cut(s) in the time delimiting the areas of assignments for copying only affects gates, they are less likely to have activities between the two intersection sites than in the case of the other crossover operators used here, which clearly affects activities are copied from each parent.

The time an aircraft expends parked at a gate has a considerable effect on the operations which take place up-stream in the overall airport operation, especially when some of the resources required, such as gates, are limited. Delays in starting the departure sequencing may have important effects on the departure itself, which in turn may also require other aircraft to extend the time during which they are held at the gates. This could well affect other flights arriving which have had the same gates assigned to them. It would be therefore advisable to account for the effect of potential disturbances in the assignment plan and so some approaches were considered. The number of conflicts in perturbed schedules were used as a means of comparing the performance of different approaches. It was concluded that the empirical results indicate that the PCBG did not provide such good results as the UESRS regarding those conflicts where there are plentiful gates to which to assign activities. PCBG performance improved as the number of gates available to service the activities is reduced. Furthermore, it was noted that the close relationship between the PCBG approach and the perturbed base schedule used to calculate the conflicts, and which provides some bias in favour of the PCBG, may be reduced or removed if the buffer times, considered for the other robustness approaches, are modified accordingly. The combination of UESRS and PCBG with TRS provides good solutions, and there is still some potential for combining UESRS and PCBG with other approaches, such as ARS and BSARS, which take account of other problem characteristics which both UESRS and PCBG do not, so potentially further improving the robustness of the solutions reached.

To establish the validity of the model (original model) a different model was also considered (new model) where the extra constraint for the parking activity does not exist. The empirical results for the new model when compared with the original model show that both models find good solutions and it cannot be said that either is better. Nevertheless the new model performs well in a wider range of parameter values, making it preferable. However, as the number of parking activities increases, so the search space also increases. This will increase the time required by the new model to find good solutions, so the use of the extra constraint (original model) should reduce the time taken to find good solutions, making it preferable.

Chapter 9

Conclusions and Future Directions

The main focus of this thesis is on investigation of the Airport Baggage Sorting Station Assignment Problem (ABSSAP) using real life examples from London Heathrow airport. This research was then extended to the Airport Gate Assignment Problem (AGAP) showing its more general applicability. This involved the analysis of other approaches previously presented in the AGAP literature, the definition of new approaches, the investigation of exact approaches, large scale simulations to estimate the operational performance of the assignments, and rigorous analysis of the results.

The research presented in this thesis was driven by the desire to understand the ABSSAP, an area of the airport operation left unexamined until now, in order to provide better solutions, robustness assignments, and to understand better the influence of expectations, i.e. the interaction and trade-off between the multiple objectives, the robustness and the characteristics influencing the assignments. The research included in this thesis has contributed toward a better understanding of the assignment of baggage sorting stations (BSSs) to flights at a passenger airport, and the fresh approaches presented here have also been shown to be appropriate for the AGAP.

The remainder of this chapter summarises the main contributions of this thesis and draws conclusions from the work presented.

9.1 Contributions

9.1.1 Constructive Algorithms

Constructive algorithms have been used previously in both the ABSSAP (Abdelghany et al (2006)) and the AGAP (Ding et al (2004)) as feasible initial solutions to algorithms for solving these problems. Order by Departure Time (ODT) and Order by Departure Time Lookahead and Improvement (ODTLI) flight ordering methods, no reduction in service time and assignment algorithm '*E*' (so that all resources are considered, rather than only those on the preferred pier) and Last In First Out (LIFO) ordering, guarantees maximum assignments by minimising the wasted/idle time between flights, Ding et al (2004) and Cormen et al (2001). However, this does not consider a variable service time, restrictions, as represented by the assignment algorithm.

There is an advantage in using longer service times, by adding a buffer time to the base service duration. The buffer time may be reduced if it helps to increase the number of assignments, since longer service duration helps to absorb potential disruptions on the day of operation and ease the workload, so reducing the chance of mistakes.

A framework for constructive algorithms was presented and used to generate some specific constructive algorithms tailored to take account of the airport topology and the position of the BSSs. This framework can easily be applied to generate more algorithms where other considerations may need to be taken into account such as alternative grouping strategies, in the same way as when extended for application to the AGAP in Chapter 7. The constructive algorithms considered take account of various different conflicting objectives normally present in the ABSSAP and AGAP, allowing the generation of diverse solutions. They are able to provide high quality solutions quickly, which may be used as initial solutions in further algorithms such as exact methods (Branch and Bound (B&B)), and metaheuristics (Tabu Search (TS), Canonical Genetic Algorithm (CGA) and Steady State Evolutionary Algorithm (SSEA)). The ability to obtain solutions favouring the different objectives typically present in some types of problem may be advantageous, particularly in population based algorithms such as those belonging to the Genetic Algorithms (GAs) group, as was seen when they were used as initial solutions to different Evolutionary Algorithms (EAs). The baggage sorting station selection methods Order Between Times (OBT) and Order Between Times Lookahead and Improvement (OBTLI) group into one the different constructive algorithms controlled by an extra parameter, which simplifies and speeds up the generation of solutions, taking the different objectives into account. They order the activities based on a point in the activities service time.

A useful contribution has been the identification of two useful points from the flight density distribution, which measure the number of resources required to achieve a full assignment of activities, when no buffer time is used and when buffer time cannot be reduced. These points are the Lower Maximum Assignment Point (LMAP) and the Upper Maximum Assignment Point (UMAP), respectively. They divide the range of numbers of BSSs into three zones. The first zone is when there is an insufficient number of resources to achieve full assignment of activities to resources (a number of resources lower than the LMAP). The second zone is where there are sufficient resources but only at the expense of reducing the buffer time (an equal or greater number of resources than the LMAP but fewer than the UMAP). The third zone is where there is a sufficient number of resources to achieve full assignment without the need to reduce the buffer time (the number of resources is greater than or equal to the UMAP). The reduction in the buffer time corresponds to a decrease in the robustness of the assignments, which may easily be affected by disruptions on the day of operation. These points were seen to assist in identifying characteristics of the different algorithms and their parameters throughout this thesis.

In summation, the constructive algorithms presented provide high quality solutions in a very short time, which have proved useful as initial solutions for some other algorithms, particularly those which benefit from a population of diverse solutions. The performance of the constructive algorithms has also been seen to vary depending on the number of resources available, but their fast generation and fitness make them particularly interesting as initial solutions, especially in those cases where a very quick generation of solutions is required in a very short time.

9.1.2 Steady State Evolutionary Algorithm

One of the aims of this thesis was to develop and study algorithms and search operators for use in a decision support tool to assist airport resource managers in the assignment of flights to BSSs and gates. The SSEA presented in this thesis is, to the best of the author's knowledge, a new EA. The SSEA combines a search strategy with operators and a fitness function, which take account of the different objectives, such as in the GAs.

However, EAs also use a population of solutions and in the case of GAs a search strategy based on natural evolution. The new algorithm presented (SSEA) allows the inclusion in its search strategy of some of the processes used in the classical GAs.

Whereas in a GA a generation involves all the population of parent solutions and the children constitute the next population of parents, in the SSEA a generation may not involve all the population and the children from the previous generation will join those from which the new population is obtained. Additionally, the operators used do not necessarily include the crossover operator followed by a mutation, which is characteristic in GAs. Furthermore, the operators introduced do not make use of a binary or integer representation of the problem which it is typical of the classical GAs.

Evolutionary Algorithms are population based search methods which rely on the population to provide search direction. Genetic diversity is regarded as necessary to spread the search to other areas of the search space having potential, but eventually requiring convergence to the optimal or a good solution. The convergence is necessary but insufficient, since the type and rate of convergence is more likely to be the cause of failures. Some of the methods to control the diversity and rate of convergence available within an EA are to vary the number of iterations in a generation, the selection methods, or the operators which are responsible for finding promising solutions.

It was shown in Section 5 that the SSEA provides fitter solutions than those provided by other algorithms, such as the CGA and TS, for the ABSSAP and results were similar when used for the AGAP. The AGAP model used is more restrictive since the assignment of some activities has extra constraints to be complied with. Furthermore the real data sets used are composed of fewer flights requiring assignment than those data sets used for the ABSSAP, which increase the suitability of exact methods such as B&B. The SSEA was shown in Section 5 to provide fitter solutions than the B&B when using CPLEX for a wide range of BSSs. It was important to consider which operators were being used when deciding on the appropriate population size. An analysis of the performance of different operators when the number of BSSs changes from few to when there are plentiful was conducted for the SSEA, CGA and TS. This gives an idea of which operators were preferred, based on the number of BSSs. This study was later extended to the AGAP to determine the validity of the SSEA and the performance of these operators. The SSEA not only provides fitter solutions when run for long time, but fitter solutions are also found in very short run times, which suggest a potential for use in solving the dynamic problem.

9.1.3 Operators

Some new operators were proposed, which are based on those presented in the AGAP literature. These operators were also used throughout this thesis in different metaheuristics. They can be grouped into mutation and crossover operators. The problems considered could be generalised as Activity Assignment Problems, where there are some resources, e.g. BSSs or gates, and some activities which require servicing by a resource for a period of time, where some constraints need to be complied with and compliance with some objectives is desirable. In this thesis the activities are flights which have already been scheduled and the resources are BSSs for the ABSSAP or gates for the AGAP.

New mutation operators

Multiple new mutation operators were introduced, which are all local search (guided mutation) operators, and which generate feasible solutions. They exchange assignments between different resources within a time range, but cannot increase the number of assignments. These solutions were shown to help the algorithm improve on the original solutions. Therefore, when used it may be advisable to combine them with other operators which have the ability to increase the number of assignments, so further enhancing the solution provided. An alternative version was that where following application of the base mutation operator, a recovery stage attempted to assign the remaining unassigned activities to any of the resources, potentially increasing the number of activities assigned. This improved version was applied to the AGAP and performed better than its base version.

It was seen that the solution method tends to benefit from these mutation operators, with a higher number of resources between which to exchange assignments when there are a low number of resources. These mutation operators are also very fast and have been shown to provide quick improvements when applied to the problems studied using the SSEA. This ability is desirable if the algorithm has a limited time in which to run, as is normally the case for the dynamic problem.

Different implementation of crossover operators

A new implementation of the crossover operators was presented in Section 5.4.2 wherein the cuts in time, from which the assignments are copied from the parent to the child, are made on each resource. This could be seen as the execution of the typical crossover on each resource followed by a final recovery stage. These operators were seen to perform better for very low numbers of BSS in the ABSSAP.

9.1.4 Replacement Strategies

New replacement strategies were presented to improve the performance of some typical selection methods presented in the literature. It was seen that the Elitist Selection performed well but showed a tendency to stagnate. Thus a new selection method was presented to reduce the number of solutions with the same fitness. This was mainly deduced from the hypothesis that solutions with the same fitness will generally be very similar, so too many would significantly reduce the diversity and effectiveness of the population size. This selection method was later extended to examine other base selection methods as well as the elitist, and also to remove only those solutions exceeding the population size. These new selection methods improved the solutions by balancing the need for diversity to extend the search to other areas of the search space, and the need to converge to optimal or good solutions provided by the search pressure.

9.1.5 Robust Scheduling

The time during which an activity requires a gate has a considerable effect on the operations which take place up-stream in the overall operation, especially when some of the resources required are limited, such as BSSs and gates. Delays in completing service of an activity may have important effects on the operations which follow. This could well affect other activities, to which those resources are also assigned. This is particularly true for the problems studied in this thesis.

Multiple disrupted schedules were used to obtain a measure of the robustness of all of the robustness approaches considered, the measure itself being the average number of conflicts a solution has when applied to those disrupted schedules.

Multiple approaches to account for robustness

Multiple approaches were presented and studied which take account of potential disruptions on the day of operation. It was noticed particularly that the Unsupervised Estimated Stochastic Reduction in Service (UESRS) provided the best results but given that these approaches make use of the buffer time it is necessary that the buffer time reflects the real problem case in order to take full advantage of these approaches, as discussed in Chapter 6. Further improvement was seen when the UESRS was combined with the Total Reduction in Service Time (TRS) and it is envisaged that this may be improved even further when combined with the new robustness approaches introduced in Sections 6.3.4 and 6.3.5, which take account of the time of day by using the flight density at that time.

Many of the approaches assessed use the idea of introducing buffer times between two assignments to reduce the chance that disruptions in the system will not require flights to be reassigned, or where this is necessary the reassignments are kept to a minimum and ensure that such reassignments do not affect other assignments. One of these approaches is the TRS, which is the total reduction in the buffer time between assignments, and which has been seen to perform worse than the other approaches. However, long gaps between assignments not only reduce the chance of the assignments being affected by perturbations on the day of operation, but may also reduce the chance of some reassignments affecting other existing assignments, since they may be reassigned to these gaps once the gaps are sufficiently long. The approaches in this thesis which consider the full gap, and in some cases take account of a buffer time, are the 'Minimise Reduction in Service Time' using arctangent (Section 6.3.1), the UESRS (Section 6.3.6), and the Probability of Conflict Based on the Gap (PCBG) (Section 6.3.8). Furthermore, some experiments using real data from London Heathrow airport showed in general that the Area of Reduction in Service (ARS) and the Base Sub-Area Reduction in Service (BSARS) in general when used alone performed better than TRS, and improved when each was combined with TRS.

New robustness approaches

Many approaches consider the disruption to be independent of the number of assignments required at each time on a given day, but this is not realistic since disruptions in periods of high activity are more likely to propagate throughout the rest of the day, with potentially expensive consequences, as shown in Section 6.4.1. This must therefore be taken into account when assigning the resources to the activities. Three new approaches were introduced which take account of these potential future disruptions in the assignments based on the time of the day, and make use of the flight density to calculate the cost of disruptions. These approaches were seen to perform better than the TRS when used alone and further improved the robustness when considered together with the TRS.

9.2 Extensions and future directions

The empirical investigation in this thesis suggests many possible directions for future research.

9.2.1 Model

Dynamically calculate the number of BSSs required to service each flight

The BSSs may not all have the same capacity, so this should be borne in mind when selecting them. Furthermore, better results and robustness may be obtained if the number of BSSs required for each flight is not fixed, but depends on the capacity of the BSSs assigned to each flight and the expected checked-in baggage load on each occasion. This means that the model not only evaluates the BSSs assigned to each flight, but decides when each assignment should commence, since they may not all start at the same time, thus increasing the BSSs availability when servicing other flights or absorbing disruption on the day of operation. An acceptable assumption would be that the end of the service time for all BSSs assigned to the same flight will be the same, since it is anticipated that the volume of checked-in baggage increases as it nears the check-in desk closing time and the flight departure time.

9.2.2 Evolutionary Algorithm

The results in Chapter 5 demonstrate several areas for improving the solutions.

Approaches to considering the operator preferred population size when using multiple operators

When using combined operators the different operators' preferred population sizes should be taken into account. It may be the case that one of many operators is selected at each iteration, and that each operator may perform well for very different population sizes. It may not therefore suffice to take a compromised population size obtained from that of the preferred population size for each of the combined operators used. It is accordantly suggested that a population size equal to the maximum preferred size be derived from all the operators considered, and that the member selector takes account of the operator using the selected members to generate the new solutions.

If the population of parent solutions is ordered using the Index Selection with Elitist Selection (ISxES) (in descending fitness order), the number of parent solutions required by the operator are selected from within the first solutions in the population of parents equal to the population size for that operator. In such an approach, the solutions within the population may change from one generation to another since new fitter solutions will be closer to the beginning of the solutions list constituting the population, irrespective of which operators generated them, so they will also have more chance of selection as parents for the next generation.

Study the performance of the Improved Multi Exchange Mutation Operators for the ABSSAP

The favourable results seen in the improved version of the Multi Exchange Mutation Operators for the AGAP in Section 8.3 also suggest that this could potentially be used in the ABSSAP. This would remove the need to combine the base Multi Exchange Mutation Operators with other operators which allows the number of assignments to be increased.

Steady State Evolutionary Algorithm with Ageing

The Steady State algorithm includes the population of parent solutions in a generation of the population derived from that generation, which may potentially reduce the diversity too much. The population diversity of an EA is an important factor in the avoidance of premature convergence Michalewicz (1996). An ageing factor may be incorporated in the individuals which affects their fitness, reducing the effect of retaining the parents between generations, such that with the same base fitness the individual of a greater age will have a lower real fitness than a younger one. Alternatively, individuals on reaching a certain age could be removed from the population straight away. This assists in maintaining diversity in the population.

9.2.3 Robustness

Improvement on the robustness by combining different approaches

The combination of the UESRS, specifically with the exponential estimation function, and one of the new robustness approaches, ARS (Section 6.3.4), BSARS (Section 6.3.5) and Sub-Area of Reduction in Service (SARS) (Section 6.3.5), presented in this thesis, has the potential to further improve the robustness of the solutions obtained, as identified in Section 6.4.2.

Improvement of the robustness by using different information to obtain better buffer times

Further work should consider improving the buffer time used for each flight, and which may also be different for each of the assignments required by a flight. An analysis of historical data may be very useful for better identification of good buffer time values.

Improve the distributions of delays

Future work should consider the use of multiple distributions, in general one per flight, based on the particular characteristics of each flight, such as aircraft, airline, destination, route, season, time of the day, etc. This would be particularly interesting for both the PCBG and Reduction in the Number of Conflicts (RNC) approaches, which were presented in Chapter 6. This was not used since such information was not available at the time this study was conducted.

Bibliography

Abdelghany A, Abdelghany K, Narasimhan R (2006) Scheduling baggage-handling facilities in congested airports. Journal of Air Transport Management 12(2):76 – 81, DOI 10.1016/j.jairtraman.2005.10.001

Administration FA (2012) Air Traffic Organization Policy. U.S. Department of Transportation

Air Transport Users Council (2009) Annual report 2008/09. Tech. rep., AUC

Alam S, Nguyen M, Abbass H, Lokan C, Ellejmi M, Kirby S (2010) A dynamic continuous descent approach methodology for low noise and emission. In: Digital Avionics Systems Conference (DASC), 2010 IEEE/AIAA 29th, pp 1.E.5–1–1.E.5–18, DOI 10.1109/DASC.2010.5655502

Anagnostakis I, Clarke JP (2003) Runway operations planning: A two-stage solution methodology. Hawaii International Conference on System Sciences 3:79a, DOI 10.1109/HICSS.2003.1174196

Andersson K, Carr F, Feron E, Hall WD (2000) Analysis and modeling of ground operations at hub airports. In: 3rd USA/Europe Air Traffic Management R&D Seminar (ed) 3rd USA/Europe Air Traffic Management R&D Seminar, pp 1 – 15

Arabas J, Michalewicz Z, Mulawka J (1994) GAVaPS-a genetic algorithm with varying population size, vol 1, IEEE Press, pp 73–78

Artiouchine K, Baptiste P, Dürr C (2008) Runway sequencing with holding patterns. European Journal of Operational Research 189(3):1254 – 1266, DOI 10.1016/j.ejor.2006.06.076

Ascó A, Atkin JAD, Burke EK (2011) The airport baggage sorting station allocation problem. In: Fowler J, Kendall G, McCollum B (eds) In proceedings of the 5th Multidisciplinary International Conference on Scheduling : Theory and Applications (MISTA 2011), 9-11 August 2011, Phoenix, Arizona, USA, MISTA, pp 419–444

Ascó A, Atkin JAD, Burke EK (2012) An evolutionary algorithm for the over-constrained airport baggage sorting station assignment problem. In: Bui L, Ong Y, Hoai N, Ishibuchi H, Suganthan P (eds) 9th Intl. Conf. on Simulated Evolution And Learning, SEAL2012, Lecture Notes in Computer Science, vol 7673, Springer Berlin Heidelberg, pp 32–41, DOI 10.1007/978-3-642-34859-4_4

Ascó A, Atkin JAD, Burke EK (2013) An analysis of constructive algorithms for airport baggage sorting station assignment. Journal of Scheduling

Atkin JAD (2008) On-line decision support for take-off runway scheduling at london heathrow airport. PhD thesis, School of Computer Science, University of Nottingham

Atkin JAD, Burke EK, Greenwood JS, Reeson D (2007) Hybrid metaheuristics to aid runway scheduling at london heathrow airport. Transportation Science 41(1):90 – 106, DOI 10.1.1.64.3782

Atkin JAD, Burke EK, Greenwood JS, Reeson D (2008) On-line decision support for take-off runway scheduling at london heathrow airport. Journal of Scheduling 11(5):323 – 346, DOI 10.1007/s10951-008-0065-9

Atkin JAD, Burke EK, Greenwood JS, Reeson D (2009) An examination of take-off scheduling constraints at london heathrow airport. Public Transport 1(3):169 – 187, DOI 10.1007/s12469-009-0011-z

Atkin JAD, Burke EK, Ravizza S (2010) The airport ground movement problem: Past and current research and future directions. In: 4th International Conference on Research in Air Transportation, Budapest, Hungary

Atkin JAD, Burke EK, Greenwood JS (2011) A comparison of two methods for reducing take-off delay at london heathrow airport. Journal of Scheduling 14(5):409 – 421, DOI 10.1007/s10951-011-0228-y

Babic O, Teodorovic D, Tosic V (1984) Aircraft stand assignment to minimize walking. Journal of Transportation Engineering 110(1):55 – 66, DOI 10.1061/(ASCE) 0733-947X(1984)110:1(55)

Bäck T, Hoffmeister F (1991) Extended selection mechanisms in genetic algorithms. Morgan Kaufmann, pp 92–99, DOI 10.1.1.42.2192

Baker JE (1987) Reducing bias and inefficiency in the selection algorithm. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, pp 14–21

Baker JE (1989) An analysis of the effect of selection in genetic algorithms. PhD thesis, Graduate School of Vanderbilt University, Nashville, Tennessee

Baron P (1969) A simulation analysis of airport terminal operations. Transportation Research UK 3(4):481 – 91

Barth T, Pisinger D (2011) Scheduling of outbound luggage handling at airports. In: Operations Research Proceedings, Operations Research, Springer-Verlag Berlin Heidelberg, pp 251 – 255, DOI 10.1007/978-3-642-29210-1_40

Bertsimas D, Sim M (2004) The price of robustness. Operational Research 52(1):35 – 53, DOI 10.1287/opre.1030.0065

Bertsimas D, Brown DB, Caramanis C (2011) Theory and applications of robust optimization. Society for Industrial and Applied Mathematics 53(3):464–501, DOI 10.1137/080734510

Bianco L, DellOlmo P, Giordani S (2006) Scheduling models for air traffic control in terminal areas. Journal of Scheduling 9(3):223 – 253, DOI 10.1007/s10951-006-6779-7

Bihr RA (1990) A conceptual solution to the aircraft gate assignment problem using 0, 1 linear programming. Computers & Industrial Engineering 19(1–4):280 – 284, DOI 10.1016/0360-8352(90)90122-3

Blickle T, Thiele L (1996) A comparison of selection schemes used in evolutionary algorithms. Evolutionary Computation 4(4):361 – 394, DOI 10.1162/evco.1996.4.4.361

Böhme D, Brucherseifer R, Christoffels L (2007) Coordinated arrival departure management. 7th USA/Europe Air Traffic Management R&D Seminar, Barcelona, Spain

Bolat A (1999) Assigning arriving flights at an airport to the available gates. Journal of the Operational Research Society 50(1):23 – 34

Bolat A (2000) Procedures for providing robust gate assignments for arriving aircrafts. European Journal of Operational Research 120(1):63 – 80

Bolat A (2001) Models and a genetic algorithm for static aircraft-gate assignment problem. The Journal of the Operational Research Society 52:1107–1120, DOI 10.1057/palgrave.jors.2601190

Bolender MA (2000) Scheduling and control strategies for the departure problem in air traffic control. PhD thesis, University of Cincinnati ETDs

Burgain P, Feron E, Clarke JP (2009) Collaborative virtual queue benefits analysis of a collaborative decision making concept applied to congested airport departure operations. Air Trafic ControlQuarterly 17(2):195 – 222

Burke E, Gustafson S, Kendall G (2004) Diversity in genetic programming: An analysis of measures and correlation with fitness. IEEE Transactions on Evolutionary Computation 8:47–62, DOI 10.1109/TEVC.2003.819263

Burke EK, Kendall G (eds) (2005) Search Methodologies Introductory Tutorials in Optimization and Decision Support Techniques, 2nd edn. Springer Science+Business Media, Inc

Cant-Paz E (1998) A survey of parallel genetic algorithms. CALCULATEURS PARALLELES 10

Capozzi BJ (2003) Automated airport surface traffic control. AIAA Guidance, Navigation, and Control Conference and Exhibit, American Institute of Aeronautics and Astronautics, AIAA 2003-5772, pp 11 – 14

Caprì S, 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(5):345 – 351, DOI 10.1016/j.jairtraman.2004.05.004

Castillo Tapia M, Coello Coello CA (2007) Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp 532–539, DOI 10.1109/CEC.2007.4424516

Cela E (1998) The Quadratic Assignment Problem: Theory and Algorithms (Combinatorial Optimization). Springer

Cheng VHL, Crawford L, Menon P (1999) Air traffic control using genetic search techniques. In: Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on, vol 1, pp 249–254, DOI 10.1109/CCA.1999.806209

Cheng Y (1997) A knowledge-based airport gate assignment system integrated with mathematical programming. Computers & Industrial Engineering 32(4):837 – 852

Clare G, Richards A (2009) Airport ground operations optimizer

Clarke JPB, Ho NT, Ren L, Brown JA, Elmer KR, Tong KO, Wat JK (2004) Continuous descent approach: Design and flight test for louisville international airport. JOURNAL OF AIRCRAFT 41(5):1054 – 1066, DOI 10.2514/1.5572

Coello CAC, Lamont GB, van Veldhuizen (Author) DA (2007) Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation), 2nd edn. Springer, DOI 10.1007/978-0-387-36797-2

Commission E (2010) Commission staff working paper. Tech. rep., European Commission

Cormen TH, Leiserson CE, Rivest RL, Stein C (eds) (2001) Introduction to Algorithms. The Massachusetts Institute of Technology Press

CSD (1999) Surface management system research & development plan volume i technical plan sms-101. Tech. rep., National Aeronautics and Space Administration, AMES Research Center, Computational Sciences Division

Darwin C (1859) On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life. Mr. Murray

Darwin C, Wallace AR (1858) On the tendency of species to form varieties; and on the perpetuation of varieties and species by natural means of selection". Journal of the Proceedings of the Linnean Society of London 3:46–50

Deb K, Goldberg DE (1989) An investigation of niche and species formation in genetic function optimization. In: Proceedings of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 42–50

DeJong K (1975) An analysis of the behavior of a class of genetic adaptive systems. PhD thesis, Dept. of Computer and Communication Sciences, University of Michigan

Diepen JMHJASJW G Akker (2012) Finding a robust assignment of flights to gates at amsterdam airport schiphol. Journal of Scheduling 15(15):703

Ding H, Lim A, Rodrigues B, Zhu Y (2003) The airport gate assignment problem. Computers & Operations Research 32:1867–1880, DOI 10.1.1.5.5375

Ding H, Lim A, Rodrigues B, Zhu Y (2004) Aircraft and gate scheduling optimization at airports. In: System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on, pp 1530 – 1605, DOI 10.1109/HICSS.2004.1265219

Ding H, Rodrigues AL, Zhu Y (2005) The over-constrained airport gate assignment problem. Computers and Operations Research 32(7):1867 – 1880

Dorndorf U (2002) Project Scheduling with Time Windows: From Theory to Applications. Contributions to Management Science, Physica-Verlag

Dorndorf U, Pesch E, Phan-Huy T (2000) Constraint propagation techniques for the disjunctive scheduling problem. Artificial Intelligence 122(12):189 – 240, DOI 10.1016/S0004-3702(00)00040-0

Dorndorf U, Drexl A, Nikulin Y, Pesch E (2007a) Flight gate scheduling: State-of-the-art and recent developments. Omega - The International Journal of Management Science 35(3):326 – 334

Dorndorf U, Jaehn F, Lin C, Ma H, Pesch E (2007b) Disruption management in flight gate scheduling. Statistica Neerlandica 61(1):92 – 114

Dorndorf U, Jaeh F, Pesch E (2008) Modelling robust flight-gate scheduling as a clique partitioning problem. Transportation Science 42(3):292 – 301, DOI 10.1287/trsc.1070.0211

Dorndorf U, Jaehn F, Pesch E (2010) Flight gate scheduling with respect to a reference schedule. Annals of Operations Research pp 1–11, DOI 10.1007/s10479-010-0809-8

Drexla A, Nikulina Y (2008) Multicriteria airport gate assignment and pareto simulated annealing. IIE Transactions 40(4):385 – 397, DOI 10.1080/07408170701416673

Du Y, Zhang Q, Chen Q (2008) Aco–ih: An improved ant colony optimization algorithm for airport ground service scheduling. IEEE International Conference on Industrial Technology, vol 8, pp 1 – 6

Eberhart R, Shi Y, Kennedy J (2001) Swarm intelligence. Morgan Kaufmann

Eiben AE (2003) Advances in Evolutionary Computing, Springer-Verlag, Berlin, Heidelberg, chap Multiparent Recombination in Evolutionary Computing, pp 175 – 192. Computer Science, DOI 10.1007/978-3-642-18965-4_6

Eiben AE, Raué PE, Ruttkay Z (1994) Genetic algorithms with multi-parent recombination. In: Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature, Springer-Verlag, London, UK, PPSN III, pp 78–87

Eiben AE, Kemenade CHM, Kok JN (1995) Orgy in the computer: Multi-parent reproduction in genetic algorithms. In: In Third European Conference on Artificial Life, Springer-Verlag, pp 934–945

Fan HSL (1990) Fuel conservation by controlling aircraft ground operations. Transportation Planning and Technology 15(1):1 – 11, DOI 10.1080/03081069008717436

Federal Aviation Administration (2010) Faa aerospace forecast fiscal years 20102030 - forecast highlights. Tech. rep., Federal Aviation Administration

Federal Aviation Administration (2012) Faa aerospace forecast fiscal years 2012. Tech. rep., U.S. Department of Transportation Federal Aviation Administration Aviation Policy and Plans

Filar JA, Manyem P, White K (2001) How airlines and airports recover from schedule perturbations: A survey. Annals of Operations Research 108(1-4):315 – 333, DOI 10.1023/A:1016079600083

Frey M, Artigues C, Kolisch R, Lopez P (2010) Scheduling and planning the outbound baggage process at international airports. In: Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on, pp 2129 – 2133, DOI 10.1109/IEEM.2010.5675613

Gelinas C, Fan H (1979) Reducing air pollutant emissions at airports by controlling aircraft ground operations. Journal of the Air Pollution Control Association 29(2):125 – 128, DOI 10.1080/00022470.1979.10470766

Ghosh A, Tsutsui S, Tanaka H (1998) Function optimization in nonstationary environment using steady state genetic algorithms with aging of individuals. In: Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, pp 666–671, DOI 10.1109/ICEC.1998.700119

Glover F (1990) Tabusetabu - part ii. ORSA Journal on Computing 2(1):4–32

Glover FW (1989) Tabu search - part i. ORSA Journal on Computing 1(3):190–210

Goldberg D, Deb K (1991) A Comparative Analysis of Selection Schemes Used in Genetic Algorithms. Foundations of Genetic Algorithms, Morgan-Kaufmann

Goldberg DE (1989) Genetic Algorithms in Search, Optimization and Machine Learning, 1st edn. Addison-Wesley Longman Publishing Co., Inc., Boston, MA

Goldberg DE (1990) A Note on Boltzmann Tournament Selection for Genetic Algorithms and Population-Oriented Simulated Annealing, Complex Systems Publications, Inc., pp 445 – 460

Gosling GD (1990) Design of an expert system for aircraft gate assignment. Transportation Research Part A: General 24(1):59 – 69, DOI 10.1016/0191-2607(90) 90071-D

Gotteland JB, Durand N (2003) Genetic algorithms applied to airport ground traffic optimization. CEC 063:1121

Gu Y, Chung CA (1999) Genetic algorithm approach to aircraft gate reassignment problem. Journal of Transportation Engineering 125(5):384–389, DOI 10.1061/(ASCE)0733-947X(1999)125:5(384)

Haghani A, Chen MC (1998) Optimizing gate assignments at airport terminals. Transportation Research Part A: Policy and Practice 32(6):437 – 454

Haimes YY, Lasdon LS, Wismer DA (1971) On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. IEEE Transactions on Systems, Man, and Cybernetics 1(3):296–297, DOI 10.1109/tsmc.1971.4308298

Hanne T (2009) On utilizing infeasibility in multiobjective evolutionary algorithms. In: Barichard V, Ehrgott M, Gandibleux X, T'Kindt V (eds) Multiobjective Programming and Goal Programming, Lecture Notes in Economics and Mathematical Systems, vol 618, Springer Berlin Heidelberg, pp 113–122, DOI 10.1007/978-3-540-85646-7_11

Hansen JV (2004) Genetic search methods in air traffic control. Computers and operations research 31(3):445 – 459, DOI 10.1016/S0305-0548(02)00228-9

Haralick RM, Elliott GL (1980) Increasing tree search efficiency for constraint satisfaction problems. Artificial Intelligence 14:263 – 313, DOI 10.1016/0004-3702(80)90051-X

Hassounah MI, Steuart GN (1993) Demand for aircraft gates. Transportation Research Record 1423(1423):26 – 33

Heathrow (2010) Arrivals. Internet

Hinton GE, Nowlan SJ (1987) How learning can guide evolution. Complex Systems 1:495 – 502

Holland JH (1975) Adaptation In Natural and Artificial Systems. University of Michigan Press

Hu X, Di Paolo E (2007) An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem. In: Evolutionary Computation, 2007. CEC 2007. IEEE Congress on, pp 55 –62, DOI 10.1109/CEC.2007.4424454

Hu XB, Chen WH (2005) Receding horizon control for aircraft arrival sequencing and scheduling. Intelligent Transportation Systems, IEEE Transactions on 6(2):189 – 197, DOI 10.1109/TITS.2005.848365

Hu XB, Di Paolo E (2009) A ripple-spreading genetic algorithm for the airport gate assignment problem. In: Evolutionary Computation, 2009. CEC '09. IEEE Congress on, pp 1857 –1864, DOI 10.1109/CEC.2009.4983167

ICAO (2010) Icao news release - pio 06/10. Tech. rep., International Civil Aviation Organization

Ichoua S, Gendreau M, Potvin JY (2006) Exploiting knowledge about future demands for real-time vehicle dispatching. Transportation Science 40:211 – 225, DOI 10.1287/trsc.1050.0114

Jaehn F (2010) Solving the flight gate assignment problem using dynamic programming. Z Betriebswirtsch 80:1027–1039, DOI 10.1007/s11573-010-0396-9

Janic M (2007) A heuristic algorithm for the allocation of airport runway system capacity. Transportation Planning & Technology 30(5):501 – 520, DOI 10.1080/03081060701599995

Johnstone M, Creighton D, Nahavandi S (2010) Status-based routing in baggage handling systems: Searching verses learning. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 40(2):189 – 200, DOI 10.1109/TSMCC.2009.2035519

Kalyanmoy Deb SA Amrit Pratap, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION 6(2)

Keinrath C, Pouw F, Wise JA, Luuk A, Sedaoui A, Gauthereau V (2008) ifly: Human factors in autonomous aircraft operations (airborne self-separation environment). Tech. rep., University of Tartu

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proc. IEEE International Conference on Neural Networks, IEEE, Perth, Australia, vol 4, pp 1942 – 1948, DOI 10.1109/ICNN.1995.488968

Kim SH, Feron E (2012) Impact of gate assignment on gate-holding departure control strategies. In: Digital Avionics Systems Conference (DASC), 2012 IEEE/AIAA 31st, pp 4E3–1 –4E3–8, DOI 10.1109/DASC.2012.6382350

Knowles J, Corne D (1999a) Local search, multiobjective optimization and the pareto archived evolution strategy. In: Ashikaga Institute of Technology, pp 209–216

Knowles J, Corne D (1999b) The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, vol 1, pp –105 Vol. 1, DOI 10.1109/CEC.1999.781913

Knowles J, Corne D (2003) Properties of an adaptive archiving algorithm for storing nondominated vectors. Evolutionary Computation, IEEE Transactions on 7(2):100–116, DOI 10.1109/TEVC.2003.810755

Knowles JD, Corne DW (2000) Approximating the nondominated front using the pareto archived evolution strategy. Evol Comput 8(2):149–172, DOI 10.1162/106365600568167

Knysh D, Kureichik V (2010) Parallel genetic algorithms: a survey and problem state of the art. Journal of Computer and Systems Sciences International 49(4):579–589, DOI 10.1134/S1064230710040088

Kolischa R, Hartmann S (2006) Experimental investigation of heuristics for resource-constrained project scheduling: An update. European Journal of Operational Research 175:23 – 37

Kubota N, Fukuda T (1997) Genetic algorithms with age structure. Soft Computing 1(4):155–161, DOI 10.1007/s005000050017

Lawler EL (1963) The qadratic assignment problem. JSTOR 9(4):586 – 599, DOI 10.1287/mnsc.9.4.586

Levinthal DA, March JG (1993) The myopia of learning. Strategic Management Journal 14(S2):95–112, DOI 10.1002/smj.4250141009

Li C (2009) Airport gate assignment – a hybrid model and implementation. CoRR abs/0903.2528:5

Lim A, Wang F (2005) Robust airport gate assignment. In: Tools with Artificial Intelligence, 2005. ICTAI 05. 17th IEEE International Conference on, pp 8 pp. –81, DOI 10.1109/ICTAI.2005.110

Lim A, Rodrigues B, Zhu Y (2005) Airport gate scheduling with time windows. Artificial Intelligence Review 24(1):5 – 31, DOI 10.1007/s10462-004-7190-4

Lockwood M, Harrison RG, Woollings T, Solanki SK (2010) Are cold winters in europe associated with low solar activity? ENVIRONMENTAL RESEARCH LETTERS pp 1 – 7, DOI doi:10.1088/1748-9326/5/2/024001

Mangoubi RS, Mathaisel DFX (1985) Optimizing gate assignments at airport terminals. TRANSPORTATION SCIENCE 19:173 – 188, DOI 10.1287/trsc.19.2.173

March JG (1991) Exploration and exploitation in organizational learning. Organization science 2(1):71–87

Marn n (2006) Airport management: taxi planning. Annals of Operations Research 143(1):191–202, DOI 10.1007/s10479-006-7381-2

Martin WN, Lienig J, Cohoon JP (1997) Island (migration) models: evolutionary algorithms based on punctuated equilibria 6:3+

Mendel G (1865) Experiments in Plant Hybridisation

Mengshoel OJ, Goldberg DE (2008) The crowding approach to niching in genetic algorithms. Evol Comput 16(3):315–354, DOI 10.1162/evco.2008.16.3.315

Merlin P (1983) Energy savings in transport. Transportation Planning and Technology 8:39 – 52, DOI 10.1080/03081068308717236

Michalewicz Z (1996) Genetic Algorithms + Data Structures = Evolution Programs, 3rd edn

Michalewicz Z, Fogel DB (2002) How to Solve It: Modern Heuristics, 2nd edn. Springer

Miller B, Clarke JP (2004) Emissions reductions enabled by improved airport surface operations. Air Traffic Control Quarterly 12(2):147 – 164

Mueller ER, Chatterji GB (2002) Analysis of aircraft arrival and departure delay characteristics. In: AIAA's Aircraft Technology, Integration and Operations (ATIO), American Institute of Aeronautics and Astronautics, American Institute of Aeronautics and Astronautics, Los Angeles, California, vol 5866, pp 1 – 14

Mulvey JM, Vanderbei RJ, Zenios SA (1995) Robust optimization of large-scale systems. Operational Research 43:254 – 281, DOI 10.1287/opre.43.2.264

Nikulin Y (2006) Robustness in combinatorial optimization and scheduling theory: An extended annotated bibliography

Noël M, Florent JP, Lagios E, Howarth G, Huegel C, Eleftheriou V, Egido A, Lane R, Wennerberg A, Hogg D, Richard M (2009) Airport cdm applications guide. Tech. rep., EUROCONTROL

Norin A, Andersson T, Värbrand P, Yuan D (2007) A grasp heuristic for scheduling de-icing trucks at stockholm arlanda airport. INO workshop

Norin A, Granberg TA, Yuan PVD (2009) Integrating optimization and simulation to gain more efficient airport logistics. In: Eighth USA/Europe Air Traffic Management Research and Development Seminar (ATM2009), pp 1 – 10

Obata T (1979) The quadratic assignment problem: Evaluation of exact and heuristic algorithms. Troy, New York

Pardalos PM, Rendl F, Wolkowicz H (1994) The quadratic assignment problem: A survey and recent developments. In: In Proceedings of the DIMACS Workshop on Quadratic Assignment Problems, American Mathematical Society, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol 16, pp 1–42

Pareto V (1909) MANUEL S'ECONOMIE POLITIQUE, 1st edn. BIBLIOTEQUE INTERNATIONAL D'ECONOMIE POLITIQUE

Pesch E, Dorndorf U, Jaehn F (2008) Flight gate allocation: Modells, methods and robust solutions. In: Practice and Theory of Automated Timetabling (PATAT)

Pintea CM, Pop PC, Chira C, Dumitrescu D (2008) A hybrid ant-based system for gate assignment problem. In: E Corchado AA, Pedrycz W (eds) Hybrid Artificial Intelligence Systems, Lecture Notes in Computer Science, vol 5271, Springer-Verlag Berlin Heidelberg, pp 273 – 280, DOI 10.1007/978-3-540-87656-4_34

Pitt M, Wai FK, Teck PC (2002) Technology selection in airport passenger and baggage systems. Facilities 20(10):314 – 326, DOI 10.1108/02632770210442992

Prem Kumar V, Bierlaire M (2013) Multi-objective airport gate assignment problem in planning and operations. Journal of Advanced Transportation pp n/a–n/a, DOI 10.1002/atr.1235

Prugel-Bennett A, Shapiro JL (1994) Analysis of genetic algorithms using statistical mechanics. Physical Review Letters 72(9):1305 – 1309, DOI 10.1103/PhysRevLett.72.1305

Razali NM, Wah YB (2011) Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests. Journal of Statistical Modeling and Analytics 1(2):21–33

Rechenberg I (1971) Evolutionsstrategie - optimierung technischer systeme nach prinzipien der biologischen evolution. stuttgart: Frommann-holzboog. PhD thesis, Technical University of Berlin

Research and Innovative Technology Administration (2012) Airline on-time statistics and delay causes. Online

Rijsenbrij JC, Ottjes JA (2007) New developments in airport baggage handling systems. Transportation Planning and Technology 30(4):417 – 430, DOI 10.1080/03081060701461899

Robinson GL (1969) Simulation models for evaluation of airport baggage-handling systems. In: Proceedings of the third conference on Applications of simulation, Winter Simulation Conference, pp 226 – 235

Roussos G, Kyriakopoulos KJ (2009) Towards constant velocity navigation and collision avoidance for autonomous nonholonomic aircraft-like vehicles. In: Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, IEEE Conference on Decision and Control, pp 5661–5666, DOI 10.1109/CDC.2009.5400814

Ruszczynski A, Shapiro A (2003) Stochastic programming, Handbooks in Operations Research and Management Science, vol 10. Elsevier

Schaffer JD (1984) Multiple objective optimization with vector evaluated genetic algorithms. PhD thesis, Vanderbit University, Nashville, TN (USA)

Schaffer JD (1985) Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the 1st International Conference on Genetic Algorithms, L. Erlbaum Associates Inc., Hillsdale, NJ, USA, pp 93–100

Schwefel HP, Rudolph G (1995) Contemporary evolution strategies. In: Advance in Artificial Life, Springer Berlin Heidelberg, Lecture Notes in Computer Science, vol 929, pp 891–907, DOI 10.1007/3-540-59496-5_351

Seidenkrantz MS, Kuijpers A, Schmith T (2009) Comparing past and present climate a tool to distinguish between natural and human-induced climate change. IOP Conference Series: Earth and Environmental Science 8(1):1 – 6, DOI 10.1088/1755-1315/8/1/012012, iOP Publishing

Seker M, Noyan N (2012) Stochastic optimization models for the airport gate assignment problem. Transportation Research Part E: Logistics and Transportation Review 48(2):438 – 459, DOI 10.1016/j.tre.2011.10.008

Sheng W, Liu Y, Meng X, Zhang T (2012) An improved strength pareto evolutionary algorithm 2 with application to the optimization of distributed generations. Computers & Mathematics with Applications 64(5):944 – 955, DOI 10.1016/j.camwa.2012.01.063, ¡ce:title¿Advanced Technologies in Computer, Consumer and Control¡/ce:title¿

Sokolov A, Whitley D (2005) Unbiased tournament selection. In: Proceedings of the 2005 conference on Genetic and evolutionary computation, ACM, New York, NY, USA, GECCO '05, pp 1131–1138, DOI 10.1145/1068009.1068198

Srinivas N, Deb K (1994) Multiobjective optimization using nondominated sorting in genetic algorithms. Evolutionary Computation 2:221–248

Swenson H, Barhydt R, Landis M (2006) Next generation air transportation system (ngats) air traffic management (atm)-airspace project. Tech. rep., National Aeronautics and Space Administration (NASA)

Syswerda G (1989) Uniform crossover in genetic algorithms. In: Proceedings of the 3rd International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp 2–9

Syswerda G (1990) A study of reproduction in generational and steady state genetic algorithms. In: FOGA'90, pp 94–101

Tang CH, Yan S, Hou YZ (2009) A gate reassignment framework for real time flight delays. 4OR: A Quarterly Journal of Operations Research DOI 10.1007/s10288-009-0112-1

Tarascio VJ (1968) Pareto's methodological approach to economics: A study in the history of some scientific aspects of economic thought

Tittsworth JA, Lang SR, Johnson EJ, Barnes S (2012) Federal aviation administration wake turbulence program - recent highlights. In: Air Traffic Control Association (ATCA) Annual Conference & Exposition

Tsutsui S, Jain LC (1998) On the effect of multi-parents recombination in genetic algorithms. In: Second Intemational Conference on Knowledge-Based Intelligent Electronic Systems, IEEE

UNIQUE (2005) Aircraft engine emission reduction programme zurich airport. Tech. rep., UNIQUE, Unique (Flughafen Zrich AG), P.O. Box, CH-8058 Zurich

US Department of Transportation (2010) Air travel consumer report. Tech. rep., U.S. Department of Transportation

Vose MD, Liepins GE (1991) Punctuated equilibria in genetic search. Complex Systems 5(1):31 – 44

Voß S, Fink A, Duin C (2005) Looking ahead with the pilot method. Annals of Operations Research 136(1):285 – 302, DOI 10.1007/s10479-005-2060-2

Wei D, Liu C (2009) Fuzzy model and optimization for airport gate assignment problem. In: IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS, IEEE, Shanghai, vol 2, pp 828 – 832, DOI 10.1109/ICICISYS.2009.5358268

Wei D, Liu C (2013) Multi-objective model and optimization for airport gate assignment problem. In: Lu W, Cai G, Liu W, Xing W (eds) Proceedings of the 2012 International Conference on Information Technology and Software Engineering, Lecture Notes in Electrical Engineering, vol 211, Springer Berlin Heidelberg, pp 597–606, DOI 10.1007/978-3-642-34522-7_64

Whitley D (1989) The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In: Proceedings of the Third International Conference on Genetic Algorithms, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, vol 1, pp 116–121

Whitley D, Kauth J (1988) Genitor: A different genetic algorithm. In: Proceedings of the Rocky Mountain Conference on Artificial Intelligence, Rocky Mountain Conference on Artificial Intelligence, Colorado State University, Fort Collins, CO, vol 1, pp 118–130

Whitley LD (1991) Fundamental principles of deception in genetic search. In: Foundations of Genetic Algorithms, Morgan Kaufmann Publishers Inc., pp 221–241

Wilson GC, Intyre AM, Heywood MI (2004) Resource Review: Three Open Source Systems for Evolving ProgramsLilgp, ECJ and Grammatical Evolution, vol 3, Kluwer Academic Publishers, chap Genetic Programming and Evolvable Machines, pp 103–105. DOI 10.1023/B:GENP.0000017053.10351.dc

Wolsey LA (1998) Integer Programming. Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons, New York

Wu CL, Caves RE (2000) Aircraft operational costs and turnaround efficiency at airports. Journal of Air Transport Management 6(4):201 – 208, DOI 10.1016/S0969-6997(00)00014-4

Wu CL, Caves RE (2004) Modelling and optimization of aircraft turnaround time at an airport. Transportation Planning and Technology 27(1):47 – 66, DOI 10.1080/0308106042000184454

Xiangwei M, Ping Z, Chunjin L (2010) Aircraft category based genetic algorithm for aircraft arrival sequencing and scheduling. In: Control Conference (CCC), 2010 29th Chinese, pp 5188 –5193

Xiujuan L, Ali F, Zhongke S (2008) The aircraft departure scheduling based on particle swarm optimization combined with simulated annealing algorithm. In: IEEE Congress on Evolutionary Computation, Hong Kong, pp 1399 – 1403, DOI 10.1109/CEC.2008.4630976

Xu J, Bailey G (2001) The airport gate assignment problem: mathematical model and a tabu search algorithm. In: System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on, p 10 pp., DOI 10.1109/HICSS.2001.926327

Yan S, Chang CM (1997) A network model for gate assignment. Journal of Advanced Transportation 32(2):176 – 189, DOI 10.1002/atr.5670320204

Yan S, Huo CM (2001) Optimization of multiple objective gate assignments. Transportation Research Part A: Policy and Practice 35(5):413 – 432, DOI 10.1016/S0965-8564(99)00065-8

Yan S, Tang CH (2007) A heuristic approach for airport gate assignments for stochastic flight delays. European Journal of Operational Research 180(2):547 – 567, DOI 10.1016/j.ejor.2006.05.002

Yan S, Shieh CY, Chen M (2002) A simulation framework for evaluating airport gate assignments. Transportation Research Part A: Policy and Practice 36(10):885 – 898, DOI 10.1016/S0965-8564(01)00045-3

Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhangd Q (2011) Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 1:32 – 49

Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. Evolutionary Computation, IEEE Transactions on 3(4):257–271, DOI 10.1109/4235.797969

Zitzler E, Laumanns M, Thiele L (2001) Spea2: Improving the strength pareto evolutionary algorithm. Tech. rep.

Appendix A

Data for Heathrow

A.1 Stands

The stand codes at London Heathrow airport are shown in Table 7.5. The stand codes for Terminal 4 at London Heathrow airport are shown in Table A.1.

Stand	Code	Pier	Full gate ID
01	D(767-300)	1	4101
02	E2	1	4102
03	E2	1	4103
05	F	1	4105
06	F	1	4106
07	E3	2	4207
08	E2	2	4208
09	E2	2	4209
10	E3	2	4210
11	E3	2	4211
12	E2	2	4212
14	E1	2	4214
15	C(A321)	2	4215
16	D(767-300)	2	4216
17	C(A319)	2	4217
19	C(A321)	2	4219
20	C(A321)	2	4220
21	D(767-300)	2	4221
22	E2	3	4322
23	E2	3	4323
24	E2	3	4324
25	E2	3	4325
29	E2	3	4029

Table A.1: Stand codes for London Heathrow airport (LHR) Terminal 4.

From data provided by London Heathrow airport and available maps the stands that can accommodate several aircraft are as follow:

1. **Terminal 1** In Terminal 1 (T1) piers 4 and 4a seems to be small stands based

Terminal 1			
Gate	Left side	Right side	Whole
121	L	R	W
233	L	R	W
247	L	R	W
258	L	R	

Table A.2: T1 multiple usage stands.

   on map from March 2011. Where pier 3 seems to have stand 121 which it is big enough to accommodate a large aircraft. Finally Europier seems to be composed just of large stands.

   The position of the stands in Table A.2 are not side by side so ‘shadow’ restrictions do not apply.

2. **Terminal 4**: It seems that all Terminal 4 (T4) stands are of the same size, large. From data provided by London Heathrow airport they are just used by one flight at a time.

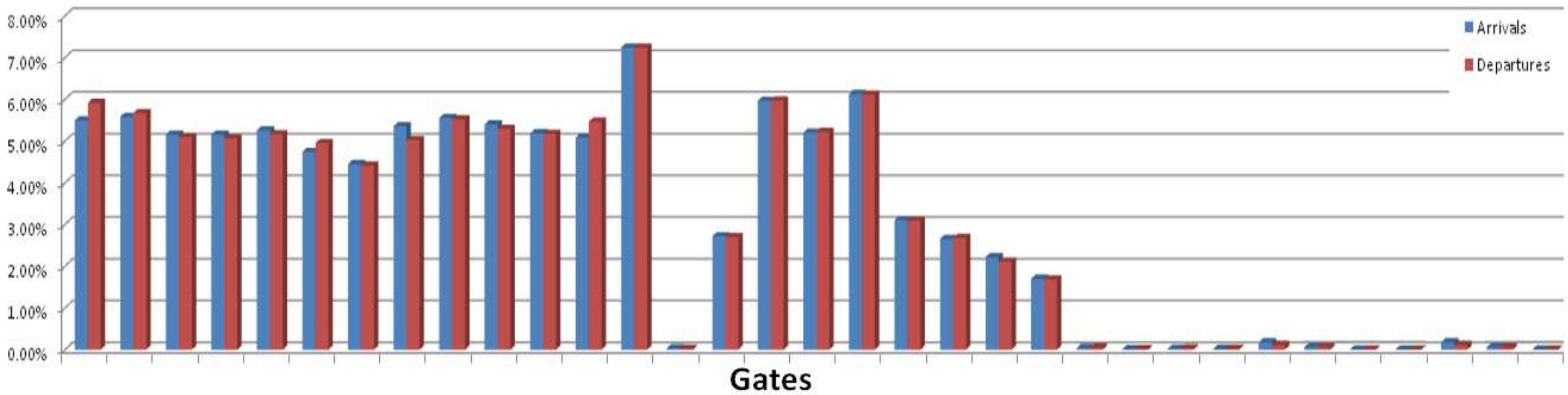
Some other stands that are used are BMA (by T1), BB (by T1, T3 and T5) and NO1 (T1, T3 and T5). It is noted that Terminal 2 was not in operation at the time of this study.

The percentage of overall flights assigned to each stand, from 6^{th} September 2010 to 12^{th} September 2010 for London Heathrow airport Terminal 4, is show in Figure A.1.

A.2 Airlines Gate Preferences

2.1 Terminal 4

The Figure A.2 shows the overall number of flights assigned to each gate, to each airline for the period from 6^{th} September 2010 to 12^{th} September 2010 for London Heathrow airport Terminal 4.



	401	402	403	405	406	407	408	409	410	411	412	414	415	416	417	419
Arrivals	5.49%	5.58%	5.15%	5.15%	5.26%	4.74%	4.45%	5.36%	5.55%	5.40%	5.19%	5.08%	7.24%	0.03%	2.72%	5.97%
Departures	5.91%	5.67%	5.10%	5.06%	5.17%	4.95%	4.42%	5.02%	5.52%	5.29%	5.17%	5.46%	7.24%	0.03%	2.70%	5.97%

	420	421	422	423	424	425	429	430	431	432	451	452	453	455	461	463	601
Arrivals	5.20%	6.13%	3.11%	2.65%	2.22%	1.70%	0.04%	0.01%	0.02%	0.02%	0.18%	0.06%	0.00%	0.00%	0.25%	0.05%	0.00%
Departures	5.22%	6.11%	3.10%	2.68%	2.11%	1.68%	0.05%	0.02%	0.03%	0.02%	0.11%	0.06%	0.00%	0.00%	0.11%	0.00%	0.00%

Figure A.1: Percentage of assignments per stand at London Heathrow airport Terminal 4.

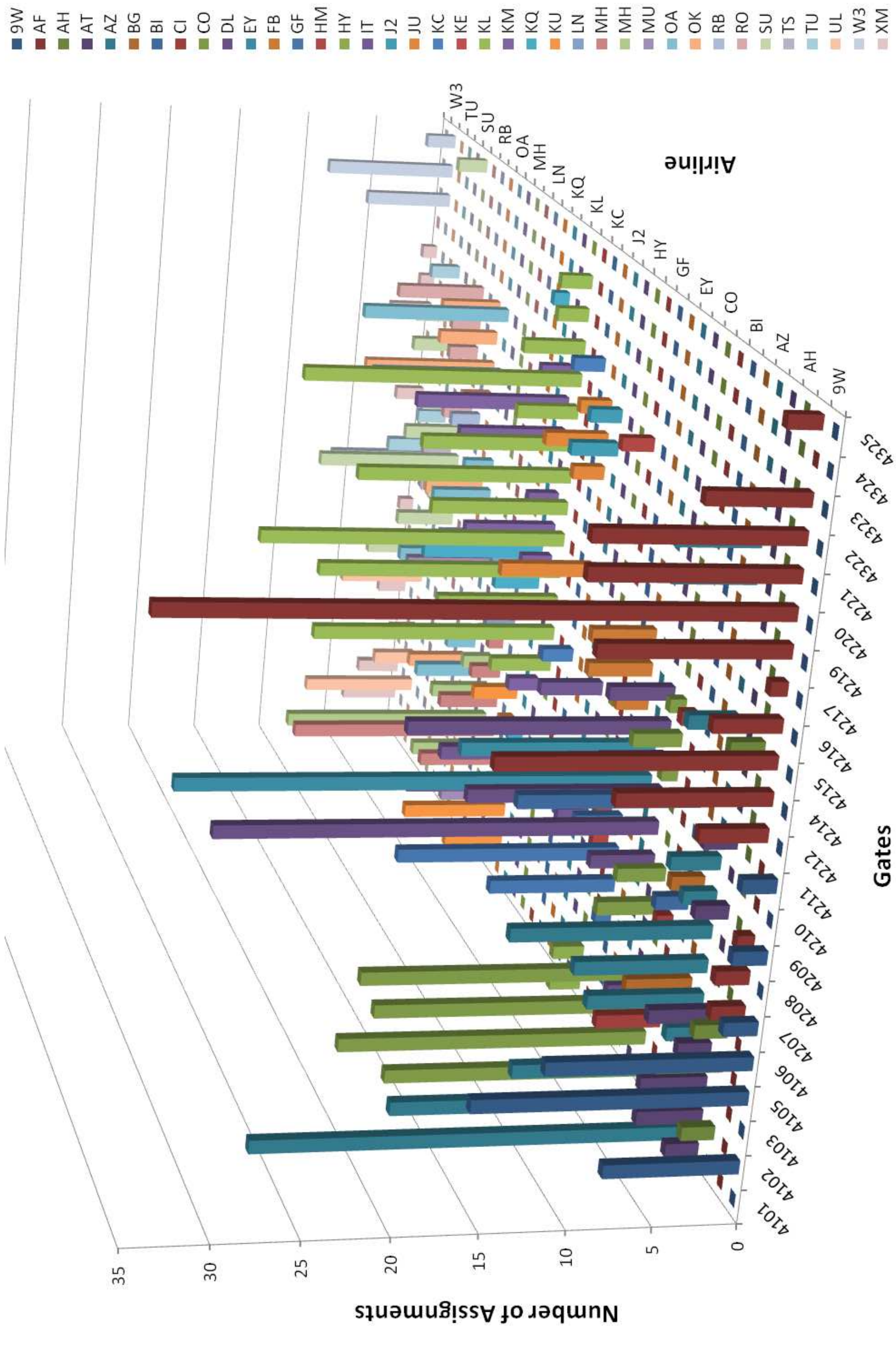
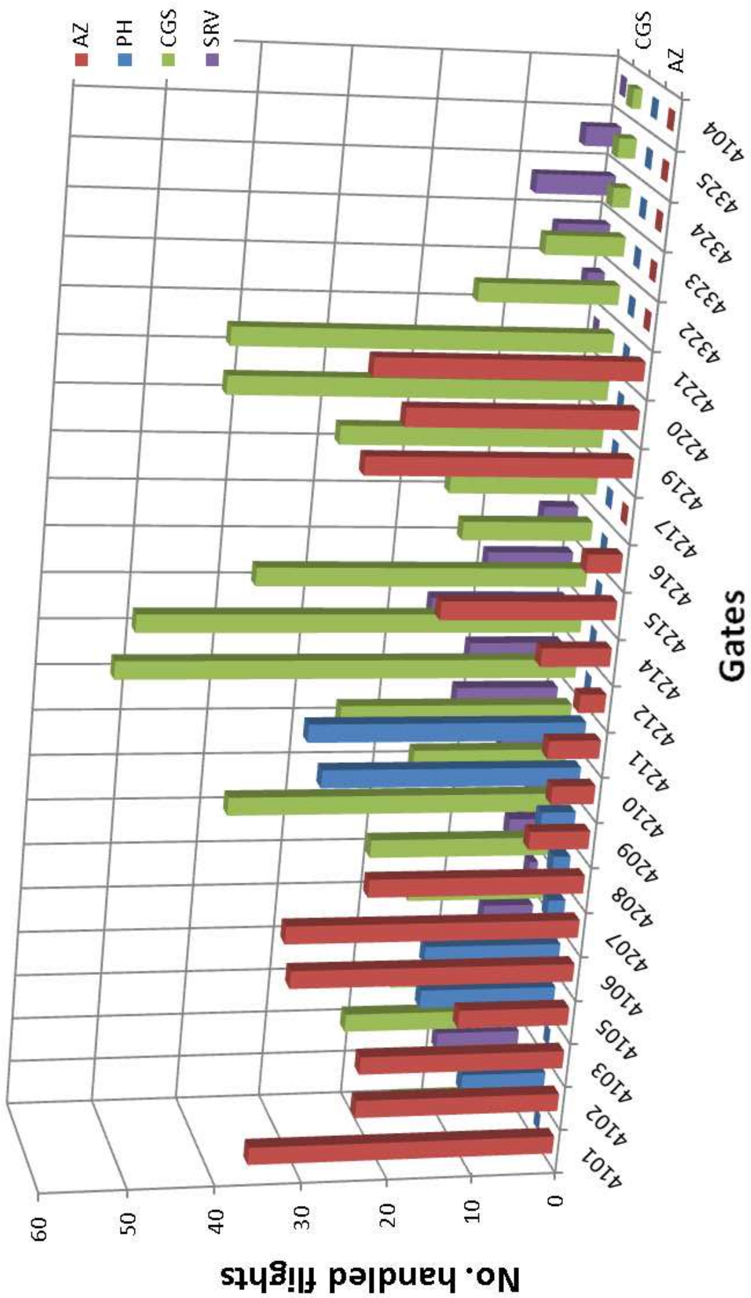


Figure A.2: Airlines preference at London Heathrow airport Terminal 4.

A.3 Handlers Gate Preferences

3.1 Terminal 4

The Figure A.3 shows the overall number of flights assigned to each gate, to each handler for the period from 6^{th} September 2009 to 12^{th} September 2009 for London Heathrow airport Terminal 4.



	4101	4102	4103	4105	4106	4207	4208	4209	4210	4211	4212	4214	4215	4216	4217	4219	4220	4221	4322	4323	4324	4325	4104
AZ	36	24	24	13	33	34	25	7	5	6	3	8	20	4	0	30	26	30	0	0	0	0	0
PH	0	10	0	16	16	2	2	4	30	32	0	0	0	0	0	0	0	0	0	0	0	0	0
CGS	16	20	23	18	2	17	22	39	18	27	53	51	38	15	17	30	43	43	16	9	2	2	1
SRV	2	10	11	6	6	1	4	2	6	12	11	16	10	4	0	4	4	0	2	6	9	4	0

Figure A.3: Handlers preference at London Heathrow airport Terminal 4.

Appendix B

Statistics for the Airport Baggage Sorting Stations Assignment Problem

If the data follows a normal distribution the most appropriate statistical significance corresponds to the t-test, otherwise the Mann-Whitney U test is used. Razali and Wah (2011) compared some normality tests and concluded that Shapiro-Wilk is the most powerful normality test. So Shapiro-Wilk normality test is used when it is required to determine whether the data can be said to follow a normal distribution, such that the appropriate statistical significance test is used.

The following contractions are used.

1. NE corresponds to **not equal**
2. LT corresponds to **less than**
3. GT corresponds to **greater than**

The value between brackets is the significance probability, e.g. "**NE (1)**" indicates that both cannot be said to be different. The significance probability is specified from 1 (100%) to 0 (0%) with the precision set to four decimals.

B.1 Constructive Algorithms

This section presents the statistical significance test conducted for the data sets obtained from the British Airports Authority (BAA)'s website composed of 142 flights for 16^{th} December 2009 and 270 flights for 1^{st} March 2010 for a 3-pier topology and 48 gates, given that they do not contain information regarding the assignment of gates to flights. This was not required for the data sets provided by NATS as they contain the assignment of each flight to a gate.

Two more Baggage Sorting Station Assignment Algorithms (BSSAAs) are considered: '*B*' and '*D*' which are presented below.

Algorithm '*B*': Baggage Sorting Station Assignment Algorithm '*B*'

```
begin
    Order all flights based on the current flight choice algorithm (Section 4.2.1);
    forall the flights do
        if a feasible baggage sorting station exists on the flight's own pier then
            Select a baggage sorting station using the selection algorithm;
        else if a feasible baggage sorting station exists in the airport then
            Select a baggage sorting station using the selection algorithm;
        end
    end
    forall the unassigned flights do
        Reduce the flight service time by the maximum reduction allowed;
        if a feasible baggage sorting station exists on the flight's own pier then
            Select a baggage sorting station using the selection algorithm;
        else if a feasible baggage sorting station exists in the airport then
            Select a baggage sorting station using the selection algorithm;
        else
            Assign the flight to the dummy baggage sorting station;
        end
    end
end
```

Algorithm '*D*': Baggage Sorting Station Assignment Algorithm '*D*'

```
begin
    Order all flights based on the current flight choice algorithm (Section 4.2.1);
    forall the flights do
        if a feasible baggage sorting station exists on the flight's own pier then
            Select a baggage sorting station using the selection algorithm;
        else if a feasible baggage sorting station exists in the airport then
            Select a baggage sorting station using the selection algorithm;
        else
            Reduce the flight service time by the maximum reduction allowed;
            if a feasible baggage sorting station exists on the flight's own pier then
                Select a baggage sorting station using the selection algorithm;
            else
                Reduce the flight service time by the maximum reduction allowed;
                if a feasible baggage sorting station exists in the airport then
                    Select a baggage sorting station using the selection algorithm;
                else
                    Assign the flight to the dummy baggage sorting station;
                end
            end
        end
    end
end
```

The BSSAA considered are: '*A*', '*B*', '*C*', '*D*' and '*E*'. The Baggage Sorting Station Selection Methods (BSSSM), which are presented in Section 4.1, and considered here are: ODT, ODTLI and Order by Starting Time (OST).

Algorithm			42				
			Max	Min	Mean	Std	p
ODT	A	Closest	115.00	108.00	112.93	1.27	0.0006
		FIFO	115.00	108.00	112.04	1.24	0.0022
		LIFO	115.00	115.00	115.00	0.00	NA
	E	Closest	115.00	115.00	115.00	0.00	NA
		FIFO	115.00	115.00	115.00	0.00	NA
		LIFO	115.00	109.00	113.07	1.17	0.0001
Algorithm			48				
			Max	Min	Mean	Std	p
ODT	A	Closest	125.00	119.00	122.66	1.22	0.0164
		FIFO	123.00	117.00	120.32	1.13	0.0093
		LIFO	125.00	119.00	122.94	1.29	0.0343
	E	Closest	125.00	124.00	124.92	0.27	NA
		FIFO	121.00	121.00	121.00	0.00	NA
		LIFO	125.00	125.00	125.00	0.00	NA
Algorithm			54				
			Max	Min	Mean	Std	p
ODT	A	Closest	133.00	127.00	130.41	1.48	0.0090
		FIFO	130.00	124.00	126.99	1.16	0.0022
		LIFO	134.00	127.00	131.08	1.56	0.0063
	E	Closest	133.00	130.00	131.78	0.83	0.0102
		FIFO	127.00	127.00	127.00	0.00	NA
		LIFO	133.00	133.00	133.00	0.00	NA
Algorithm			60				
			Max	Min	Mean	Std	p
ODT	A	Closest	140.00	133.00	136.92	1.56	0.0094
		FIFO	136.00	128.00	132.51	1.31	0.0148
		LIFO	141.00	133.00	137.85	1.64	0.0009
	E	Closest	141.00	136.0000	139.21	0.96	0.0022
		FIFO	133.00	133.00	133.00	0.00	NA
		LIFO	142.00	142.00	142.00	0.00	NA
Algorithm			84				
			Max	Min	Mean	Std	p
ODT	A	Closest	142.00	142.00	142.00	0.00	NA
		FIFO	142.00	140.00	141.74	0.54	0.0119
		LIFO	142.00	142.00	142.00	0.00	NA
	E	Closest	142.00	142.00	142.00	0.00	NA
		FIFO	142.00	142.00	142.00	0.00	NA
		LIFO	142.00	142.00	142.00	0.00	NA

Table B.1: Results of the Shapiro-Wilk normality tests.

'Closest Min Open' is a version of the 'Closest' which uses LIFO to select between different BSSs with the same distance.

The flights for each data set (from the BAA's website) were assigned to gates, 100 times per data set, using a random constructive algorithm without any restriction ('*A*' Baggage Sorting Station Assignment Algorithm). The Shapiro-Wilk normality test was run using these new sets of data, the results of which are shown in Tables B.1, indicating that the data cannot be said to follow a normal distribution. So the Mann-Whitney U test was adopted with an alpha level of 5% to obtain the statistical significant of the number of assignments obtained by the constructive algorithms which are presented in the following sections. Many of the results for a higher number of BSSs have a zero standard deviation, which is a clear indication that they cannot be said to be normally distributed. The Shapiro-Wilk normality test cannot therefore be applied (represented by 'NA').

B.1.1 Without Reduction in Service Extra Results

Data Set for 142 flights

Constructive algorithms results for data set with 142 flights, Tables B.2, B.3 and B.4.

'*A*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	0.2146/0.7853	GT (0.0037)	GT (0.0000)	0.541/0.459	GT (0.0477)	GT (0.0226)
54	GT (0.0095)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
72	0.0765/0.9235	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0011)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*C*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0047)	GT (0.002)	GT (0.0000)	0.4988/0.4988	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.4992/0.4992	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0222)	GT (0.0000)
78	GT (0.0000)	GT (0.0414)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0011)

'*E*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0414)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

Table B.2: ODTLI and ODT for 142 flights.

'*A*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0006)	0.9463/0.0537	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0023)	LT (0.0000)	LT (0.0000)	GT (0.0000)	0.5339/0.4661	LT (0.0000)
60	GT (0.0000)	LT (0.0000)	LT (0.0000)	0.7253/0.2747	LT (0.0000)	LT (0.0000)
66	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0057)	LT (0.0000)	LT (0.0000)
72	LT (0.0268)	LT (0.0000)	LT (0.0000)	GT (0.0000)	LT (0.0000)	LT (0.0000)
78	GT (0.0000)	LT (0.0011)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0000)

'*C*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0066)	LT (0.0000)	LT (0.0000)
54	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0001)	LT (0.0000)	LT (0.0000)
72	GT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0222)	LT (0.0000)
78	GT (0.0000)	LT (0.0414)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0011)

'*E*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	
54	LT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	LT (0.0000)	LT (0.0000)
72	GT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0000)
78	GT (0.0000)	LT (0.0414)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0000)

Table B.3: ODT and OST for 142 flights.

'*A*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0008)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0255)	LT (0.0002)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.005)	LT (0.0000)	LT (0.0000)	0.7253/0.2747	LT (0.006)	LT (0.0000)
66	LT (0.0000)	LT (0.0000)	LT (0.0000)	0.9943/ 0.0057	LT (0.0000)	LT (0.0000)
72	0.7211/0.2789	LT (0.0000)	LT (0.0000)	GT (0.0000)	LT (0.0037)	LT (0.0000)
78	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0066)

'*C*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0066)	LT (0.0000)	LT (0.0000)
54	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0001)	LT (0.0000)	LT (0.0000)
72	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	0.8389/0.1611	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*E*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

Table B.4: ODTLI and OST for 142 flights.

Data Set for 270 flights

Constructive algorithms results for data set with 270 flights, Figure B.1 and Tables B.5, B.6 and B.7.

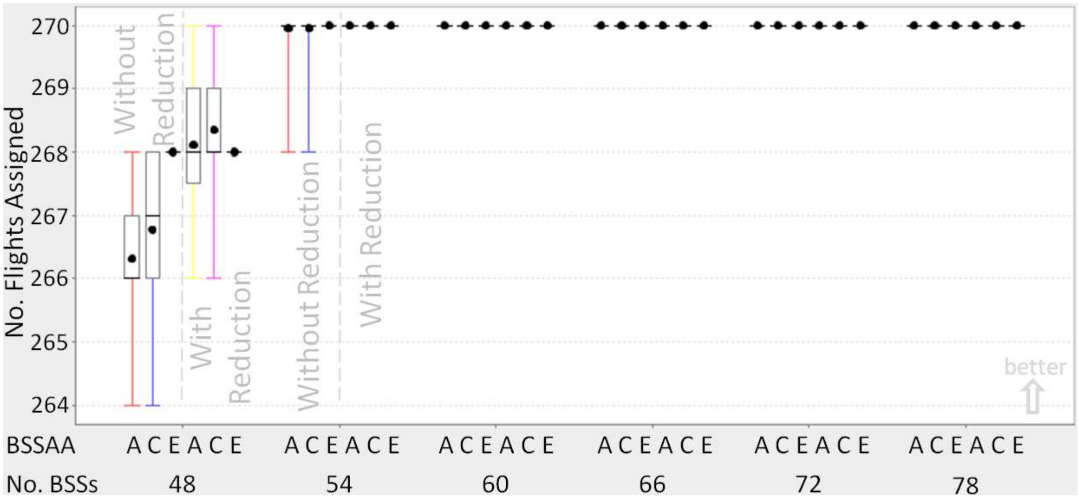


Figure B.1: Assignments to Terminal 1, BAA's website of 270 flights with LIFO and OST.

'A' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0316)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
54	GT (0.0056)	GT (0.0000)	GT (0.0000)	0.6299/0.3701	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.1611/0.8389
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0121)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'C' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0012)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0032)	GT (0.0000)
54	GT (0.0059)	GT (0.0000)	GT (0.0000)	0.4986/0.4986	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.0792/0.9208
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'E' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

Table B.5: ODTLI and ODT for 270 flights.

'A' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	LT (0.0000)	LT (0.0000)	LT (0.0000)	0.9123/0.0877	LT (0.0000)	LT (0.0000)
60	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0000)
66	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	0.8389/0.1611
72	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	LT (0.0120)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'C' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0222)	LT (0.0000)	LT (0.0000)
60	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	0.9208/0.0792
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'E' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	LT (0.0000)	LT (0.0000)
54	LT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0000)
60	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

Table B.6: ODT and OST for 270 flights.

'A' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	LT (0.0119)	LT (0.0013)	LT (0.0000)	LT (0.0499)	LT (0.0028)	LT (0.0000)
60	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	0.4972/0.497
66	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'C' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	LT (0.0011)	LT (0.0003)	LT (0.0000)	LT (0.0222)	LT (0.0020)	LT (0.0000)
60	GT (0.0000)	GT (0.0000)	0.8389/0.1611	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'E' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

Table B.7: ODTLI and OST for 270 flights.

B.1.2 With Reduction in Service Extra Results

Data Sets provided by NATS

Some extra results for the data sets provided by NATS when using the constructive algorithms with reduction in service and 3-pier topology (Section 4.4.1), Figures B.2, B.3 and B.4.

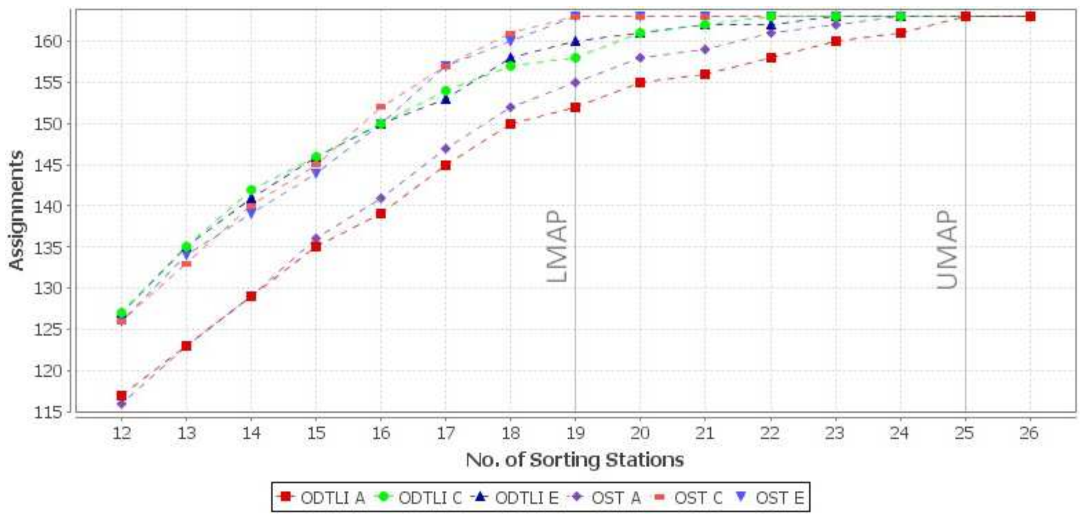


Figure B.2: Assignments to Terminal 1, with reduction of 163 flights and Closest.

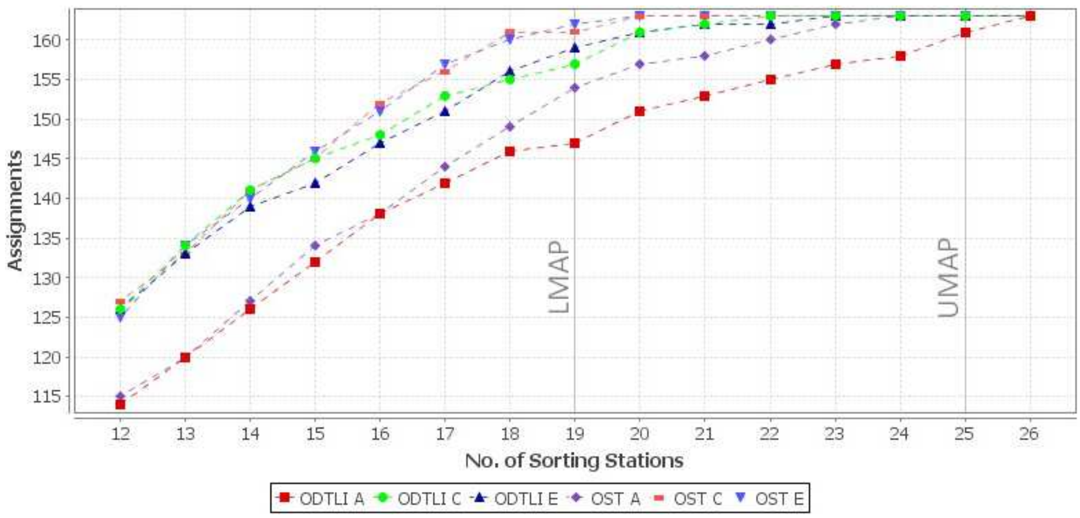


Figure B.3: Assignments to Terminal 1, with reduction of 163 flights and FIFO.

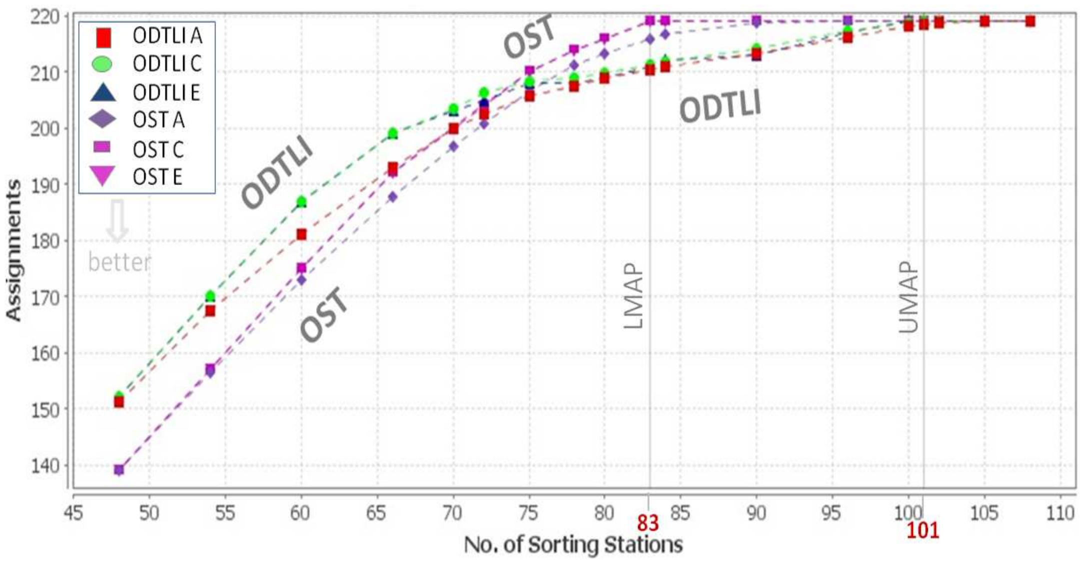


Figure B.4: Assignments to Terminal 1, with reduction of 219 flights and LIFO.

Data Sets for 142 flights

Constructive algorithms results for data set with 142 flights, Tables B.8, B.9 and B.10.

'*A*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	0.1581/0.8419	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0239)	GT (0.0001)
54	0.0598/0.9402	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0040)	GT (0.0000)
60	GT (0.0462)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0016)	GT (0.0000)
66	GT (0.0359)	GT (0.0000)	GT (0.0000)	0.4992/0.4992	GT (0.0093)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.162/0.838	GT (0.0046)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*B*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0056)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
66	GT (0.0385)	GT (0.0000)	GT (0.0000)	0.4972/0.4972	GT (0.0016)	GT (0.0000)
72	GT (0.0000)	GT (0.0011)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0121)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*C*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	0.2418/0.7582	GT (0.0000)	GT (0.0000)	0.4993/0.4993	GT (0.0165)	GT (0.0000)
54	GT (0.0397)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0038)	GT (0.0000)
60	GT (0.0283)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0001)	GT (0.0000)
66	GT (0.0228)	GT (0.0000)	GT (0.0000)	0.4984/0.4984	0.0613/0.9387	GT (0.0000)
72	GT (0.0000)	GT (0.0414)	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.1611/0.8389
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*D*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0263)	GT (0.0000)	GT (0.0000)	0.4972/0.4972	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	0.4994/0.4994	GT (0.0000)	GT (0.0000)
60	GT (0.0016)	GT (0.0000)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
66	0.0767/0.9233	GT (0.0000)	GT (0.0000)	0.4972/0.4972	GT (0.0153)	GT (0.0000)
72	GT (0.0000)	GT (0.0121)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0414)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*E*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0020)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0011)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0121)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

Table B.8: ODTLI and ODT for 142 flights.

‘A’ Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	0.5281/0.4719	LT (0.0000)	GT (0.0000)	GT (0.0000)	0.1510/0.8490
60	LT (0.0049)	LT (0.0000)	LT (0.0000)	0.6765/0.3235	LT (0.0000)	LT (0.0000)
66	LT (0.0018)	LT (0.0000)	LT (0.0000)	0.9251/0.0748	LT (0.0000)	LT (0.0000)
72	GT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	0.8389/0.1611	LT (0.0001)
78	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

‘B’ Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	0.8959/0.1041
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0001)	LT (0.0000)	LT (0.0000)	0.8389/0.1611	LT (0.0000)	LT (0.0000)
72	GT (0.0000)	LT (0.0011)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0121)
78	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

‘C’ Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0006)	LT (0.0000)	LT (0.0000)	LT (0.0414)	LT (0.0020)	LT (0.0000)
72	GT (0.0000)	LT (0.0414)	LT (0.0000)	GT (0.0000)	GT (0.0000)	0.8389/0.1611
78	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

‘D’ Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0011)	LT (0.0000)	LT (0.0000)	0.8389/0.1611	LT (0.0011)	LT (0.0000)
72	GT (0.0000)	LT (0.0121)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0414)
78	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

‘E’ Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0011)	LT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0121)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

Table B.9: ODT and OST for 142 flights.

'A' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0025)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	0.8593/0.1407	LT (0.0002)	LT (0.0000)	0.6765/0.3235	0.8917/0.1083	LT (0.0454)
66	0.8856/0.1144	LT (0.0000)	LT (0.0000)	0.9251/0.0749	LT (0.0027)	LT (0.0144)
72	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0414)
78	GT (0.0000)	GT (0.0000)	LT (0.0020)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'B' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0003)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0121)	LT (0.0000)	LT (0.0000)	0.8389/0.1611	0.9208/0.0792	LT (0.0000)
72	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'C' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0414)	LT (0.0000)	LT (0.0000)	LT (0.0414)	LT (0.0414)	LT (0.0000)
72	GT (0.0000)	GT (0.0000)	LT (0.0006)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'D' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0222)	LT (0.0000)	LT (0.0000)	0.8389/0.1611	0.9208/0.0792	LT (0.0000)
72	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'E' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

Table B.10: ODTLI and OST for 142 flights.

B.1.3 Data Sets for 270 flights

Constructive algorithms results for data set with 270 flights, Tables B.11, B.12 and B.13.

'A' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	0.1495/0.8505	GT (0.0463)	GT (0.0000)	0.4995/0.4995	GT (0.0000)	GT (0.0000)
54	GT (0.0495)	GT (0.0144)	GT (0.0000)	LT (0.0000)	GT (0.0277)	GT (0.0008)
60	LT (0.0000)	LT (0.0000)	GT (0.0002)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0000)	LT (0.0000)	0.2053/0.7947	LT (0.0000)	LT (0.0000)	LT (0.0000)
72	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
78	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)

'B' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	0.0654/0.9345	GT (0.0036)	GT (0.0000)	0.4994/0.4994	GT (0.0000)	GT (0.0000)
54	LT (0.0000)	0.3273/0.6727	GT (0.0000)	LT (0.0000)	LT (0.0000)	GT (0.0018)
60	LT (0.0000)	LT (0.0000)	GT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
72	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
78	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)

'C' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	0.1537/0.8463	0.1571/0.8429	GT (0.0000)	0.4995/0.4995	GT (0.0071)	GT (0.0003)
54	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
72	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
78	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)

'D' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	0.0937/0.9063	GT (0.0074)	GT (0.0000)	0.4993/0.4993	GT (0.0000)	GT (0.0000)
54	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
72	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
78	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)

'E' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	GT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)
54	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
60	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
66	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
72	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
78	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)

Table B.11: ODTLI and ODT for 270 flights.

'*A*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	LT (0.0121)	LT (0.0000)	LT (0.0000)	GT (0.0000)	LT (0.0066)	LT (0.0000)
60	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	LT (0.0222)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*B*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	GT (0.0000)	LT (0.0414)	LT (0.0000)	GT (0.0000)	GT (0.0000)	LT (0.0001)
60	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*C*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*D*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*E*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

Table B.12: ODT and OST for 270 flights.

'*A*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	0.8389/0.1611	LT (0.0121)	LT (0.0000)	GT (0.0000)	0.8389/0.1611	LT (0.0000)
60	GT (0.0000)	GT (0.0000)	LT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	0.9208/0.0792	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*B*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	GT (0.0000)	0.9208/0.0792	LT (0.0000)	GT (0.0000)	GT (0.0000)	0.9208/0.0792
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*C*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*D*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

'*E*' Baggage Sorting Station Assignment Algorithm

No. BSSs	Closest	Closest Min Open	FIFO	LIFO	Middle	Random 0
48	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)	LT (0.0000)
54	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
60	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
66	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
72	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)
78	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)	GT (0.0000)

Table B.13: ODTLI and OST for 270 flights.

B.2 Steady State Evolutionary Algorithms

B.2.1 SSEA Population Sizes Fitness Statistical Results

Summary tables of the Mann-Whitney tests run against the considered operators for the considered number of population sizes and Replacement Strategies 1. The values presented in the tables corresponds to the number of cases for the numbers of BSSs smaller than LMAP within [LMAP ... UMAP[and greater or equal to UMAP respectively where it is not statistically significantly less fit than any of the other cases for the same conditions.

3-pier topology

16^{th} December 2009 with 194 flights

Max. (9, 5, 3)	15	30	50	100
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (9, 5, 3)	200	500	1000	**2000**
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	1 (0, 1, 0)	2 (0, 1, 1)	3 (0, 2, 1)	16 (8, *5*, *3*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	**17** (*9*, *5*, *3*)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.14: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and C1P.

Max. (9, 5, 3)	15	30	50	100
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (9, 5, 3)	200	500	1000	**2000**
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	1 (0, 1, 0)	1 (0, 1, 0)	1 (0, 1, 0)	**16** (*9*, 4, *3*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	15 (7, *5*, *3*)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.15: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and C2P.

Max. (9, 5, 3)	15	30	50	100
ES	6 (2, 1, *3*)	6 (2, 1, *3*)	6 (2, 1, *3*)	6 (2, 1, *3*)
IS1ES	6 (2, 1, *3*)	6 (2, 1, *3*)	7 (3, 1, *3*)	9 (5, 1, *3*)
IS1SUMS	0 (0, 0, 0)	8 (4, 1, *3*)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	6 (2, 1, *3*)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (9, 5, 3)	200	500	**800**	1000
ES	6 (2, 1, *3*)	7 (3, 1, *3*)	7 (3, 1, *3*)	7 (3, 1, *3*)
IS1ES	12 (6, 3, *3*)	14 (7, 4, *3*)	**16** (8, *5*, *3*)	15 (8, 4, *3*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	15 (8, 4, *3*)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	5 (1, 1, *3*)	0 (0, 0, 0)

Table B.16: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and DSEMO.

Max. (9, 5, 3)	1	5	**10**	15	30
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	10 (5, 2, *3*)	11 (5, 3, *3*)	**12** (7, 3, 2)	9 (6, 1, 2)	5 (3, 0, 2)
IS1SUMS	8 (2, 3, *3*)	2 (0, 0, 2)	2 (0, 0, 2)	0 (0, 0, 0)	5 (3, 0, 2)
SUMS	3 (1, 0, 2)	3 (1, 0, 2)	3 (1, 0, 2)	0 (0, 0, 0)	3 (1, 0, 2)
Max. (9, 5, 3)	50	100	200	500	1000
ES	0 (0, 0, 0)	1 (1, 0, 0)	2 (2, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	4 (2, 0, 2)	2 (2, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.17: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and MEBPFNR3.

Max. (9, 5, 3)	**1**	**5**	10	15	30
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	**13** (5, *5*, *3*)	**13** (6, 4, *3*)	11 (6, 3, 2)	12 (7, 3, 2)	4 (3, 0, 1)
IS1SUMS	9 (4, 2, *3*)	5 (1, 2, 2)	3 (1, 0, 2)	0 (0, 0, 0)	3 (1, 0, 2)
SUMS	4 (2, 0, 2)	3 (1, 0, 2)	4 (1, 2, 1)	0 (0, 0, 0)	2 (0, 0, 2)
Max. (9, 5, 3)	50	100	200	500	1000
ES	0 (0, 0, 0)	1 (1, 0, 0)	2 (2, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
IS1ES	3 (2, 0, 1)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.18: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and MEFNR3.

Max. (9, 5, 3)	1	5	10	**15**	30
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	6 (2, 2, 2)	9 (3, 4, 2)	12 (4, *5*, *3*)	**15** (7, *5*, *3*)	11 (5, 3, *3*)
IS1SUMS	6 (2, 2, 2)	4 (1, 1, 2)	3 (1, 0, 2)	0 (0, 0, 0)	3 (2, 0, 1)
SUMS	6 (2, 2, 2)	5 (1, 2, 2)	5 (1, 2, 2)	0 (0, 0, 0)	2 (0, 0, 2)
Max. (9, 5, 3)	50	100	200	500	1000
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	6 (3, 1, 2)	2 (2, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.19: Number of occurrences which cannot be said to be statistically significantly less fit than the others for 3-piers topology, 194 flights and RMEFNR2.

13	**MEBPFNR3**	**IS1ES**	**10, 15, 50**
	MEFNR3	**IS1ES**	**5, 10, 15, 50**
		IS1SUMS	**30**
14	**MEBPFNR3**	**IS1ES**	**1, 10, 15, 30**
	MEFNR3	**IS1ES**	**1, 5, 15**
15	**MEBPFNR3**	**ES**	**200**
	RMEFNR2	**IS1ES**	**30, 50, 100**
16	**MEBPFNR3**	**IS1ES**	**1, 10, 15, 30, 50**
		IS1SUMS	**30**
	MEFNR3	**IS1ES**	**10, 15, 30**
		IS1SUMS	**1, 5**
	RMEFNR2	**IS1ES**	**5, 15**
17	**MEFNR3**	**ES**	**200**
	RMEFNR2	**IS1ES**	**30, 100**
18	**MEBPFNR3**	**IS1ES**	**5, 15**
		IS1SUMS	**1**
		SUMS	**5**
	MEFNR3	**IS1ES**	**1, 10, 15**
		IS1SUMS	**1**
		SUMS	**1**
	RMEFNR2	**IS1ES**	**1, 5, 10, 15, 50**
		IS1SUMS	**1, 5**
		SUMS	**1, 4**
19	**RMEFNR2**	**IS1ES**	**10, 15, 30**
20	**RMEFNR2**	**IS1ES**	**10, 15**
21	**RMEFNR2**	**IS1ES**	**1, 5, 10, 15**
		IS1SUMS	**1**
		SUMS	**1, 10**
22	**RMEFNR2**	**IS1ES**	**5, 10, 15**
23	**RMEFNR2**	**IS1ES**	**5, 10, 15, 30, 50**
24	**RMEFNR2**	**IS1ES**	**10, 15**
25	**MEBPFNR3**	**IS1ES**	**1**
	MEFNR3	**IS1ES**	**1**
		IS1SUMS	**1**
	RMEFNR2	**IS1ES**	**1, 5, 10, 15, 30**
		IS1SUMS	**1, 5**
		SUMS	**1, 5, 10**
26	**MEFNR3**	**IS1ES**	**1**
		IS1SUMS	**1**
	RMEFNR2	**IS1ES**	**1, 5, 10, 15, 30**
		IS1SUMS	**1**
		SUMS	**1, 5, 10**
27	**RMEFNR2**	**IS1ES**	**10, 15, 30, 50**
28	**MEBPFNR3**	**IS1ES**	**1, 5, 10, 15, 30, 50**
		IS1SUMS	**1, 5, 10, 30**
		SUMS	**1, 5, 10, 30**
	MEFNR3	**IS1ES**	**1, 5, 10, 15, 30, 50**
		IS1SUMS	**1, 5, 10, 30**
		SUMS	**1, 5, 30**
	RMEFNR2	**IS1ES**	**1, 5, 10, 15, 30, 50**
		IS1SUMS	**1, 5, 10, 30**
		SUMS	**1, 5, 10, 30**
29	**MEBPFNR3**	**IS1ES**	**1, 5, 10, 15, 30, 50**
		IS1SUMS	**1, 5, 10, 30**
		SUMS	**1, 5, 10, 30**
	MEFNR3	**IS1ES**	**1, 5, 10, 15**
		IS1SUMS	**1, 5, 10, 30**
		SUMS	**1, 5, 10, 30**
	RMEFNR2	**IS1ES**	**1, 5, 10, 15, 30**
		IS1SUMS	**1, 5, 10**
		SUMS	**1, 5, 10, 30**

Table B.20: 3-pier topology, 194 flights with operators C1P, C2P, DSEMO, MEBPFNR3, MEFNR3 and RMEFNR2.

1st March 2010 with 163 flights

Max. (6, 6, 5)	15	30	50	100
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1SUMS	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (6, 6, 5)	200	500	1000	**2000**
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1ES	1 (0, 0, 1)	1 (0, 0, 1)	5 (0, 3, 2)	16 (5, *6*, *5*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	**17** (*6*, *6*, *5*)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)

Table B.21: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and C1P.

Max. (6, 6, 5)	15	30	50	100
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1SUMS	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (6, 6, 5)	200	500	1000	**2000**
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1ES	1 (0, 0, 1)	2 (0, 1, 1)	4 (0, 1, 3)	**17** (*6*, *6*, *5*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	**17** (*6*, *6*, *5*)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)

Table B.22: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and C2P.

Max. (6, 6, 5)	1	5	10	15	30	50
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	8 (0, 3, *5*)	8 (0, 3, *5*)	8 (0, 3, *5*)
IS1ES	0 (0, 0, 0)	1 (0, 1, 0)	1 (0, 1, 0)	9 (1, 3, *5*)	9 (1, 3, *5*)	9 (1, 3, *5*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	9 (1, 3, *5*)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	8 (0, 3, *5*)	0 (0, 0, 0)

Max. (6, 6, 5)	100	200	500	800	**1000**
ES	8 (0, 3, *5*)	8 (0, 3, *5*)	8 (0, 3, *5*)	8 (0, 3, *5*)	9 (1, 3, *5*)
IS1ES	11 (3, 3, *5*)	13 (5, 3, *5*)	14 (4, 5, *5*)	**15** (4, *6*, *5*)	10 (2, 3, *5*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	13 (3, 5, *5*)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	8 (0, 3, *5*)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.23: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and DSEMO.

Max. (6, 6, 5)	1	**5**	10	15	30
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1ES	10 (1, 5, 4)	**13** (3, 5, *5*)	12 (4, 4, 4)	11 (4, 4, 3)	5 (2, 0, 3)
IS1SUMS	6 (1, 3, 2)	2 (0, 0, 2)	2 (0, 0, 2)	0 (0, 0, 0)	5 (3, 0, 2)
SUMS	4 (1, 1, 2)	5 (0, 1, 4)	3 (0, 0, 3)	0 (0, 0, 0)	5 (0, 1, 4)
Max. (6, 6, 5)	50	100	200	500	1000
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1ES	3 (2, 0, 1)	2 (1, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.24: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and MEBPFNR3.

Max. (6, 6, 5)	1	5	**10**	15	30
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1ES	13 (3, *6*, 4)	12 (3, 5, 4)	**14** (4, 5, *5*)	10 (5, 2, 3)	4 (2, 0, 2)
IS1SUMS	10 (1, 4, *5*)	2 (0, 0, 2)	5 (2, 1, 2)	0 (0, 0, 0)	6 (3, 0, 3)
SUMS	4 (2, 1, 1)	4 (0, 1, 3)	4 (0, 1, 3)	0 (0, 0, 0)	5 (0, 1, 4)
Max. (6, 6, 5)	50	100	200	500	1000
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1ES	3 (2, 0, 1)	3 (2, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.25: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and MEFNR3.

Max. (6, 6, 5)	1	5	10	**15**	30
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1ES	4 (0, 0, 4)	14 (5, 5, 4)	14 (5, 5, 4)	**16** (5, *6*, *5*)	10 (3, 4, 3)
IS1SUMS	3 (0, 0, 3)	2 (0, 0, 2)	1 (0, 0, 1)	0 (0, 0, 0)	6 (1, 2, 3)
SUMS	4 (0, 0, 4)	4 (0, 0, 4)	3 (0, 0, 3)	0 (0, 0, 0)	4 (0, 0, 4)
Max. (6, 6, 5)	50	100	200	500	1000
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1ES	8 (3, 1, 4)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.26: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and RMEFNR2.

13	MEBPFNR3	IS1ES	30
		IS1SUMS	30
	MEFNR3	IS1ES	15, 50
14	MEBPFNR3	IS1ES	10, 15, 50
	MEFNR3	IS1ES	1, 5, 10, 15, 30, 100
		IS1SUMS	30
		SUMS	1
15	MEBPFNR3	IS1ES	50, 100
		IS1SUMS	30
	MEFNR3	IS1ES	50, 100
		IS1SUMS	30
16	MEBPFNR3	IS1ES	5, 10, 15
	MEFNR3	IS1ES	5, 10, 15
		SUMS	1
17	DSEMO	ES	1000
		IS1ES	200, 500, 800, 1000
		IS1SUMS	800
18	DSEMO	IS1ES	500, 800, 1000
		IS1SUMS	800
19	MEBPFNR3	IS1ES	1, 5, 10, 15
		IS1SUMS	1
	MEFNR3	IS1ES	1, 5, 10
		IS1SUMS	1
		SUMS	5
20	MEBPFNR3	IS1ES	5, 10, 15
	MEFNR3	IS1ES	1, 5, 10, 15
		IS1SUMS	1
		SUMS	1
21	MEBPFNR3	IS1ES	1
	MEFNR3	IS1ES	1, 5, 10, 15
22	MEBPFNR3	IS1ES	1
		IS1SUMS	1
		SUMS	1, 5
	MEFNR3	IS1ES	1
		SUMS	30
23	MEBPFNR3	IS1ES	1, 5, 10, 15
		IS1SUMS	1
	MEFNR3	IS1ES	1, 5, 10
		IS1SUMS	1
		SUMS	10
24	MEBPFNR3	IS1ES	1, 5, 10
	MEFNR3	IS1ES	1, 5, 10
25	MEBPFNR3	IS1ES	1, 5
	MEFNR3	IS1ES	1, 5
		IS1SUMS	1
26	MEBPFNR3	IS1ES	10, 30
	MEFNR3	IS1ES	1, 5
		SUMS	5, 10
	RMEFNR2	IS1ES	10, 50
		IS1SUMS	1, 10, 30
		SUMS	1, 30
27	MEBPFNR3	IS1ES	15
		SUMS	30
	MEFNR3	IS1SUMS	1
		SUMS	30
	RMEFNR2	IS1ES	1, 10, 15, 30, 50
		IS1SUMS	1, 5, 10, 30
		SUMS	1, 30
28	MEBPFNR3	IS1ES	1, 5, 10, 15, 30
		IS1SUMS, SUMS	1, 5, 10, 30
	MEFNR3	IS1ES	1, 5, 10, 15, 30
		IS1SUMS	1, 5, 10, 30
		SUMS	5, 10, 30
	RMEFNR2	IS1ES	1, 5, 10, 15
		IS1SUMS	1, 5, 10
		SUMS	1, 5, 10, 30
29	C1P	ES, IS1ES	15, 30, 50, 100, 200, 500, 1000, 2000
		IS1SUMS	2000
		SUMS	2000
	C2P	ES, IS1ES	15, 30, 50, 100, 200, 500, 1000, 2000
		IS1SUMS	2000
		SUMS	2000
	DSEMO	ES, IS1ES	15, 30, 50, 100, 200, 500, 1000, 2000
		IS1SUMS	800
		SUMS	800
	MEBPFNR3	ES, IS1ES	1, 5, 10, 15, 30, 50, 100, 200, 500, 1000
		IS1SUMS, SUMS	1, 5, 10, 30
	MEFNR3	ES, IS1ES	1, 5, 10, 15, 30, 50, 100, 200, 500, 1000
		IS1SUMS, SUMS	1, 5, 10, 30
	RMEFNR2	ES	1, 5, 10, 15, 30, 50, 100, 200, 500, 1000
		IS1ES	1, 5, 10, 15, 30, 50, 100, 200, 500, 1000
		IS1SUMS	1, 5, 10, 30
		SUMS	1, 5, 10, 30

Table B.27: 3-pier topology and 163 flights with operators C1P, C2P, DSEMO, MEBPFNR3, MEFNR3 and RMEFNR2.

4-pier topology

16^{th} December 2009 with 194 flights

Max. (9, 5, 3)	15	30	50	100
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (9, 5, 3)	200	500	1000	**2000**
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	15 (8, 4, *3*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	**16** (8, *5*, *3*)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.28: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and C1P.

Max. (9, 5, 3)	15	30	50	100
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (9, 5, 3)	200	500	1000	**2000**
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)	16 (8, *5*, *3*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	**17** (*9*, *5*, *3*)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.29: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and C2P.

Max. (9, 5, 3)	15	30	50	100
ES	4 (0, 1, *3*)	4 (0, 1, *3*)	4 (0, 1, *3*)	5 (1, 1, *3*)
IS1ES	5 (1, 1, *3*)	5 (1, 1, *3*)	6 (2, 1, *3*)	6 (2, 1, *3*)
IS1SUMS	0 (0, 0, 0)	6 (2, 1, *3*)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	4 (0, 1, *3*)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (9, 5, 3)	200	**500**	800	1000
ES	5 (1, 1, *3*)	5 (1, 1, *3*)	5 (1, 1, *3*)	5 (1, 1, *3*)
IS1ES	8 (4, 1, *3*)	**15** (7, *5*, *3*)	14 (7, 4, *3*)	14 (7, 4, *3*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	13 (7, 3, *3*)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	4 (0, 1, *3*)	0 (0, 0, 0)

Table B.30: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and DSEMO.

Max. (9, 5, 3)	1	**5**	10	**15**	30
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	5 (3, 2, 0)	**11** (4, *5*, 2)	9 (4, 3, 2)	**11** (7, 3, 1)	8 (6, 1, 1)
IS1SUMS	1 (1, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	5 (5, 0, 0)
SUMS	1 (1, 0, 0)	0 (0, 0, 0)	1 (0, 1, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Max. (9, 5, 3)	50	100	200	500	1000
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	3 (3, 0, 0)	2 (2, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.31: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and MEBPFNR3.

Max. (9, 5, 3)	1	5	**10**	15	30
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	9 (5, 3, 1)	13 (6, *5*, 2)	**15** (8, *5*, 2)	12 (6, 4, 2)	6 (5, 0, 1)
IS1SUMS	5 (3, 2, 0)	3 (3, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)	6 (5, 1, 0)
SUMS	3 (3, 0, 0)	2 (1, 1, 0)	1 (0, 1, 0)	0 (0, 0, 0)	3 (1, 2, 0)

Max. (9, 5, 3)	50	100	200	500	1000
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	5 (5, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.32: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and MEFNR3.

Max. (9, 5, 3)	1	5	**10**	15	30
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	1 (0, 1, 0)	7 (3, 2, 2)	**15** (7, *5*, *3*)	14 (8, 3, *3*)	8 (8, 0, 0)
IS1SUMS	1 (0, 1, 0)	3 (3, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)	4 (4, 0, 0)
SUMS	0 (0, 0, 0)	1 (0, 1, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Max. (9, 5, 3)	50	100	200	500	1000
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	8 (5, 2, 1)	3 (3, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.33: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and RMEFNR2.

13	MEBPFNR3	IS1ES	10, 30, 50, 100, 200
		IS1SUMS	30
		SUMS	1
	MEFNR3	IS1ES	10, 15, 50, 100
		IS1SUMS	10, 30
14	MEBPFNR3	IS1ES	5, 10, 30, 100
		IS1SUMS	30
	MEFNR3	IS1ES	1, 5, 10, 15, 30, 50, 100
		IS1SUMS	10, 30
15	MEBPFNR3	IS1ES	30, 50
		IS1SUMS	10, 30
		SUMS	1
	MEFNR3	IS1ES	1, 15, 30, 50
		IS1SUMS	5
		SUMS	1
16	MEBPFNR3	IS1ES	1, 5, 15, 30, 50
		IS1SUMS	5, 30
		SUMS	5
	MEFNR3	IS1ES	5, 10, 15, 30, 50, 100
		IS1SUMS	5, 10, 30
17	MEBPFNR3	IS1ES	15
		IS1SUMS	1
		SUMS	1
	MEFNR3	IS1ES	10
18	MEBPFNR3	IS1ES	1, 5, 10, 15, 30
		SUMS	5
	MEFNR3	IS1ES	5, 10
		IS1SUMS	1
		SUMS	5
19	MEBPFNR3	IS1ES	5, 10, 15, 50
		IS1SUMS	1
	MEFNR3	IS1ES	1, 5, 10, 15
20	MEBPFNR3	IS1ES	5, 15
	MEFNR3	IS1ES	1, 10
		IS1SUMS	1
	RMEFNR2	IS1ES	10, 15
21	MEBPFNR3	IS1ES	1, 5, 10, 15
		IS1SUMS	30
	MEFNR3	IS1ES	5, 10
	RMEFNR2	IS1ES	30
22	MEBPFNR3	IS1SUMS	1
	RMEFNR2	IS1ES	1, 10
		IS1SUMS	1
		SUMS	5
23	MEBPFNR3	IS1ES	1, 5, 15
	RMEFNR2	IS1ES	5, 10, 50
24	RMEFNR2	IS1ES	10, 15
25	MEBPFNR3	IS1ES	5, 15
	MEFNR3	IS1ES	5, 10
	RMEFNR2	IS1ES	5, 10, 15, 50
26	RMEFNR2	IS1ES	10, 15
27	MEBPFNR3	IS1ES	5
	MEFNR3	IS1ES	5
	RMEFNR2	IS1ES	5, 10, 15
28	RMEFNR2	IS1ES	5, 10, 15
29	RMEFNR2	IS1ES	10, 15, 50

Table B.34: 4-pier topology, 194 flights with operators C1P, C2P, DSEMO, MEBPFNR3, MEFNR3 and RMEFNR2.

1st March 2010 with 163 flights

Max. (6, 6, 5)	15	30	50	100
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (6, 6, 5)	200	500	1000	**2000**
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)	**17** (*6*, *6*, *5*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	**17** (*6*, *6*, *5*)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.35: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and C1P.

Max. (6, 6, 5)	15	30	50	100
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (6, 6, 5)	200	500	1000	**2000**
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	1 (0, 0, 1)	0 (0, 0, 0)	2 (0, 0, 2)	**17** (*6*, *6*, *5*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	**17** (*6*, *6*, *5*)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.36: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and C2P.

Max. (6, 6, 5)	15	30	50	100
ES	7 (0, 2, *5*)	7 (0, 2, *5*)	7 (0, 2, *5*)	7 (0, 2, *5*)
IS1ES	7 (0, 2, *5*)	7 (0, 2, *5*)	8 (1, 2, *5*)	11 (3, 3, *5*)
IS1SUMS	0 (0, 0, 0)	8 (1, 2, *5*)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	7 (0, 2, *5*)	0 (0, 0, 0)	0 (0, 0, 0)
Max. (6, 6, 5)	200	**500**	800	1000
ES	7 (0, 2, *5*)	7 (0, 2, *5*)	7 (0, 2, *5*)	7 (0, 2, *5*)
IS1ES	12 (4, 3, *5*)	**17** (*6*, *6*, *5*)	14 (4, 5, *5*)	15 (4, *6*, *5*)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	14 (3, *6*, *5*)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	7 (0, 2, *5*)	0 (0, 0, 0)

Table B.37: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and DSEMO.

Max. (6, 6, 5)	1	**5**	10	15	30
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	12 (3, 5, 4)	**14** (5, *6*, 3)	13 (5, 5, 3)	12 (*6*, 3, 3)	4 (3, 1, 0)
IS1SUMS	4 (1, 3, 0)	2 (0, 2, 0)	2 (1, 1, 0)	0 (0, 0, 0)	5 (2, 3, 0)
SUMS	2 (2, 0, 0)	2 (0, 2, 0)	1 (0, 1, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Max. (6, 6, 5)	50	100	200	500	1000
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	4 (4, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.38: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and MEBPFNR3.

Max. (6, 6, 5)	1	5	**10**	15	30
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	13 (3, 5, *5*)	15 (5, *6*, 4)	**16** (*6*, 5, *5*)	13 (*6*, *6*, 1)	5 (4, 1, 0)
IS1SUMS	7 (2, 3, 2)	2 (0, 2, 0)	0 (0, 0, 0)	0 (0, 0, 0)	5 (3, 2, 0)
SUMS	4 (2, 1, 1)	2 (0, 1, 1)	4 (0, 2, 2)	0 (0, 0, 0)	3 (0, 2, 1)

Max. (6, 6, 5)	50	100	200	500	1000
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	3 (3, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.39: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and MEFNR3.

Max. (6, 6, 5)	1	5	10	**15**	30
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	4 (0, 3, 1)	8 (1, 4, 3)	10 (2, *6*, 2)	**11** (4, 4, 3)	7 (3, 2, 2)
IS1SUMS	2 (0, 1, 1)	3 (2, 1, 0)	4 (3, 1, 0)	0 (0, 0, 0)	7 (5, 2, 0)
SUMS	2 (0, 1, 1)	2 (0, 1, 1)	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)

Max. (6, 6, 5)	50	100	200	500	1000
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
IS1ES	7 (3, 2, 2)	3 (3, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
IS1SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
SUMS	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.40: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and RMEFNR2.

The Figures B.5, B.6, B.7 and B.8 show the average percentage of improvement in fitness for different population sizes, where 0% refers to the best initial solution and 100% is the upper bound obtained when running CPLEX with the Mixed Integer Linear Programming (MILP) introduced in Section 3.3.

13	**DSEMO**	**IS1ES**	**100, 200, 500, 800**
14	**DSEMO**	**IS1ES**	**200, 500**
15	**MEBPFNR3**	**IS1ES**	**5, 10, 15, 30, 50, 100**
		IS1SUMS	**1, 10, 30**
	MEFNR3	**IS1ES**	**1, 10, 15, 30, 50, 100**
		IS1SUMS	**1, 30**
16	**MEBPFNR3**	**IS1ES**	**1, 5, 15, 30, 50**
		IS1SUMS	**5, 10**
		SUMS	**1**
	MEFNR3	**IS1ES**	**1, 5, 10, 15, 30, 50**
17	**DSEMO**	**IS1ES**	**50, 100, 200, 500, 1000**
18	**MEBPFNR3**	**IS1ES**	**5, 10, 15**
	MEFNR3	**IS1ES**	**5, 10, 15**
19	**MEBPFNR3**	**IS1ES**	**1, 5, 10, 15**
		IS1SUMS	**30**
20	**MEBPFNR3**	**IS1ES**	**1, 5, 10, 15**
		IS1SUMS	**1, 10, 30**
		SUMS	**5**
	MEFNR3	**IS1ES**	**1, 5, 10, 15, 30**
		IS1SUMS	**1, 30**
		SUMS	**1, 5**
	RMEFNR2	**IS1ES**	**1, 10, 15, 50**
21	**MEBPFNR3**	**IS1ES**	**10, 15**
	MEFNR3	**IS1ES**	**1, 5, 10, 15**
		IS1SUMS	**5**
	RMEFNR2	**IS1ES**	**10,15, 30**
		IS1SUMS	**30**
22	**MEBPFNR3**	**IS1ES**	**1, 5, 10**
		IS1SUMS	**1**
	MEFNR3	**IS1ES**	**1, 5, 10**
23	**MEBPFNR3**	**IS1ES**	**5, 10**
	MEFNR3	**IS1ES**	**1, 5**
24	**MEBPFNR3**	**IS1ES**	**1, 5**
		IS1SUMS	**1**
		SUMS	**5**
	MEFNR3	**IS1ES**	**1, 5, 10, 15**
		IS1SUMS	**1**
25	**MEBPFNR3**	**IS1ES**	**1, 5, 10, 15**
		IS1SUMS	**1**
	MEFNR3	**IS1ES**	**1, 10**
		IS1SUMS	**1**
	RMEFNR2	**IS1ES**	**15, 30**
26	**MEBPFNR3**	**IS1ES**	**5, 10, 15**
	MEFNR3	**IS1ES**	**1, 10**
27	**RMEFNR2**	**IS1ES**	**5, 10, 15**
28	**MEBPFNR3**	**IS1ES**	**1**
	RMEFNR2	**IS1SUMS**	**1**
		SUMS	**1**
29	**MEBPFNR3**	**IS1ES**	**1, 5**
	MEFNR3	**IS1ES**	**1**
	RMEFNR2	**IS1ES**	**1, 10**
		SUMS	**10**

Table B.41: 4-pier topology, 163 flight for operators C1P, C2P, DSEMO, MEBPFNR3, MEFNR3 and RMEFNR2.

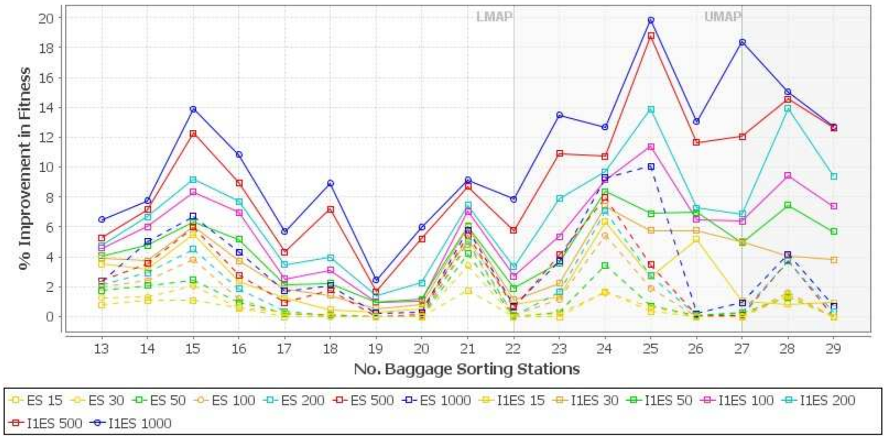


Figure B.5: Average percent improvement on average fitness for 4-pier topology, 194 flights (16^{th} December 2009) and 1-point crossover (C1P) and population sizes.

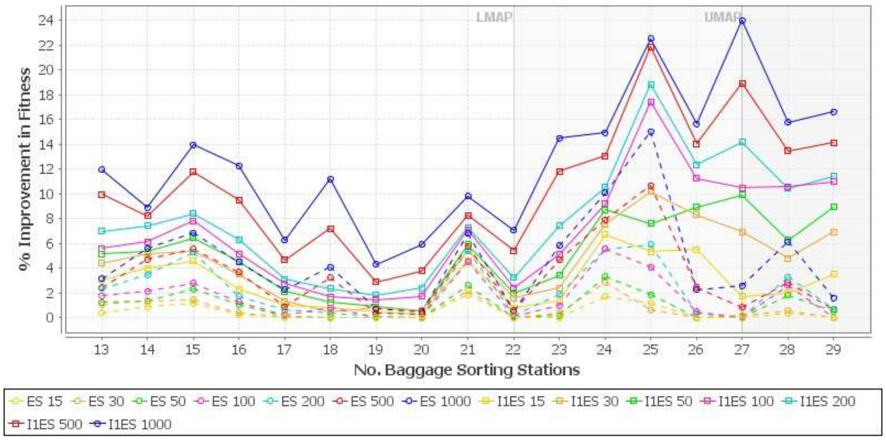


Figure B.6: Average percent improvement on average fitness for 4-pier topology, 194 flights (16^{th} December 2009) and 2-point crossover (C2P) and population sizes.

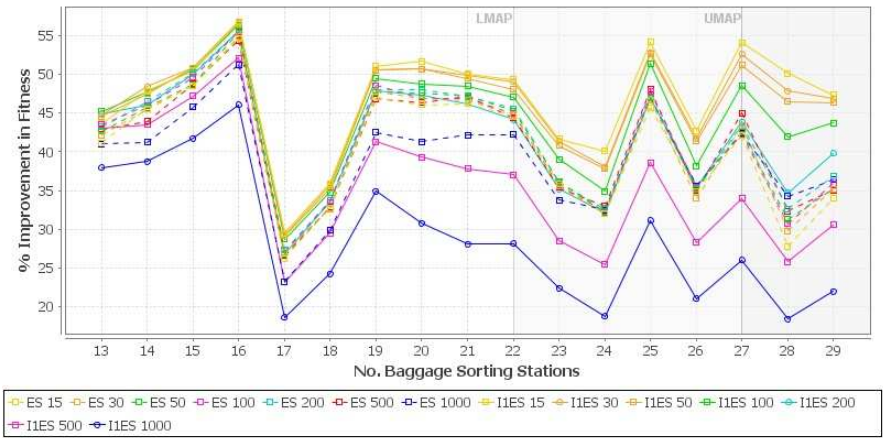


Figure B.7: Average percent improvement on average fitness for 4-pier topology, 194 flights (16^{th} December 2009) and Multi Exchange By Pier between a Fixed Number of 3 Resources (MEBPFNR3) and population sizes.

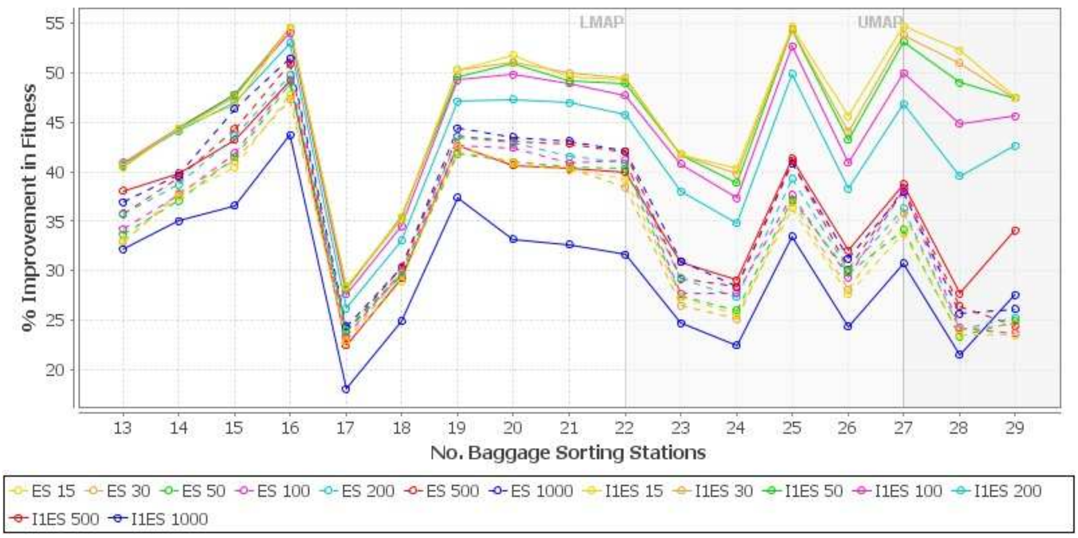


Figure B.8: Average percent improvement on average fitness for 4-pier topology, 194 flights (16^{th} December 2009) and Range Multi Exchange between Fixed Number of 2 Resources (RMEFNR2) and population sizes.

B.2.2 Crossover Operators Population Sizes

This section provides the summary results of the comparison between the different versions of crossover operators considered in this thesis (Section 5.4.2) for the two data sets of 16^{th} December 2009 and $1^{st} March 2010$, and 3-pier and 4-pier topologies.

The 1-point serial crossover generating two children (SC1P(2)) performs statistically significantly better in many more instances than 1-point crossover (C1P) and 2-point crossover (C2P) as shown in the following tables for both data sets of London Heathrow airport Terminal 1 and for both topologies considered.

3-pier topology

13	**C2P**	**IS1ES**	**2000**
14	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**2000**
15	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**2000**
16	**C2P**	**IS1ES**	**2000**
17	**C1P**	**IS1ES**	**2000**
	C2P	**IS1ES**	**2000**
	SC1P(2)	**IS1ES**	**2000**
		IS1SUMS	**2000**
18	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**500, 2000**
19	**SC1P(2)**	**IS1ES**	**500, 2000**
		IS1SUMS	**1000, 2000**
20	**SC1P(2)**	**IS1ES**	**1000, 2000**
		IS1SUMS	**2000**
21	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**2000**
22	**SC1P(2)**	**IS1ES**	**1000, 2000**
		IS1SUMS	**500, 1000, 2000**
23	**SC1P(2)**	**ES**	**2000**
		IS1ES	**500, 1000, 2000**
		IS1SUMS	**500, 1000**
24	**SC1P(2)**	**IS1ES**	**500, 1000, 2000**
		IS1SUMS	**500**
25	**SC1P(2)**	**IS1ES**	**500, 2000**
		IS1SUMS	**1000, 2000**
26	**SC1P(2)**	**IS1ES**	**500, 1000**
		IS1SUMS	**500, 1000, 2000**
27	**SC1P(2)**	**IS1ES**	**1000**
		IS1SUMS	**500**
28	**SC1P(2)**	**IS1ES**	**500, 1000**
		IS1SUMS	**500, 2000**
29	**SC1P(2)**	**IS1ES**	**500, 1000, 2000**
		IS1SUMS	**500, 1000, 2000**

Table B.42: Instances statistically significantly not less fit than the others for a 3-pier topology and data set of 16^{th} December 2009.

The SC1P(2) performs better than C1P and C2P for the data set of 16^{th} December 2009, and population selectors Index Selection with Elitist Selection and a group size of 1 (IS1ES) and Index Selection with Stochastic Universal Modified Sampling and group size of 1 (IS1SUMS). A higher population size of 2000 solutions appears to perform better for lower number of BSSs where the problem is more difficult to solve, as shown in Tables B.42 and B.43. Furthermore, the C2P provides better results for very low number of BSSs (N << LMAP).

Max. (9, 5, 3)	Selector	500	1000	2000
C1P	IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)
C2P	IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	3 (3, 0, 0)
SC1P(2)	ES	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 1, 0)
	IS1ES	8 (1, **<u>5</u>**, 2)	8 (1, 4, **<u>3</u>**)	<u>12</u> (<u>7</u>, 4, 1)
	IS1SUMS	7 (0, 4, **<u>3</u>**)	6 (1, 4, 1)	<u>12</u> (<u>7</u>, 3, 2)

Table B.43: Summary: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and data set of 16^{th} December 2009.

The SC1P(2) with selector IS1SUMS and data set of 1^{st} March 2010 performs overall better, as shown in Tables B.44 and B.45.

Max. (6, 6, 5)	Selector	500	1000	2000
C1P	ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
	IS1ES	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 1, 1)
	IS1SUMS	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
	SUMS	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C2P	ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
	IS1ES	1 (0, 0, 1)	1 (0, 0, 1)	6 (3, 2, 1)
	IS1SUMS	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
	SUMS	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
SC1P(2)	ES	2 (0, 0, 2)	4 (0, 0, <u>4</u>)	5 (0, 1, <u>4</u>)
	IS1ES	6 (0, 2, <u>4</u>)	7 (0, 3, <u>4</u>)	10 (3, 4, 3)
	IS1SUMS	4 (0, 1, 3)	7 (1, 2, <u>4</u>)	**12** (<u>4</u>, <u>5</u>, 3)
	SUMS	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)

Table B.44: Summary: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and data set of 1^{st} March 2010.

13	C2P	IS1ES	2000
14	C2P	IS1ES	2000
15	C2P	IS1ES	2000
	SC1P(2)	IS1SUMS	2000
16	SC1P(2)	IS1ES	2000
		IS1SUMS	2000
17	SC1P(2)	IS1ES	2000
		IS1SUMS	1000, 2000
18	SC1P(2)	IS1ES	2000
		IS1SUMS	2000
19	SC1P(2)	IS1ES	2000
		IS1SUMS	2000
20	C2P	IS1ES	2000
	SC1P(2)	ES	2000
		IS1SUMS	2000
21	C2P	IS1ES	2000
22	C1P	IS1ES	2000
	SC1P(2)	IS1ES	1000, 2000
		IS1SUMS	500, 2000
23	SC1P(2)	IS1ES	500, 1000, 2000
		IS1SUMS	1000, 2000
24	SC1P(2)	IS1ES	500, 1000, 2000
		IS1SUMS	1000, 2000
25	SC1P(2)	ES	1000, 2000
		IS1ES	500, 1000, 2000
		IS1SUMS	1000, 2000
26	SC1P(2)	ES	2000
		IS1ES	500, 1000, 2000
		IS1SUMS	500, 1000, 2000
27	SC1P(2)	ES	1000
		IS1ES	500, 1000
		IS1SUMS	500, 1000
28	SC1P(2)	ES	500, 1000, 2000
29	C1P	ES	500, 1000, 2000
		IS1ES	500, 1000, 2000
		IS1SUMS	500, 1000, 2000
		SUMS	500, 1000, 2000
	C2P	ES	500, 1000, 2000
		IS1ES	500, 1000, 2000
		IS1SUMS	500, 1000, 2000
		SUMS	500, 1000, 2000
	SC1P(2)	ES	500, 1000, 2000
		IS1ES	500, 1000, 2000
		IS1SUMS	500, 1000, 2000
		SUMS	500, 1000, 2000

Table B.45: Instances statistically significantly not less fit than the others for a 3-pier topology and data set of 1^{st} March 2010.

4-pier topology

Overall the SC1P(2) with selector IS1SUMS for a population size of 2000 and data set of 16^{th} December 2009 performs better, as shown in Tables B.46 and B.47.

Max. (9, 5, 3)	Selector	500	1000	2000
C1P	IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	3 (2, 0, 1)
C2P	IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	4 (4, 0, 0)
SC1P(2)	IS1ES	1 (0, 1, 0)	4 (0, 3, 1)	13 (6, 4, **3**)
	IS1SUMS	2 (0, 1, 1)	6 (0, 3, **3**)	14 (6, **5**, **3**)

Table B.46: Summary: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 20091216.

13	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**2000**
14	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**2000**
15	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**2000**
16	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**2000**
17	**C2P**	**IS1ES**	**2000**
18	**C1P**	**IS1ES**	**2000**
	C2P	**IS1ES**	**2000**
19	**C1P**	**IS1ES**	**2000**
	C2P	**IS1ES**	**2000**
20	**C2P**	**IS1ES**	**2000**
	SC1P(2)	**IS1ES**	**2000**
		IS1SUMS	**2000**
21	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**2000**
22	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**2000**
23	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**2000**
24	**SC1P(2)**	**IS1ES**	**1000**
		IS1SUMS	**500, 1000, 2000**
25	**SC1P(2)**	**IS1ES**	**500, 1000, 2000**
		IS1SUMS	**1000, 2000**
26	**SC1P(2)**	**IS1ES**	**1000, 2000**
		IS1SUMS	**1000, 2000**
27	**SC1P(2)**	**IS1ES**	**2000**
		IS1SUMS	**1000, 2000**
28	**SC1P(2)**	**IS1ES**	**1000, 2000**
		IS1SUMS	**1000, 2000**
29	**C1P**	**IS1ES**	**2000**
	SC1P(2)	**IS1ES**	**2000**
		IS1SUMS	**500, 1000, 2000**

Table B.47: Instances statistically significantly not less fit than the others for a 4-pier topology and data set of 16^{th} December 2009.

Overall the SC1P(2) with selector IS1ES for a population size of 2000 and data set of 1^{st} March 2010 performs better but for the range of number of BSSs for a real problem, i.e. $N \geq$ LMAP, the SC1P(2) with selector IS1SUMS is preferable with a lower population size of 500 or 1000 solutions, as shown in Tables B.48 and B.49.

Max. (6, 6, 5)	Selector	500	1000	2000
C1P	IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	3 (0, 3, 0)
C2P	IS1ES	0 (0, 0, 0)	0 (0, 0, 0)	5 (2, 3, 0)
SC1P(2)	ES	1 (0, 0, 1)	5 (0, 1, 4)	4 (0, 0, 4)
	IS1ES	5 (0, 1, 4)	7 (0, 3, 4)	$\underline{10}$ ($\underline{5}$, 2, 3)
	IS1SUMS	7 (0, 2, $\underline{5}$)	7 (0, 2, $\underline{5}$)	9 (4, 2, 3)

Table B.48: Summary: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and data set of 20100301.

13	SC1P(2)	IS1ES	2000
14	SC1P(2)	IS1ES	2000
14	SC1P(2)	IS1SUMS	2000
15	SC1P(2)	IS1ES	2000
15	SC1P(2)	IS1SUMS	2000
16	SC1P(2)	IS1ES	2000
17	C2P	IS1ES	2000
17	SC1P(2)	IS1ES	2000
17	SC1P(2)	IS1SUMS	2000
18	C2P	IS1ES	2000
18	SC1P(2)	IS1SUMS	2000
19	C2P	IS1ES	2000
20	C1P	IS1ES	2000
20	C2P	IS1ES	2000
21	C2P	IS1ES	2000
22	SC1P(2)	IS1ES	500, 1000, 2000
22	SC1P(2)	IS1SUMS	1000, 2000
23	C1P	IS1ES	2000
23	SC1P(2)	IS1ES	1000
23	SC1P(2)	IS1SUMS	500, 1000, 2000
24	C1P	IS1ES	2000
24	SC1P(2)	ES	1000
24	SC1P(2)	IS1ES	1000, 2000
24	SC1P(2)	IS1SUMS	500
25	SC1P(2)	ES	1000, 2000
25	SC1P(2)	IS1ES	500, 2000
25	SC1P(2)	IS1SUMS	500, 1000
26	SC1P(2)	ES	1000, 2000
26	SC1P(2)	IS1ES	500, 1000, 2000
26	SC1P(2)	IS1SUMS	500, 1000, 2000
27	SC1P(2)	ES	1000, 2000
27	SC1P(2)	IS1ES	1000
27	SC1P(2)	IS1SUMS	500, 1000, 2000
28	SC1P(2)	ES	500, 1000, 2000
28	SC1P(2)	IS1ES	500, 1000, 2000
28	SC1P(2)	IS1SUMS	500, 1000, 2000
29	SC1P(2)	IS1ES	500, 1000
29	SC1P(2)	IS1SUMS	500, 1000

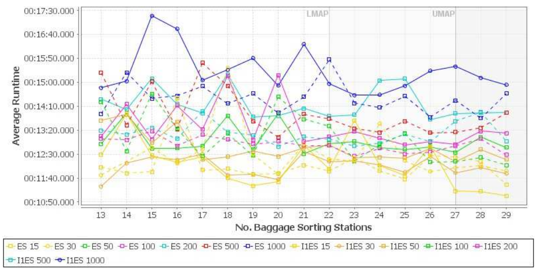
Table B.49: Instances statistically significantly not less fit than the others for a 4-pier topology and data set of 20100301.

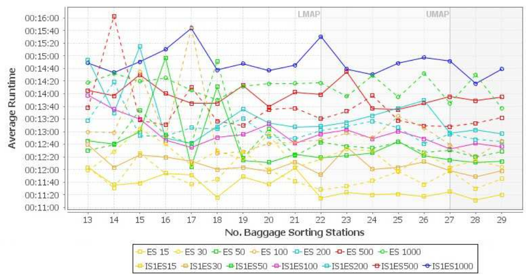
B.2.3 Run time Results for the Different Population Sizes

The results for the runtime for different population sizes and the ABSSAP are presented here for 3-pier topology and 4-pier topology results.

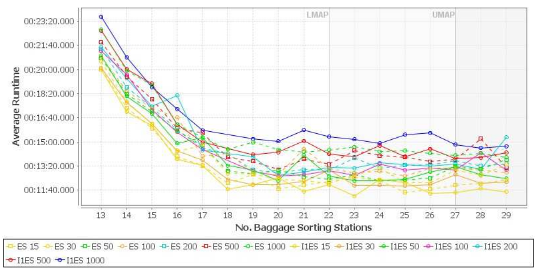
3-pier topology 4-pier topology

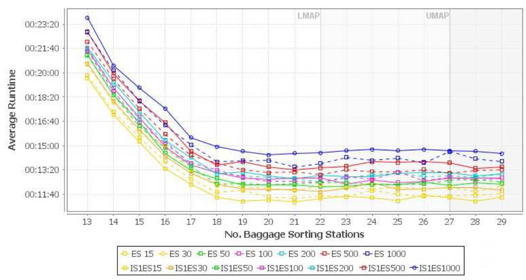
C1P



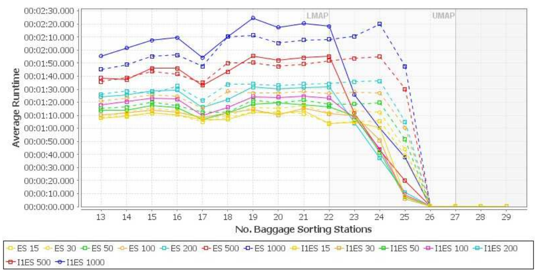


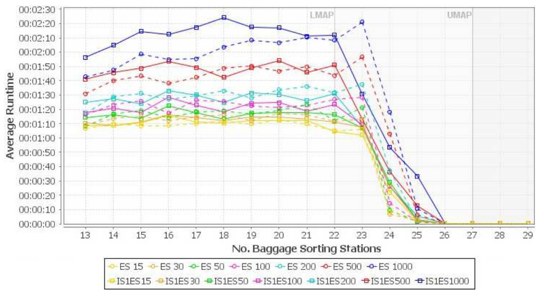
C2P



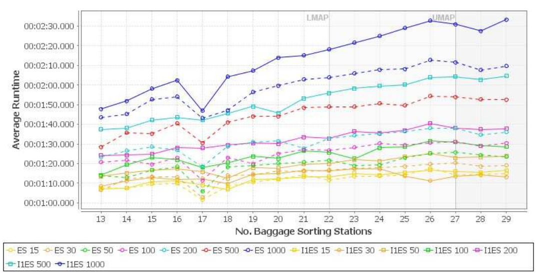


DSEMO





MEBPFNR3



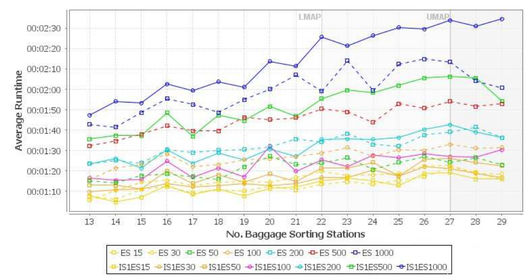
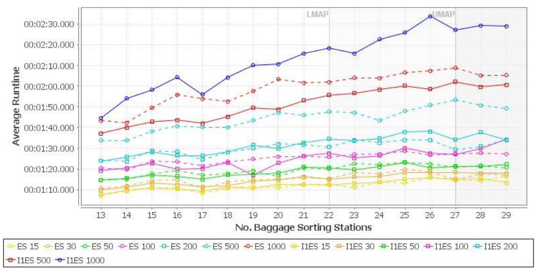
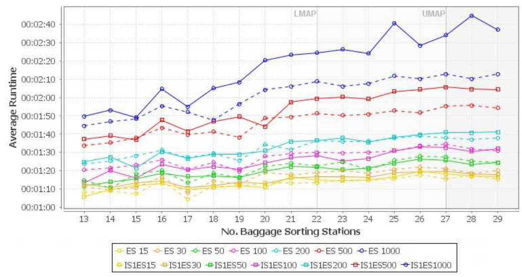


Table B.50: (a). 16^{th} December 2009 with 194 Flights.

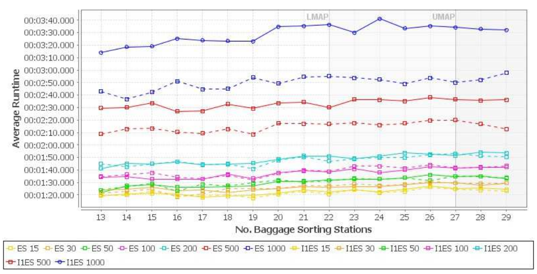
3-pier topology | 4-pier topology

MEFNR3





RMEFNR2



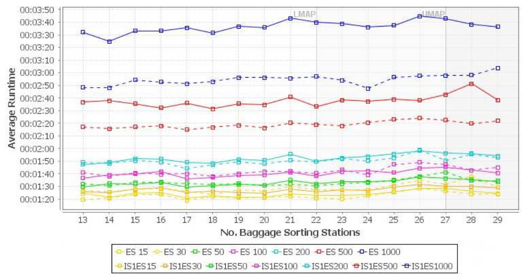
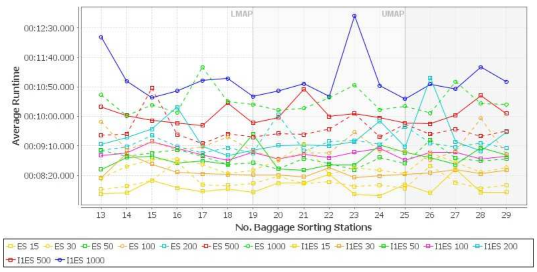
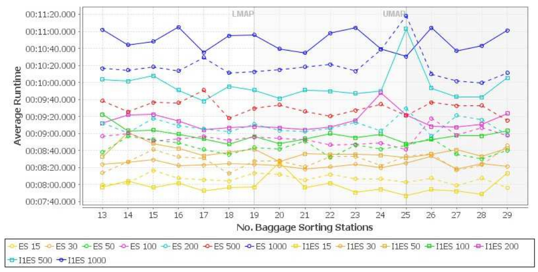


Table B.51: (b). 16^{th} December 2009 with 194 Flights.

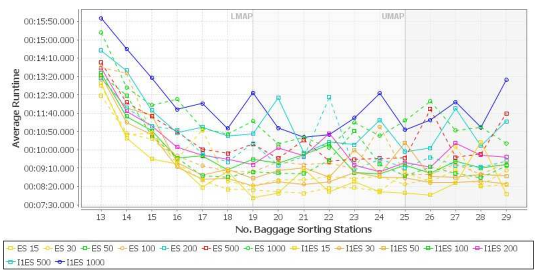
3-pier topology | 4-pier topology

C1P





C2P



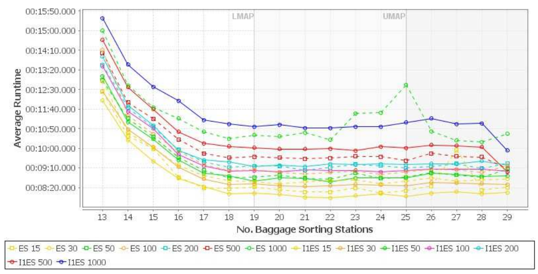
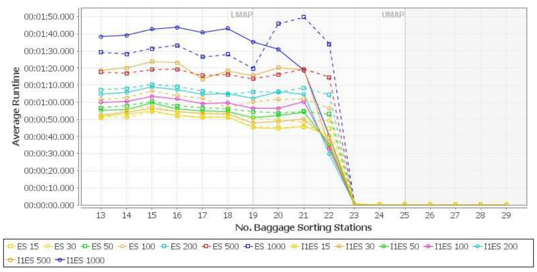
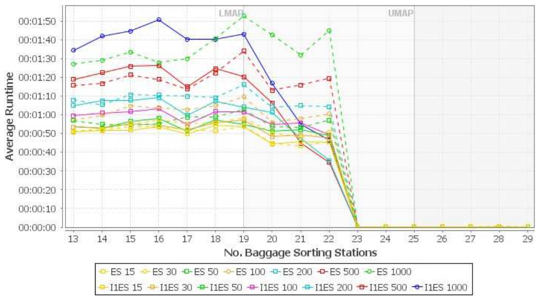


Table B.52: (a). 1^{st} March 2010 with 163 Flights.

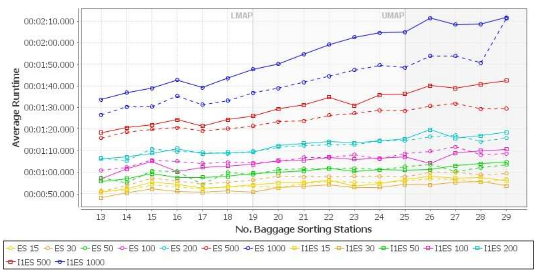
3-pier topology	4-pier topology

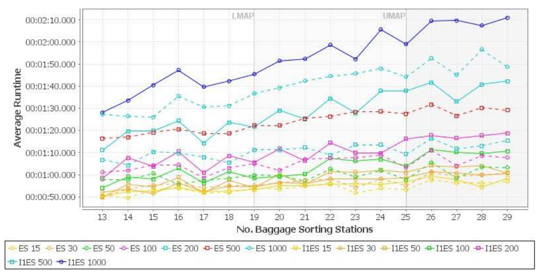
DSEMO



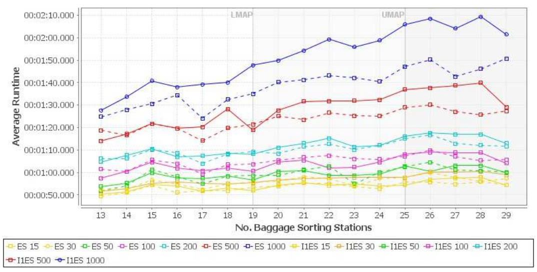


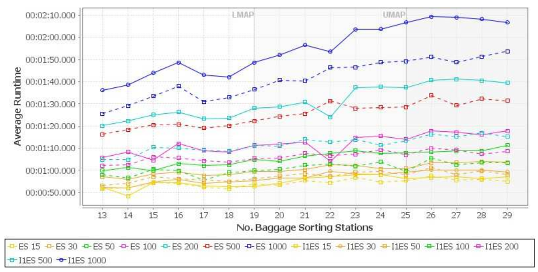
MEBPFNR3



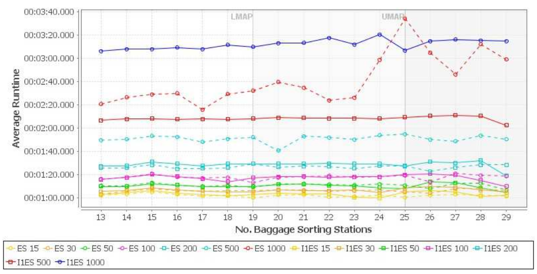


MEFNR3





RMEFNR2



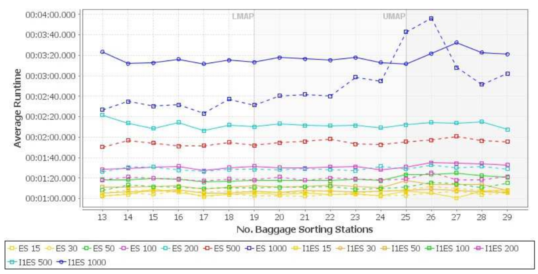


Table B.53: (b). 1^{st} March 2010 with 163 Flights.

B.3 Results for the Number of Iterations in a Generation

B.3.1 Graphical Representation of the Results

An image per considered operator and an image with all considered single operators showing the results for the different population sizes studied, per topology and data set. The replacement strategy I1ES is the same than IS1ES.

3-pier topology

16th December 2009 with 194 flights: Results in Figures B.9, B.10 B.11, B.12, B.13, B.14 and B.15.

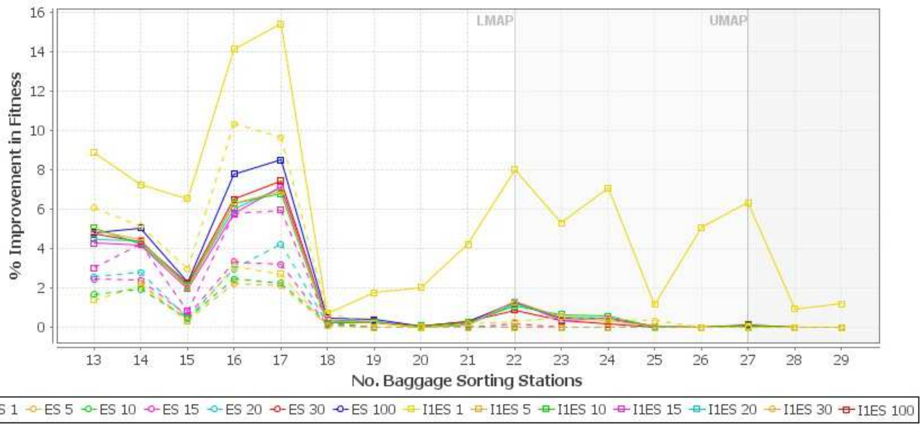


Figure B.9: 1-point crossover (C1P) with population size of 1000.

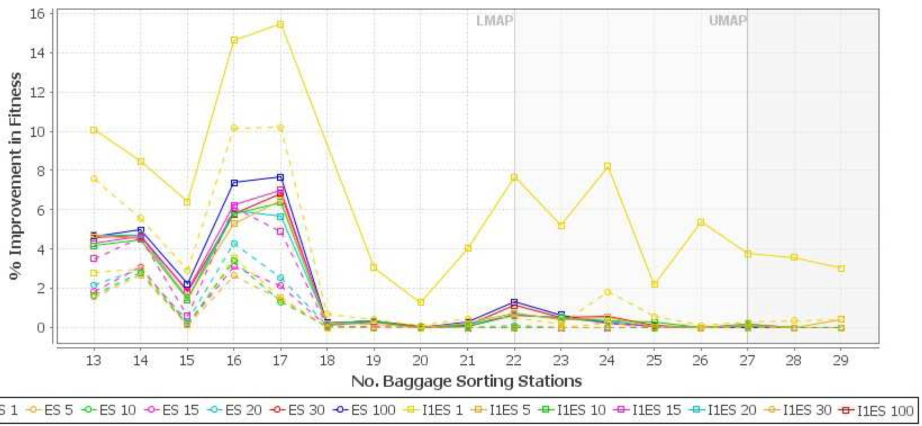


Figure B.10: 2-point crossover (C2P) with population size of 1000.

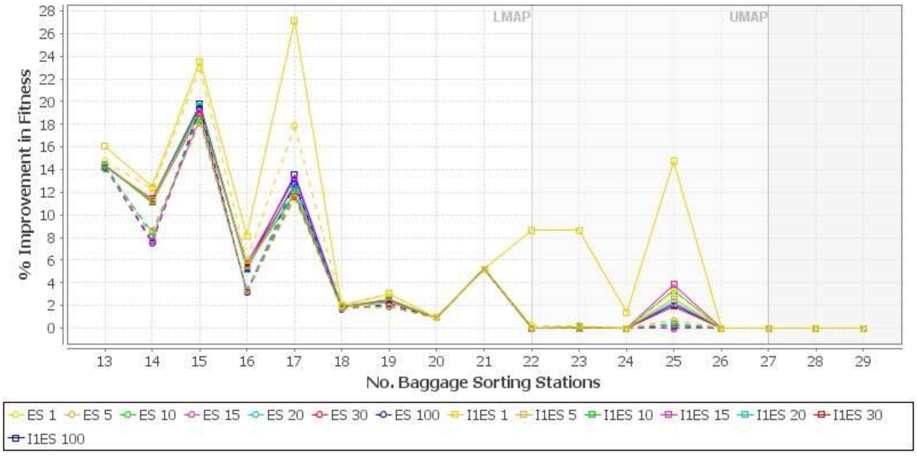


Figure B.11: DSEMO with population size of 1000.

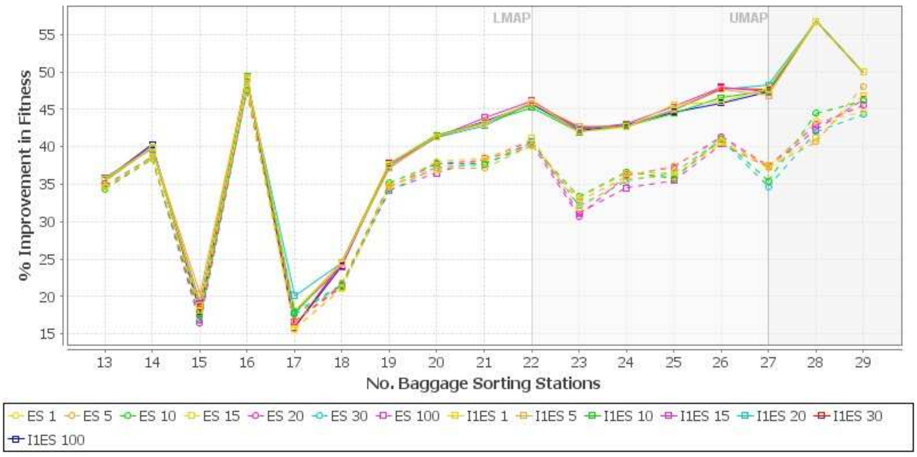


Figure B.12: MEBPFNR3 with population size of 15.

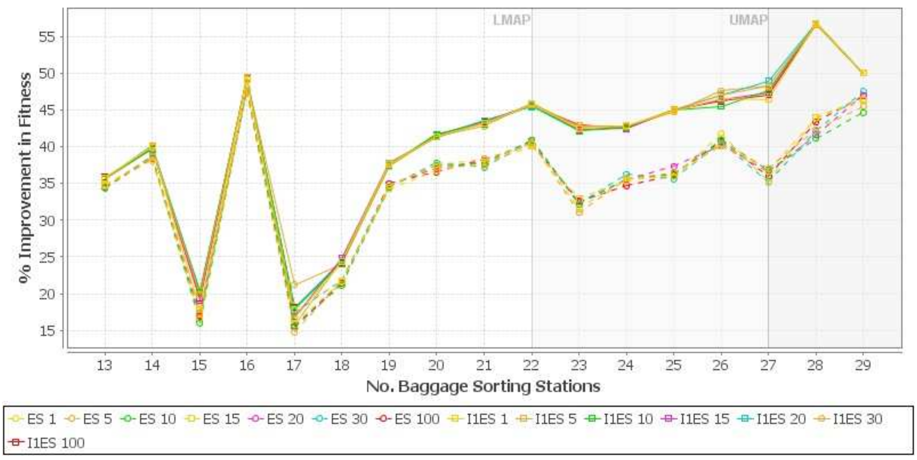


Figure B.13: MEFNR3 with population size of 15.

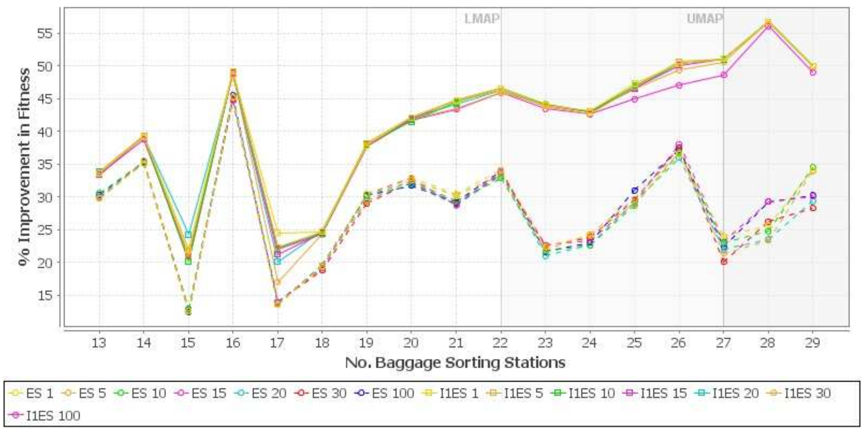


Figure B.14: RMEFNR2 with population size of 15.

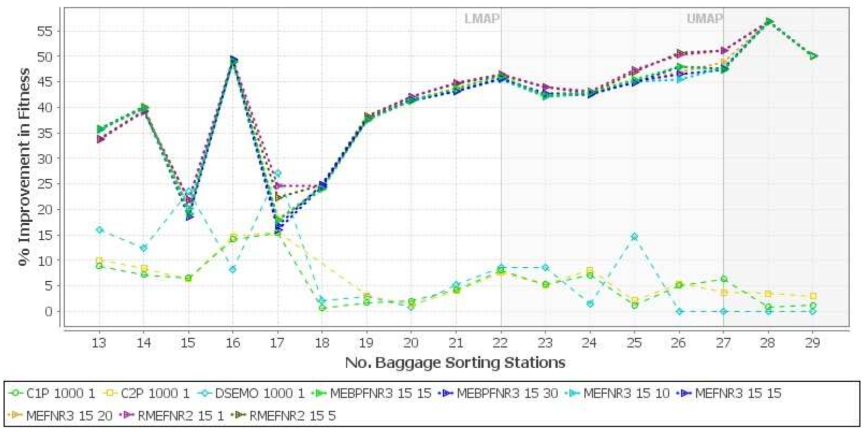


Figure B.15: IS1ES selector for operators with their best population size and ℓ.

1st *March 2010 with 163 flights*

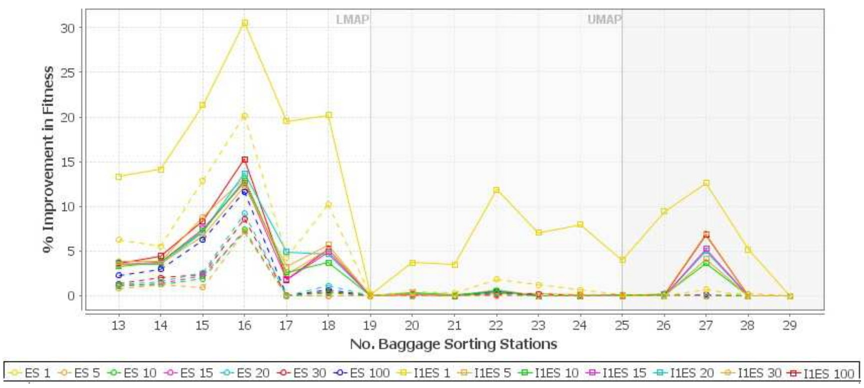


Figure B.16: 1-point crossover (C1P) with population size of 1000.

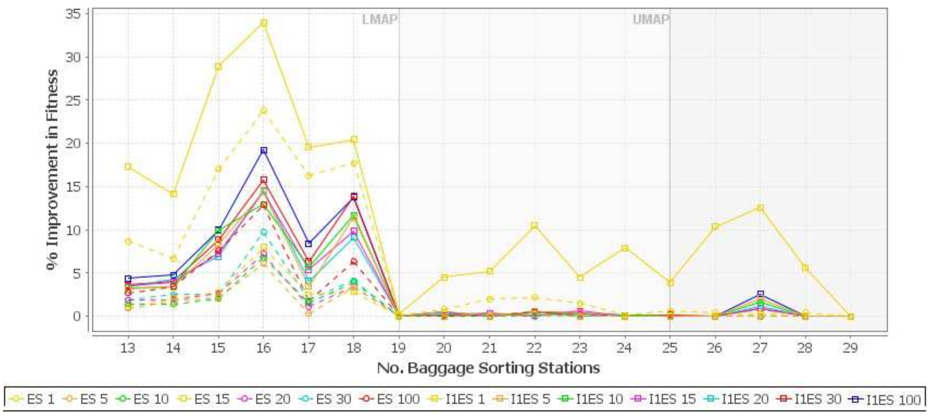


Figure B.17: 2-point crossover (C2P) with population size of 1000.

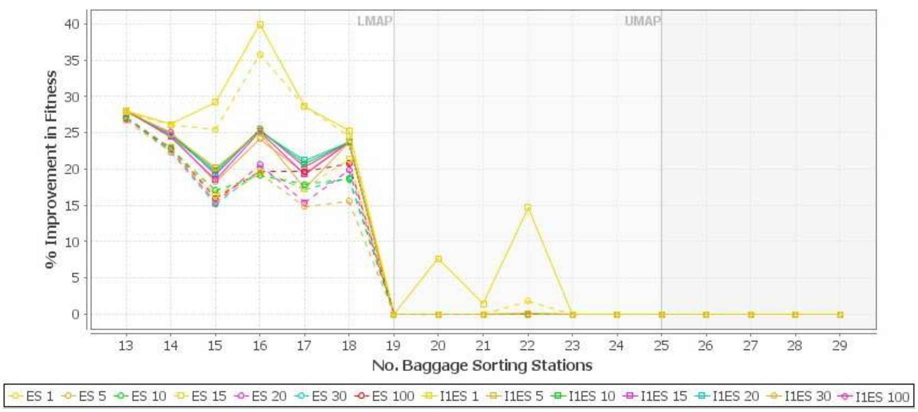


Figure B.18: DSEMO with population size of 1000.

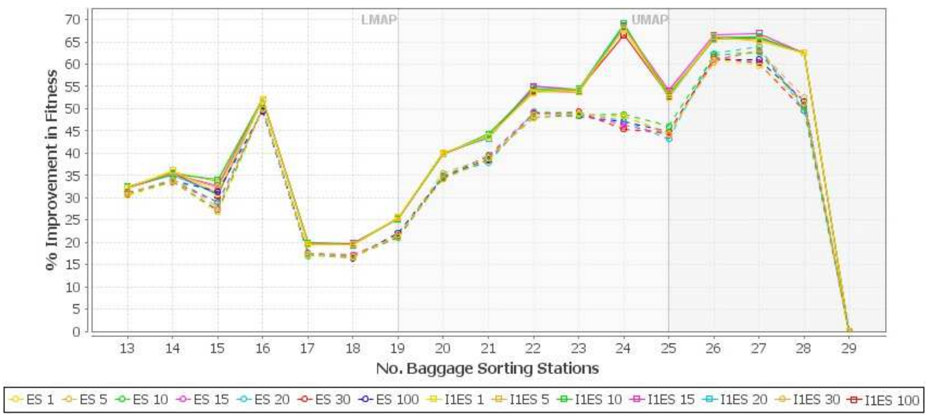


Figure B.19: MEBPFNR3 with population size of 15.

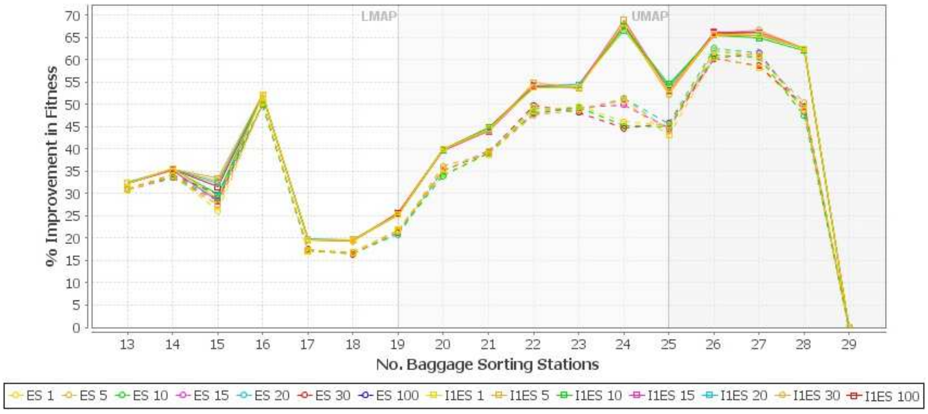


Figure B.20: MEFNR3 with population size of 15.

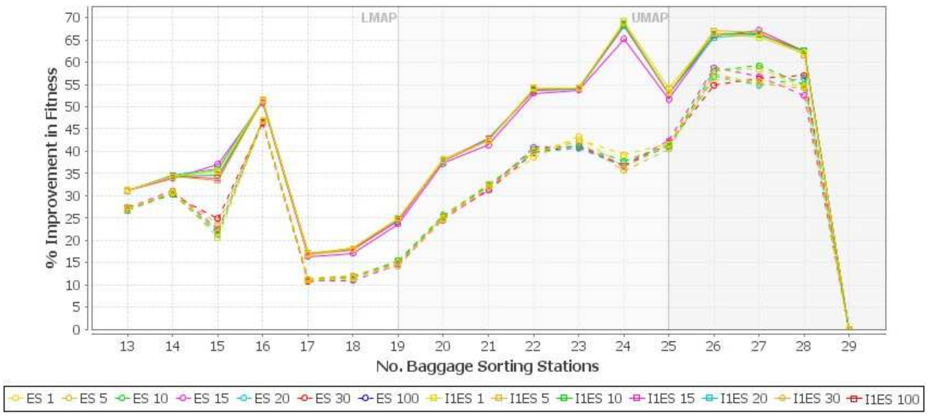


Figure B.21: RMEFNR2 with population size of 15.

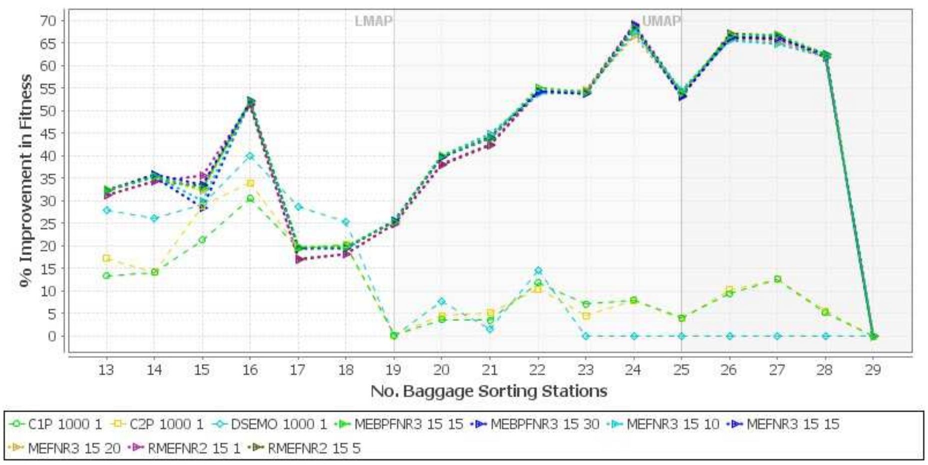


Figure B.22: IS1ES selector for operators with their best population size and ℓ.

4-pier topology

16^{th} December 2009 with 194 flights

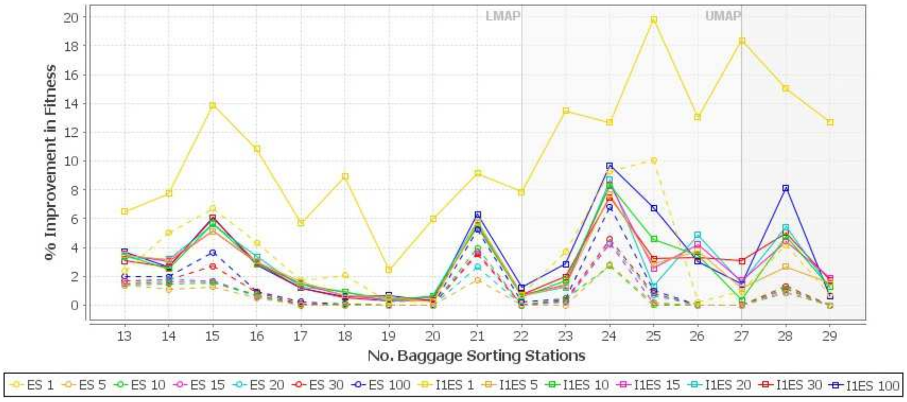


Figure B.23: 1-point crossover (C1P) with population size of 1000.

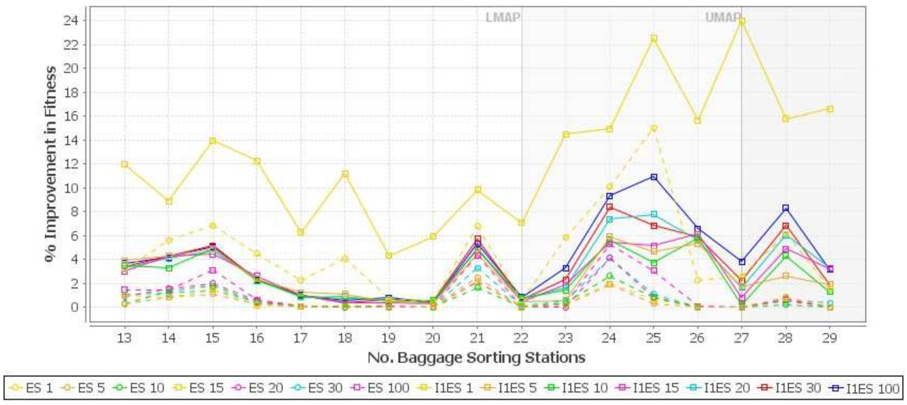


Figure B.24: 2-point crossover (C2P) with population size of 1000.

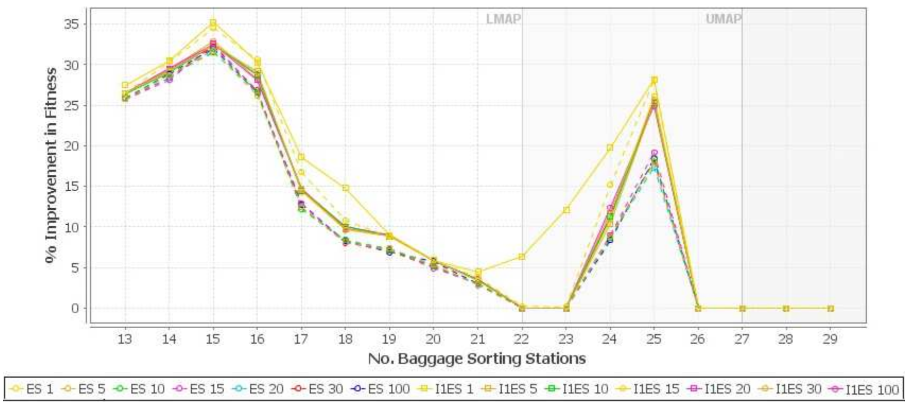


Figure B.25: DSEMO with population size of 1000.

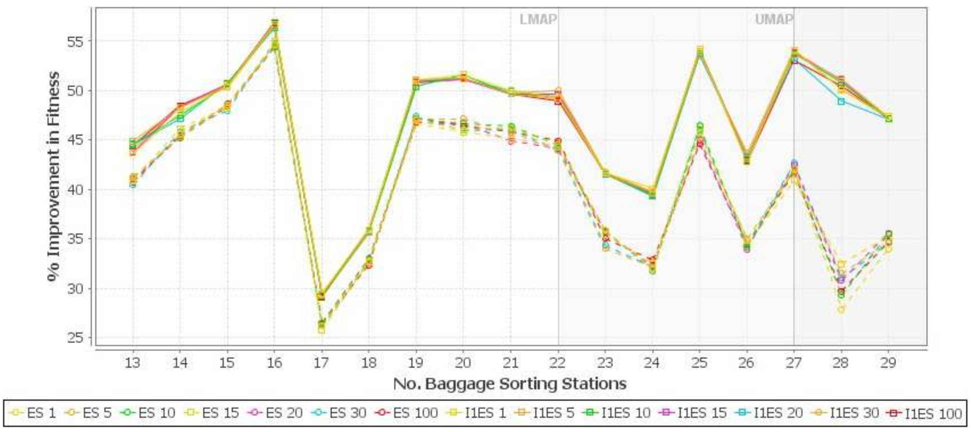


Figure B.26: MEBPFNR3 with population size of 15.

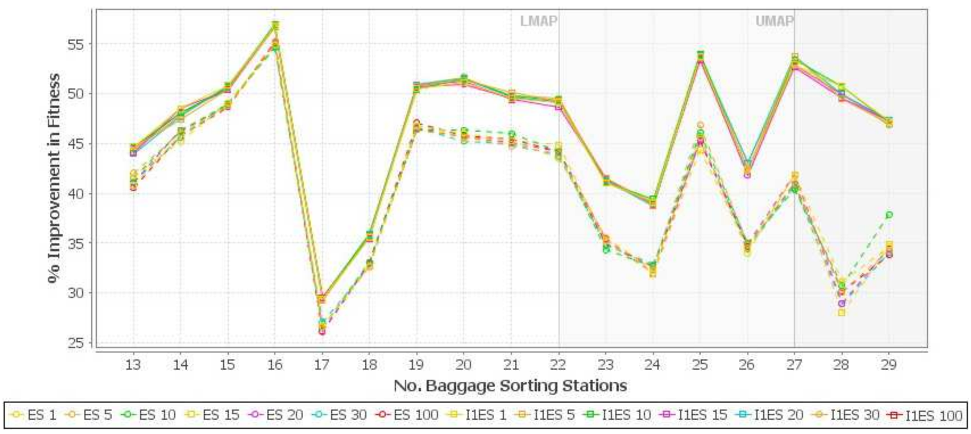


Figure B.27: MEBPFNR3 with population size of 15.

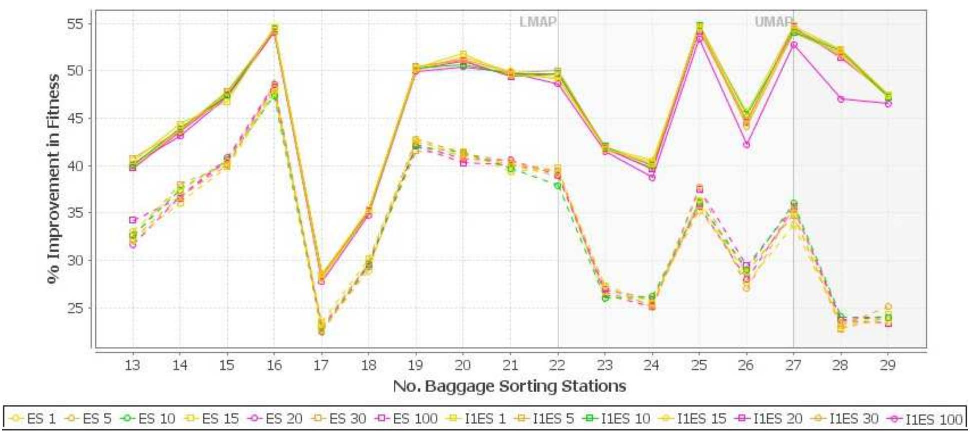


Figure B.28: RMEFNR2 with population size of 15.

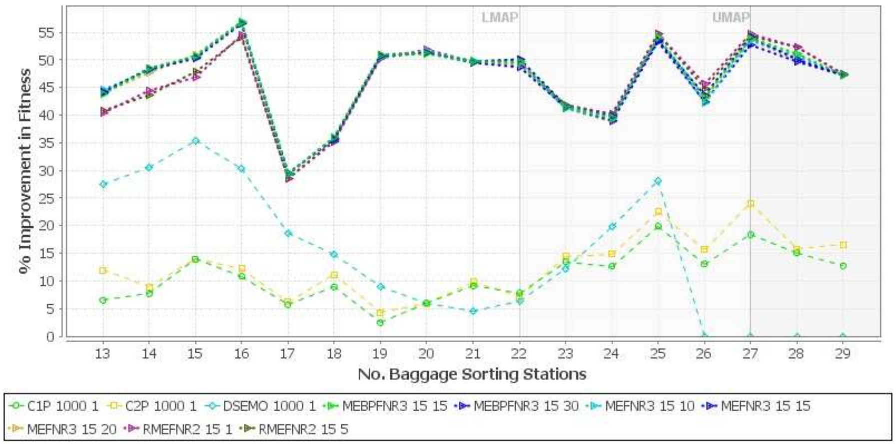


Figure B.29: IS1ES selector for operators with their best population size and ℓ.

1^{st} March 2010 with 163 flights

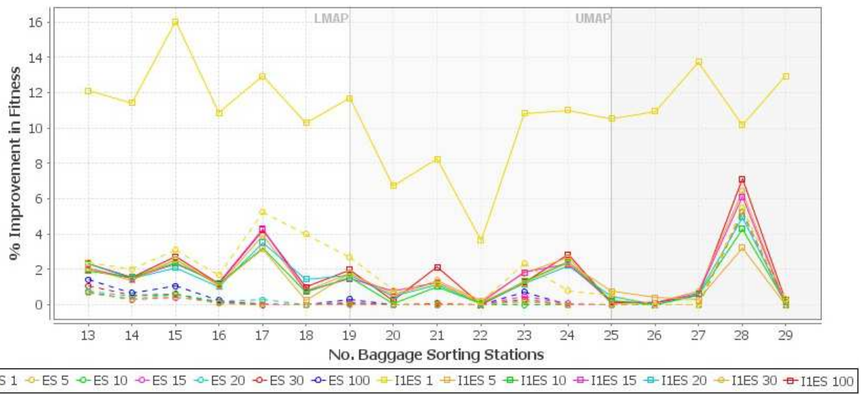


Figure B.30: 1-point crossover (C1P) with population size of 1000.

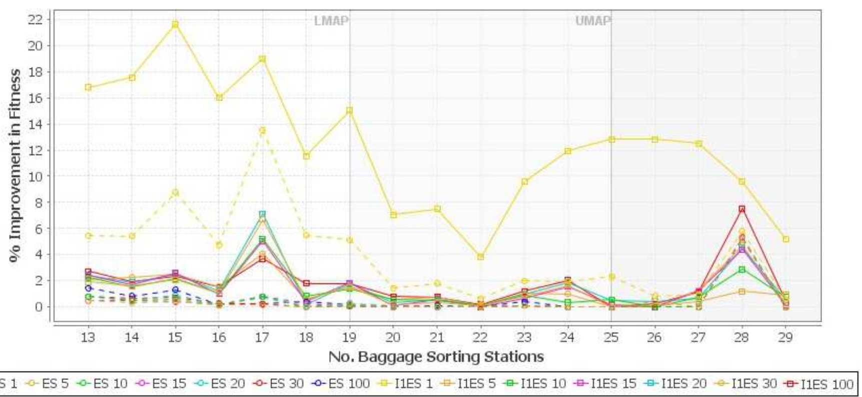


Figure B.31: 2-point crossover (C2P) with population size of 1000.

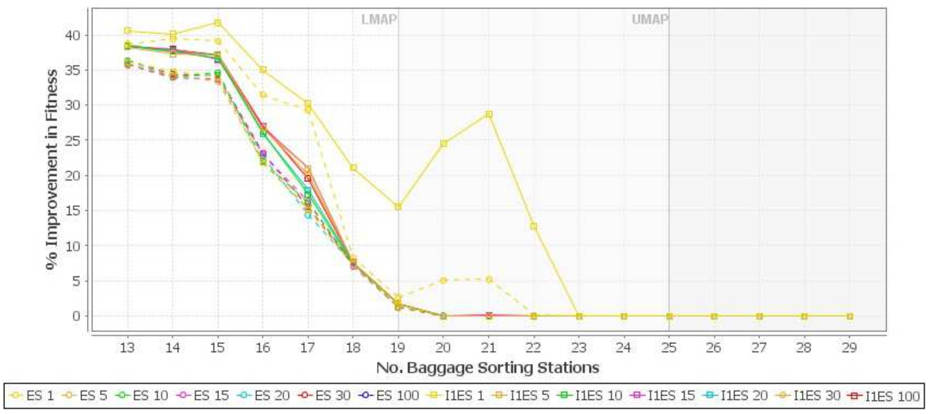


Figure B.32: DSEMO with population size of 1000.

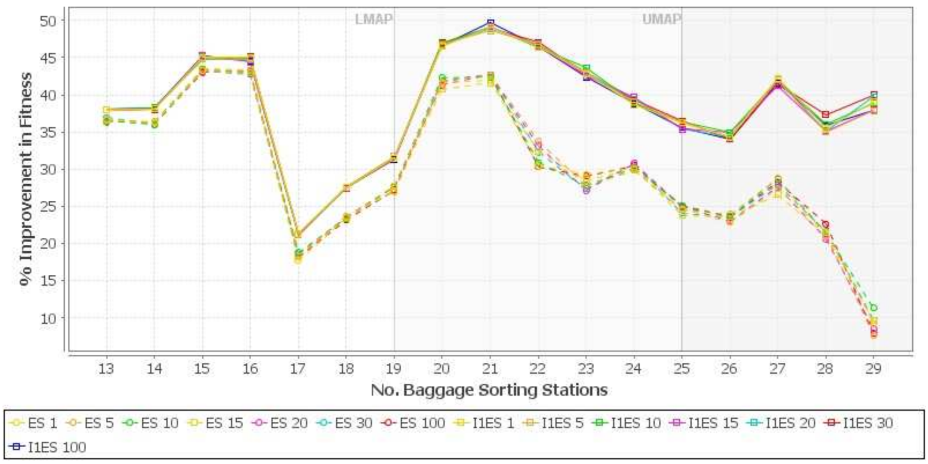


Figure B.33: MEBPFNR3 with population size of 15.

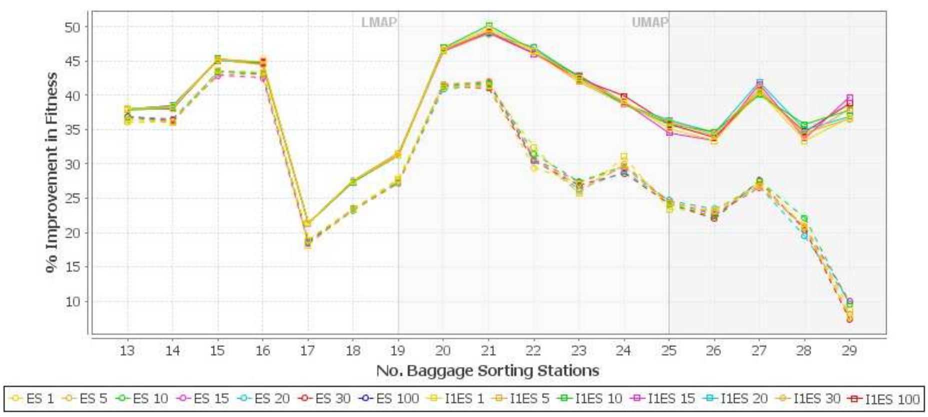


Figure B.34: MEFNR3 with population size of 15.

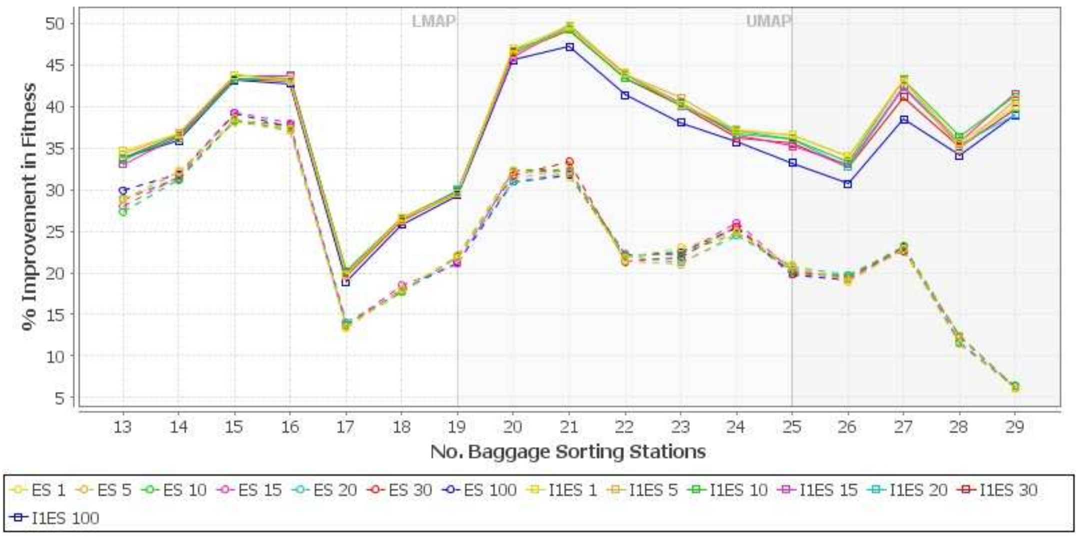


Figure B.35: RMEFNR2 with population size of 15.

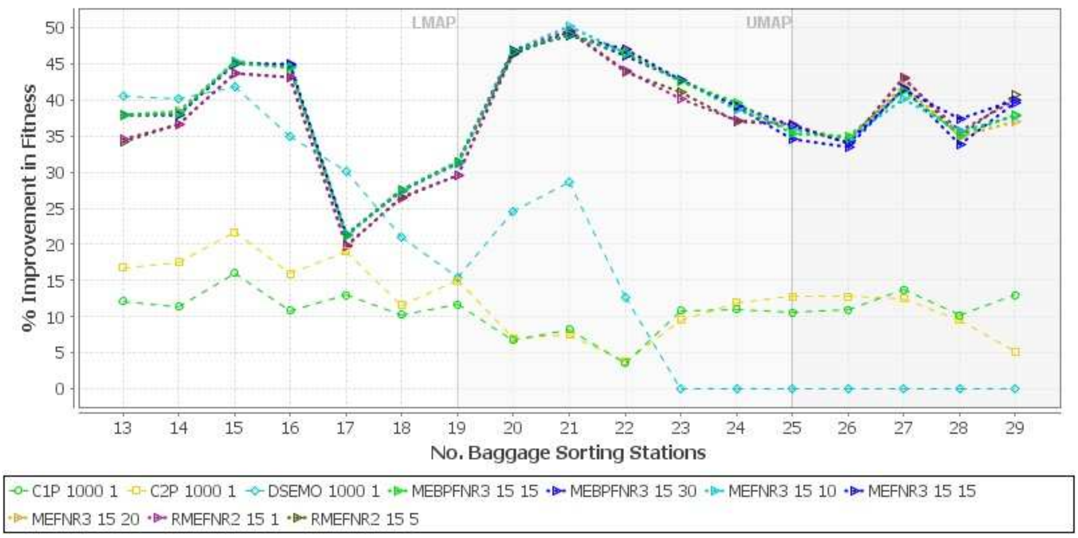


Figure B.36: IS1ES selector for operators with their best population size and ℓ.

B.3.2 Statistical Results

Summary tables of the Mann-Whitney tests run against the considered operators for the considered number of iterations.

3-pier topology

16^{th} December 2009 with 194 flights

Both C1P and C2P provide statistically significantly fitter solutions for all considered data sets, topologies and number of BSSs for $\ell = 1$, as shown in Tables B.54, B.55, B.56 and B.57.

13	**C1P 1000**	**I1ES**	**1**
14	**C1P 1000**	**I1ES**	**1**
15	**C1P 1000**	**I1ES**	**1**
16	**C1P 1000**	**I1ES**	**1**
17	**C1P 1000**	**I1ES**	**1**
18	**C1P 1000**	**ES**	**1**
		I1ES	**1**
19	**C1P 1000**	**I1ES**	**1**
20	**C1P 1000**	**I1ES**	**1**
21	**C1P 1000**	**I1ES**	**1**
22	**C1P 1000**	**I1ES**	**1**
23	**C1P 1000**	**I1ES**	**1**
24	**C1P 1000**	**I1ES**	**1**
25	**C1P 1000**	**I1ES**	**1**
26	**C1P 1000**	**I1ES**	**1**
27	**C1P 1000**	**I1ES**	**1**
28	**C1P 1000**	**I1ES**	**1**
29	**C1P 1000**	**I1ES**	**1**

Table B.54: 3-pier, 194 flights, 48 stands and C1P 1000.

Max. (9, 5, 3)	**1**	5	10	15	20
ES	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	**17** (*9, 5, 3*)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Max. (9, 5, 3)	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	0 (0, 0, 0)	0 (0, 0, 0)

Table B.55: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and C1P 1000 for significance level 0.05.

13	**C2P 1000**	**I1ES**	**1**
14	**C2P 1000**	**I1ES**	**1**
15	**C2P 1000**	**I1ES**	**1**
16	**C2P 1000**	**I1ES**	**1**
17	**C2P 1000**	**I1ES**	**1**
18	**C2P 1000**	**ES**	**1**
19	**C2P 1000**	**I1ES**	**1**
20	**C2P 1000**	**I1ES**	**1**
21	**C2P 1000**	**I1ES**	**1**
22	**C2P 1000**	**I1ES**	**1**
23	**C2P 1000**	**I1ES**	**1**
24	**C2P 1000**	**I1ES**	**1**
25	**C2P 1000**	**I1ES**	**1**
26	**C2P 1000**	**I1ES**	**1**
27	**C2P 1000**	**I1ES**	**1**
28	**C2P 1000**	**I1ES**	**1**
29	**C2P 1000**	**I1ES**	**1**

Table B.56: 3-pier, 194 flights, 48 stands and C2P 1000.

Max. (9, 5, 3)	**1**	5	10	15	20
ES	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	**16** (8, *5*, *3*)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Max. (9, 5, 3)	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	0 (0, 0, 0)	0 (0, 0, 0)

Table B.57: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and C2P 1000 for significance level 0.05.

Dummy Single Exchange Mutation Operator (DSEMO) provides statistically significantly fitter solutions for all considered data sets, topologies and number of BSSs for $\ell = 1$, as shown in Tables B.58 and B.59.

13	**DSEMO 1000**	**I1ES**	**1**
14	**DSEMO 1000**	**I1ES**	**1**
15	**DSEMO 1000**	**I1ES**	**1**
16	**DSEMO 1000**	**I1ES**	**1**
17	**DSEMO 1000**	**I1ES**	**1**
18	**DSEMO 1000**	**I1ES**	**1**
19	**DSEMO 1000**	**ES**	**1**
		I1ES	**1**
20	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
21	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
22	**DSEMO 1000**	**I1ES**	**1**
23	**DSEMO 1000**	**I1ES**	**1**
24	**DSEMO 1000**	**I1ES**	**1**
25	**DSEMO 1000**	**I1ES**	**1**
26	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
27	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
28	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
29	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**

Table B.58: 3-pier, 194 flights, 48 stands and DSEMO 1000.

Max. (9, 5, 3)	**1**	5	10	15	20
ES	7 (3, 1, *3*)	6 (2, 1, *3*)	6 (2, 1, *3*)	6 (2, 1, *3*)	6 (2, 1, *3*)
I1ES	**17** (*9*, *5*, *3*)	6 (2, 1, *3*)	6 (2, 1, *3*)	6 (2, 1, *3*)	6 (2, 1, *3*)

Max. (9, 5, 3)	30	100
ES	6 (2, 1, *3*)	6 (2, 1, *3*)
I1ES	6 (2, 1, *3*)	6 (2, 1, *3*)

Table B.59: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and DSEMO 1000 for significance level 0.05.

13	**MEBPFNR3 15**	**I1ES**	**1, 5, 15, 20, 30**
14	**MEBPFNR3 15**	**I1ES**	**100**
15	**MEBPFNR3 15**	**I1ES**	**1, 5, 15, 20**
16	**MEBPFNR3 15**	**I1ES**	**1, 10, 30, 100**
17	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
18	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 20, 30**
19	**MEBPFNR3 15**	**I1ES**	**1, 10, 15, 20, 30, 100**
20	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
21	**MEBPFNR3 15**	**I1ES**	**15, 100**
22	**MEBPFNR3 15**	**I1ES**	**1, 15**
23	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
24	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
25	**MEBPFNR3 15**	**I1ES**	**1, 5, 15, 20, 30**
26	**MEBPFNR3 15**	**I1ES**	**5, 10, 15, 20, 30**
27	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
28	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
29	**MEBPFNR3 15**	**I1ES**	**1, 10, 15, 20, 30, 100**

Table B.60: 3-pier, 194 flights, 48 stands and MEBPFNR3 15.

Max. (9, 5, 3)	**1**	5	10	**15**	20
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	**14** (7, 4, *3*)	11 (5, 4, 2)	11 (5, 3, *3*)	**14** (6, *5*, *3*)	13 (6, 4, *3*)

Max. (9, 5, 3)	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	13 (6, 4, *3*)	11 (6, 2, *3*)

Table B.61: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and MEBPFNR3 15 for significance 0.05.

13	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
14	**MEFNR3 15**	**I1ES**	**1, 5, 15, 20**
15	**MEFNR3 15**	**I1ES**	**5, 10, 15, 30**
16	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
17	**MEFNR3 15**	**I1ES**	**1, 10, 15, 20, 30, 100**
18	**MEFNR3 15**	**I1ES**	**1, 5, 15, 20**
19	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20**
20	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
21	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 100**
22	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
23	**MEFNR3 15**	**I1ES**	**1, 5, 15, 20, 30, 100**
24	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
25	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
26	**MEFNR3 15**	**I1ES**	**1, 5, 15, 20, 30, 100**
27	**MEFNR3 15**	**I1ES**	**5, 10, 15, 20, 30**
28	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
29	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**

Table B.62: 3-pier, 194 flights, 48 stands and MEFNR3 15.

Max. (9, 5, 3)	1	5	10	**15**	20	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	15 (8, *5*, 2)	16 (8, *5*, *3*)	13 (7, 3, *3*)	**17** (*9*, *5*, *3*)	16 (8, *5*, *3*)	13 (5, *5*, *3*)	12 (5, *5*, 2)

Table B.63: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and MEFNR3 15 for significance level 0.05.

13	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 20, 30, 100**
14	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
15	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 20, 30, 100**
16	**RMEFNR2 15**	**I1ES**	**1, 10, 15, 30**
17	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
18	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20**
19	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
20	**RMEFNR2 15**	**I1ES**	**1, 5, 15, 20, 30**
21	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20**
22	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20**
23	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
24	**RMEFNR2 15**	**I1ES**	**1, 10, 15, 20, 30**
25	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
26	**RMEFNR2 15**	**I1ES**	**1, 5, 15, 20**
27	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
28	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
29	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**

Table B.64: 3-pier, 194 flights, 48 stands and RMEFNR2 15.

Max. (9, 5, 3)	**1**	5	10	15	20
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	**17** (*9*, *5*, *3*)	15 (8, 4, *3*)	15 (8, 4, *3*)	15 (7, *5*, *3*)	16 (8, *5*, *3*)

Max. (9, 5, 3)	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	13 (7, 3, *3*)	4 (4, 0, 0)

Table B.65: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 194 flights and RMEFNR2 15 for significance level of 0.05.

13	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
14	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**15, 20**
15	**DSEMO 1000**	**1**
16	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
	RMEFNR2 15	**1**
17	**DSEMO 1000**	**1**
18	**MEBPFNR3 15**	**30**
	MEFNR3 15	**15, 20**
	RMEFNR2 15	**1, 5**
19	**RMEFNR2 15**	**1, 5**
20	**RMEFNR2 15**	**1, 5**
21	**RMEFNR2 15**	**1, 5**
22	**MEBPFNR3 15**	**15**
	RMEFNR2 15	**1, 5**
23	**RMEFNR2 15**	**1, 5**
24	**RMEFNR2 15**	**1**
25	**RMEFNR2 15**	**1, 5**
26	**RMEFNR2 15**	**1, 5**
27	**RMEFNR2 15**	**1, 5**
28	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
	RMEFNR2 15	**1, 5**
29	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
	RMEFNR2 15	**1, 5**

Table B.66: 194 flights, 3-pier and 48 stands.

Max. (9, 5, 3)	**1**	5	10	15	20	30
C1P 1000	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P 1000	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 1000	2 (2, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 15	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	6 (3, 1, 2)	0 (0, 0, 0)	6 (4, 0, 2)
MEFNR3 15	0 (0, 0, 0)	0 (0, 0, 0)	4 (2, 0, 2)	6 (4, 0, 2)	6 (4, 0, 2)	0 (0, 0, 0)
RMEFNR2 15	**13** (5, *5*, *3*)	11 (4, 4, *3*)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.67: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 194 flights for significance level 0.05.

1^{st} March 2010 with 163 flights

Both C1P and C2P provide statistically significantly fitter solutions for all considered data sets, topologies and number of BSSs for $\ell = 1$, as shown in Tables B.68, B.69, B.70 and B.71.

13	**C1P 1000**	**I1ES**	**1**
14	**C1P 1000**	**I1ES**	**1**
15	**C1P 1000**	**I1ES**	**1**
16	**C1P 1000**	**I1ES**	**1**
17	**C1P 1000**	**I1ES**	**1**
18	**C1P 1000**	**I1ES**	**1**
19	**C1P 1000**	**I1ES**	**1**
20	**C1P 1000**	**I1ES**	**1**
21	**C1P 1000**	**I1ES**	**1**
22	**C1P 1000**	**I1ES**	**1**
23	**C1P 1000**	**I1ES**	**1**
24	**C1P 1000**	**I1ES**	**1**
25	**C1P 1000**	**I1ES**	**1**
26	**C1P 1000**	**I1ES**	**1**
27	**C1P 1000**	**I1ES**	**1**
28	**C1P 1000**	**I1ES**	**1**
29	**C1P 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**

Table B.68: 3-pier, 163 flights, 48 stands and C1P 1000.

Max. (6, 6, 5)	**1**	5	10	15	20
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
I1ES	**17** (*6*, *6*, *5*)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)

Max. (6, 6, 5)	30	100
ES	1 (0, 0, 1)	1 (0, 0, 1)
I1ES	1 (0, 0, 1)	1 (0, 0, 1)

Table B.69: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and C1P 1000 for significance level 0.05.

13	**C2P 1000**	**I1ES**	**1**
14	**C2P 1000**	**I1ES**	**1**
15	**C2P 1000**	**I1ES**	**1**
16	**C2P 1000**	**I1ES**	**1**
17	**C2P 1000**	**I1ES**	**1**
18	**C2P 1000**	**I1ES**	**1**
19	**C2P 1000**	**I1ES**	**1**
20	**C2P 1000**	**I1ES**	**1**
21	**C2P 1000**	**I1ES**	**1**
22	**C2P 1000**	**I1ES**	**1**
23	**C2P 1000**	**I1ES**	**1**
24	**C2P 1000**	**I1ES**	**1**
25	**C2P 1000**	**I1ES**	**1**
26	**C2P 1000**	**I1ES**	**1**
27	**C2P 1000**	**I1ES**	**1**
28	**C2P 1000**	**I1ES**	**1**
29	**C2P 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**

Table B.70: 3-pier, 163 flights, 48 stands and C2P 1000.

Max. (6, 6, 5)	**1**	5	10	15	20
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
I1ES	**17** (*6*, *6*, *5*)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)

Max. (6, 6, 5)	30	100
ES	1 (0, 0, 1)	1 (0, 0, 1)
I1ES	1 (0, 0, 1)	1 (0, 0, 1)

Table B.71: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and C2P 1000 for significance level 0.05.

DSEMO provides statistically significantly fitter solutions for all considered data sets, topologies and number of BSSs for $\ell = 1$, as shown in Tables B.72 and B.73.

13	**DSEMO 1000**	**ES**	**1**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
14	**DSEMO 1000**	**ES**	**1**
		I1ES	**1**
15	**DSEMO 1000**	**I1ES**	**1**
16	**DSEMO 1000**	**I1ES**	**1**
17	**DSEMO 1000**	**ES**	**1**
		I1ES	**1**
18	**DSEMO 1000**	**I1ES**	**1**
19	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
20	**DSEMO 1000**	**I1ES**	**1**
21	**DSEMO 1000**	**I1ES**	**1**
22	**DSEMO 1000**	**I1ES**	**1**
23	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
24	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
25	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
26	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
27	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
28	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
29	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**

Table B.72: 3-pier, 163 flights, 48 stands and DSEMO 1000.

Max. (6, 6, 5)	**1**	5	10	15	20
ES	11 (3, 3, *5*)	8 (0, 3, *5*)	8 (0, 3, *5*)	8 (0, 3, *5*)	8 (0, 3, *5*)
I1ES	**17** (*6*, *6*, *5*)	9 (1, 3, *5*)	9 (1, 3, *5*)	9 (1, 3, *5*)	9 (1, 3, *5*)

Max. (6, 6, 5)	30	100
ES	8 (0, 3, *5*)	8 (0, 3, *5*)
I1ES	9 (1, 3, *5*)	9 (1, 3, *5*)

Table B.73: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and DSEMO 1000 for significance level 0.05.

13	**MEBPFNR3 15**	**I1ES**	**5, 10, 30, 100**
14	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 30, 100**
15	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
16	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
17	**MEBPFNR3 15**	**I1ES**	**10, 15, 20**
18	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
19	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
20	**MEBPFNR3 15**	**I1ES**	**1, 5, 15, 20, 100**
21	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 30, 100**
22	**MEBPFNR3 15**	**I1ES**	**10, 15, 20, 30, 100**
23	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
24	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
25	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 30, 100**
26	**MEBPFNR3 15**	**I1ES**	**1, 10, 15, 20, 30, 100**
27	**MEBPFNR3 15**	**I1ES**	**10, 15, 30**
28	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
29	**MEBPFNR3 15**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**

Table B.74: 3-pier, 163 flights, 48 stands and MEBPFNR3 15.

Max. (6, 6, 5)	1	5	**10**	**15**	20
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
I1ES	13 (4, 5, 4)	13 (5, 5, 3)	**16** (*6*, 5, *5*)	**16** (5, *6*, *5*)	12 (4, 5, 3)

Max. (6, 6, 5)	30	100
ES	1 (0, 0, 1)	1 (0, 0, 1)
I1ES	15 (5, 5, *5*)	13 (4, 5, 4)

Table B.75: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and MEBPFNR3 15 for significance level 0.05.

13	**MEFNR3 15**	**I1ES**	**5, 15, 20, 30, 100**
14	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
15	**MEFNR3 15**	**I1ES**	**5, 10, 20, 30, 100**
16	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
17	**MEFNR3 15**	**I1ES**	**5, 10, 20, 100**
18	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 100**
19	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
20	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
21	**MEFNR3 15**	**I1ES**	**1, 10, 20, 30, 100**
22	**MEFNR3 15**	**I1ES**	**5, 15, 20, 30, 100**
23	**MEFNR3 15**	**I1ES**	**1, 10, 15, 20, 30, 100**
24	**MEFNR3 15**	**I1ES**	**5, 10, 15, 30**
25	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 100**
26	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
27	**MEFNR3 15**	**I1ES**	**1, 15, 20, 30, 100**
28	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
29	**MEFNR3 15**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**

Table B.76: 3-pier, 163 flights, 48 stands and MEFNR3 15.

Max. (6, 6, 5)	1	5	10	15	**20**	30	**100**
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
I1ES	12 (3, 4, *5*)	14 (*6*, 4, 4)	14 (5, 5, 4)	14 (4, 5, *5*)	**16** (*6*, 5, *5*)	14 (4, *6*, 4)	**16** (*6*, 5, *5*)

Table B.77: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and MEFNR3 15 for significance level 0.05.

13	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
14	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
15	**RMEFNR2 15**	**I1ES**	**1, 10, 15, 20, 30, 100**
16	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
17	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20**
18	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
19	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 20, 30**
20	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15**
21	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
22	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
23	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
24	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20**
25	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 30**
26	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 30**
27	**RMEFNR2 15**	**I1ES**	**5, 10, 15, 30, 100**
28	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
29	**RMEFNR2 15**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**

Table B.78: 3-pier, 163 flights, 48 stands and RMEFNR2 15.

Max. (6, 6, 5)	1	5	**10**	15	20
ES	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
I1ES	16 (*6*, *6*, 4)	16 (5, *6*, *5*)	**17** (*6*, *6*, *5*)	16 (*6*, 5, *5*)	13 (*6*, 5, 2)

Max. (6, 6, 5)	30	100
ES	1 (0, 0, 1)	1 (0, 0, 1)
I1ES	14 (5, 4, *5*)	6 (2, 1, 3)

Table B.79: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology, 163 flights and RMEFNR2 15 for significance level 0.05.

13	**MEBPFNR3 15**	**30**
	MEFNR3 15	**10, 15, 20**
14	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
15	**C2P 1000**	**1**
	DSEMO 1000	**1**
	MEBPFNR3 15	**15, 30**
	MEFNR3 15	**20**
	RMEFNR2 15	**1**
16	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
17	**DSEMO 1000**	**1**
18	**DSEMO 1000**	**1**
19	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
20	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
21	**MEFNR3 15**	**10, 20**
22	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**15, 20**
23	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
	RMEFNR2 15	**1, 5**

24	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15**
	RMEFNR2 15	**1, 5**
25	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
	RMEFNR2 15	**1**
26	**MEBPFNR3 15**	**15**
	MEFNR3 15	**15**
	RMEFNR2 15	**1, 5**
27	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**15,**
	RMEFNR2 15	**5**
28	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
	RMEFNR2 15	**1, 5**
29	**C1P 1000**	**1**
	C2P 1000	**1**
	DSEMO 1000	**1**
	MEBPFNR3 15	**15, 30**
	MEFNR3 15	**10, 15, 20**
	RMEFNR2 15	**1, 5**

Table B.80: 163 flights, 3-pier and 48 stands.

Max. (6, 6, 5)	1	5	10	**15**	**20**	**30**
C1P 1000	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P 1000	2 (1, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 1000	4 (3, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 15	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	**13** (3, 5, *5*)	**0** (0, 0, 0)	**13** (4, 5, 4)
MEFNR3 15	0 (0, 0, 0)	0 (0, 0, 0)	11 (3, 5, 3)	**13** (3, 5, *5*)	**13** (4, 5, 4)	**0** (0, 0, 0)
RMEFNR2 15	7 (1, 2, 4)	6 (0, 2, 4)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.81: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 163 flights for significance level 0.05.

4-pier topology

16^{th} December 2009 with 194 flights

Both C1P and C2P provide statistically significantly fitter solutions for all considered data sets, topologies and number of BSSs for $\ell = 1$, as shown in Tables B.82, B.83, B.84 and B.85.

13	**C1P 1000**	**I1ES**	**1**
14	**C1P 1000**	**I1ES**	**1**
15	**C1P 1000**	**I1ES**	**1**
16	**C1P 1000**	**I1ES**	**1**
17	**C1P 1000**	**I1ES**	**1**
18	**C1P 1000**	**I1ES**	**1**
19	**C1P 1000**	**I1ES**	**1**
20	**C1P 1000**	**I1ES**	**1**
21	**C1P 1000**	**I1ES**	**1**
22	**C1P 1000**	**I1ES**	**1**
23	**C1P 1000**	**I1ES**	**1**
24	**C1P 1000**	**I1ES**	**1**
25	**C1P 1000**	**I1ES**	**1**
26	**C1P 1000**	**I1ES**	**1**
27	**C1P 1000**	**I1ES**	**1**
28	**C1P 1000**	**I1ES**	**1**
29	**C1P 1000**	**I1ES**	**1**

Table B.82: 4-pier, 194 flights, 46 stands and C1P 1000.

Max. (9, 5, 3)	**1**	5	10	15	20
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	**17** (*9, 5, 3*)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Max. (9, 5, 3)	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	0 (0, 0, 0)	0 (0, 0, 0)

Table B.83: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and C1P 1000 for significance level 0.05.

13	**C2P 1000**	**I1ES**	**1**
14	**C2P 1000**	**I1ES**	**1**
15	**C2P 1000**	**I1ES**	**1**
16	**C2P 1000**	**I1ES**	**1**
17	**C2P 1000**	**I1ES**	**1**
18	**C2P 1000**	**I1ES**	**1**
19	**C2P 1000**	**I1ES**	**1**
20	**C2P 1000**	**I1ES**	**1**
21	**C2P 1000**	**I1ES**	**1**
22	**C2P 1000**	**I1ES**	**1**
23	**C2P 1000**	**I1ES**	**1**
24	**C2P 1000**	**I1ES**	**1**
25	**C2P 1000**	**I1ES**	**1**
26	**C2P 1000**	**I1ES**	**1**
27	**C2P 1000**	**I1ES**	**1**
28	**C2P 1000**	**I1ES**	**1**
29	**C2P 1000**	**I1ES**	**1**

Table B.84: 4-pier, 194 flights, 46 stands and C2P 1000.

Max. (9, 5, 3)	**1**	5	10	15	20
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	**17** (*9*, *5*, *3*)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Max. (9, 5, 3)	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	0 (0, 0, 0)	0 (0, 0, 0)

Table B.85: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and C2P 1000 for significance level 0.05.

DSEMO provides statistically significantly fitter solutions for all considered data sets, topologies and number of BSSs for $\ell = 1$, as shown in Tables B.86 and B.87.

13	**DSEMO 1000**	**I1ES**	**1**
14	**DSEMO 1000**	**I1ES**	**1**
15	**DSEMO 1000**	**I1ES**	**1**
16	**DSEMO 1000**	**I1ES**	**1**
17	**DSEMO 1000**	**I1ES**	**1**
18	**DSEMO 1000**	**I1ES**	**1**
19	**DSEMO 1000**	**I1ES**	**1**, **100**
20	**DSEMO 1000**	**ES**	**1**
		I1ES	**1**, **10**, **15**, **20**, **30**, **100**
21	**DSEMO 1000**	**I1ES**	**1**
22	**DSEMO 1000**	**I1ES**	**1**
23	**DSEMO 1000**	**I1ES**	**1**
24	**DSEMO 1000**	**I1ES**	**1**
25	**DSEMO 1000**	**I1ES**	**1**
26	**DSEMO 1000**	**ES**	**1**, **5**, **10**, **15**, **20**, **30**, **100**
		I1ES	**1**, **5**, **10**, **15**, **20**, **30**, **100**
27	**DSEMO 1000**	**ES**	**1**, **5**, **10**, **15**, **20**, **30**, **100**
		I1ES	**1**, **5**, **10**, **15**, **20**, **30**, **100**
28	**DSEMO 1000**	**ES**	**1**, **5**, **10**, **15**, **20**, **30**, **100**
		I1ES	**1**, **5**, **10**, **15**, **20**, **30**, **100**
29	**DSEMO 1000**	**ES**	**1**, **5**, **10**, **15**, **20**, **30**, **100**
		I1ES	**1**, **5**, **10**, **15**, **20**, **30**, **100**

Table B.86: 4-pier, 194 flights, 46 stands and DSEMO 1000.

Max. (9, 5, 3)	**1**	5	10	15	20
ES	5 (1, 1, *3*)	4 (0, 1, *3*)	4 (0, 1, *3*)	4 (0, 1, *3*)	4 (0, 1, *3*)
I1ES	**17** (*9*, *5*, *3*)	4 (0, 1, *3*)	5 (1, 1, *3*)	5 (1, 1, *3*)	5 (1, 1, *3*)

Max. (9, 5, 3)	30	100
ES	4 (0, 1, *3*)	4 (0, 1, *3*)
I1ES	5 (1, 1, *3*)	6 (2, 1, *3*)

Table B.87: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and DSEMO 1000 for significance level 0.05.

13	**MEBPFNR3 15**	**I1ES**	**5, 10, 20, 30, 100**
14	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
15	**MEBPFNR3 15**	**I1ES**	**5, 10, 15, 20, 30, 100**
16	**MEBPFNR3 15**	**I1ES**	**1, 15, 100**
17	**MEBPFNR3 15**	**I1ES**	**1, 5, 15, 30**
18	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
19	**MEBPFNR3 15**	**I1ES**	**1, 5, 15, 20, 30, 100**
20	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
21	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
22	**MEBPFNR3 15**	**I1ES**	**15, 30**
23	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
24	**MEBPFNR3 15**	**I1ES**	**1, 5, 15, 30**
25	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
26	**MEBPFNR3 15**	**I1ES**	**5, 20, 30**
27	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
28	**MEBPFNR3 15**	**I1ES**	**5, 10, 15, 100**
29	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 30, 100**

Table B.88: 4-pier, 194 flights, 46 stands and MEBPFNR3 15.

Max. (9, 5, 3)	1	**5**	10	**15**	20
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	12 (7, 3, 2)	**15** (8, 4, *3*)	11 (6, 2, *3*)	**15** (8, 4, *3*)	11 (7, 3, 1)

Max. (9, 5, 3)	**30**	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	**15** (8, *5*, 2)	12 (8, 2, 2)

Table B.89: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and MEBPFNR3 15 for significance level of 0.05.

13	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 30, 100**
14	**MEFNR3 15**	**I1ES**	**1, 10, 15, 20, 30, 100**
15	**MEFNR3 15**	**I1ES**	**1, 5, 10, 20, 30, 100**
16	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
17	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
18	**MEFNR3 15**	**I1ES**	**10, 20, 30, 100**
19	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
20	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
21	**MEFNR3 15**	**I1ES**	**5, 10, 20, 30**
22	**MEFNR3 15**	**I1ES**	**1, 5, 10, 20, 30, 100**
23	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
24	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
25	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
26	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
27	**MEFNR3 15**	**I1ES**	**5, 10, 20, 30**
28	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 30, 100**
29	**MEFNR3 15**	**I1ES**	**1, 10, 15, 20, 100**

Table B.90: 4-pier, 194 flights, 46 stands and MEFNR3 15.

Max. (9, 5, 3)	1	5	**10**	15	20
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	14 (7, *5*, 2)	14 (7, *5*, 2)	**17** (*9*, *5*, *3*)	12 (6, 4, 2)	15 (8, *5*, 2)

Max. (9, 5, 3)	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	16 (*9*, *5*, 2)	13 (7, 4, 2)

Table B.91: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and MEFNR3 15 for significance level 0.05.

13	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 30, 100**
14	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
15	**RMEFNR2 15**	**I1ES**	**5, 10, 15, 20, 30, 100**
16	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
17	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 20**
18	**RMEFNR2 15**	**I1ES**	**1, 5, 15, 20, 30**
19	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
20	**RMEFNR2 15**	**I1ES**	**1, 15**
21	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 100**
22	**RMEFNR2 15**	**I1ES**	**5, 10, 20, 30**
23	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
24	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15**
25	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 30**
26	**RMEFNR2 15**	**I1ES**	**1, 5, 10**
27	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
28	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 20, 30**
29	**RMEFNR2 15**	**I1ES**	**1, 5, 15, 20, 30**

Table B.92: 4-pier, 194 flights, 46 stands and RMEFNR2 15.

Max. (9, 5, 3)	1	**5**	10	15	20
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	15 (8, 4, *3*)	**16** (8, *5*, *3*)	14 (7, *5*, 2)	13 (8, 3, 2)	11 (6, 2, *3*)

Max. (9, 5, 3)	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	12 (6, 3, *3*)	3 (3, 0, 0)

Table B.93: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 194 flights and RMEFNR2 15 for significance of 0.05.

13	**MEBPFNR3 15**	**30**
	MEFNR3 15	**10, 15**
14	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
15	**MEBPFNR3 15**	**15**
	MEFNR3 15	**10, 15, 20**
16	**MEFNR3 15**	**10, 15, 20**
17	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
18	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 20**
19	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
20	**MEFNR3 15**	**10, 20**
	RMEFNR2 15	**1**

21	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
	RMEFNR2 15	**1, 5**
22	**MEBPFNR3 15**	**15, 30**
	RMEFNR2 15	**5**
23	**MEBPFNR3 15**	**30**
	MEFNR3 15	**15**
	RMEFNR2 15	**1, 5**
24	**MEBPFNR3 15**	**15, 30**
	RMEFNR2 15	**1, 5**
25	**RMEFNR2 15**	**1, 5**
26	**RMEFNR2 15**	**1, 5**
27	**MEBPFNR3 15**	**30**
	RMEFNR2 15	**1, 5**
28	**RMEFNR2 15**	**1, 5**
29	**RMEFNR2 15**	**1, 5**

Table B.94: 194 flights, 4-pier and 46 stands.

Max. (9, 5, 3)	1	5	10	15	20	**30**
C1P 1000	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P 1000	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 1000	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 15	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	8 (6, 2, 0)	0 (0, 0, 0)	**10** (6, 3, 1)
MEFNR3 15	0 (0, 0, 0)	0 (0, 0, 0)	9 (*9*, 0, 0)	8 (7, 1, 0)	8 (8, 0, 0)	0 (0, 0, 0)
RMEFNR2 15	9 (2, 4, *3*)	9 (1, *5*, *3*)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.95: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 194 flights for significance level of 0.05.

1^{st} March 2010 with 163 flights

Both C1P and C2P provide statistically significantly fitter solutions for all considered data sets, topologies and number of BSSs for $\ell = 1$, as shown in Tables B.96, B.97, B.98 and B.99.

13	**C1P 1000**	**I1ES**	**1**
14	**C1P 1000**	**I1ES**	**1**
15	**C1P 1000**	**I1ES**	**1**
16	**C1P 1000**	**I1ES**	**1**
17	**C1P 1000**	**I1ES**	**1**
18	**C1P 1000**	**I1ES**	**1**
19	**C1P 1000**	**I1ES**	**1**
20	**C1P 1000**	**I1ES**	**1**
21	**C1P 1000**	**I1ES**	**1**
22	**C1P 1000**	**I1ES**	**1**
23	**C1P 1000**	**I1ES**	**1**
24	**C1P 1000**	**I1ES**	**1**
25	**C1P 1000**	**I1ES**	**1**
26	**C1P 1000**	**I1ES**	**1**
27	**C1P 1000**	**I1ES**	**1**
28	**C1P 1000**	**I1ES**	**1**
29	**C1P 1000**	**I1ES**	**1**

Table B.96: 4-pier, 163 flights, 46 stands and C1P 1000.

Max. (6, 6, 5)	**1**	5	10	15	20
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	**17** (*6*, *6*, *5*)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Max. (6, 6, 5)	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	0 (0, 0, 0)	0 (0, 0, 0)

Table B.97: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and C1P 1000 for significance level of 0.05.

13	**C2P 1000**	**I1ES**	**1**
14	**C2P 1000**	**I1ES**	**1**
15	**C2P 1000**	**I1ES**	**1**
16	**C2P 1000**	**I1ES**	**1**
17	**C2P 1000**	**I1ES**	**1**
18	**C2P 1000**	**I1ES**	**1**
19	**C2P 1000**	**I1ES**	**1**
20	**C2P 1000**	**I1ES**	**1**
21	**C2P 1000**	**I1ES**	**1**
22	**C2P 1000**	**I1ES**	**1**
23	**C2P 1000**	**I1ES**	**1**
24	**C2P 1000**	**I1ES**	**1**
25	**C2P 1000**	**I1ES**	**1**
26	**C2P 1000**	**I1ES**	**1**
27	**C2P 1000**	**I1ES**	**1**
28	**C2P 1000**	**I1ES**	**1**
29	**C2P 1000**	**I1ES**	**1**

Table B.98: 4-pier, 163 flights, 46 stands and C2P 1000.

Max. (6, 6, 5)	**1**	5	10	15	20
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	**17** (*6*, *6*, *5*)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Max. (6, 6, 5)	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	0 (0, 0, 0)	0 (0, 0, 0)

Table B.99: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and C2P 1000 for significance level of 0.05.

DSEMO provides statistically significantly fitter solutions for all considered data sets, topologies and number of BSSs for $\ell = 1$, as shown in Tables B.72 and B.73.

13	**DSEMO 1000**	**I1ES**	**1**
14	**DSEMO 1000**	**I1ES**	**1**
15	**DSEMO 1000**	**I1ES**	**1**
16	**DSEMO 1000**	**I1ES**	**1**
17	**DSEMO 1000**	**I1ES**	**1**
18	**DSEMO 1000**	**I1ES**	**1**
19	**DSEMO 1000**	**I1ES**	**1**
20	**DSEMO 1000**	**I1ES**	**1**
21	**DSEMO 1000**	**I1ES**	**1**
22	**DSEMO 1000**	**I1ES**	**1**
23	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
24	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
25	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
26	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
27	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
28	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**
29	**DSEMO 1000**	**ES**	**1, 5, 10, 15, 20, 30, 100**
		I1ES	**1, 5, 10, 15, 20, 30, 100**

Table B.100: 4-pier, 163 flights, 46 stands and DSEMO 1000.

Max. (6, 6, 5)	**1**	5	10	15	20
ES	7 (0, 2, *5*)	7 (0, 2, *5*)	7 (0, 2, *5*)	7 (0, 2, *5*)	7 (0, 2, *5*)
I1ES	**17** (*6*, *6*, *5*)	7 (0, 2, *5*)	7 (0, 2, *5*)	7 (0, 2, *5*)	7 (0, 2, *5*)

Max. (6, 6, 5)	30	100
ES	7 (0, 2, *5*)	7 (0, 2, *5*)
I1ES	7 (0, 2, *5*)	7 (0, 2, *5*)

Table B.101: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and DSEMO 1000 for significance level of 0.05.

13	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
14	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 20, 30, 100**
15	**MEBPFNR3 15**	**I1ES**	**1, 5, 15, 20, 30, 100**
16	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 30**
17	**MEBPFNR3 15**	**I1ES**	**1, 10, 20, 30, 100**
18	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
19	**MEBPFNR3 15**	**I1ES**	**5, 20**
20	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
21	**MEBPFNR3 15**	**I1ES**	**1, 10, 15, 100**
22	**MEBPFNR3 15**	**I1ES**	**1, 5, 15, 20, 30, 100**
23	**MEBPFNR3 15**	**I1ES**	**5, 10, 20**
24	**MEBPFNR3 15**	**I1ES**	**5, 15, 20, 30**
25	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 30, 100**
26	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
27	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
28	**MEBPFNR3 15**	**I1ES**	**30**
29	**MEBPFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**

Table B.102: 4-pier, 163 flights, 46 stands and MEBPFNR3 15.

Max. (6, 6, 5)	1	**5**	10	15	20
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	13 (*6*, 3, 4)	**14** (5, 5, 4)	12 (5, 3, 4)	11 (3, 4, 4)	13 (5, 5, 3)

Max. (6, 6, 5)	**30**	100
ES	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	**14** (*6*, 3, *5*)	12 (5, 3, 4)

Table B.103: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and MEBPFNR3 15 for significance level of 0.05.

13	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
14	**MEFNR3 15**	**I1ES**	**1, 5, 10, 20, 100**
15	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
16	**MEFNR3 15**	**I1ES**	**1, 10, 15, 20, 100**
17	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
18	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
19	**MEFNR3 15**	**I1ES**	**30**
20	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
21	**MEFNR3 15**	**I1ES**	**1, 10**
22	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
23	**MEFNR3 15**	**I1ES**	**1, 5, 10, 15, 20, 30, 100**
24	**MEFNR3 15**	**I1ES**	**100**
25	**MEFNR3 15**	**I1ES**	**1, 5, 10, 20, 30, 100**
26	**MEFNR3 15**	**I1ES**	**5, 10, 20, 30, 100**
27	**MEFNR3 15**	**I1ES**	**5, 15, 20, 100**
28	**MEFNR3 15**	**I1ES**	**10, 20, 30, 100**
29	**MEFNR3 15**	**I1ES**	**5, 10, 15, 100**

Table B.104: 4-pier, 163 flights, 46 stands and MEFNR3 15.

Max. (6, 6, 5)	1	5	**10**	15	20	30	**100**
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	11 (*6*, 4, 1)	12 (5, 3, 4)	**14** (*6*, 4, 4)	10 (5, 3, 2)	13 (*6*, 3, 4)	11 (4, 4, 3)	**14** (*6*, 3, *5*)

Table B.105: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and MEFNR3 15 for significance level of 0.05.

13	**RMEFNR2 15**	**I1ES**	**1, 5**
14	**RMEFNR2 15**	**I1ES**	**1, 5, 15, 20, 30**
15	**RMEFNR2 15**	**I1ES**	**1, 5, 15, 20, 30**
16	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
17	**RMEFNR2 15**	**I1ES**	**10, 15, 20**
18	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20**
19	**RMEFNR2 15**	**I1ES**	**10, 15, 20**
20	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 30**
21	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
22	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
23	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
24	**RMEFNR2 15**	**I1ES**	**1, 5, 20**
25	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 20, 30**
26	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15**
27	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 20**
28	**RMEFNR2 15**	**I1ES**	**1, 5, 10, 15, 20, 30**
29	**RMEFNR2 15**	**I1ES**	**5, 10, 15**

Table B.106: 4-pier, 163 flights, 46 stands and RMEFNR2 15.

Max. (6, 6, 5)	1	**5**	10	15	20	30	100
ES	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
I1ES	14 (5, 5, 4)	**15** (5, 5, 5)	13 (3, 5, 5)	12 (5, 4, 3)	13 (5, 5, 3)	9 (3, 4, 2)	0 (0, 0, 0)

Table B.107: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology, 163 flights and RMEFNR2 15 for significance level of 0.05.

13	**DSEMO 1000**	**1**
14	**DSEMO 1000**	**1**
15	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
16	**MEBPFNR3 15**	**30**
	MEFNR3 15	**15**
17	**DSEMO 1000**	**1**
18	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
19	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
20	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 20**
	RMEFNR2 15	**1**
21	**MEFNR3 15**	**10**
	RMEFNR2 15	**1, 5**

22	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 20**
23	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 15, 20**
24	**MEBPFNR3 15**	**15, 30**
25	**MEBPFNR3 15**	**30**
	MEFNR3 15	**10, 20**
	RMEFNR2 15	**1, 5**
26	**MEBPFNR3 15**	**15, 30**
	MEFNR3 15	**10, 20**
	RMEFNR2 15	**1, 5**
27	**RMEFNR2 15**	**1, 5**
28	**MEBPFNR3 15**	**30**
29	**MEBPFNR3 15**	**30**
	MEFNR3 15	**15**
	RMEFNR2 15	**1, 5**

Table B.108: 163 flights, 4-pier and 46 stands.

Max. (6, 6, 5)	1	5	10	15	20	**30**
C1P 1000	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P 1000	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 1000	3 (3, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 15	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	8 (2, 5, 1)	0 (0, 0, 0)	**12** (3, 5, 4)
MEFNR3 15	0 (0, 0, 0)	0 (0, 0, 0)	9 (2, 5, 2)	6 (3, 2, 1)	8 (2, 4, 2)	0 (0, 0, 0)
RMEFNR2 15	6 (0, 2, 4)	5 (0, 1, 4)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.109: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 163 flights for significance level of 0.05.

B.4 Results Index for ISxES

In this section are presented the results for the performance of the ISxES for $x \in (1 \ldots 15)$.

B.4.1 Graphical Representation of Results

3-pier topology

16^{th} December 2009 with 194 flights

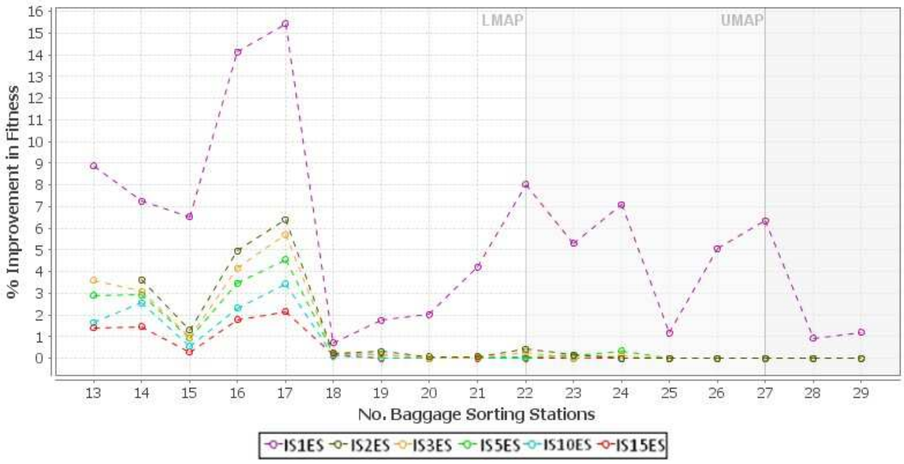


Figure B.37: ISxES, 3-pier, 194 flights and 1-point crossover with 1000 population size (C1P 1000).

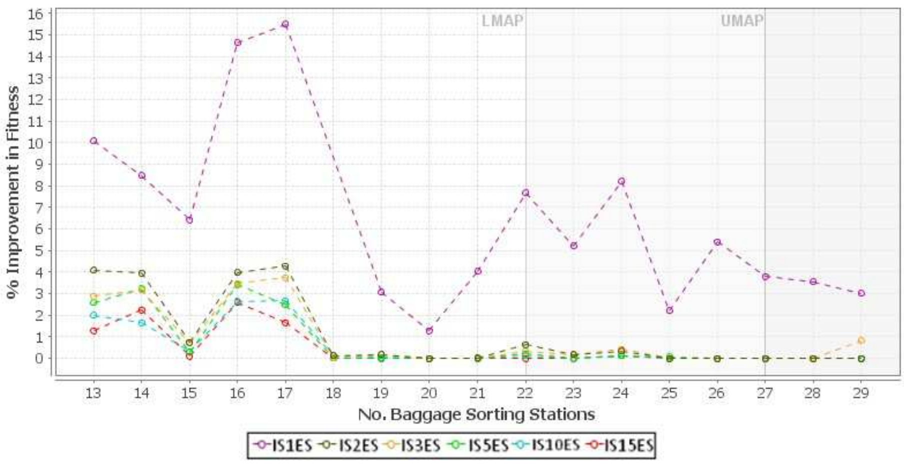


Figure B.38: ISxES, 3-pier, 194 flights and 2-point crossover with 1000 population size (C2P 1000).

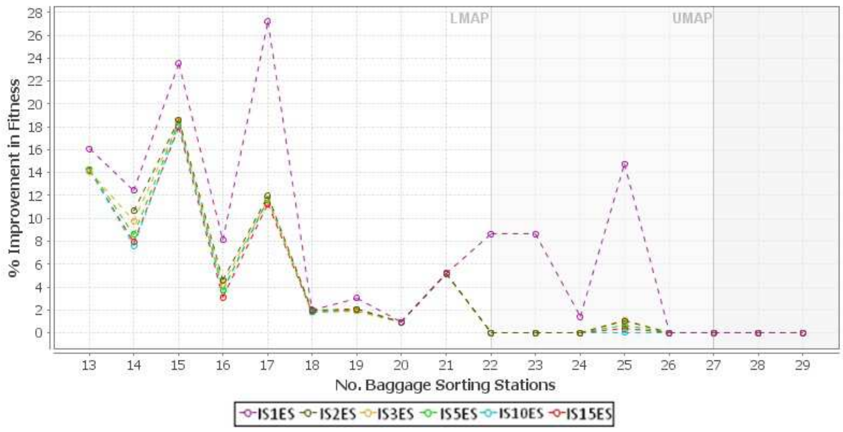


Figure B.39: ISxES, 3-pier, 194 flights and DSEMO with 1000 population size.

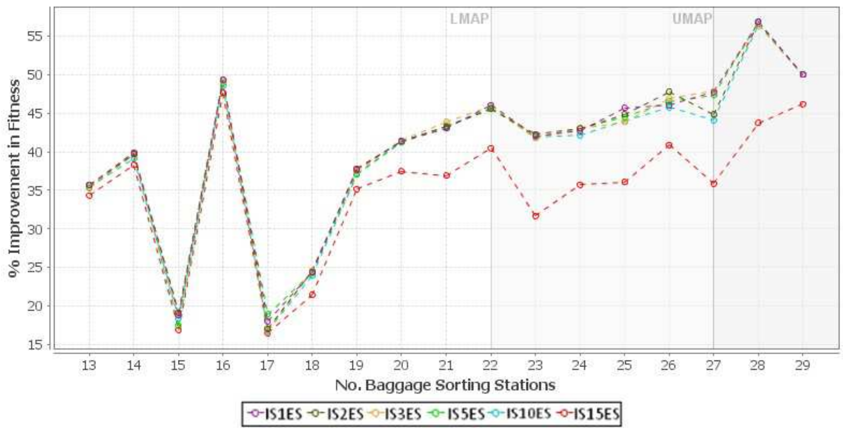


Figure B.40: ISxES, 3-pier, 194 flights and MEBPFNR3 with 15 population size.

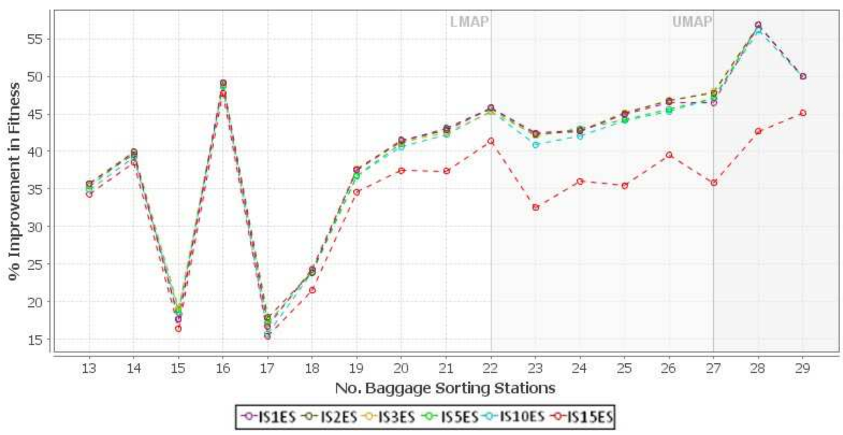


Figure B.41: ISxES, 3-pier, 194 flights and MEFNR3 with 15 population size.

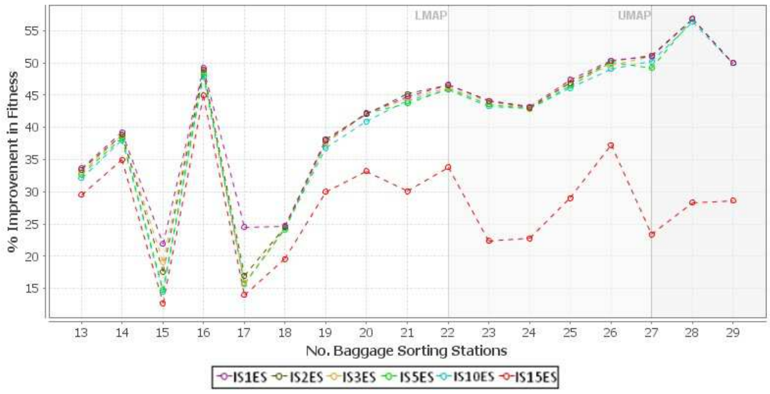


Figure B.42: ISxES, 3-pier, 194 flights and RMEFNR2 with 15 population size.

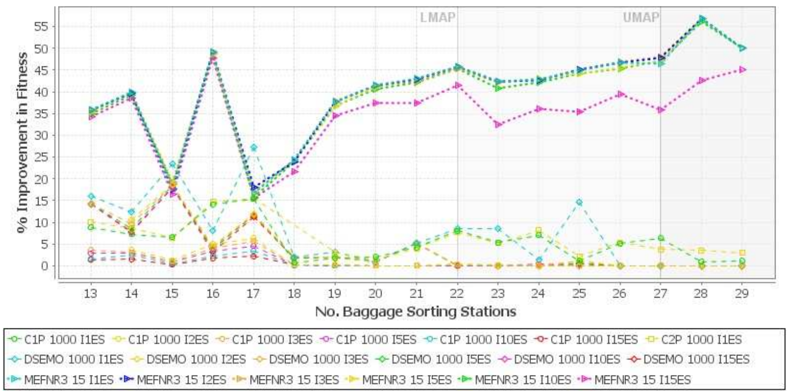


Figure B.43: ISxES, 3-pier, 194 flights for C1P 1000, C2P 1000, DSEMO 1000 and MEFNR3 15

1^{st} March 2010 with 163 flights

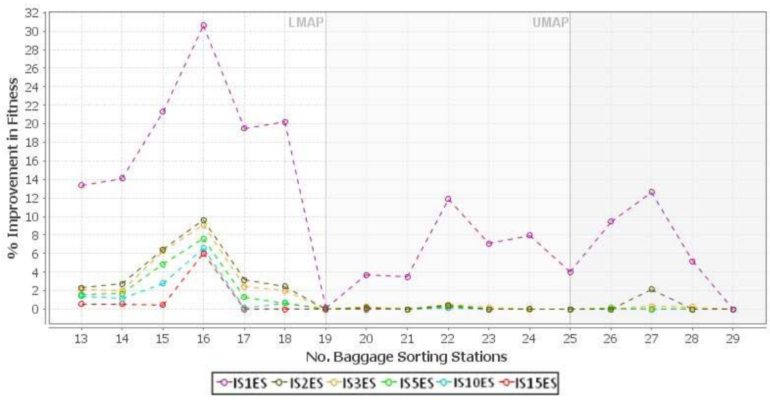


Figure B.44: ISxES, 3-pier, 163 flights and 1-point crossover with 1000 population size (C1P 1000).

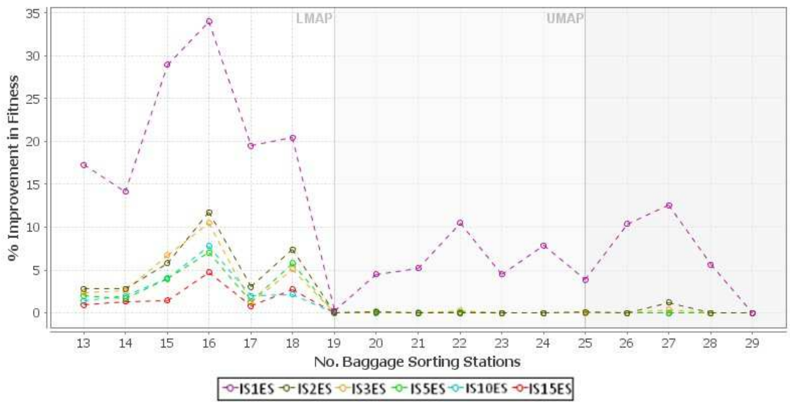


Figure B.45: ISxES, 3-pier, 163 flights and 2-point crossover with 1000 population size (C2P 1000).

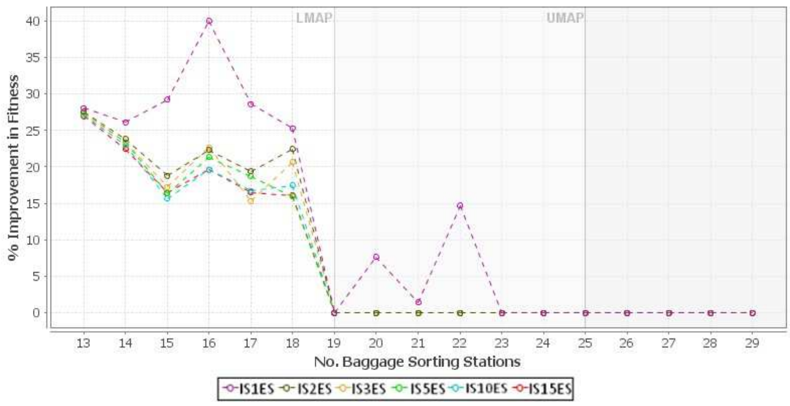


Figure B.46: ISxES, 3-pier, 163 flights and DSEMO with 1000 population size.

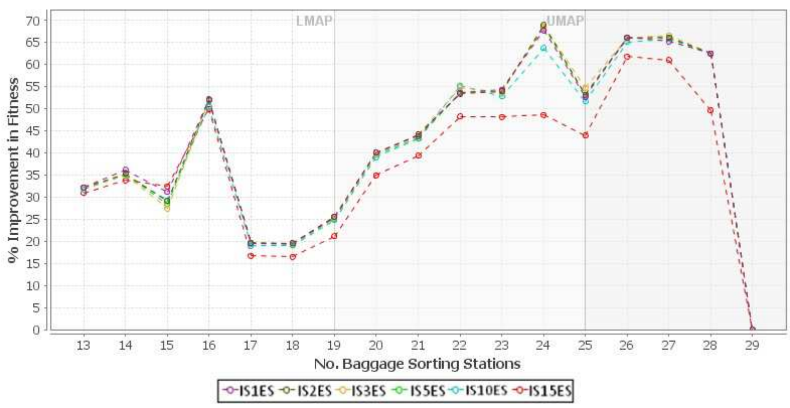


Figure B.47: ISxES, 3-pier, 163 flights and MEBPFNR3 with 15 population size.

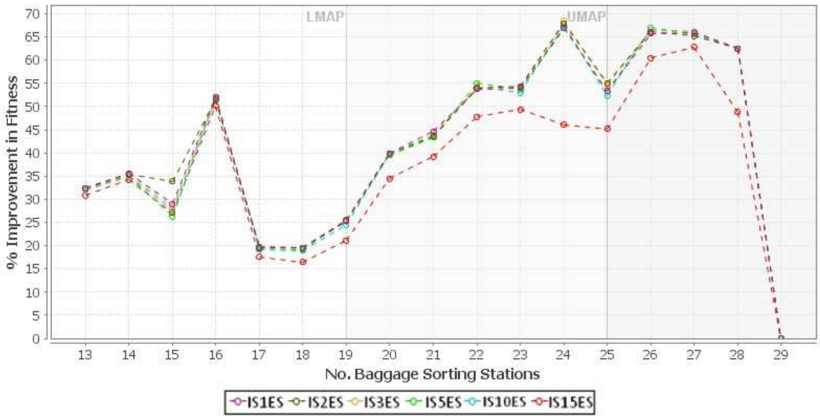


Figure B.48: ISxES, 3-pier, 163 flights and MEFNR3 with 15 population size.

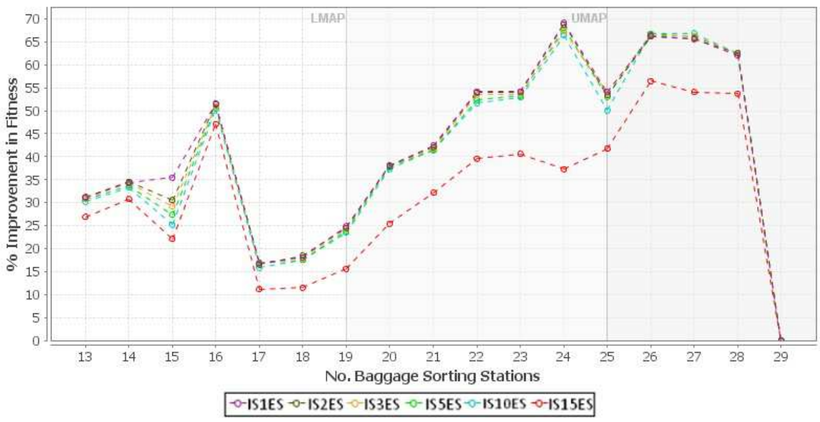


Figure B.49: ISxES, 3-pier, 163 flights and RMEFNR2 with 15 population size.

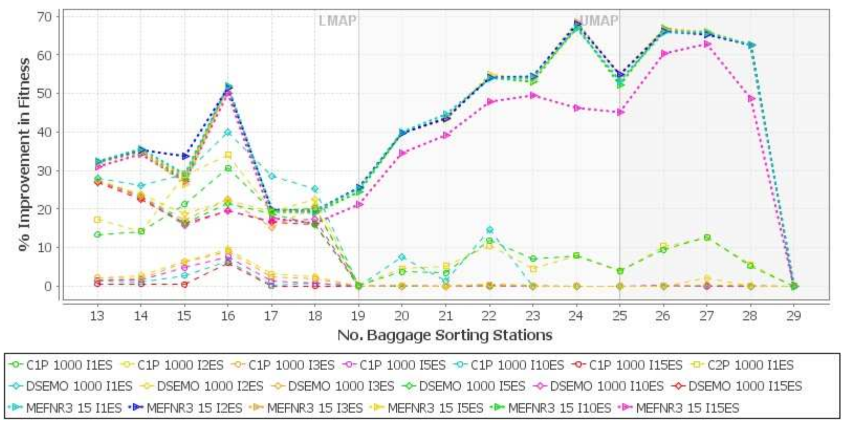


Figure B.50: ISxES, 3-pier, 163 flights for C1P 1000, C2P 1000, DSEMO 1000 and MEFNR3 15.

4-pier topology

16^{th} December 2009 with 194 flights

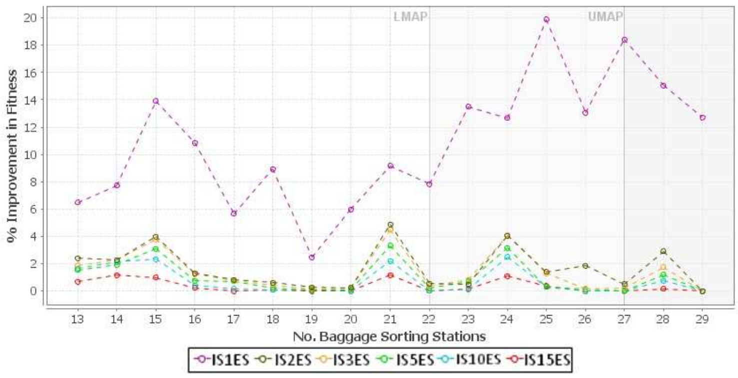


Figure B.51: ISxES for 1-point crossover with 1000 population size (C1P 1000).

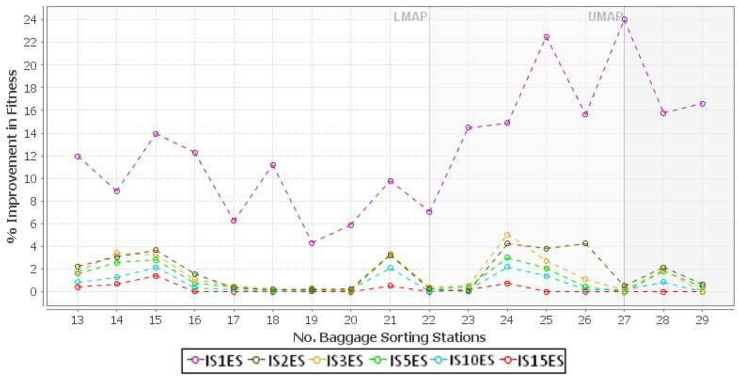


Figure B.52: ISxES for 2-point crossover with 1000 population size (C2P 1000).

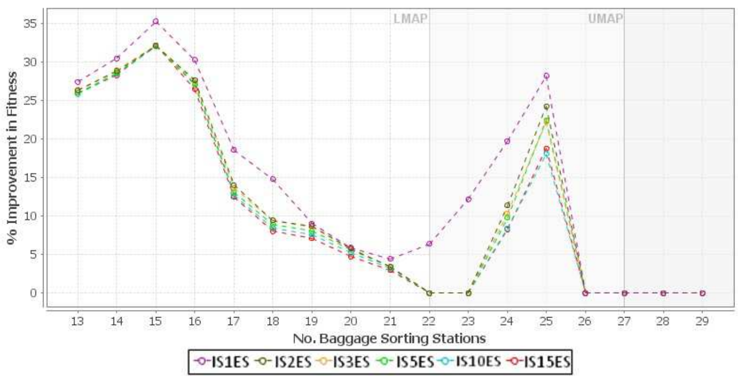


Figure B.53: ISxES for DSEMO with 1000 population size.

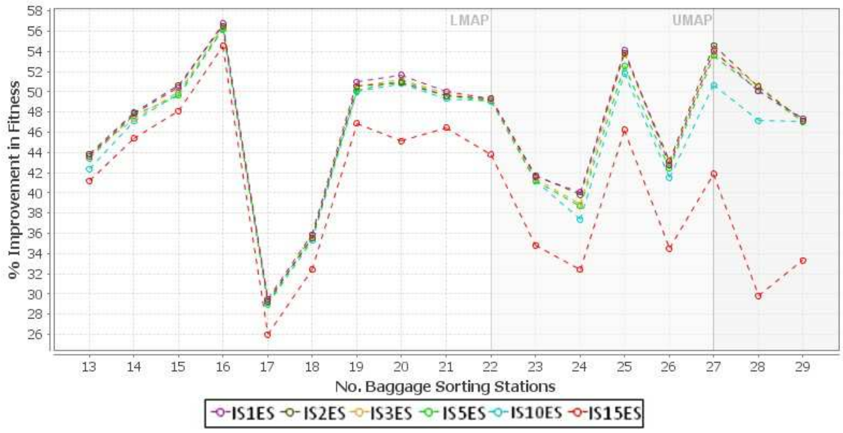


Figure B.54: ISxES, 4-pier, 194 flights for MEBPFNR3 with 15 population size.

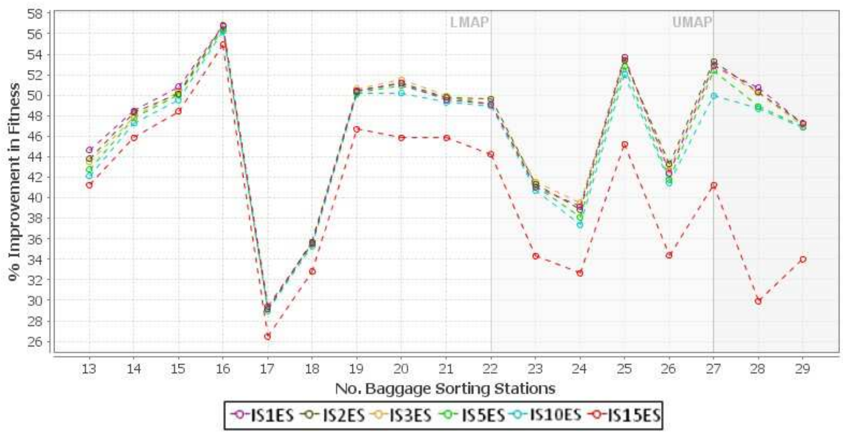


Figure B.55: ISxES, 4-pier, 194 flights for MEFNR3 with 15 population size.

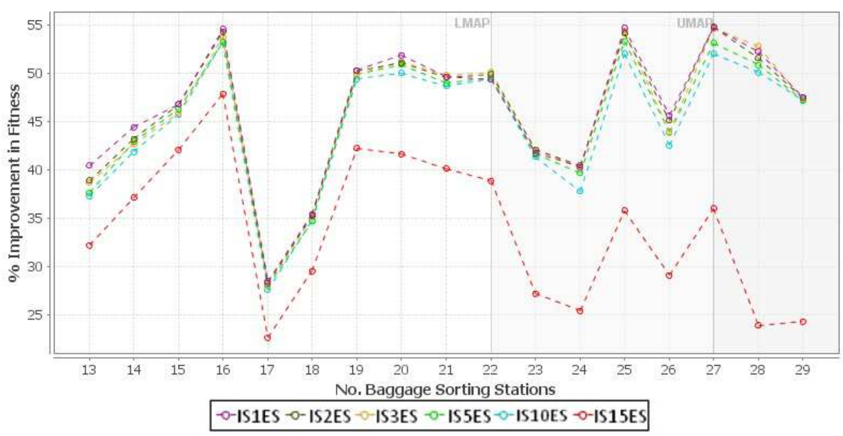


Figure B.56: ISxES, 4-pier, 194 flights for RMEFNR2 with 15 population size.

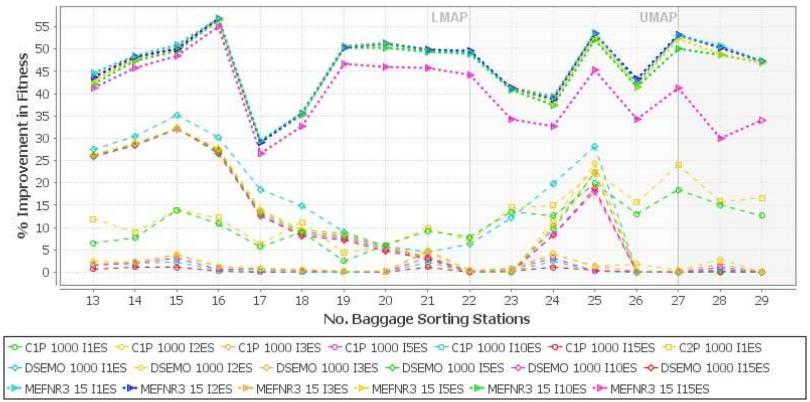


Figure B.57: ISxES, 4-pier, 194 flights for C1P 1000, C2P 1000, DSEMO 1000 and MEFNR3 15.

1^{st} March 2010 with 163 flights

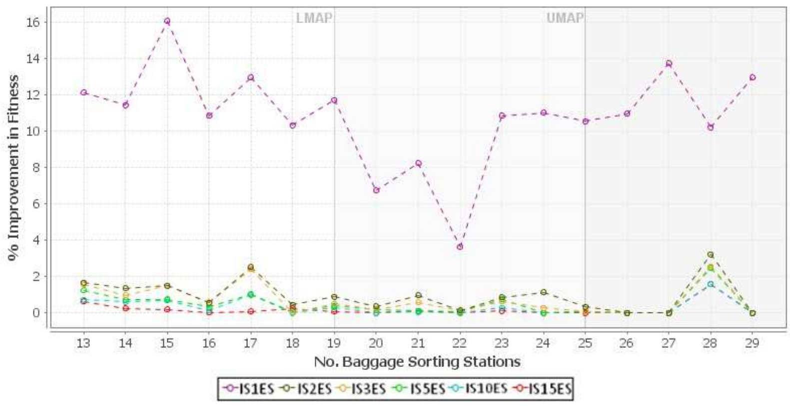


Figure B.58: ISxES for 1-point crossover with 1000 population size (C1P 1000).

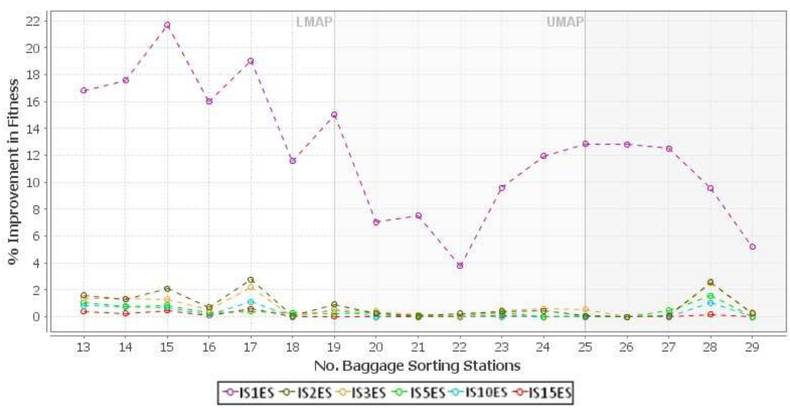


Figure B.59: ISxES for 2-point crossover with 1000 population size (C2P 1000).

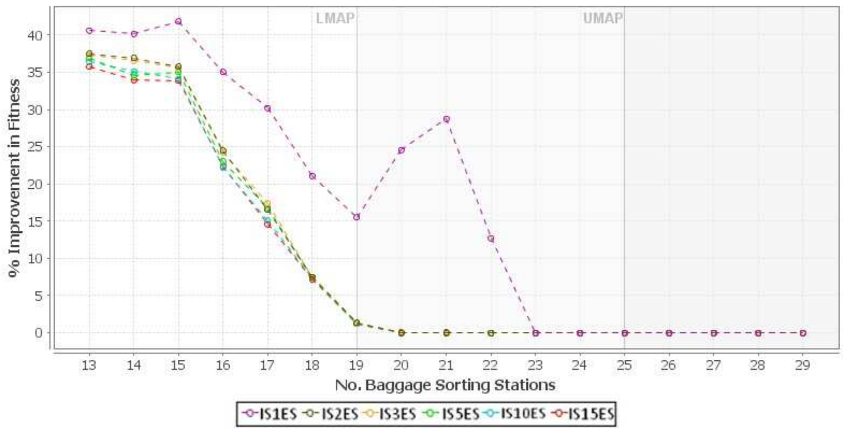


Figure B.60: ISxES, 4-pier, 163 flights for DSEMO with 1000 population size.

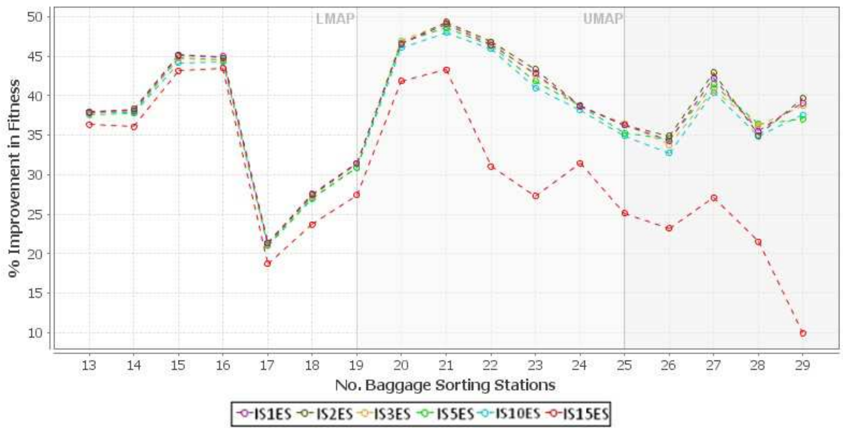


Figure B.61: ISxES, 4-pier, 163 flights for MEBPFNR3 with 15 population size.

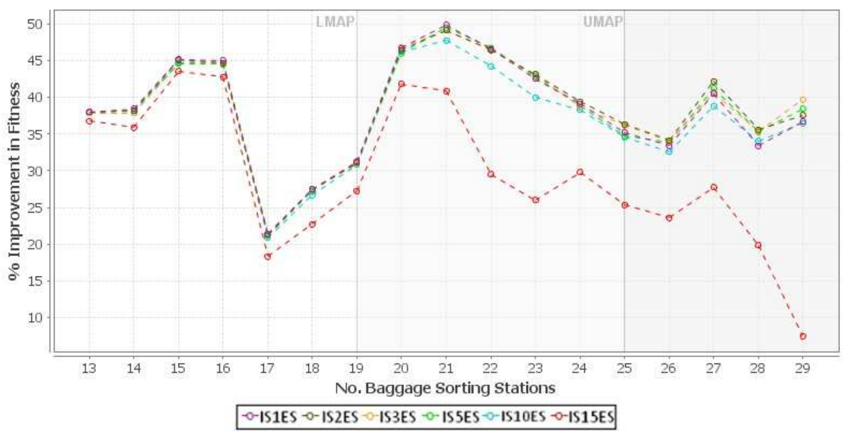


Figure B.62: ISxES, 4-pier, 163 flights for MEFNR3 with 15 population size.

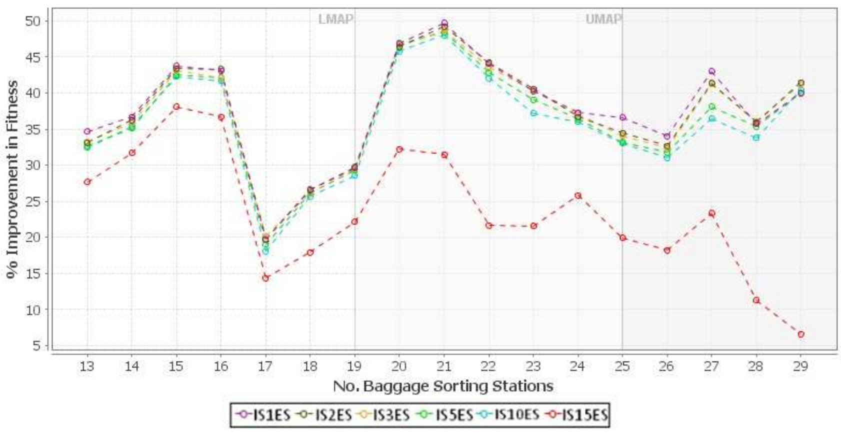


Figure B.63: ISxES, 4-pier, 163 flights for RMEFNR2 with 15 population size.

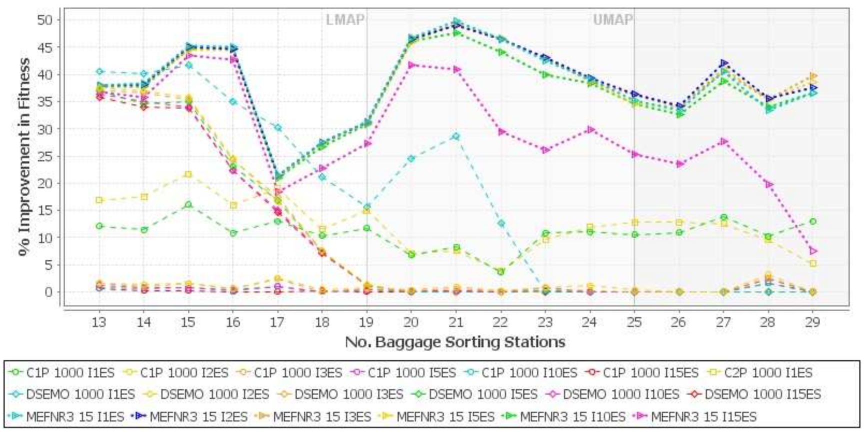


Figure B.64: ISxES, 4-pier, 163 flights for C1P 1000, C2P 1000, DSEMO 1000 and MEFNR3 15.

B.4.2 Statistical Results

Summary tables of the Mann-Whitney tests for the experiments conducted for the considered operators, topologies and ISxES indexes, $x \in (1, 2, 3, 5, 10, 15)$.

16^{th} December 2009 with 194 flights

3-pier topology

13	C1P 1000	IS1ES
14	C1P 1000	IS1ES
15	C1P 1000	IS1ES
16	C1P 1000	IS1ES
17	C1P 1000	IS1ES
18	C1P 1000	IS1ES
19	C1P 1000	IS1ES
20	C1P 1000	IS1ES
21	C1P 1000	IS1ES
22	C1P 1000	IS1ES
23	C1P 1000	IS1ES
24	C1P 1000	IS1ES
25	C1P 1000	IS1ES
26	C1P 1000	IS1ES
27	C1P 1000	IS1ES
28	C1P 1000	IS1ES
29	C1P 1000	IS1ES

4-pier topology

13	C1P 1000	IS1ES
14	C1P 1000	IS1ES
15	C1P 1000	IS1ES
16	C1P 1000	IS1ES
17	C1P 1000	IS1ES
18	C1P 1000	IS1ES
19	C1P 1000	IS1ES
20	C1P 1000	IS1ES
21	C1P 1000	IS1ES
22	C1P 1000	IS1ES
23	C1P 1000	IS1ES
24	C1P 1000	IS1ES
25	C1P 1000	IS1ES
26	C1P 1000	IS1ES
27	C1P 1000	IS1ES
28	C1P 1000	IS1ES
29	C1P 1000	IS1ES

Table B.110: ISxES and 194 flights with operator C1P for 1000 population size.

3-pier topology

13	C2P 1000	IS1ES
14	C2P 1000	IS1ES
15	C2P 1000	IS1ES
16	C2P 1000	IS1ES
17	C2P 1000	IS1ES
18	C2P 1000	IS2ES
		IS3ES
		IS10ES
		IS15ES
19	C2P 1000	IS1ES
20	C2P 1000	IS1ES
21	C2P 1000	IS1ES
22	C2P 1000	IS1ES
23	C2P 1000	IS1ES
24	C2P 1000	IS1ES
25	C2P 1000	IS1ES
26	C2P 1000	IS1ES
27	C2P 1000	IS1ES
28	C2P 1000	IS1ES
29	C2P 1000	IS1ES

4-pier topology

13	C2P 1000	IS1ES
14	C2P 1000	IS1ES
15	C2P 1000	IS1ES
16	C2P 1000	IS1ES
17	C2P 1000	IS1ES
18	C2P 1000	IS1ES
19	C2P 1000	IS1ES
20	C2P 1000	IS1ES
21	C2P 1000	IS1ES
22	C2P 1000	IS1ES
23	C2P 1000	IS1ES
24	C2P 1000	IS1ES
25	C2P 1000	IS1ES
26	C2P 1000	IS1ES
27	C2P 1000	IS1ES
28	C2P 1000	IS1ES
29	C2P 1000	IS1ES

Table B.111: ISxES and 194 flights with operator C2P for 1000 population size.

3-pier topology

13	DSEMO 1000	IS1ES
14	DSEMO 1000	IS1ES
15	DSEMO 1000	IS1ES
16	DSEMO 1000	IS1ES
17	DSEMO 1000	IS1ES
18	DSEMO 1000	IS1ES
19	DSEMO 1000	IS1ES
20	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
21	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
22	DSEMO 1000	IS1ES
23	DSEMO 1000	IS1ES
24	DSEMO 1000	IS1ES
25	DSEMO 1000	IS1ES
26	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
27	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
28	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
29	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES

4-pier topology

13	DSEMO 1000	IS1ES
14	DSEMO 1000	IS1ES
15	DSEMO 1000	IS1ES
16	DSEMO 1000	IS1ES
17	DSEMO 1000	IS1ES
18	DSEMO 1000	IS1ES
19	DSEMO 1000	IS1ES
20	DSEMO 1000	IS1ES
21	DSEMO 1000	IS1ES
22	DSEMO 1000	IS1ES
23	DSEMO 1000	IS1ES
24	DSEMO 1000	IS1ES
25	DSEMO 1000	IS1ES
26	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
27	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
28	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
29	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES

Table B.112: ISxES and 194 flights with operator DSEMO for 1000 population size.

3-pier topology

13	MEBPFNR3 15	IS1ES
		IS2ES
14	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
15	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
16	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
17	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
18	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
19	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
20	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
21	MEBPFNR3 15	IS3ES
		IS5ES
		IS10ES
22	MEBPFNR3 15	IS1ES
		IS3ES
		IS5ES
23	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
24	MEBPFNR3 15	IS2ES
		IS3ES
		IS5ES
25	MEBPFNR3 15	IS1ES
		IS2ES
26	MEBPFNR3 15	IS2ES
		IS3ES
		IS5ES
27	MEBPFNR3 15	IS1ES
		IS3ES
		IS5ES
28	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
29	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES

4-pier topology

13	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
14	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
15	MEBPFNR3 15	IS1ES
		IS2ES
16	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
17	MEBPFNR3 15	IS1ES
		IS3ES
18	MEBPFNR3 15	IS1ES
		IS3ES
		IS5ES
19	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
20	MEBPFNR3 15	IS1ES
		IS3ES
		IS5ES
21	MEBPFNR3 15	IS1ES
		IS3ES
		IS5ES
22	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
23	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
24	MEBPFNR3 15	IS1ES
		IS2ES
25	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
26	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
27	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
28	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
29	MEBPFNR3 15	IS1ES
		IS2ES

Table B.113: ISxES and 194 flights with operator MEBPFNR3 for 15 population size.

3-pier topology

13	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
14	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
15	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
16	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
17	MEFNR3 15	IS5ES
18	MEFNR3 15	IS1ES
		IS3ES
19	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
20	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
21	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
22	MEFNR3 15	IS1ES
		IS2ES
		IS5ES
23	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
24	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
25	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
26	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
27	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
28	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
29	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES

4-pier topology

13	MEFNR3 15	IS1ES
14	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
15	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
16	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
17	MEFNR3 15	IS1ES
		IS5ES
18	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
19	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
20	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
21	MEFNR3 15	IS2ES
		IS3ES
		IS5ES
22	MEFNR3 15	IS2ES
		IS5ES
23	MEFNR3 15	IS2ES
		IS3ES
		IS5ES
24	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
25	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
26	MEFNR3 15	IS2ES
27	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
28	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
29	MEFNR3 15	IS1ES
		IS2ES
		IS3ES

Table B.114: ISxES and 194 flights with operator MEFNR3 for 15 population size.

3-pier topology

13	RMEFNR2 15	IS1ES
		IS2ES
14	RMEFNR2 15	IS1ES
		IS3ES
15	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
16	RMEFNR2 15	IS1ES
		IS2ES
17	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
18	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS10ES
19	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
20	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
21	RMEFNR2 15	IS1ES
		IS2ES
22	RMEFNR2 15	IS1ES
		IS2ES
23	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
24	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
25	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
26	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
27	RMEFNR2 15	IS1ES
		IS2ES
28	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
29	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES

4-pier topology

13	RMEFNR2 15	IS1ES
14	RMEFNR2 15	IS1ES
15	RMEFNR2 15	IS1ES
		IS2ES
		IS5ES
16	RMEFNR2 15	IS1ES
		IS2ES
17	RMEFNR2 15	IS1ES
		IS2ES
18	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
19	RMEFNR2 15	IS1ES
		IS2ES
		IS5ES
20	RMEFNR2 15	IS1ES
21	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
22	RMEFNR2 15	IS2ES
		IS3ES
23	RMEFNR2 15	IS2ES
		IS3ES
		IS10ES
24	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
25	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
26	RMEFNR2 15	IS1ES
		IS2ES
27	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
28	RMEFNR2 15	IS1ES
		IS3ES
29	RMEFNR2 15	IS1ES

Table B.115: ISxES and 194 flights with operator RMEFNR2 for 15 population size.

3-pier topology

13	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES, IS2ES, IS3ES
14	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
15	DSEMO 1000	IS1ES
16	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
	RMEFNR2 15	IS1ES
17	DSEMO 1000	IS1ES
18	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES, IS3ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS10ES
19	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
20	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
21	RMEFNR2 15	IS1ES, IS2ES
22	RMEFNR2 15	IS1ES, IS2ES
23	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
24	MEFNR3 15	IS5ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
25	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
26	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
27	RMEFNR2 15	IS1ES, IS2ES
28	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
29	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES

4-pier topology

13	MEFNR3 15	IS1ES
14	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
15	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
16	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
17	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
18	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS2ES
		IS3ES
19	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS3ES
20	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS3ES
	RMEFNR2 15	IS1ES
21	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS2ES
		IS3ES
		IS5ES
	RMEFNR2 15	IS1ES
		IS3ES
22	RMEFNR2 15	IS2ES
		IS3ES
23	MEBPFNR3 15	IS1ES
	RMEFNR2 15	IS2ES
		IS3ES
		IS10ES
24	MEBPFNR3 15	IS1ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
25	MEBPFNR3 15	IS1ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
26	RMEFNR2 15	IS1ES
		IS2ES
27	MEBPFNR3 15	IS1ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
28	RMEFNR2 15	IS1ES
		IS3ES
29	RMEFNR2 15	IS1ES

Table B.116: ISxES and 194 flights with operators C1P 1000, C1P 1000, DSEMO 1000, MEBPFNR3 15, MEFNR3 15 and RMEFNR2 15.

1^{st} March 2010 with 163 flights

3-pier topology

13	C1P 1000	IS1ES
14	C1P 1000	IS1ES
15	C1P 1000	IS1ES
16	C1P 1000	IS1ES
17	C1P 1000	IS1ES
18	C1P 1000	IS1ES
19	C1P 1000	IS1ES
20	C1P 1000	IS1ES
21	C1P 1000	IS1ES
22	C1P 1000	IS1ES
23	C1P 1000	IS1ES
24	C1P 1000	IS1ES
25	C1P 1000	IS1ES
26	C1P 1000	IS1ES
27	C1P 1000	IS1ES
28	C1P 1000	IS1ES
29	C1P 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES

4-pier topology

13	C1P 1000	IS1ES
14	C1P 1000	IS1ES
15	C1P 1000	IS1ES
16	C1P 1000	IS1ES
17	C1P 1000	IS1ES
18	C1P 1000	IS1ES
19	C1P 1000	IS1ES
20	C1P 1000	IS1ES
21	C1P 1000	IS1ES
22	C1P 1000	IS1ES
23	C1P 1000	IS1ES
24	C1P 1000	IS1ES
25	C1P 1000	IS1ES
26	C1P 1000	IS1ES
27	C1P 1000	IS1ES
28	C1P 1000	IS1ES
29	C1P 1000	IS1ES

Table B.117: ISxES and 163 flights with operator C1P for 1000 population size.

3-pier topology

13	C2P 1000	IS1ES
14	C2P 1000	IS1ES
15	C2P 1000	IS1ES
16	C2P 1000	IS1ES
17	C2P 1000	IS1ES
18	C2P 1000	IS1ES
19	C2P 1000	IS1ES
20	C2P 1000	IS1ES
21	C2P 1000	IS1ES
22	C2P 1000	IS1ES
23	C2P 1000	IS1ES
24	C2P 1000	IS1ES
25	C2P 1000	IS1ES
26	C2P 1000	IS1ES
27	C2P 1000	IS1ES
28	C2P 1000	IS1ES
29	C2P 1000	IS1ES
		IS2ES
		IS10ES
		IS15ES

4-pier topology

13	C2P 1000	IS1ES
14	C2P 1000	IS1ES
15	C2P 1000	IS1ES
16	C2P 1000	IS1ES
17	C2P 1000	IS1ES
18	C2P 1000	IS1ES
19	C2P 1000	IS1ES
20	C2P 1000	IS1ES
21	C2P 1000	IS1ES
22	C2P 1000	IS1ES
23	C2P 1000	IS1ES
24	C2P 1000	IS1ES
25	C2P 1000	IS1ES
26	C2P 1000	IS1ES
27	C2P 1000	IS1ES
28	C2P 1000	IS1ES
29	C2P 1000	IS1ES

Table B.118: ISxES and 163 flights with operator C2P for 1000 population size.

3-pier topology

13	DSEMO 1000	IS1ES
14	DSEMO 1000	IS1ES
15	DSEMO 1000	IS1ES
16	DSEMO 1000	IS1ES
17	DSEMO 1000	IS1ES
18	DSEMO 1000	IS1ES
19	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
20	DSEMO 1000	IS1ES
21	DSEMO 1000	IS1ES
22	DSEMO 1000	IS1ES
23	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
24	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
25	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
26	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
27	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
28	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
29	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES

4-pier topology

13	DSEMO 1000	IS1ES
14	DSEMO 1000	IS1ES
15	DSEMO 1000	IS1ES
16	DSEMO 1000	IS1ES
17	DSEMO 1000	IS1ES
18	DSEMO 1000	IS1ES
19	DSEMO 1000	IS1ES
20	DSEMO 1000	IS1ES
21	DSEMO 1000	IS1ES
22	DSEMO 1000	IS1ES
23	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
24	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
25	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
26	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
27	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
28	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
29	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES

Table B.119: ISxES and 163 flights with operator DSEMO for 1000 population size.

3-pier topology

13	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
14	MEBPFNR3 15	IS1ES
15	MEBPFNR3 15	IS1ES
		IS2ES
16	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
17	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
18	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
19	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
20	MEBPFNR3 15	IS1ES
		IS2ES
21	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
22	MEBPFNR3 15	IS3ES
		IS10ES
23	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
24	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
25	MEBPFNR3 15	IS2ES
		IS3ES
		IS5ES
26	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
27	MEBPFNR3 15	IS2ES
		IS3ES
		IS5ES
		IS10ES
28	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
29	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES

4-pier topology

13	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
14	MEBPFNR3 15	IS1ES
		IS2ES
15	MEBPFNR3 15	IS1ES
		IS2ES
16	MEBPFNR3 15	IS1ES
		IS2ES
		IS5ES
17	MEBPFNR3 15	IS1ES
		IS2ES
18	MEBPFNR3 15	IS1ES
		IS2ES
19	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
20	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
21	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
22	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
23	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
24	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
25	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
26	MEBPFNR3 15	IS1ES
		IS2ES
		IS5ES
27	MEBPFNR3 15	IS1ES
		IS2ES
28	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
29	MEBPFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES

Table B.120: ISxES and 163 flights with operator MEBPFNR3 for 15 population size.

3-pier topology

13	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
14	MEFNR3 15	IS1ES
		IS2ES
15	MEFNR3 15	IS1ES
		IS2ES
16	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
17	MEFNR3 15	IS2ES
18	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
19	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
20	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
21	MEFNR3 15	IS1ES
22	MEFNR3 15	IS3ES
		IS5ES
		IS10ES
23	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
24	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
25	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
26	MEFNR3 15	IS2ES
		IS3ES
		IS5ES
		IS10ES
27	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
28	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
29	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES

4-pier topology

13	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
14	MEFNR3 15	IS1ES
		IS2ES
		IS5ES
15	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
16	MEFNR3 15	IS1ES
		IS2ES
		IS10ES
17	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
18	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
19	MEFNR3 15	IS1ES
		IS2ES
		IS5ES
20	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
21	MEFNR3 15	IS1ES
		IS5ES
22	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
23	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
24	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
25	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
26	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
27	MEFNR3 15	IS2ES
		IS5ES
28	MEFNR3 15	IS2ES
		IS3ES
		IS5ES
		IS10ES
29	MEFNR3 15	IS3ES
		IS5ES

Table B.121: ISxES and 163 flights with operator MEFNR3 for 15 population size.

3-pier topology

13	RMEFNR2 15	IS1ES
		IS2ES
14	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
15	RMEFNR2 15	IS1ES
16	RMEFNR2 15	IS1ES
		IS2ES
17	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
18	RMEFNR2 15	IS1ES
		IS2ES
19	RMEFNR2 15	IS1ES
		IS2ES
20	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
21	RMEFNR2 15	IS1ES
		IS2ES
22	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
23	RMEFNR2 15	IS1ES
		IS2ES
24	RMEFNR2 15	IS1ES
		IS2ES
25	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
26	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
27	RMEFNR2 15	IS3ES
		IS5ES
		IS10ES
28	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
29	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES

4-pier topology

13	RMEFNR2 15	IS1ES
14	RMEFNR2 15	IS1ES
15	RMEFNR2 15	IS1ES
		IS2ES
16	RMEFNR2 15	IS1ES
		IS2ES
17	RMEFNR2 15	IS3ES
18	RMEFNR2 15	IS1ES
		IS2ES
19	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
20	RMEFNR2 15	IS1ES
		IS3ES
		IS5ES
21	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
22	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
23	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
24	RMEFNR2 15	IS1ES
		IS3ES
25	RMEFNR2 15	IS1ES
26	RMEFNR2 15	IS1ES
27	RMEFNR2 15	IS1ES
28	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
29	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES

Table B.122: ISxES and 163 flights with operator RMEFNR2 for 15 population size.

13	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
14	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
15	C2P 1000	IS1ES
	DSEMO 1000	IS1ES
	MEFNR3 15	IS2ES
	RMEFNR2 15	IS1ES
16	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
17	DSEMO 1000	IS1ES
18	DSEMO 1000	IS1ES
19	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
20	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
21	MEFNR3 15	IS1ES
22	MEFNR3 15	IS3ES
		IS5ES
		IS10ES
23	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
	RMEFNR2 15	IS1ES
		IS2ES
24	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS3ES
		IS5ES
		IS10ES
	RMEFNR2 15	IS1ES
		IS2ES
25	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
	RMEFNR2 15	IS1ES
		IS3ES
26	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS2ES
		IS3ES
		IS5ES
		IS10ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
27	MEFNR3 15	IS1ES
		IS5ES
	RMEFNR2 15	IS3ES
		IS5ES
		IS10ES
28	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
29	C1P 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
	C2P 1000	IS1ES
		IS2ES
		IS10ES
		IS15ES
	DSEMO 1000	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES
		IS15ES

Table B.123: ISxES, 3-pier topology and 163 flights with operators C1P 1000, C2P 1000, DSEMO 1000, MEBPFNR3 15, MEFNR3 15 and RMEFNR2 15.

13	DSEMO 1000	IS1ES
14	DSEMO 1000	IS1ES
15	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
16	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS10ES
17	DSEMO 1000	IS1ES
18	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
19	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
20	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
	RMEFNR2 15	IS1ES
		IS3ES
		IS5ES
21	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS5ES
	RMEFNR2 15	IS1ES
		IS2ES
22	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
23	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
24	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
25	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS2ES
		IS3ES
	RMEFNR2 15	IS1ES
26	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
	RMEFNR2 15	IS1ES
27	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS2ES
	RMEFNR2 15	IS1ES
28	MEBPFNR3 15	IS1ES
	MEFNR3 15	IS2ES
		IS3ES
		IS5ES
		IS10ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
29	MEFNR3 15	IS3ES
	RMEFNR2 15	IS1ES
		IS2ES
		IS3ES
		IS5ES
		IS10ES

Table B.124: ISxES, 4-pier topology and 163 flights with operators C1P 1000, C2P 1000, DSEMO 1000, MEBPFNR3 15, MEFNR3 15 and RMEFNR2 15.

B.5 Probability Single Multi Operator

The following subsections provides the summaries of the statistical significance for the experiments conducted for Probability Single Multi Operator (PSMO), operator described in Section 5.4.3.

B.5.1 Statistical Results for Two Operators

Summary tables of the Mann-Whitney tests with significance level of 0.005 for the results when using an operator composed of two sub-operators and one single operator.

3-pier topology

16^{th} December 2009 with 194 flights

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 20% + C2P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 30% + C2P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 50% + C2P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 70% + C2P 30%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 80% + C2P 20%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 20% + C1P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 20% + C2P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 30% + C1P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 30% + C2P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 50% + C1P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 50% + C2P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 70% + C1P 30%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 70% + C2P 30%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 80% + C1P 20%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 80% + C2P 20%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C1P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.125: (a) Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 194 flights.

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
MEBPFNR3 20% + C1P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 20% + C2P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEBPFNR3 20% + DSEMO 80%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 30% + C1P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 30% + C2P 70%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEBPFNR3 30% + DSEMO 70%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEBPFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEBPFNR3 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEBPFNR3 50% + C1P 50%	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 50% + C2P 50%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEBPFNR3 50% + DSEMO 50%	0 (0, 0, 0)	3 (3, 0, 0)	1 (1, 0, 0)	2 (0, 0, 2)
MEBPFNR3 60% + C1P 40%	0 (0, 0, 0)	2 (0, 0, 2)	0 (0, 0, 0)	2 (0, 0, 2)
MEBPFNR3 60% + C2P 40%	0 (0, 0, 0)	1 (0, 0, 1)	1 (1, 0, 0)	2 (0, 0, 2)
MEBPFNR3 60% + DSEMO 40%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEBPFNR3 70% + C1P 30%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 70% + C2P 30%	0 (0, 0, 0)	3 (1, 0, 2)	2 (0, 0, 2)	1 (0, 0, 1)
MEBPFNR3 70% + DSEMO 30%	0 (0, 0, 0)	5 (4, 0, 1)	4 (3, 0, 1)	2 (0, 0, 2)
MEBPFNR3 80% + C1P 20%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 80% + C2P 20%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 80% + DSEMO 20%	0 (0, 0, 0)	4 (3, 0, 1)	3 (3, 0, 0)	2 (0, 0, 2)
MEBPFNR3 90% + C1P 10%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 90% + C2P 10%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 90% + DSEMO 10%	0 (0, 0, 0)	4 (2, 0, 2)	4 (3, 0, 1)	2 (0, 0, 2)
MEFNR3	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + MEBPFNR3 90%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	0 (0, 0, 0)
MEFNR3 10% + RMEFNR2 90%	0 (0, 0, 0)	4 (0, 2, 2)	1 (0, 0, 1)	6 (1, 3, 2)
MEFNR3 20% + C1P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 20% + C2P 80%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 20% + DSEMO 80%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 20% + MEBPFNR3 80%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEFNR3 30% + C1P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 30% + C2P 70%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 30% + DSEMO 70%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 30% + MEBPFNR3 70%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	1 (0, 0, 1)
MEFNR3 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 40% + DSEMO 60%	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + MEBPFNR3 60%	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)	0 (0, 0, 0)
MEFNR3 40% + RMEFNR2 60%	0 (0, 0, 0)	2 (0, 1, 1)	1 (0, 0, 1)	3 (0, 1, 2)
MEFNR3 50% + C1P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 50% + C2P 50%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEFNR3 50% + DSEMO 50%	0 (0, 0, 0)	1 (1, 0, 0)	2 (2, 0, 0)	1 (0, 0, 1)
MEFNR3 50% + MEBPFNR3 50%	0 (0, 0, 0)	2 (0, 0, 2)	1 (0, 0, 1)	2 (0, 0, 2)
MEFNR3 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 60% + C2P 40%	0 (0, 0, 0)	2 (1, 0, 1)	2 (1, 0, 1)	2 (0, 0, 2)
MEFNR3 60% + DSEMO 40%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEFNR3 60% + MEBPFNR3 40%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	0 (0, 0, 0)
MEFNR3 60% + RMEFNR2 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	3 (0, 1, 2)
MEFNR3 70% + C1P 30%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)

Table B.126: (b) Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 194 flights.

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
MEFNR3 70% + C2P 30%	0 (0, 0, 0)	1 (1, 0, 0)	2 (0, 0, 2)	1 (0, 0, 1)
MEFNR3 70% + DSEMO 30%	0 (0, 0, 0)	3 (2, 0, 1)	4 (3, 0, 1)	1 (0, 0, 1)
MEFNR3 70% + MEBPFNR3 30%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	1 (0, 0, 1)
MEFNR3 80% + C1P 20%	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)	1 (0, 0, 1)
MEFNR3 80% + C2P 20%	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 80% + DSEMO 20%	0 (0, 0, 0)	5 (4, 0, 1)	4 (4, 0, 0)	1 (0, 0, 1)
MEFNR3 80% + MEBPFNR3 20%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 90% + C1P 10%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 90% + C2P 10%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 90% + DSEMO 10%	0 (0, 0, 0)	5 (3, 0, 2)	1 (1, 0, 0)	2 (0, 0, 2)
MEFNR3 90% + MEBPFNR3 10%	0 (0, 0, 0)	3 (0, 1, 2)	2 (0, 0, 2)	0 (0, 0, 0)
MEFNR3 90% + RMEFNR2 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2	0 (0, 0, 0)	8 (0, **<u>5</u>**, **<u>3</u>**)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + MEBPFNR3 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
RMEFNR2 20% + C1P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 20% + C2P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 20% + DSEMO 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 20% + MEBPFNR3 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
RMEFNR2 30% + C1P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 30% + C2P 70%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
RMEFNR2 30% + DSEMO 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 30% + MEBPFNR3 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)
RMEFNR2 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)
RMEFNR2 40% + C2P 60%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	3 (0, 1, 2)
RMEFNR2 40% + DSEMO 60%	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)
RMEFNR2 40% + MEBPFNR3 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)
RMEFNR2 50% + C1P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)
RMEFNR2 50% + C2P 50%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	2 (0, 0, 2)
RMEFNR2 50% + DSEMO 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	3 (1, 0, 2)
RMEFNR2 50% + MEBPFNR3 50%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	3 (0, 1, 2)
RMEFNR2 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)	4 (1, 1, 2)
RMEFNR2 60% + C2P 40%	0 (0, 0, 0)	3 (1, 1, 1)	1 (1, 0, 0)	4 (1, 1, 2)
RMEFNR2 60% + DSEMO 40%	0 (0, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)	3 (1, 0, 2)
RMEFNR2 60% + MEBPFNR3 40%	0 (0, 0, 0)	2 (0, 2, 0)	1 (0, 0, 1)	3 (0, 1, 2)
RMEFNR2 70% + C1P 30%	0 (0, 0, 0)	3 (0, 2, 1)	2 (0, 0, 2)	3 (0, 1, 2)
RMEFNR2 70% + C2P 30%	0 (0, 0, 0)	3 (1, 1, 1)	3 (1, 0, 2)	4 (0, 2, 2)
RMEFNR2 70% + DSEMO 30%	0 (0, 0, 0)	4 (2, 0, 2)	1 (1, 0, 0)	4 (2, 0, 2)
RMEFNR2 70% + MEBPFNR3 30%	0 (0, 0, 0)	7 (1, 4, 2)	1 (0, 0, 1)	4 (1, 1, 2)
RMEFNR2 80% + C1P 20%	0 (0, 0, 0)	6 (0, 4, 2)	3 (0, 2, 1)	4 (1, 1, 2)
RMEFNR2 80% + C2P 20%	0 (0, 0, 0)	3 (0, 3, 0)	1 (1, 0, 0)	4 (1, 1, 2)
RMEFNR2 80% + DSEMO 20%	0 (0, 0, 0)	<u>11</u> (<u>6</u>, 3, 2)	2 (1, 0, 1)	4 (1, 1, 2)
RMEFNR2 80% + MEBPFNR3 20%	0 (0, 0, 0)	5 (0, 3, 2)	1 (0, 0, 1)	5 (1, 2, 2)
RMEFNR2 90% + C1P 10%	0 (0, 0, 0)	8 (1, 4, **<u>3</u>**)	3 (1, 1, 1)	4 (1, 1, 2)
RMEFNR2 90% + C2P 10%	0 (0, 0, 0)	9 (2, 4, **<u>3</u>**)	2 (1, 1, 0)	4 (1, 1, 2)
RMEFNR2 90% + DSEMO 10%	0 (0, 0, 0)	9 (5, 1, **<u>3</u>**)	4 (2, 0, 2)	4 (1, 1, 2)
RMEFNR2 90% + MEBPFNR3 10%	0 (0, 0, 0)	5 (1, 3, 1)	1 (0, 0, 1)	4 (0, 2, 2)

Table B.127: (c) Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 194 flights.

1^{st} March 2010 with 163 flights

Max. (6, 6, 5)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 10% + C2P 90%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 20% + C2P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 30% + C2P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 40% + C2P 60%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 50% + C2P 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 60% + C2P 40%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 70% + C2P 30%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 80% + C2P 20%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 90% + C2P 10%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 10% + C1P 90%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C2P 90%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 20% + C1P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 20% + C2P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 30% + C1P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 30% + C2P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 40% + C1P 60%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C2P 60%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 50% + C1P 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 50% + C2P 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 60% + C1P 40%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C2P 40%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 70% + C1P 30%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 70% + C2P 30%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 80% + C1P 20%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 80% + C2P 20%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 90% + C1P 10%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C2P 10%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	1 (0, 0, 1)	3 (0, 1, 2)	1 (0, 0, 1)	4 (0, 1, 3)
MEBPFNR3 10% + C1P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 10% + C2P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 10% + DSEMO 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 20% + C1P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 20% + C2P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 20% + DSEMO 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 30% + C1P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 30% + C2P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 30% + DSEMO 70%	1 (0, 0, 1)	1 (0, 0, 1)	2 (1, 0, 1)	1 (0, 0, 1)
MEBPFNR3 40% + C1P 60%	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 40% + C2P 60%	1 (0, 0, 1)	3 (1, 0, 2)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 40% + DSEMO 60%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 50% + C1P 50%	1 (0, 0, 1)	3 (0, 0, 3)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 50% + C2P 50%	1 (0, 0, 1)	3 (1, 0, 2)	5 (2, 0, 3)	3 (0, 1, 2)
MEBPFNR3 50% + DSEMO 50%	1 (0, 0, 1)	3 (2, 0, 1)	3 (2, 0, 1)	2 (0, 0, 2)
MEBPFNR3 60% + C1P 40%	1 (0, 0, 1)	3 (0, 0, 3)	2 (0, 0, 2)	3 (0, 1, 2)
MEBPFNR3 60% + C2P 40%	1 (0, 0, 1)	5 (2, 0, 3)	2 (1, 0, 1)	2 (0, 0, 2)
MEBPFNR3 60% + DSEMO 40%	1 (0, 0, 1)	3 (2, 0, 1)	4 (3, 0, 1)	2 (0, 0, 2)

Table B.128: (a) Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 163 flights.

Max. (6, 6, 5)	ES	**IS1ES**	IS1SUMS	SUMS
MEBPFNR3 70% + C1P 30%	1 (0, 0, 1)	5 (0, 1, 4)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 70% + C2P 30%	1 (0, 0, 1)	4 (1, 1, 2)	6 (2, 1, 3)	4 (0, 1, 3)
MEBPFNR3 70% + DSEMO 30%	1 (0, 0, 1)	3 (2, 0, 1)	6 (5, 0, 1)	2 (0, 0, 2)
MEBPFNR3 80% + C1P 20%	1 (0, 0, 1)	4 (0, 0, 4)	5 (0, 2, 3)	4 (0, 2, 2)
MEBPFNR3 80% + C2P 20%	1 (0, 0, 1)	6 (1, 0, $\underline{5}$)	5 (2, 1, 2)	3 (0, 1, 2)
MEBPFNR3 90% + DSEMO 10%	1 (0, 0, 1)	6 (5, 0, 1)	4 (3, 0, 1)	2 (0, 0, 2)
MEFNR3	1 (0, 0, 1)	3 (0, 1, 2)	4 (0, 2, 2)	5 (0, 2, 3)
MEFNR3 10% + C1P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 10% + C2P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 10% + DSEMO 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 10% + MEBPFNR3 90%	0 (0, 0, 0)	5 (0, 3, 2)	5 (0, 3, 2)	0 (0, 0, 0)
MEFNR3 10% + RMEFNR2 90%	0 (0, 0, 0)	2 (0, 0, 2)	1 (0, 0, 1)	3 (0, 0, 3)
MEFNR3 20% + C1P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 20% + C2P 80%	1 (0, 0, 1)	1 (0, 0, 1)	2 (1, 0, 1)	1 (0, 0, 1)
MEFNR3 20% + DSEMO 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 20% + MEBPFNR3 80%	1 (0, 0, 1)	3 (0, 1, 2)	3 (0, 2, 1)	1 (0, 0, 1)
MEFNR3 30% + C1P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 30% + C2P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 30% + DSEMO 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 30% + MEBPFNR3 70%	1 (0, 0, 1)	3 (0, 2, 1)	3 (0, 2, 1)	2 (0, 0, 2)
MEFNR3 40% + C1P 60%	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 40% + C2P 60%	1 (0, 0, 1)	2 (1, 0, 1)	3 (1, 0, 2)	3 (0, 1, 2)
MEFNR3 40% + DSEMO 60%	1 (0, 0, 1)	1 (0, 0, 1)	2 (1, 0, 1)	1 (0, 0, 1)
MEFNR3 40% + MEBPFNR3 60%	0 (0, 0, 0)	6 (0, 3, 3)	4 (0, 3, 1)	0 (0, 0, 0)
MEFNR3 40% + RMEFNR2 60%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	3 (0, 1, 2)
MEFNR3 50% + C1P 50%	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)	1 (0, 0, 1)
MEFNR3 50% + C2P 50%	1 (0, 0, 1)	1 (0, 0, 1)	2 (1, 0, 1)	2 (0, 0, 2)
MEFNR3 50% + DSEMO 50%	1 (0, 0, 1)	4 (3, 0, 1)	2 (1, 0, 1)	1 (0, 0, 1)
MEFNR3 50% + MEBPFNR3 50%	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 60% + C1P 40%	1 (0, 0, 1)	2 (0, 0, 2)	3 (0, 0, 3)	2 (0, 0, 2)
MEFNR3 60% + C2P 40%	1 (0, 0, 1)	5 (0, 1, 4)	4 (1, 1, 2)	2 (0, 0, 2)
MEFNR3 60% + DSEMO 40%	1 (0, 0, 1)	4 (3, 0, 1)	2 (1, 0, 1)	2 (0, 0, 2)
MEFNR3 60% + MEBPFNR3 40%	0 (0, 0, 0)	5 (0, 2, 3)	6 (0, 3, 3)	0 (0, 0, 0)
MEFNR3 60% + RMEFNR2 40%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEFNR3 70% + C1P 30%	1 (0, 0, 1)	4 (0, 1, 3)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 70% + C2P 30%	1 (0, 0, 1)	6 (2, 1, 3)	6 (1, 1, 4)	3 (0, 1, 2)
MEFNR3 70% + DSEMO 30%	1 (0, 0, 1)	$\underline{7}$ ($\underline{\mathbf{6}}$, 0, 1)	6 (5, 0, 1)	2 (0, 0, 2)
MEFNR3 70% + MEBPFNR3 30%	1 (0, 0, 1)	4 (0, 2, 2)	2 (0, 1, 1)	4 (0, 1, 3)
MEFNR3 80% + C1P 20%	1 (0, 0, 1)	6 (0, 1, $\underline{\mathbf{5}}$)	4 (0, 1, 3)	6 (0, 4, 2)
MEFNR3 80% + C2P 20%	1 (0, 0, 1)	6 (0, 2, 4)	3 (1, 0, 2)	3 (0, 1, 2)
MEFNR3 80% + DSEMO 20%	1 (0, 0, 1)	4 (3, 0, 1)	4 (3, 0, 1)	2 (0, 0, 2)
MEFNR3 80% + MEBPFNR3 20%	1 (0, 0, 1)	4 (0, 2, 2)	1 (0, 0, 1)	4 (0, 1, 3)
MEFNR3 90% + C1P 10%	1 (0, 0, 1)	6 (0, 1, $\underline{\mathbf{5}}$)	3 (0, 1, 2)	3 (0, 1, 2)
MEFNR3 90% + C2P 10%	1 (0, 0, 1)	5 (0, 2, 3)	4 (1, 1, 2)	5 (0, 3, 2)
MEFNR3 90% + DSEMO 10%	1 (0, 0, 1)	5 (3, 0, 2)	6 (5, 0, 1)	2 (0, 0, 2)
MEFNR3 90% + MEBPFNR3 10%	0 (0, 0, 0)	$\underline{7}$ (0, $\underline{5}$, 2)	5 (0, 4, 1)	0 (0, 0, 0)
MEFNR3 90% + RMEFNR2 10%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2	1 (0, 0, 1)	3 (0, 1, 2)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 10% + C1P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 10% + C2P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 10% + DSEMO 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)

Table B.129: (b) Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 163 flights.

Max. (6, 6, 5)	ES	IS1ES	IS1SUMS	SUMS
RMEFNR2 10% + MEBPFNR3 90%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 20% + C1P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 20% + C2P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 20% + DSEMO 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 20% + MEBPFNR3 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 30% + C1P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 30% + C2P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 30% + DSEMO 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 30% + MEBPFNR3 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 1, 1)
RMEFNR2 40% + C1P 60%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 40% + C2P 60%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 40% + DSEMO 60%	1 (0, 0, 1)	2 (1, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 40% + MEBPFNR3 60%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 1, 1)
RMEFNR2 50% + C1P 50%	1 (0, 0, 1)	2 (0, 0, 2)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 50% + C2P 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 50% + DSEMO 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 50% + MEBPFNR3 50%	1 (0, 0, 1)	2 (0, 1, 1)	1 (0, 0, 1)	5 (0, 2, 3)
RMEFNR2 60% + C1P 40%	1 (0, 0, 1)	2 (0, 0, 2)	3 (0, 0, 3)	2 (0, 0, 2)
RMEFNR2 60% + C2P 40%	1 (0, 0, 1)	3 (0, 0, 3)	2 (0, 0, 2)	1 (0, 0, 1)
RMEFNR2 60% + DSEMO 40%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 60% + MEBPFNR3 40%	0 (0, 0, 0)	2 (0, 1, 1)	1 (0, 0, 1)	5 (0, 2, 3)
RMEFNR2 70% + C1P 30%	1 (0, 0, 1)	3 (0, 1, 2)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 70% + C2P 30%	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
RMEFNR2 70% + DSEMO 30%	1 (0, 0, 1)	2 (1, 0, 1)	2 (1, 0, 1)	2 (0, 0, 2)
RMEFNR2 70% + MEBPFNR3 30%	1 (0, 0, 1)	2 (0, 1, 1)	1 (0, 0, 1)	4 (0, 1, 3)
RMEFNR2 80% + C1P 20%	1 (0, 0, 1)	4 (0, 1, 3)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 80% + C2P 20%	1 (0, 0, 1)	3 (0, 1, 2)	4 (0, 2, 2)	2 (0, 0, 2)
RMEFNR2 80% + DSEMO 20%	1 (0, 0, 1)	3 (1, 1, 1)	2 (1, 0, 1)	1 (0, 0, 1)
RMEFNR2 80% + MEBPFNR3 20%	1 (0, 0, 1)	3 (0, 1, 2)	1 (0, 0, 1)	3 (0, 0, 3)
RMEFNR2 90% + C1P 10%	1 (0, 0, 1)	3 (0, 0, 3)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 90% + C2P 10%	1 (0, 0, 1)	4 (0, 2, 2)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 90% + DSEMO 10%	1 (0, 0, 1)	2 (1, 0, 1)	2 (1, 0, 1)	2 (0, 0, 2)
RMEFNR2 90% + MEBPFNR3 10%	0 (0, 0, 0)	2 (0, 1, 1)	2 (0, 0, 2)	3 (0, 0, 3)

Table B.130: (c) Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 163 flights.

4-pier topology

16^{th} December 2009 with 194 flights

Max. (9, 5, 3)	ES	IS1ES	IS1SUMS	SUMS
C1P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C1P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 20% + DSEMO 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 30% + DSEMO 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + C1P 60%	0 (0, 0, 0)	1 (0, 1, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + DSEMO 60%	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 50% + C1P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 50% + DSEMO 50%	0 (0, 0, 0)	1 (1, 0, 0)	2 (2, 0, 0)	0 (0, 0, 0)
MEBPFNR3 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 60% + DSEMO 40%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)
MEBPFNR3 70% + C1P 30%	0 (0, 0, 0)	1 (0, 1, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 70% + DSEMO 30%	0 (0, 0, 0)	4 (4, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)
MEBPFNR3 80% + C1P 20%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 80% + DSEMO 20%	0 (0, 0, 0)	8 (8, 0, 0)	5 (5, 0, 0)	1 (1, 0, 0)
MEBPFNR3 90% + C1P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 90% + C2P 10%	0 (0, 0, 0)	2 (1, 0, 1)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 90% + DSEMO 10%	0 (0, 0, 0)	6 (6, 0, 0)	5 (5, 0, 0)	1 (1, 0, 0)
MEFNR3	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + MEBPFNR3 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + RMEFNR2 90%	0 (0, 0, 0)	4 (0, 3, 1)	1 (0, 0, 1)	2 (0, 2, 0)
MEFNR3 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + MEBPFNR3 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + C2P 40%	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.131: (a) Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 194 flights.

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
MEFNR3 40% + RMEFNR2 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + DSEMO 40%	0 (0, 0, 0)	2 (2, 0, 0)	2 (2, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + MEBPFNR3 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + RMEFNR2 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 30%	0 (0, 0, 0)	7 (7, 0, 0)	5 (5, 0, 0)	0 (0, 0, 0)
MEFNR3 90% + C1P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 90% + C2P 10%	0 (0, 0, 0)	2 (2, 0, 0)	2 (2, 0, 0)	0 (0, 0, 0)
MEFNR3 90% + DSEMO 10%	0 (0, 0, 0)	6 (6, 0, 0)	4 (4, 0, 0)	0 (0, 0, 0)
MEFNR3 90% + MEBPFNR3 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 90% + RMEFNR2 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2	0 (0, 0, 0)	3 (0, 1, 2)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + MEBPFNR3 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + MEBPFNR3 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 1, 0)
RMEFNR2 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 60% + DSEMO 40%	0 (0, 0, 0)	2 (2, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 60% + MEBPFNR3 40%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	1 (0, 1, 0)
RMEFNR2 90% + C1P 10%	0 (0, 0, 0)	5 (0, 3, 2)	4 (0, 2, 2)	1 (0, 1, 0)
RMEFNR2 90% + C2P 10%	0 (0, 0, 0)	**9** (1, **5**, **3**)	6 (2, 2, 2)	1 (0, 1, 0)
RMEFNR2 90% + DSEMO 10%	0 (0, 0, 0)	6 (3, 1, 2)	2 (2, 0, 0)	3 (2, 1, 0)
RMEFNR2 90% + MEBPFNR3 10%	0 (0, 0, 0)	4 (0, 2, 2)	2 (0, 0, 2)	1 (0, 1, 0)

Table B.132: (b) Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 194 flights.

1^{st} March 2010 with 163 flights

Max. (6, 6, 5)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C1P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	0 (0, 0, 0)	5 (0, 3, 2)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 20% + DSEMO 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 30% + DSEMO 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 50% + DSEMO 50%	0 (0, 0, 0)	2 (2, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 60% + C1P 40%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 60% + C2P 40%	0 (0, 0, 0)	2 (1, 0, 1)	2 (1, 0, 1)	1 (0, 1, 0)
MEBPFNR3 60% + DSEMO 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 70% + C1P 30%	0 (0, 0, 0)	4 (0, 2, 2)	1 (0, 0, 1)	1 (0, 1, 0)
MEBPFNR3 70% + DSEMO 30%	0 (0, 0, 0)	2 (1, 1, 0)	2 (2, 0, 0)	0 (0, 0, 0)
MEBPFNR3 80% + C1P 20%	0 (0, 0, 0)	4 (0, 1, 3)	3 (0, 1, 2)	1 (0, 1, 0)
MEBPFNR3 80% + DSEMO 20%	0 (0, 0, 0)	5 (4, 1, 0)	3 (3, 0, 0)	1 (0, 1, 0)
MEBPFNR3 90% + C1P 10%	0 (0, 0, 0)	9 (0, **6**, 3)	3 (0, 2, 1)	2 (0, 1, 1)
MEBPFNR3 90% + C2P 10%	0 (0, 0, 0)	6 (1, 3, 2)	3 (0, 2, 1)	1 (0, 1, 0)
MEBPFNR3 90% + DSEMO 10%	0 (0, 0, 0)	4 (3, 1, 0)	3 (3, 0, 0)	0 (0, 0, 0)
MEFNR3	0 (0, 0, 0)	2 (0, 2, 0)	0 (0, 0, 0)	1 (0, 1, 0)
MEFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + MEBPFNR3 90%	0 (0, 0, 0)	5 (0, 5, 0)	1 (0, 1, 0)	0 (0, 0, 0)
MEFNR3 10% + RMEFNR2 90%	0 (0, 0, 0)	3 (0, 1, 2)	2 (0, 1, 1)	4 (0, 1, 3)
MEFNR3 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + MEBPFNR3 60%	0 (0, 0, 0)	5 (0, 5, 0)	3 (0, 3, 0)	0 (0, 0, 0)
MEFNR3 40% + RMEFNR2 60%	0 (0, 0, 0)	4 (0, 3, 1)	2 (0, 1, 1)	5 (0, 2, 3)
MEFNR3 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)
MEFNR3 60% + C2P 40%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + DSEMO 40%	0 (0, 0, 0)	2 (2, 0, 0)	4 (4, 0, 0)	0 (0, 0, 0)

Table B.133: (a) Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 163 flights.

Max. (6, 6, 5)	ES	IS1ES	IS1SUMS	SUMS
MEFNR3 60% + MEBPFNR3 40%	0 (0, 0, 0)	4 (0, 3, 1)	5 (0, 5, 0)	0 (0, 0, 0)
MEFNR3 60% + RMEFNR2 40%	0 (0, 0, 0)	1 (0, 1, 0)	0 (0, 0, 0)	3 (0, 1, 2)
MEFNR3 70% + DSEMO 30%	0 (0, 0, 0)	3 (3, 0, 0)	5 (4, 1, 0)	0 (0, 0, 0)
MEFNR3 90% + C1P 10%	0 (0, 0, 0)	3 (0, 1, 2)	0 (0, 0, 0)	1 (0, 1, 0)
MEFNR3 90% + C2P 10%	0 (0, 0, 0)	4 (2, 2, 0)	4 (2, 2, 0)	1 (0, 1, 0)
MEFNR3 90% + DSEMO 10%	0 (0, 0, 0)	1 (1, 0, 0)	3 (3, 0, 0)	1 (0, 1, 0)
MEFNR3 90% + MEBPFNR3 10%	0 (0, 0, 0)	4 (0, 3, 1)	5 (0, 5, 0)	0 (0, 0, 0)
MEFNR3 90% + RMEFNR2 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 1, 0)
RMEFNR2	0 (0, 0, 0)	3 (0, 1, 2)	1 (0, 1, 0)	0 (0, 0, 0)
RMEFNR2 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + MEBPFNR3 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 2, 0)
RMEFNR2 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + MEBPFNR3 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	4 (0, 2, 2)
RMEFNR2 60% + C1P 40%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)
RMEFNR2 60% + DSEMO 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 60% + MEBPFNR3 40%	0 (0, 0, 0)	1 (0, 1, 0)	2 (0, 1, 1)	7 (0, 3, 4)
RMEFNR2 90% + C1P 10%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	1 (0, 0, 1)
RMEFNR2 90% + C2P 10%	0 (0, 0, 0)	3 (0, 1, 2)	2 (0, 1, 1)	1 (0, 0, 1)
RMEFNR2 90% + DSEMO 10%	0 (0, 0, 0)	2 (0, 0, 2)	1 (0, 1, 0)	0 (0, 0, 0)
RMEFNR2 90% + MEBPFNR3 10%	0 (0, 0, 0)	2 (0, 2, 0)	2 (0, 1, 1)	2 (0, 0, 2)

Table B.134: (b) Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 163 flights.

B.5.2 Statistical Results for Three Operators

3-pier topology

16^{th} December 2009 with 194 flights

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	3 (1, 0, 2)	6 (4, 0, 2)	2 (1, 0, 1)
MEBPFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	7 (5, 0, 2)	5 (3, 0, 2)	3 (1, 0, 2)
MEFNR3	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 70% + DSEMO 20% + C1P 10%	0 (0, 0, 0)	6 (4, 0, 2)	6 (4, 0, 2)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 20% + C2P 10%	0 (0, 0, 0)	7 (5, 0, 2)	4 (3, 0, 1)	0 (0, 0, 0)
MEFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	6 (4, 0, 2)	4 (2, 0, 2)	2 (0, 0, 2)
MEFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	6 (4, 0, 2)	3 (2, 0, 1)	2 (1, 0, 1)
RMEFNR2	0 (0, 0, 0)	11 (3, **5**, 3)	2 (0, 1, 1)	3 (0, 1, 2)
RMEFNR2 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	14 (6, *5*, *3*)	7 (4, 2, 1)	8 (2, 4, 2)
RMEFNR2 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	9 (3, 3, *3*)	5 (3, 0, 2)	7 (2, 3, 2)

Table B.135: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 194 flights.

1^{st} March 2010 with 163 flights

Max. (6, 6, 5)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C2P	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3	1 (0, 0, 1)	4 (0, 2, 2)	2 (0, 1, 1)	5 (0, 2, 3)
MEBPFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	12 (3, 5, 4)	8 (3, 1, 4)	6 (1, 2, 3)
MEBPFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	9 (3, 2, 4)	7 (4, 2, 1)	5 (1, 2, 2)
MEFNR3	1 (0, 0, 1)	5 (0, 3, 2)	5 (0, 3, 2)	7 (0, 4, 3)
MEFNR3 70% + DSEMO 20% + C1P 10%	0 (0, 0, 0)	12 (4, 4, 4)	8 (4, 2, 2)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 20% + C2P 10%	1 (0, 0, 1)	13 (5, 4, 4)	5 (2, 1, 2)	0 (0, 0, 0)
MEFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	10 (2, 3, **5**)	5 (0, 3, 2)	5 (0, 2, 3)
MEFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	9 (4, 3, 2)	9 (4, 2, 3)	4 (2, 0, 2)
RMEFNR2	1 (0, 0, 1)	4 (0, 2, 2)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	6 (0, 2, 4)	1 (0, 0, 1)	3 (0, 0, 3)
RMEFNR2 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	7 (0, 2, **5**)	1 (0, 0, 1)	2 (0, 0, 2)

Table B.136: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 163 flights.

4-pier topology

16^{th} December 2009 with 194 flights

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 1, 0)
MEBPFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	10 (<u>7</u>, 3, 0)	9 (7, 2, 0)	2 (1, 1, 0)
MEBPFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	7 (4, 3, 0)	7 (6, 1, 0)	3 (2, 1, 0)
MEFNR3	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 1, 0)
MEFNR3 70% + DSEMO 20% + C1P 10%	0 (0, 0, 0)	7 (6, 1, 0)	5 (4, 1, 0)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 20% + C2P 10%	0 (0, 0, 0)	6 (5, 1, 0)	4 (4, 0, 0)	0 (0, 0, 0)
MEFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	6 (5, 1, 0)	1 (1, 0, 0)	2 (1, 1, 0)
MEFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	7 (5, 2, 0)	4 (4, 0, 0)	1 (1, 0, 0)
RMEFNR2	0 (0, 0, 0)	7 (0, <u>**5**</u>, 2)	0 (0, 0, 0)	1 (0, 1, 0)
RMEFNR2 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	8 (2, 4, 2)	4 (1, 2, 1)	2 (0, 2, 0)
RMEFNR2 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	<u>11</u> (3, <u>**5**</u>, <u>**3**</u>)	4 (1, 2, 1)	3 (2, 1, 0)

Table B.137: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 194 flights.

1^{st} March 2010 with 163 flights

Max. (6, 6, 5)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	0 (0, 0, 0)	7 (0, 4, 3)	3 (0, 2, 1)	4 (0, 2, 2)
MEBPFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	<u>14</u> (4, <u>**6**</u>, 4)	10 (4, 3, 3)	3 (0, 2, 1)
MEBPFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	<u>14</u> (<u>**6**</u>, <u>**6**</u>, 2)	9 (5, 3, 1)	4 (1, 1, 2)
MEFNR3	0 (0, 0, 0)	5 (0, 5, 0)	2 (0, 2, 0)	3 (0, 3, 0)
MEFNR3 70% + DSEMO 20% + C1P 10%	0 (0, 0, 0)	13 (5, 4, 4)	8 (4, 2, 2)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 20% + C2P 10%	0 (0, 0, 0)	<u>14</u> (<u>**6**</u>, 5, 3)	7 (5, 1, 1)	0 (0, 0, 0)
MEFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	12 (5, 3, 4)	6 (3, 3, 0)	1 (0, 1, 0)
MEFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	10 (4, 4, 2)	7 (3, 2, 2)	2 (1, 1, 0)
RMEFNR2	0 (0, 0, 0)	7 (0, 2, <u>**5**</u>)	4 (0, 1, 3)	1 (0, 0, 1)
RMEFNR2 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	4 (0, 1, 3)	2 (0, 0, 2)	2 (0, 0, 2)
RMEFNR2 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	5 (0, 2, 3)	3 (0, 0, 3)	2 (0, 0, 2)

Table B.138: Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 163 flights.

B.5.3 Statistical Results for One, Two and Three Operators

The tables in this section summarise all of the result for 'Probability Single Multi Operator' (Section 5.4.3) composed of two and three operators together with the single operator for SSEA with $\ell = 1$.

The tables used in the following subsections are a summary of the statistical significance tests conducted which show between parenthesis the number of instances and separated by a comma, where the combined operator cannot be said to be statistically significantly worse than any of the other operators studied here for each of the number of BSSs grouped into ranges of $N <$ LMAP, LMAP $\leq N <$ UMAP and UMAP $\leq N$, ranges which are separated by a comma, e.g. (1, 3, 0) means that there are 1, 3 and 3 within the ranges of $N <$ LMAP, LMAP $\leq N <$ UMAP and UMAP $\leq N$ respectively where the operators cannot be said to be less fit than any of the other operators. Furthermore, in the headers of the table starting with 'Max.' and between parenthesis it is specified the number of set of BSSs part of the range which only depend on the data set, e.g. for the data set of 16^{th} December 2009 there are 9 instances where $N <$ LMAP, 5 with LMAP $\leq N <$ UMAP, and 3 with UMAP $\geq N$. These values give an idea of how much an operator covers a range of numbers of BSSs, where full coverage happens when the number in the 'Max.' for the range is the same than for the operator, which in the example considered up to now happens only for the last range, where UMAP $\leq N$.

3-pier topology

16^{th} December 2009 with 194 flights

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 20% + C2P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 30% + C2P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 50% + C2P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 70% + C2P 30%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 80% + C2P 20%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 20% + C1P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 20% + C2P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 30% + C1P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 30% + C2P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 50% + C1P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 50% + C2P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 70% + C1P 30%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 70% + C2P 30%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.139: (a). Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 194 flights.

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
DSEMO 80% + C1P 20%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 80% + C2P 20%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C1P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 20% + C1P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 20% + C2P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEBPFNR3 20% + DSEMO 80%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 30% + C1P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 30% + C2P 70%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEBPFNR3 30% + DSEMO 70%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEBPFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEBPFNR3 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEBPFNR3 50% + C1P 50%	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 50% + C2P 50%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEBPFNR3 50% + DSEMO 50%	0 (0, 0, 0)	3 (3, 0, 0)	1 (1, 0, 0)	2 (0, 0, 2)
MEBPFNR3 60% + C1P 40%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	2 (0, 0, 2)
MEBPFNR3 60% + C2P 40%	0 (0, 0, 0)	1 (0, 0, 1)	1 (1, 0, 0)	2 (0, 0, 2)
MEBPFNR3 60% + DSEMO 40%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEBPFNR3 70% + C1P 30%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 70% + C2P 30%	0 (0, 0, 0)	3 (1, 0, 2)	2 (0, 0, 2)	1 (0, 0, 1)
MEBPFNR3 70% + DSEMO 30%	0 (0, 0, 0)	5 (4, 0, 1)	4 (3, 0, 1)	2 (0, 0, 2)
MEBPFNR3 80% + C1P 20%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 80% + C2P 20%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	3 (1, 0, 2)	2 (0, 0, 2)	2 (1, 0, 1)
MEBPFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	3 (1, 0, 2)	3 (1, 0, 2)	3 (1, 0, 2)
MEBPFNR3 80% + DSEMO 20%	0 (0, 0, 0)	4 (3, 0, 1)	3 (3, 0, 0)	2 (0, 0, 2)
MEBPFNR3 90% + C1P 10%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 90% + C2P 10%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 90% + DSEMO 10%	0 (0, 0, 0)	4 (2, 0, 2)	4 (3, 0, 1)	2 (0, 0, 2)
MEFNR3	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + MEBPFNR3 90%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	0 (0, 0, 0)
MEFNR3 10% + RMEFNR2 90%	0 (0, 0, 0)	4 (0, 2, 2)	1 (0, 0, 1)	6 (1, 3, 2)
MEFNR3 20% + C1P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 20% + C2P 80%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 20% + DSEMO 80%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 20% + MEBPFNR3 80%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEFNR3 30% + C1P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 30% + C2P 70%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 30% + DSEMO 70%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 30% + MEBPFNR3 70%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	1 (0, 0, 1)
MEFNR3 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 40% + DSEMO 60%	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + MEBPFNR3 60%	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)	0 (0, 0, 0)
MEFNR3 40% + RMEFNR2 60%	0 (0, 0, 0)	2 (0, 1, 1)	1 (0, 0, 1)	3 (0, 1, 2)
MEFNR3 50% + C1P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 50% + C2P 50%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
MEFNR3 50% + DSEMO 50%	0 (0, 0, 0)	1 (1, 0, 0)	2 (2, 0, 0)	1 (0, 0, 1)
MEFNR3 50% + MEBPFNR3 50%	0 (0, 0, 0)	2 (0, 0, 2)	1 (0, 0, 1)	2 (0, 0, 2)
MEFNR3 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 60% + C2P 40%	0 (0, 0, 0)	2 (1, 0, 1)	2 (1, 0, 1)	2 (0, 0, 2)
MEFNR3 60% + DSEMO 40%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)

Table B.140: (b). Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 194 flights.

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
MEFNR3 60% + MEBPFNR3 40%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	0 (0, 0, 0)
MEFNR3 60% + RMEFNR2 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	3 (0, 1, 2)
MEFNR3 70% + C1P 30%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
MEFNR3 70% + C2P 30%	0 (0, 0, 0)	1 (1, 0, 0)	2 (0, 0, 2)	1 (0, 0, 1)
MEFNR3 70% + DSEMO 20% + C1P 10%	0 (0, 0, 0)	2 (0, 0, 2)	3 (2, 0, 1)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 20% + C2P 10%	0 (0, 0, 0)	2 (0, 0, 2)	1 (0, 0, 1)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 30%	0 (0, 0, 0)	3 (2, 0, 1)	4 (3, 0, 1)	1 (0, 0, 1)
MEFNR3 70% + MEBPFNR3 30%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	1 (0, 0, 1)
MEFNR3 80% + C1P 20%	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)	1 (0, 0, 1)
MEFNR3 80% + C2P 20%	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	3 (1, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	2 (0, 0, 2)	1 (0, 0, 1)	2 (1, 0, 1)
MEFNR3 80% + DSEMO 20%	0 (0, 0, 0)	5 (4, 0, 1)	4 (4, 0, 0)	1 (0, 0, 1)
MEFNR3 80% + MEBPFNR3 20%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 90% + C1P 10%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 90% + C2P 10%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 90% + DSEMO 10%	0 (0, 0, 0)	5 (3, 0, 2)	1 (1, 0, 0)	2 (0, 0, 2)
MEFNR3 90% + MEBPFNR3 10%	0 (0, 0, 0)	3 (0, 1, 2)	2 (0, 0, 2)	0 (0, 0, 0)
MEFNR3 90% + RMEFNR2 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2	0 (0, 0, 0)	8 (0, **5**, **3**)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + MEBPFNR3 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
RMEFNR2 20% + C1P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 20% + C2P 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 20% + DSEMO 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 20% + MEBPFNR3 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)
RMEFNR2 30% + C1P 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 30% + C2P 70%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	1 (0, 0, 1)
RMEFNR2 30% + DSEMO 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 30% + MEBPFNR3 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)
RMEFNR2 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)
RMEFNR2 40% + C2P 60%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	3 (0, 1, 2)
RMEFNR2 40% + DSEMO 60%	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)
RMEFNR2 40% + MEBPFNR3 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)
RMEFNR2 50% + C1P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 0, 2)
RMEFNR2 50% + C2P 50%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	2 (0, 0, 2)
RMEFNR2 50% + DSEMO 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	3 (1, 0, 2)
RMEFNR2 50% + MEBPFNR3 50%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	3 (0, 1, 2)
RMEFNR2 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)	4 (1, 1, 2)
RMEFNR2 60% + C2P 40%	0 (0, 0, 0)	3 (1, 1, 1)	1 (1, 0, 0)	4 (1, 1, 2)
RMEFNR2 60% + DSEMO 40%	0 (0, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)	3 (1, 0, 2)
RMEFNR2 60% + MEBPFNR3 40%	0 (0, 0, 0)	2 (0, 2, 0)	1 (0, 0, 1)	3 (0, 1, 2)
RMEFNR2 70% + C1P 30%	0 (0, 0, 0)	2 (0, 1, 1)	2 (0, 0, 2)	3 (0, 1, 2)
RMEFNR2 70% + C2P 30%	0 (0, 0, 0)	2 (1, 0, 1)	3 (1, 0, 2)	3 (0, 1, 2)
RMEFNR2 70% + DSEMO 30%	0 (0, 0, 0)	4 (2, 0, 2)	1 (1, 0, 0)	4 (2, 0, 2)
RMEFNR2 70% + MEBPFNR3 30%	0 (0, 0, 0)	6 (1, 3, 2)	1 (0, 0, 1)	4 (1, 1, 2)
RMEFNR2 80% + C1P 20%	0 (0, 0, 0)	5 (0, 3, 2)	3 (0, 2, 1)	4 (1, 1, 2)
RMEFNR2 80% + C2P 20%	0 (0, 0, 0)	3 (0, 3, 0)	1 (1, 0, 0)	4 (1, 1, 2)
RMEFNR2 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	8 (2, 4, 2)	2 (1, 0, 1)	5 (2, 1, 2)
RMEFNR2 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	7 (1, 3, **3**)	1 (0, 0, 1)	4 (2, 0, 2)
RMEFNR2 80% + DSEMO 20%	0 (0, 0, 0)	11 (6, 3, 2)	2 (1, 0, 1)	4 (1, 1, 2)
RMEFNR2 80% + MEBPFNR3 20%	0 (0, 0, 0)	5 (0, 3, 2)	1 (0, 0, 1)	5 (1, 2, 2)
RMEFNR2 90% + C1P 10%	0 (0, 0, 0)	8 (1, 4, **3**)	3 (1, 1, 1)	4 (1, 1, 2)
RMEFNR2 90% + C2P 10%	0 (0, 0, 0)	9 (2, 4, **3**)	2 (1, 1, 0)	4 (1, 1, 2)
RMEFNR2 90% + DSEMO 10%	0 (0, 0, 0)	9 (5, 1, **3**)	4 (2, 0, 2)	4 (1, 1, 2)
RMEFNR2 90% + MEBPFNR3 10%	0 (0, 0, 0)	5 (1, 3, 1)	1 (0, 0, 1)	4 (0, 2, 2)

Table B.141: (c). Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 194 flights.

1^{st} March 2010 with 163 flights

Max. (6, 6, 5)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 10% + C2P 90%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 20% + C2P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 30% + C2P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 40% + C2P 60%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 50% + C2P 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 60% + C2P 40%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 70% + C2P 30%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 80% + C2P 20%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
C1P 90% + C2P 10%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 10% + C1P 90%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C2P 90%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 20% + C1P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 20% + C2P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 30% + C1P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 30% + C2P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 40% + C1P 60%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C2P 60%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 50% + C1P 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 50% + C2P 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 60% + C1P 40%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C2P 40%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 70% + C1P 30%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 70% + C2P 30%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 80% + C1P 20%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 80% + C2P 20%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
DSEMO 90% + C1P 10%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C2P 10%	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	1 (0, 0, 1)	3 (0, 1, 2)	1 (0, 0, 1)	4 (0, 1, 3)
MEBPFNR3 10% + C1P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 10% + C2P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 10% + DSEMO 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 20% + C1P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 20% + C2P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 20% + DSEMO 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 30% + C1P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 30% + C2P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 30% + DSEMO 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEBPFNR3 40% + C1P 60%	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 40% + C2P 60%	1 (0, 0, 1)	2 (0, 0, 2)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 40% + DSEMO 60%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 50% + C1P 50%	1 (0, 0, 1)	3 (0, 0, 3)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 50% + C2P 50%	1 (0, 0, 1)	2 (0, 0, 2)	3 (0, 0, 3)	3 (0, 1, 2)
MEBPFNR3 50% + DSEMO 50%	1 (0, 0, 1)	1 (0, 0, 1)	2 (1, 0, 1)	2 (0, 0, 2)
MEBPFNR3 60% + C1P 40%	1 (0, 0, 1)	3 (0, 0, 3)	2 (0, 0, 2)	3 (0, 1, 2)
MEBPFNR3 60% + C2P 40%	1 (0, 0, 1)	3 (0, 0, 3)	1 (0, 0, 1)	2 (0, 0, 2)
MEBPFNR3 60% + DSEMO 40%	1 (0, 0, 1)	1 (0, 0, 1)	2 (1, 0, 1)	2 (0, 0, 2)
MEBPFNR3 70% + C1P 30%	1 (0, 0, 1)	5 (0, 1, 4)	2 (0, 0, 2)	2 (0, 0, 2)
MEBPFNR3 70% + C2P 30%	1 (0, 0, 1)	3 (0, 1, 2)	4 (0, 1, 3)	4 (0, 1, 3)
MEBPFNR3 70% + DSEMO 30%	1 (0, 0, 1)	3 (2, 0, 1)	4 (3, 0, 1)	2 (0, 0, 2)
MEBPFNR3 80% + C1P 20%	1 (0, 0, 1)	4 (0, 0, 4)	5 (0, 2, 3)	3 (0, 1, 2)
MEBPFNR3 80% + C2P 20%	1 (0, 0, 1)	5 (0, 0, **<u>5</u>**)	3 (0, 1, 2)	3 (0, 1, 2)
MEBPFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	8 (2, 3, 3)	3 (0, 0, 3)	3 (0, 1, 2)
MEBPFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	7 (2, 2, 3)	4 (2, 1, 1)	2 (0, 0, 2)
MEBPFNR3 80% + DSEMO 20%	1 (0, 0, 1)	4 (3, 0, 1)	5 (4, 0, 1)	2 (0, 0, 2)
MEBPFNR3 90% + C1P 10%	1 (0, 0, 1)	7 (0, 3, 4)	3 (0, 0, 3)	3 (0, 1, 2)

Table B.142: (a). Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 163 flights.

Max. (6, 6, 5)	ES	**IS1ES**	IS1SUMS	SUMS
MEBPFNR3 90% + C2P 10%	1 (0, 0, 1)	5 (0, 3, 2)	2 (0, 0, 2)	6 (0, 4, 2)
MEBPFNR3 90% + DSEMO 10%	1 (0, 0, 1)	4 (3, 0, 1)	3 (2, 0, 1)	2 (0, 0, 2)
MEFNR3	1 (0, 0, 1)	3 (0, 1, 2)	4 (0, 2, 2)	5 (0, 2, 3)
MEFNR3 10% + C1P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 10% + C2P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 10% + DSEMO 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 10% + MEBPFNR3 90%	0 (0, 0, 0)	5 (0, 3, 2)	5 (0, 3, 2)	0 (0, 0, 0)
MEFNR3 10% + RMEFNR2 90%	0 (0, 0, 0)	2 (0, 0, 2)	1 (0, 0, 1)	3 (0, 0, 3)
MEFNR3 20% + C1P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 20% + C2P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 20% + DSEMO 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 20% + MEBPFNR3 80%	1 (0, 0, 1)	3 (0, 1, 2)	3 (0, 2, 1)	1 (0, 0, 1)
MEFNR3 30% + C1P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 30% + C2P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 30% + DSEMO 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 30% + MEBPFNR3 70%	1 (0, 0, 1)	3 (0, 2, 1)	3 (0, 2, 1)	2 (0, 0, 2)
MEFNR3 40% + C1P 60%	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 40% + C2P 60%	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)	3 (0, 1, 2)
MEFNR3 40% + DSEMO 60%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 40% + MEBPFNR3 60%	0 (0, 0, 0)	6 (0, 3, 3)	4 (0, 3, 1)	0 (0, 0, 0)
MEFNR3 40% + RMEFNR2 60%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	3 (0, 1, 2)
MEFNR3 50% + C1P 50%	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)	1 (0, 0, 1)
MEFNR3 50% + C2P 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEFNR3 50% + DSEMO 50%	1 (0, 0, 1)	2 (1, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
MEFNR3 50% + MEBPFNR3 50%	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 60% + C1P 40%	1 (0, 0, 1)	2 (0, 0, 2)	3 (0, 0, 3)	2 (0, 0, 2)
MEFNR3 60% + C2P 40%	1 (0, 0, 1)	5 (0, 1, 4)	3 (0, 1, 2)	2 (0, 0, 2)
MEFNR3 60% + DSEMO 40%	1 (0, 0, 1)	2 (1, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEFNR3 60% + MEBPFNR3 40%	0 (0, 0, 0)	5 (0, 2, 3)	6 (0, 3, 3)	0 (0, 0, 0)
MEFNR3 60% + RMEFNR2 40%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
MEFNR3 70% + C1P 30%	1 (0, 0, 1)	4 (0, 1, 3)	2 (0, 0, 2)	2 (0, 0, 2)
MEFNR3 70% + C2P 30%	1 (0, 0, 1)	4 (0, 1, 3)	5 (0, 1, 4)	3 (0, 1, 2)
MEFNR3 70% + DSEMO 20% + C1P 10%	0 (0, 0, 0)	8 (2, 4, 2)	3 (1, 1, 1)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 20% + C2P 10%	1 (0, 0, 1)	8 (2, 3, 3)	2 (0, 0, 2)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 30%	1 (0, 0, 1)	5 (<u>4</u>, 0, 1)	4 (3, 0, 1)	2 (0, 0, 2)
MEFNR3 70% + MEBPFNR3 30%	1 (0, 0, 1)	4 (0, 2, 2)	2 (0, 1, 1)	4 (0, 1, 3)
MEFNR3 80% + C1P 20%	1 (0, 0, 1)	6 (0, 1, <u>**5**</u>)	4 (0, 1, 3)	6 (0, 4, 2)
MEFNR3 80% + C2P 20%	1 (0, 0, 1)	6 (0, 2, 4)	2 (0, 0, 2)	3 (0, 1, 2)
MEFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	<u>9</u> (2, 3, 4)	3 (0, 1, 2)	4 (0, 1, 3)
MEFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	4 (2, 1, 1)	6 (2, 2, 2)	2 (0, 0, 2)
MEFNR3 80% + DSEMO 20%	1 (0, 0, 1)	4 (3, 0, 1)	4 (3, 0, 1)	2 (0, 0, 2)
MEFNR3 80% + MEBPFNR3 20%	1 (0, 0, 1)	4 (0, 2, 2)	1 (0, 0, 1)	4 (0, 1, 3)
MEFNR3 90% + C1P 10%	1 (0, 0, 1)	6 (0, 1, <u>**5**</u>)	3 (0, 1, 2)	3 (0, 1, 2)
MEFNR3 90% + C2P 10%	1 (0, 0, 1)	5 (0, 2, 3)	3 (0, 1, 2)	5 (0, 3, 2)
MEFNR3 90% + DSEMO 10%	1 (0, 0, 1)	5 (3, 0, 2)	5 (<u>4</u>, 0, 1)	2 (0, 0, 2)
MEFNR3 90% + MEBPFNR3 10%	0 (0, 0, 0)	7 (0, <u>5</u>, 2)	5 (0, 4, 1)	0 (0, 0, 0)
MEFNR3 90% + RMEFNR2 10%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2	1 (0, 0, 1)	2 (0, 0, 2)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 10% + C1P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 10% + C2P 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 10% + DSEMO 90%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 10% + MEBPFNR3 90%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 20% + C1P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 20% + C2P 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 20% + DSEMO 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 20% + MEBPFNR3 80%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 30% + C1P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 30% + C2P 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)

Table B.143: (b). Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 163 flights.

Max. (6, 6, 5)	ES	**IS1ES**	IS1SUMS	SUMS
RMEFNR2 30% + DSEMO 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 30% + MEBPFNR3 70%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 1, 1)
RMEFNR2 40% + C1P 60%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 40% + C2P 60%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 40% + DSEMO 60%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 40% + MEBPFNR3 60%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 50% + C1P 50%	1 (0, 0, 1)	2 (0, 0, 2)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 50% + C2P 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 50% + DSEMO 50%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 50% + MEBPFNR3 50%	1 (0, 0, 1)	2 (0, 1, 1)	1 (0, 0, 1)	5 (0, 2, 3)
RMEFNR2 60% + C1P 40%	1 (0, 0, 1)	2 (0, 0, 2)	3 (0, 0, 3)	2 (0, 0, 2)
RMEFNR2 60% + C2P 40%	1 (0, 0, 1)	3 (0, 0, 3)	2 (0, 0, 2)	1 (0, 0, 1)
RMEFNR2 60% + DSEMO 40%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 60% + MEBPFNR3 40%	0 (0, 0, 0)	2 (0, 1, 1)	1 (0, 0, 1)	5 (0, 2, 3)
RMEFNR2 70% + C1P 30%	1 (0, 0, 1)	2 (0, 0, 2)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 70% + C2P 30%	1 (0, 0, 1)	2 (0, 0, 2)	2 (0, 0, 2)	2 (0, 0, 2)
RMEFNR2 70% + DSEMO 30%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 70% + MEBPFNR3 30%	1 (0, 0, 1)	2 (0, 1, 1)	1 (0, 0, 1)	4 (0, 1, 3)
RMEFNR2 80% + C1P 20%	1 (0, 0, 1)	4 (0, 1, 3)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 80% + C2P 20%	1 (0, 0, 1)	2 (0, 0, 2)	3 (0, 1, 2)	2 (0, 0, 2)
RMEFNR2 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	3 (0, 1, 2)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	5 (0, 1, 4)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 80% + DSEMO 20%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)
RMEFNR2 80% + MEBPFNR3 20%	1 (0, 0, 1)	3 (0, 1, 2)	1 (0, 0, 1)	3 (0, 0, 3)
RMEFNR2 90% + C1P 10%	1 (0, 0, 1)	3 (0, 0, 3)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 90% + C2P 10%	1 (0, 0, 1)	4 (0, 2, 2)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 90% + DSEMO 10%	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 90% + MEBPFNR3 10%	0 (0, 0, 0)	1 (0, 0, 1)	2 (0, 0, 2)	3 (0, 0, 3)

Table B.144: (c). Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 3-pier topology and 163 flights.

4-pier topology

16^{th} December 2009 with 194 flights

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C1P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 20% + DSEMO 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 30% + DSEMO 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)

Table B.145: (a). Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 194 flights.

Max. (9, 5, 3)	ES	**IS1ES**	IS1SUMS	SUMS
MEBPFNR3 40% + C1P 60%	0 (0, 0, 0)	1 (0, 1, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + DSEMO 60%	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	3 (2, 1, 0)	2 (2, 0, 0)	0 (0, 0, 0)
MEBPFNR3 50% + C1P 50%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 50% + DSEMO 50%	0 (0, 0, 0)	1 (1, 0, 0)	2 (2, 0, 0)	0 (0, 0, 0)
MEBPFNR3 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 60% + DSEMO 40%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 70% + C1P 30%	0 (0, 0, 0)	1 (0, 1, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 70% + DSEMO 30%	0 (0, 0, 0)	3 (3, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)
MEBPFNR3 80% + C1P 20%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	1 (1, 0, 0)	3 (3, 0, 0)	2 (2, 0, 0)
MEBPFNR3 80% + DSEMO 20%	0 (0, 0, 0)	7 (7, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)
MEBPFNR3 90% + C1P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 90% + C2P 10%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 90% + DSEMO 10%	0 (0, 0, 0)	4 (4, 0, 0)	5 (5, 0, 0)	0 (0, 0, 0)
MEFNR3	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + MEBPFNR3 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + RMEFNR2 90%	0 (0, 0, 0)	4 (0, 3, 1)	0 (0, 0, 0)	2 (0, 2, 0)
MEFNR3 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + MEBPFNR3 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + RMEFNR2 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + DSEMO 40%	0 (0, 0, 0)	2 (2, 0, 0)	2 (2, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + MEBPFNR3 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + RMEFNR2 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 20% + C1P 10%	0 (0, 0, 0)	2 (2, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 20% + C2P 10%	0 (0, 0, 0)	3 (3, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 30%	0 (0, 0, 0)	5 (5, 0, 0)	5 (5, 0, 0)	0 (0, 0, 0)
MEFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	3 (2, 1, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)	1 (1, 0, 0)
MEFNR3 90% + C1P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 90% + C2P 10%	0 (0, 0, 0)	2 (2, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 90% + DSEMO 10%	0 (0, 0, 0)	4 (4, 0, 0)	4 (4, 0, 0)	0 (0, 0, 0)
MEFNR3 90% + MEBPFNR3 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 90% + RMEFNR2 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2	0 (0, 0, 0)	3 (0, 1, 2)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + MEBPFNR3 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + MEBPFNR3 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 1, 0)
RMEFNR2 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 60% + DSEMO 40%	0 (0, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 60% + MEBPFNR3 40%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	1 (0, 1, 0)
RMEFNR2 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	6 (2, 2, 2)	3 (1, 1, 1)	1 (0, 1, 0)
RMEFNR2 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	8 (3, 2, **3**)	2 (1, 0, 1)	3 (2, 1, 0)
RMEFNR2 90% + C1P 10%	0 (0, 0, 0)	5 (0, 3, 2)	4 (0, 2, 2)	1 (0, 1, 0)
RMEFNR2 90% + C2P 10%	0 (0, 0, 0)	9 (1, **5**, **3**)	6 (2, 2, 2)	1 (0, 1, 0)
RMEFNR2 90% + DSEMO 10%	0 (0, 0, 0)	4 (2, 1, 1)	1 (1, 0, 0)	1 (0, 1, 0)
RMEFNR2 90% + MEBPFNR3 10%	0 (0, 0, 0)	4 (0, 2, 2)	1 (0, 0, 1)	1 (0, 1, 0)

Table B.146: (b). Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 194 flights.

1^{st} March 2010 with 163 flights

Max. (6, 6, 5)	ES	**IS1ES**	IS1SUMS	SUMS
C1P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C1P 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
C2P	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C1P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
DSEMO 90% + C2P 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3	0 (0, 0, 0)	5 (0, 3, 2)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 20% + DSEMO 80%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 30% + DSEMO 70%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 50% + DSEMO 50%	0 (0, 0, 0)	2 (2, 0, 0)	1 (1, 0, 0)	0 (0, 0, 0)
MEBPFNR3 60% + C1P 40%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 60% + C2P 40%	0 (0, 0, 0)	1 (0, 0, 1)	1 (0, 0, 1)	1 (0, 1, 0)
MEBPFNR3 60% + DSEMO 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEBPFNR3 70% + C1P 30%	0 (0, 0, 0)	4 (0, 2, 2)	1 (0, 0, 1)	1 (0, 1, 0)
MEBPFNR3 70% + DSEMO 30%	0 (0, 0, 0)	2 (1, 1, 0)	2 (2, 0, 0)	0 (0, 0, 0)
MEBPFNR3 80% + C1P 20%	0 (0, 0, 0)	4 (0, 1, 3)	3 (0, 1, 2)	1 (0, 1, 0)
MEBPFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	5 (1, 3, 1)	2 (1, 1, 0)	1 (0, 1, 0)
MEBPFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	<u>10</u> (<u>4</u>, 5, 1)	4 (1, 2, 1)	1 (1, 0, 0)
MEBPFNR3 80% + DSEMO 20%	0 (0, 0, 0)	5 (<u>4</u>, 1, 0)	2 (2, 0, 0)	1 (0, 1, 0)
MEBPFNR3 90% + C1P 10%	0 (0, 0, 0)	9 (0, <u>6</u>, 3)	3 (0, 2, 1)	2 (0, 1, 1)
MEBPFNR3 90% + C2P 10%	0 (0, 0, 0)	5 (0, 3, 2)	3 (0, 2, 1)	1 (0, 1, 0)
MEBPFNR3 90% + DSEMO 10%	0 (0, 0, 0)	3 (2, 1, 0)	3 (3, 0, 0)	0 (0, 0, 0)
MEFNR3	0 (0, 0, 0)	2 (0, 2, 0)	0 (0, 0, 0)	1 (0, 1, 0)
MEFNR3 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 10% + MEBPFNR3 90%	0 (0, 0, 0)	5 (0, 5, 0)	1 (0, 1, 0)	0 (0, 0, 0)
MEFNR3 10% + RMEFNR2 90%	0 (0, 0, 0)	3 (0, 1, 2)	2 (0, 1, 1)	4 (0, 1, 3)
MEFNR3 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 40% + MEBPFNR3 60%	0 (0, 0, 0)	5 (0, 5, 0)	3 (0, 3, 0)	0 (0, 0, 0)
MEFNR3 40% + RMEFNR2 60%	0 (0, 0, 0)	4 (0, 3, 1)	2 (0, 1, 1)	5 (0, 2, 3)
MEFNR3 60% + C1P 40%	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)
MEFNR3 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + DSEMO 40%	0 (0, 0, 0)	2 (2, 0, 0)	3 (3, 0, 0)	0 (0, 0, 0)
MEFNR3 60% + MEBPFNR3 40%	0 (0, 0, 0)	4 (0, 3, 1)	5 (0, 5, 0)	0 (0, 0, 0)
MEFNR3 60% + RMEFNR2 40%	0 (0, 0, 0)	1 (0, 1, 0)	0 (0, 0, 0)	3 (0, 1, 2)
MEFNR3 70% + DSEMO 20% + C1P 10%	0 (0, 0, 0)	4 (2, 1, 1)	2 (0, 1, 1)	0 (0, 0, 0)
MEFNR3 70% + DSEMO 20% + C2P 10%	0 (0, 0, 0)	6 (<u>4</u>, 1, 1)	3 (2, 1, 0)	0 (0, 0, 0)

Table B.147: (a). Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 163 flights.

Max. (6, 6, 5)	ES	**IS1ES**	IS1SUMS	SUMS
MEFNR3 70% + DSEMO 30%	0 (0, 0, 0)	3 (3, 0, 0)	4 (3, 1, 0)	0 (0, 0, 0)
MEFNR3 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	6 (4, 0, 2)	1 (1, 0, 0)	0 (0, 0, 0)
MEFNR3 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	4 (3, 0, 1)	2 (1, 1, 0)	1 (0, 1, 0)
MEFNR3 90% + C1P 10%	0 (0, 0, 0)	3 (0, 1, 2)	0 (0, 0, 0)	1 (0, 1, 0)
MEFNR3 90% + C2P 10%	0 (0, 0, 0)	3 (1, 2, 0)	2 (0, 2, 0)	1 (0, 1, 0)
MEFNR3 90% + DSEMO 10%	0 (0, 0, 0)	1 (1, 0, 0)	2 (2, 0, 0)	1 (0, 1, 0)
MEFNR3 90% + MEBPFNR3 10%	0 (0, 0, 0)	4 (0, 3, 1)	5 (0, 5, 0)	0 (0, 0, 0)
MEFNR3 90% + RMEFNR2 10%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 1, 0)
RMEFNR2	0 (0, 0, 0)	3 (0, 1, 2)	1 (0, 1, 0)	0 (0, 0, 0)
RMEFNR2 10% + C1P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + C2P 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + DSEMO 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 10% + MEBPFNR3 90%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	2 (0, 2, 0)
RMEFNR2 40% + C1P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + C2P 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + DSEMO 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 40% + MEBPFNR3 60%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	4 (0, 2, 2)
RMEFNR2 60% + C1P 40%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 60% + C2P 40%	0 (0, 0, 0)	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)
RMEFNR2 60% + DSEMO 40%	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)	0 (0, 0, 0)
RMEFNR2 60% + MEBPFNR3 40%	0 (0, 0, 0)	1 (0, 1, 0)	2 (0, 1, 1)	7 (0, 3, 4)
RMEFNR2 80% + DSEMO 10% + C1P 10%	0 (0, 0, 0)	2 (0, 1, 1)	1 (0, 0, 1)	2 (0, 0, 2)
RMEFNR2 80% + DSEMO 10% + C2P 10%	0 (0, 0, 0)	1 (0, 0, 1)	0 (0, 0, 0)	1 (0, 0, 1)
RMEFNR2 90% + C1P 10%	0 (0, 0, 0)	2 (0, 0, 2)	2 (0, 0, 2)	1 (0, 0, 1)
RMEFNR2 90% + C2P 10%	0 (0, 0, 0)	3 (0, 1, 2)	2 (0, 1, 1)	1 (0, 0, 1)
RMEFNR2 90% + DSEMO 10%	0 (0, 0, 0)	2 (0, 0, 2)	1 (0, 1, 0)	0 (0, 0, 0)
RMEFNR2 90% + MEBPFNR3 10%	0 (0, 0, 0)	2 (0, 2, 0)	2 (0, 1, 1)	2 (0, 0, 2)

Table B.148: (b). Number of occurrences which cannot be said to be statistically significantly less fit than the others for a 4-pier topology and 163 flights.

B.6 Results Robustness

This section contains the summary results of some of the experiments which results have not been shown in the Chapter 6 for the ABSSAP.

B.6.1 Results Robust Approaches Using Buffer Times

Tables B.149, B.150 and B.151 show how each approach performed in respect to collisions for the different ranges of number of BSSs when combined with the TRS, where higher values correspond to wider covering and better performance within a range, i.e. fewer conflicts than others for more number of BSSs.

Approach	16^{th} December 2009		1^{st} March 2010	
	Max. (9, 5, 3)		Max. (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
ARS	(1, 2, 1)	(1, 2, **3**)	(0, 1, 4)	(1, 1, 0)
ARS + TRS	(7, 4, 0)	(2, 4, **3**)	(5, 5, **5**)	(0, 4, 2)
BSARS	(7, 4, 2)	(2, 0, **3**)	(1, 5, **5**)	(3, 5, 4)
BSARS +TRS	(8, 3, **3**)	(8, 4, **3**)	(4, **6**, 3)	(4, **6**, **5**)
SARS + TRS	(2, 4, 1)	(0, 3, 2)	(0, 1, 4)	(0, 1, 0)
TRS	(0, 1, 0)	(0, 0, 1)	(1, 3, 4)	(1, 1, 0)

Table B.149: Conflicts (σ = 10min) statistical significance for MEFRN3 operator combine robustness approaches with TRS.

Approach	16^{th} December 2009		1^{st} March 2010	
	Max. (9, 5, 3)		Max. (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
ARS	(1, 4, 3)	(1, 1, **3**)	(1, 2, 4)	(1, 1, 0)
ARS + TRS	(7, **5**, 2)	(2, **5**, **3**)	(5, 5, **5**)	(0, 5, 2)
BSARS	(7, 3, **3**)	(5, 2, **3**)	(2, **6**, **5**)	(3, **6**, 4)
BSARS +TRS	(8, 4, **3**)	(8, **5**, **3**)	(4, **6**, 3)	(4, **6**, 4)
SARS + TRS	(2, 4, 2)	(1, 2, **3**)	(1, 2, **5**)	(0, 3, 2)
TRS	(0, 2, 2)	(1, 0, 0)	(1, 3, **5**)	(1, 0, 0)

Table B.150: Conflicts (σ = 20min) statistical significance for MEFRN3 operator combine robustness approaches with TRS.

Approach	16^{th} December 2009		1^{st} March 2010	
	Max. (9, 5, 3)		Max. (6, 6, 5)	
	3-pier	4-pier	3-pier	4-pier
ARS	(2, 4, **3**)	(1, 2, **3**)	(3, 2, 4)	(1, 4, 0)
ARS + TRS	(**9**, **5**, **3**)	(3, **5**, **3**)	(5, 5, **5**)	(0, 5, 2)
BSARS	(8, 4, **3**)	(4, 3, **3**)	(2, **6**, **5**)	(3, **6**, 4)
BSARS +TRS	(7, **5**, **3**)	(7, **5**, **3**)	(4, **6**, 4)	(4, **6**, **5**)
SARS + TRS	(6, **5**, **3**)	(1, 2, **3**)	(1, 1, **5**)	(0, 2, 1)
TRS	(1, 3, **3**)	(1, 0, 0)	(1, 3, **5**)	(1, 0, 0)

Table B.151: Conflicts (σ = 30min) statistical significance for MEFRN3 operator combine robustness approaches with TRS.

Appendix C

More AGAP Results

C.1 Constructive Algorithms

Result for the first four objectives are presented in Section 7.9.

Following the 'Minimise Number of Towing Operations' objective studied in Section 7.9 the First In First Out (FIFO) might be expected to perform well for this objective, as one of its characteristics is achieving larger gaps between assignments than the other selection methods, since assignments are distributed between all available gates. This should increase the chance of assigning parking activities to the same gate as the corresponding arrival and departure flights, but on the contrary, it performs worst. This may be attributed to the way in which this selection method works, as the most recently assigned gate will be the last to be used in a new assignment, and the particular requirement for a parking activity to be assigned only to the same gate as either its associated arrival or departure activities, so reducing the chance in which that a departure will be assigned next to its associated parking activity. It must be noted that this does not happen with the arrival activity even though the gate to which the arrival activity is assigned is not the first gate to be selected for assigning to the parking activity, because the parking activity can only be assigned to the same gate as the arrival activity, and then eventually that gate will be selected and assigned to the parking activity. The characteristic of the FIFO selection method being to distribute assignments between all available gates, which is detrimental to the parking assignment, since it reduces the chance of both the parking and departure of an aircraft being assigned to the same gate. Therefore any operation able to reduce the FIFO tendency to spread assignments among the gates will obviously improve the towing objective since more restrictions are applied to the available gates, as when Algorithms 'C' and 'A' are used, and given the way in which the constraints ('Airline Preference') generate a favourable set of gates from which to select, which is corroborated by the results shown in Appendix C.1 (Figure C.1).

The 'Maximise Handling Agent Preferences' objective is shown in the second column in Figure C.1. This objective for Algorithm 'E' is best achieved by the selection method FIFO, which is not as might be expected, given that spreading the assignments should increase the chance of the aircraft being assigned to a gate away from that preferred by the handling agent, whereas concentrating the assignments on a few gates, characteristic of the Last In First Out (LIFO) selection method, might be expected to be preferable. However, this LIFO characteristic would be detrimental if the group of gates, within which the assignment is concentrated, may also be those least preferable to the handling agent, thereby decreasing this objective sufficiently to render it even worse than FIFO, which would appear to be the case here. The additional restrictions in this case show that the performance of the selection methods 'Maximise Airline Preferences' and LIFO are improved as expected, since the single group is now spread to multiple selection groups based on the topology.

The Order by Departure Time Lookahead and Improvement (ODTLI) does not achieve maximum assignments as is shown in Figures C.2 and C.3 for Algorithm 'E'. These figures show the number of assignments to gates achieved by each constructive algorithm and data set, where the results are always lower than the total number of activities shown by the upper dashed grey line. This also applies to the other algorithms, i.e. Algorithms 'C' and 'A'. Therefore ODTLI does not perform well for all those data sets considered where there are sufficient gates to which all the activities may be assigned (including parking activities, Upper Maximum Assignment Point with Parking (UMAP_p) $\leq N$).

C.2 Steady State Evolutionary Algorithm

Some operators are described in this section which extends the number of operators already provided in Section 5.4, followed by some statistical results in Section C.2.2.

C.2.1 Remote Mutation Operators

A new fictitious dummy stand, namely the remote dummy, was introduced in this problem to explore the parking activities between arrival and departure flights by the same aircraft, as was presented in Section 7.2. In order to allow these parking activities to be unassigned from the remote dummy stand, it is necessary to add another operator. The Remote Dummy Single Exchange Mutation Operator (RDSEMO) selects only one of the parking activities assigned to the remote dummy for exchange, namely the RDSEMO, described in Algorithm 9. The parking activity for reassignment may be randomly selected when there is more than one assignment to the remote dummy stand. The RDSEMO will be seen in Section 8.3 to perform poorly given that it is restricted to solutions with parking activities assigned to the remote dummy, together with the extra constraint of assigning them to the gate already assigned to either the arrival or departure flight of the same aircraft as the parking activity.

The Remote Dummy Exchange All Mutation Operator (RDEAMO) removes each parking activity from the remote dummy and assigns it to an appropriate gate, perhaps by removing one of the activities assigned to that gate, as described in Algorithm 10, so repeating the process followed by the RDSEMO for each of the parking activities assigned to the remote dummy stand.

An example of the RDSEMO operator is shown in Figure C.4a, where the problem is composed of three gates and five groups. A parking activity is selected randomly

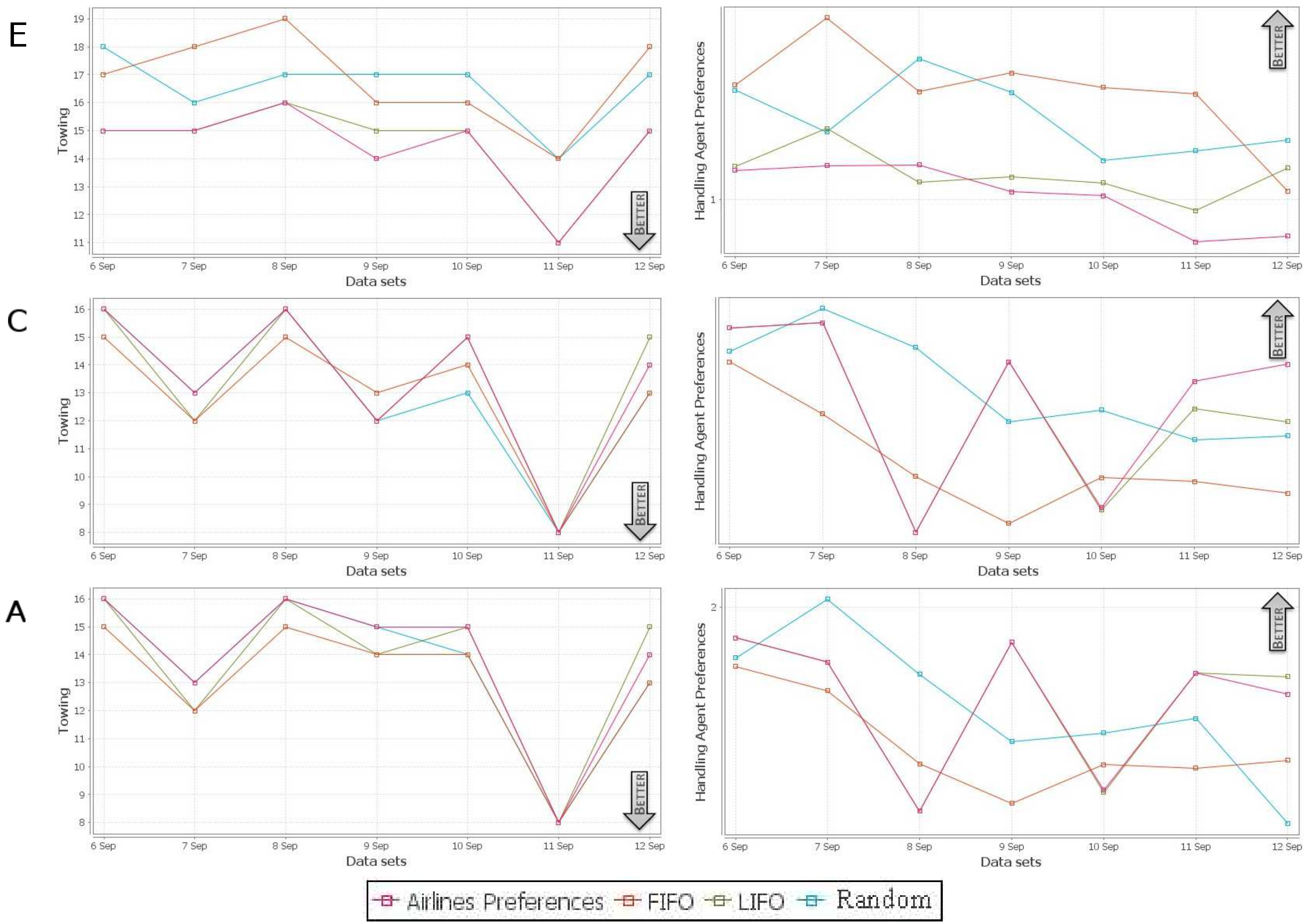


Figure C.1: Comparison of results for the fourth and last objectives for the Order by Starting Time (OST) ordering method, the four selection methods and three assignment algorithms, for 3-pier and 23 stands.

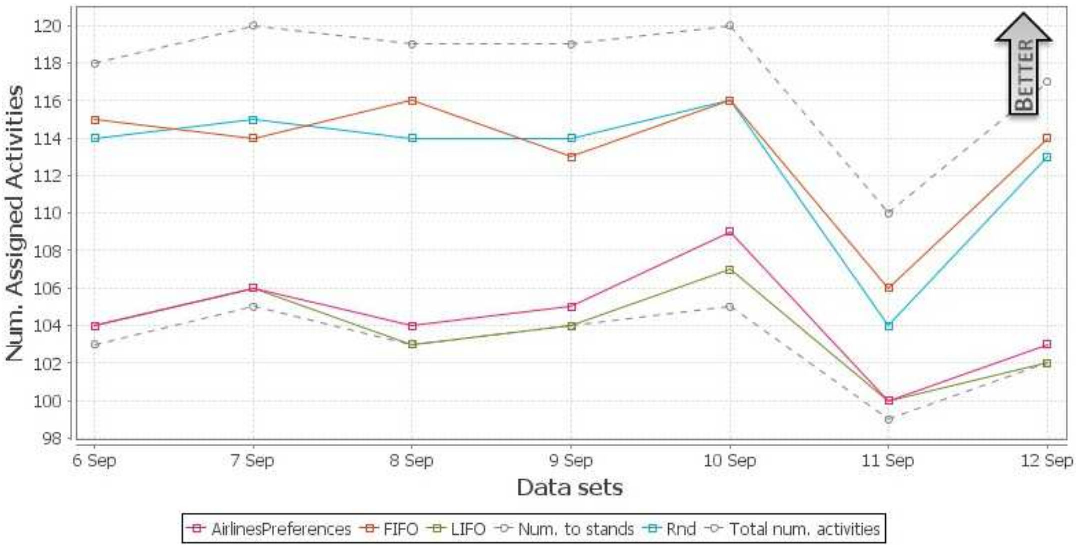


Figure C.2: Total number of assignments for ordering method ODTLI, Algorithm 'E' and With Maximum Reduction in Service Time.

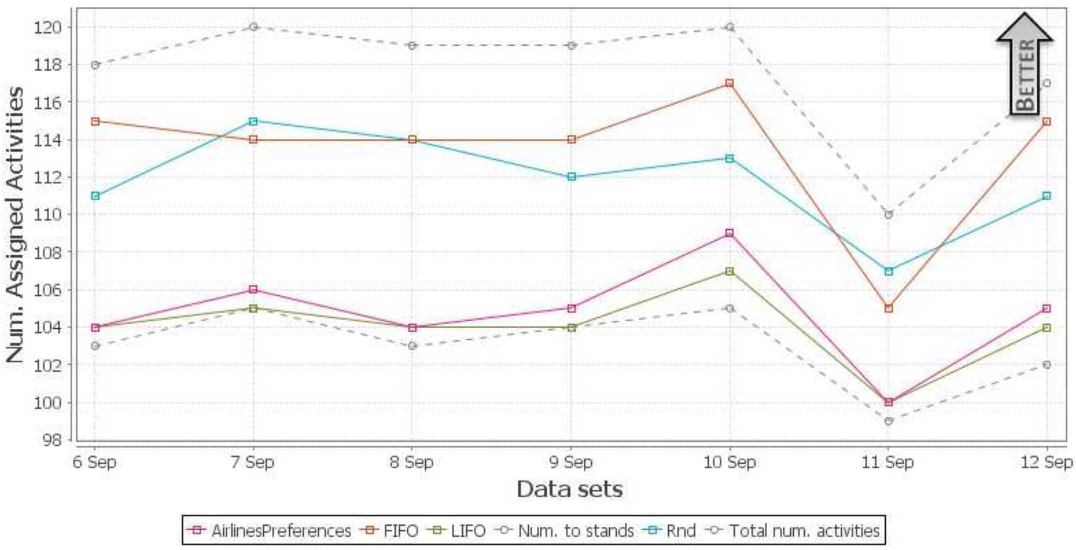


Figure C.3: Total number of assignments for ordering method ODTLI, Algorithm 'E' and With Reduction in Service Time.

from those assigned to the remote dummy, for example the parking activity of group 3. A gate is next randomly selected, e.g. gate 1, from which the search to assign the parking activity commences. As a parking activity must be assigned to the same gate as either its arrival or departure flight, then this remote activity cannot be assigned to gate 1. So the search moves to the next gate, gate 3, but the same applies to this gate so the parking activity cannot be assigned to this gate either. Finally, the next gate, gate 2, which has not yet been looked at, is now checked and the parking activity for group 3 can be assigned to it given that the flight arrival at this parking activity is also assigned to this gate, but group 4 must first be unassigned. The process is

---

Algorithm 9: Remote Dummy Single Exchange Mutation Operator

Randomly select a parking activity between all assigned to the remote dummy;
if *arrival or departure activity associated with this parking activity is assigned to a gate* **then**
    Select randomly one of the assigned gates to either the arrival or departure activity;
    if *is possible to assigned the parking activity to this gate* **then**
        Assign the parking activity to the gate;
    end
    if *is possible to assigned the parking activity to this gate once the appropriate activity is unassigned from this gate* **then**
        Assign the parking activity to the gate;
        Assign the unassigned activity to the appropriate dummy stand;
    end
end

Algorithm 10: Remote Dummy Exchange All Mutation Operator

---

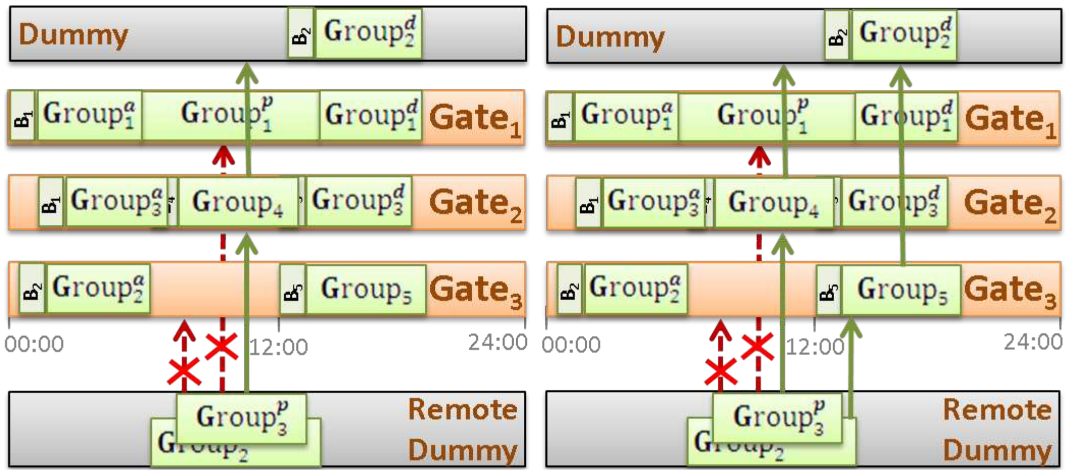
forall the *parking activities in assigned to the remote dummy stand* **do**
    if *arrival or departure activity associated with this parking activity is assigned to a gate* **then**
        Select randomly one of the assigned gates to either the arrival or departure activity;
        if *is possible to assigned the parking activity to this gate* **then**
            Assign the parking activity to the gate;
        end
        if *is possible to assigned the parking activity to this gate once the appropriate activity is unassigned from this gate* **then**
            Assign the parking activity to the gate;
            Assign the unassigned activity to the appropriate dummy stand;
        end
    end
end

---

repeated in turn for each of the other parking activities assigned to the remote dummy which has not yet been considered, e.g. parking activity 2. This parking activity can only be assigned to gate 3, but it would overlap with group 5, so firstly group 5 is unassigned and then parking activity 4 is assigned to gate 3.

Another operator moves one or multiple parking activity assignments from the remote dummy stand to appropriate gates with a sufficient gap to accommodate them all, namely Remote Dummy Move All Mutation Operator (RDMAMO). Therefore only those parking activities will be assigned where there is a gate with adjacent assignments which have a sufficient gap to accommodate the parking activity, and where one of those activities is an arrival or departure for the parking activity.

An example of the RDMAMO operator is shown in Figure C.5, where the problem



a Remote Dummy Single Exchange Mutation Operator (RDSEMO).

b Remote Dummy Exchange All Mutation Operator (RDEAMO).

Figure C.4: Examples of the process for the Remote Dummy Exchange Mutation Operators.

is composed of three gates and four groups. A parking activity is randomly selected from those assigned to the remote dummy, e.g. a parking activity of group 3. A gate is next randomly selected, e.g. gate 1, from which the search to assign the parking activity begins. As the parking activity must be assigned to the same gate as either its arrival or departure flight this remote activity cannot be assigned to gate 1. So the search moves to the next gate, gate 3, but this assignment is not possible either, as otherwise it would overlap with the activity for group 4. Finally, the next gate, 2, which has not yet been looked at, is now checked and the parking activity for group 3 can be assigned to it, given that this parking activity does not overlap with any of the activities already assigned to that gate, and the arrival flight for this parking activity is also assigned to the gate. The process is repeated for each of the other parking activities assigned to the remote dummy which have not yet been considered, for example parking activity 2. However this parking activity cannot be assigned to any of the gates as it would overlap other assigned activities.

When the remote dummy stand has no remote activity assigned to it then obviously none of the remote dummy mutation operators presented here provide a new solution, as there are no remote activities available to be unassigned from the remote dummy and assigned to a gate. These operators may therefore only be used when there are parking activities assigned to the remote dummy, and once this is no longer the case, they should not be used.

When only one or many of the mutation operators introduced in Section 5.4 are used it may be advantageous to include at least one of the remote dummy mutation operators, as the other mutation operators do not have the capability of reassigning parking activities to gates. The crossover operators may be able to reassign parking activities to gates, but only where at least one of both parents have not assigned the same parking activity to the remote dummy gate. This applies similarly to the dummy operators Dummy Single Exchange Mutation Operator (DSEMO) and

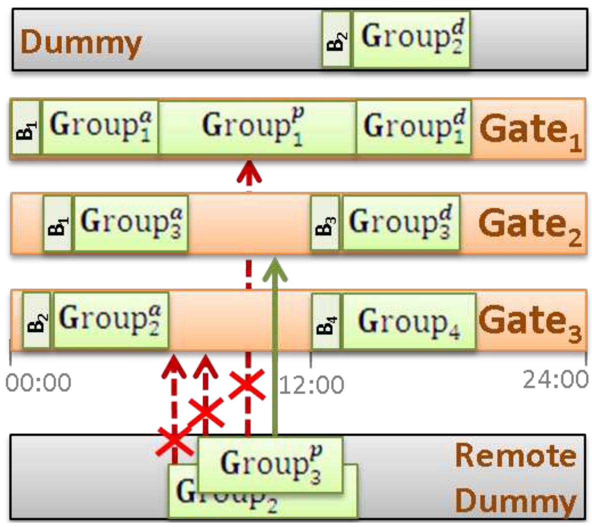


Figure C.5: Example of the process for the Remote Dummy Move All Mutation Operator (RDMAMO).

Dummy Single Move Mutation Operator (DSMMO) in respect of unassigned flight arrivals and departures.

C.2.2 Single Operator Results

The results show that the RDSEMO does not perform well when used alone, being even worse for $N <$ Lower Maximum Assignment Point (LMAP), where N is the number of gates available. This is due to the duration of the parking activity normally being very long, in our case over two hours (Section 7.6), which may overlap with multiple arrival and departure activities already assigned to the same gates where the exchange of assignments is attempted. Furthermore if these activities already assigned are unassigned in order to allow the parking activity to be assigned to that gate, the number of assignments is reduced. Similarly as in Chapter 5 for $N \geq$ Lower Maximum Assignment Point with Parking (LMAP$_p$), this operator can only improve the solutions if the population contains solutions without full assignment of activities, since there are no unassigned activities for removal from the remote dummy stand.

Table C.1 shows a summary of the population sizes for the single operators whivh cannot be said to provide statistically significantly less fit solutions than any of the other single operators considered for the Steady State Evolutionary Algorithm with $\ell = 1$ (SSEA1). A summary of the parameters is shown in Table 8.2 where only $\ell = 1$ was considered. The results give an indication of the preferred population sizes for each operator. The influence of the ℓ in the performance of each operator and preferred population size is studied in Section 8.3.1.

C.2.3 Robustness

The summary of the statistical significance of the different robustness approaches is shown in Tables C.2 and C.2, where the Probability of Conflict Based on the Gap (PCBG) uses the same standard deviation as the normal distribution which was

Operator	Selector	H4T100906	H4T100907	H4T100908	H4T100909	H4T100910	H4T100911	H4T100912
C1P	**IS1ES**	**2000**	**2000**	**2000**	**2000**	**2000**	**2000**	**2000**
	IS1SUMS	**2000**	**2000**	**2000**	1000, **2000**	**2000**	**2000**	1000, **2000**
C2P	IS1ES	1000, 500	5, 10, 15, 30, 50, 500, 2000	500, **2000**	**1000**, **2000**	1000, 2000	500, 1000, 2000	500, 100, 200
	IS1SUMS	500, 2000	5, 10, 15, 30, 100, 500, 1000, 2000	1000, **2000**	**2000**	2000	5, 50, 500, 1000, 2000	500, 1000, 2000
IMEFNR3	IS1ES		5, 10	5, 10	5, 10	**5**	5, 10, 15	10, 15
	IS1SUMS		5	30	5, 10			
	SUMS	5, **15**, 30, 100, 200, **1000**, 2000	5, 10, 15, 30, 50, 100, 200		15, 30, 50, 100. 2000	5, 15, 2000	5, 10, 15, 30, 100, 200, 500, 1000, 2000	10, 30, 1000
MEFNR3	ES	100, 500		100		50, 100		30, 200, 500, 1000
	IS1ES	5		5, 15, 30, 50, 500		15	10	30
	IS1SUMS			10, 15, 30, 100, 500		5, 30		
	SUMS	50, 200, 1000, 2000	**200**, 500	15, 50, 100, 200, 500, 2000	5, 15, 30, 200, 1000, 2000	5, 15, 30, 100, 200, 500, 1000, 2000	10, 15, 50, 200, 1000, 2000	15, 200, 500, 1000, 2000
IRMEFNR2	IS1ES	5, 10, 15, 30	5	**5**, 15	**5**, 10, 15, 500	5	10	5, 10, 30
	IS1SUMS	5, 50	5	15, 30	500			30
	SUMS	15, 500	10, 15, 30, 200		5, 10, 15, **30**, 50, 100, 200, 500, **2000**	5, 10, 15, 50, 100	5, 10, 15, 30, 50, 100, 200, 500, 2000	5, 10, 15, 50, 100, 200, 500
RMEFNR2	ES	500, **2000**	30, 2000					500, 1000, 2000
	IS1ES	5, 10, 15, 30, 50, 200, 500, 1000	50, 500, 1000	5, 10, 15, 30	5, 15, 500	5, 10, 15, 30		5, 15, 50
	IS1SUMS	10, 30, 500	10, 30, 50, 200	10, 15, 30, 50	5, 10, 15, 30	500		
	SUMS	10, 30, 50, 100, 200, 500, 2000	5, 10, 15, 30, 50, 100, 200, 500, 1000	10, 100	5, 10, 15, 30, 50, 100, 200, 500, 1000, 2000	5, 10, 15, 30, 100, 200, 500, 1000, 2000	5, 10, 15, 50, 100, 200, 500, 1000, 2000	100, 200, 500, 1000

Table C.1: SSEA1 statistically significantly fitter solutions for the data sets from 6^{th} to 12^{th} September 2010 for each operator.

used to generate the perturbed schedules. The LMAP, Upper Maximum Assignment Point (UMAP), LMAP_ps and UMAP_ps from Table 7.6 are shown between brackets in the table heading for convenience as (LMAP, UMAP, LMAP_p, UMAP_p). The table only presents those approaches which either alone or combined with others provide solutions with statistically significantly fewer collisions than other approaches studied and cannot be said to have more collisions than any of the other operators studied when used alone or in combination, which are shown with a tick. Only those approaches having at least one tick are shown.

To speed up execution of the PCBG robustness approach instead of using the density function for the distribution (folded normal distribution), a pre-generated table of the accumulative probabilities was used for up to four times the standard deviation.

It should be noted that given that the PCBG used considers standard deviations equal to those used to build the perturbed data sets it may be considered biased and be expected to perform better. However the results obtained for data sets with a sufficient number of gates to assign all the activities shows that the Unsupervised Estimated Stochastic Reduction in Service (UESRS) performs better for different unsupervised functions than PCBG for all the disruption standard deviations considered.

Approach	H4T100906 (8, 10, 17, 19)			H4T100907 (11, 14, 18, 20)			H4T100908 (7, 10, 16, 18)			H4T100909 (8, 10, 18, 20)			H4T100910 (9, 12, 15, 18)			H4T100911 (9. 10, 16, 16)			H4T100912 (11, 11, 18. 19)		
	Standard Deviation (x) in min																				
	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30
0.225PCBG(x)+0.00025TRS(2)	✓	✓		✓			✓			✓			✓			✓			✓		✓
0.3125PCBG(x)+0.00025TRS(2)	✓	✓	✓	✓		✓	✓			✓		✓	✓			✓			✓		✓
0.225PCBG(x)	✓			✓			✓			✓		✓	✓			✓			✓		✓
0.3125PCBG(x)	✓	✓	✓	✓		✓	✓			✓		✓	✓		✓	✓			✓		
0.225UESRS(E0.03)+0.00025TRS(2)	✓	✓	✓	✓			✓		✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓
0.225UESRS(E0.05)+0.00025TRS(2)	✓	✓		✓			✓			✓			✓			✓			✓		
0.225UESRS(I4)+0.00025TRS(2)	✓	✓		✓			✓			✓			✓			✓			✓		
0.225UESRS(I6)+0.00025TRS(2)	✓	✓		✓			✓			✓	✓		✓			✓			✓	✓	
0.3125UESRS(E0.03)+0.00025TRS(2)	✓	✓		✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
0.3125UESRS(E0.05)+0.00025TRS(2)	✓	✓	✓	✓			✓			✓			✓			✓			✓	✓	✓
0.3125UESRS(I4)+0.00025TRS(2)	✓	✓		✓			✓			✓			✓			✓	✓		✓	✓	✓
0.3125UESRS(I6)+0.00025TRS(2)	✓	✓		✓			✓			✓			✓			✓			✓	✓	✓
0.225UESRS(E0.03)	✓	✓	✓	✓			✓	✓	✓	✓	✓		✓	✓		✓			✓	✓	✓
0.225UESRS(E0.05)	✓	✓		✓			✓			✓	✓		✓			✓			✓	✓	✓
0.225UESRS(I4)	✓	✓		✓			✓			✓			✓			✓			✓		
0.225UESRS(I6)	✓	✓		✓			✓			✓	✓		✓			✓			✓	✓	
0.225UESRS(I15)	✓	✓					✓			✓	✓	✓	✓			✓		✓	✓	✓	✓
0.3125UESRS(E0.03)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓
0.3125UESRS(I4)	✓	✓		✓			✓			✓			✓			✓			✓		
0.3125UESRS(I6)	✓	✓	✓	✓			✓			✓			✓	✓		✓			✓	✓	✓
0.3125UESRS(I15)	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓

Table C.2: Summary Airport Gate Assignment Problem (AGAP) robustness statistical significance (significance level 0.05) using perturbed schedules generated from normal distributions of 10, 20 and 30 min standard deviations and SSEA1.

SSEA1 and population size 5 Approach	H4T100906 (17, 20, 23, 26)			H4T100907 (21, 23, 25, 28)			H4T100908 (18, 20, 23, 25)			H4T100909 (21. 21, 28, 28)			H4T100910 (19, 20, 20, 21)			H4T100911 (19. 21, 21, 21)			H4T100912 (19, 21, 23. 24)		
	Standard Deviation (x) in min																				
	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30	10	20	30
0.225PCBG(x)+0.00025TRS(2)			✓	✓	✓	✓			✓	✓	✓	✓		✓	✓					✓	
0.3125PCBG(x)+0.00025TRS(2)		✓		✓	✓	✓		✓	✓	✓	✓	✓		✓	✓			✓		✓	✓
0.225PCBG(x)			✓	✓	✓	✓			✓	✓	✓	✓		✓				✓		✓	✓
0.3125PCBG(x)		✓	✓	✓				✓	✓	✓				✓	✓		✓	✓	✓	✓	✓
0.225UESRS(E0.03)+0.00025TRS(2)				✓			✓						✓			✓			✓		
0.225UESRS(E0.05)+0.00025TRS(2)	✓			✓			✓						✓			✓			✓	✓	
0.225UESRS(I4)+0.00025TRS(2)	✓						✓												✓		
0.225UESRS(I6)+0.00025TRS(2)	✓						✓						✓			✓			✓		
0.3125UESRS(E0.03)+0.00025TRS(2)				✓			✓		✓	✓			✓			✓	✓		✓	✓	
0.3125UESRS(E0.05)+0.00025TRS(2)		✓		✓	✓	✓	✓			✓		✓	✓			✓	✓		✓	✓	
0.3125UESRS(I4)+0.00025TRS(2)	✓						✓						✓			✓			✓		
0.3125UESRS(I6)+0.00025TRS(2)	✓						✓						✓			✓	✓				
0.225UESRS(E0.03)				✓			✓						✓								
0.225UESRS(E0.05)							✓						✓			✓	✓		✓		
0.225UESRS(I4)	✓						✓						✓						✓		
0.225UESRS(I6)							✓						✓						✓		
0.225UESRS(I15)							✓						✓			✓	✓		✓		
0.3125UESRS(E0.03)	✓						✓						✓			✓	✓		✓	✓	
0.3125UESRS(I4)	✓						✓						✓			✓			✓		
0.3125UESRS(I6)							✓						✓			✓	✓		✓		
0.3125UESRS(I15)							✓						✓			✓	✓				

Table C.3: Summary AGAP robustness statistical significance (significance level = 0.05) using perturbed schedules generated from normal distributions of 10, 20 and 30 min standard deviations and SSEA1 for new data sets with 37 extra groups each.