ABSTRACT

Title of dissertation: OPTIMAL REASSIGNMENT OF FLIGHTS TO GATES FOCUSING ON TRANSFER PASSENGERS

Moschoula Pternea
Doctor of Philosophy, 2019

Dissertation directed by: Ali Haghani, Professor
Department of Civil and Environmental Engineering

This dissertation focuses on the optimal flight-to-gate assignment in cases of schedule disruptions with a focus on transfer passengers. Disruptions result from increased passenger demand, combined with tight scheduling and limited infrastructure capacity. The critical role of gate assignment, combined with the scarcity of models and algorithms to handle passenger connections, is the main motivation for this study.

Our first task is to develop a generalizable multidimensional assignment model that considers the location of gates and the required connection time to assess the success of passenger transfers. The results demonstrate that considering gate location is critical for assessing of the success of a connection, since transfer passengers contribute significantly to total cost.

We then explore the mathematical programming formulation of the problem. First, we compare different state-of-art mathematical formulations, and identify

their underlying assumptions. Then, we strengthen our time-index formulation by introducing valid inequalities. Afterwards, we express the cost of passenger connections using an aggregating formulation, which outperforms the quadratic formulation and is consistently more efficient than network flow formulations when the cost of successful connections is considered.

In the last part of the dissertation, we embed the formulation in an MIP-based metaheuristic framework using Variable Neighborhood Search with Local Branching (VNS-LB). We explore the key notion of a solution neighborhood in the context of gate assignment, given that transfer passengers are our main consideration. Our implementation produces near-optimal results in a low amount of time and responds reasonably to sensitivity analysis in operating parameters and external conditions. Furthermore, VNS-LB is shown to outperform the Local Branching heuristic in terms of solution quality. Finally, we propose a set of extensions to the algorithm which are shown to improve the quality of the final solution, as well as the progress of the optimization procedure as a whole.

This dissertation aspires to develop a versatile tool that can be adapted to the objectives and priorities of practitioners, and to provide researchers with an insight of how the features of a solution are reflected in the mathematical formulation. Every idea relying on these principles should be a promising path for future research.

OPTIMAL REASSIGNMENT OF FLIGHTS TO GATES FOCUSING ON TRANFER PASSENGERS

by

Moschoula Pternea

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2019

Advisory Committee:
Professor Ali Haghani, Chair/Advisor
Professor Gang-Len Chang
Professor Cinzia Cirillo
Professor Martin Dresner, Dean's Representative
Professor Paul Schonfeld

Dedication

This dissertation is dedicated to the memory of Professor Matthew G. Karlaftis (1969-2014), my advisor and mentor in Greece, without whom I would not even be able to locate Maryland on the map.

I still have a lot of room for improvement.

- Johan Cruyff

Acknowledgments

I would like to thank my PhD advisor, Professor Ali Haghani, for giving me the opportunity to explore a topic that I consider interesting and challenging at the same time, as well as for his guidance, suggestions, and willingness to answer the not-so-few questions that kept arising during my research. I would also like to thank my doctoral committee members, professors Paul Schonfeld, Martin Dresner, Cinzia Cirillo, and Gang-Len Chang, for their insightful observations and suggestions.

This is a perfect opportunity to thank the people who contributed to my work through our collaboration: Chien-Lun Lan, an amazing colleague and friend, from whom I learned a lot; Professor Konstantinos Kepaptsoglou of the National Technical University of Athens, who has been offering his advice to me since I was an undergraduate student; my fellow students Kiana Roshan Zamir, Bahar Zarin, Golnush Mashgati Amoli, and Sanaz Aliari, for providing their ideas.

I also want to thank the talented people that brightened up my years in College Park and who have been lucky (?) enough to share the road to the defense with me: Lida, Ioana, Niki, Eirini, Kanishka, Adi, Kostas, Jason, as well as the younger and newer additions to the Greek community of College Park, Christos, Ioanna, Leonidas, Leonidas, Panos, Christos, Eleftheria, and Sotiria. I am also happy to have been sharing my daily work breaks with the fantastic computer vision and robotics people, Greg, Chahat, Anton, Nitin, Chethan, and Snehesh.

I would not have been able to accomplish anything without my friends from Greece: the Miss Rodney Friends Association, Thodoris, Eleni, Stamatis, and Anna;

the Roman club, Silia, Nicole, Christos, Magda, Eleni, Stela, Perseas, Spyros, and Vasilis; and of course, friends who cannot be grouped under a common label (other than "Amazing Uncategorized Friends"): Nicky, Mina, Panos, and Konstantinos; the biggest amateur aviation enthusiast, Alexandros; Grigoris, with whom we have had almost parallel paths since we began our studies in Athens (although in the end he accelerated a little more); and lots of other people who kept making my day with our chats, our conversations, and our reunions in Athens.

It goes without saying that I am grateful to my family and relatives. Special thanks go to my brother, Vangelis, who was always the first one to update me on the latest events of Greek politics and Greek basketball.

Finally, I can never thank enough Konstantinos Zampogiannis, whom I objectively consider a person of unusual intelligence, skills, and proven patience, and who understands all the sacrifices required when pursuing a PhD, including watching Champions League at such an inconvenient time as 2:45 in the afternoon.

Table of Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Chapter Overview

Airport and airline operators deal with a variety of interconnected problems when planning and scheduling their operations. However, a variety of factors such as adverse weather conditions and unexpected incidents disrupt the smoothness of daily operations. The increase in air transport demand over the past years, combined with tight operation scheduling, increase the severity and duration of the disruptions by propagating the initial delay across the whole network. To restore normal operating conditions and prevent the propagation of delays, recovery procedures are developed. The assignment of flights to airport gates is a typical example of an operation which can be disrupted. Therefore, developing a new assignment that minimizes the impact of disruptions is of critical importance for airport, airlines, and passengers. In particular, transfer passengers, who comprise one of the most important but also vulnerable portion of airport users, are an indispensable part of schedule recovery procedures. Also, from the perspective of mathematical modeling, passenger connections present a significant computational challenge. The optimal assignment of flights to airport gates, given the special characteristics of transfer passengers, is the main focus of this dissertation.

Keywords: *gate reassignment; delays; transfer passengers*

1.1 Delays in Air Transportation

The airline industry is experiencing continuously growing passenger demand over the years. According to the World Bank (World Bank, 2016), the total travel demand for air transport in 2015 was 3.441 billion passengers, compared to 1.97 billion passengers in 2005; in other words, the total number of passengers carried by airlines increased by 75% in a decade, with the increasing trend obvious in both international and domestic air travel. Regarding international markets, the scheduled passenger traffic measured in RPKs (revenue passenger kilometers) increased by 8% from 2016 to 2017, according to ICAO (2018). Europe is the region with the largest international market share (37%), followed by Asia/Pacific (29%), North America (13%), Latin America and the Caribbean (4%) and Africa (3%). It is interesting to note that all regions experienced individual growth in their international market, with Latin America and Caribbean exhibiting the largest increase (10%) between 2016 and 2017. The growth rate for each region can be seen in Figure 1.1. Meanwhile, domestic markets also increased by 7% in total in 2017 (ICAO, 2018).

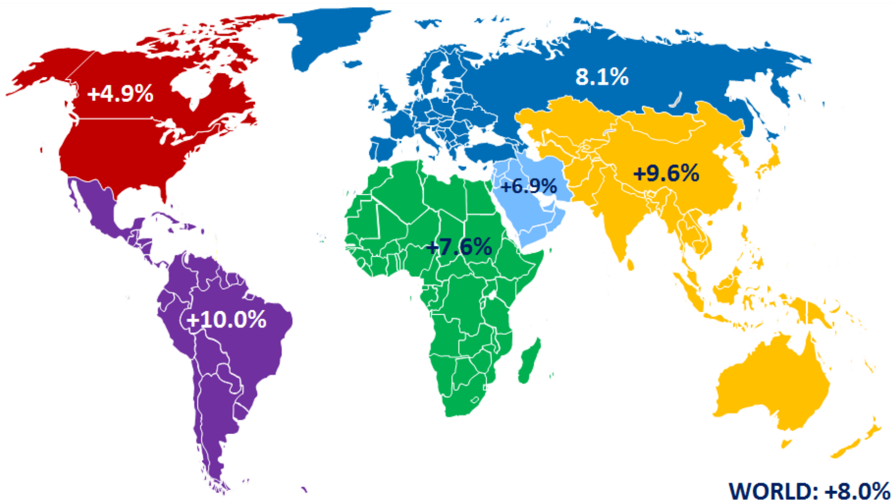

Figure 1.1: Growth rate in total airline passenger demand from 2016 to 2017 for different regions. Source: ICAO, 2018.

Not surprisingly, air travel demand and the respective provided capacity is expected to increase further, with the International Air Transport Association (IATA, 2018) estimating that the airline industry will serve 7.8 billion passengers in 2036, based on an estimated annual growth rate of 3.6%. The Asia-Pacific region is expected to be the major contributor to this increase, supplying more than 50% of the total demand. Although the value of the estimated growth rate can be decreased under conservative assumptions (e.g., trade protectionism and travel restrictions, as opposed to market liberalization and visa facilitation), the expected growth rate still remains positive, approximately equal to 2.7% per year (IATA, 2018). Predictions of passenger demand until the year 2036 can be seen in Figure 1.2.

However, largely due to the increase in travel demand, airlines experience recurrent flight delays. As a result, moderate disruptions are now part of travelers' normal routine, while more intense disruptions due to extreme weather conditions, strikes, and congestion, are also observed in the Eastern and Western United States,

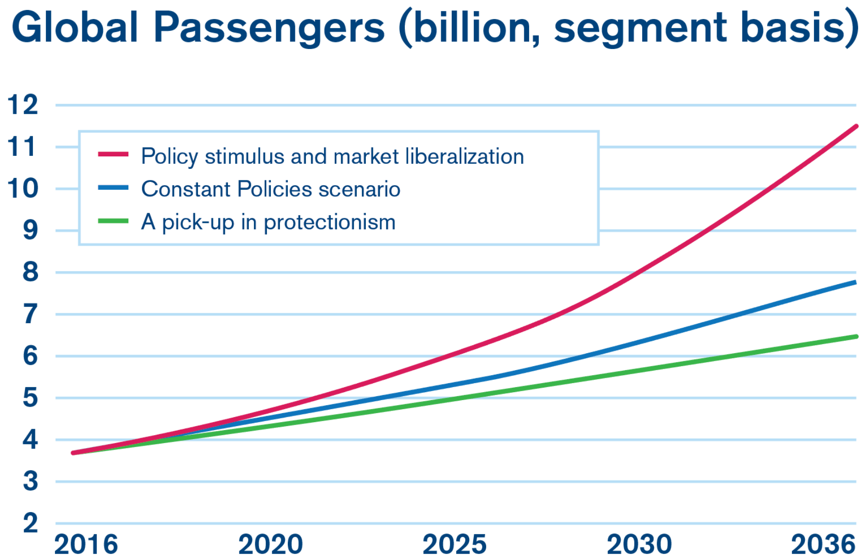

Figure 1.2: Prediction for global airline passenger demand for three alternative scenarios. Source: IATA, 2018.

Europe, China and other areas worldwide (Bendinelli et al., 2016). Statistics from both the US and Europe have quantified the average number of delayed flights, as well as the average delay duration, as follows: By default, a flight is considered delayed "if it arrived at (or departed) the gate 15 minutes or more after the scheduled arrival (departure) time as reflected in the Computerized Reservation System" (BTS, 2018). Regarding Europe, Eurocontrol (2018) identified a decrease in flight punctuality in 2017, since the percentage of flights arriving on time was 80%, as opposed to 81% in 2016. Similarly, the average departure time delays increased by 9.6%. The ADD index (average delay per delayed flight) was estimated as 28 and 29.9 minutes for departing and arriving flights, respectively. A similar trend was observed for the United States, with 18.57% of arriving flights and 17.79% of departing flights experiencing delays.

Delays are exacerbated by the combination of large travel demand and resource scarcity, which in turn make airports and airlines rely on tight scheduling to maximize their efficiency. To help us understand air transport delays, the paragraphs that follow (1.1.1 - 1.1.5) explain their fundamental causes and their most significant effects.

1.1.1 Causes of Delays

A detailed explanation of how air transport delays are generated is a complicated issue. For the purpose of this dissertation, we will present the reported causes of delays based on the classification provided by the Bureau of Transportation Statistics (BTS, 2018):

a) Air Carrier causes. This category includes factors that are within the airlines control, such as crew problems and delays in cleaning, maintenance, refueling and baggage handling (BTS, 2018).

b) Extreme Weather. This category includes actual or forecasted meteorological conditions, such as tornadoes, blizzards, and hurricanes, that may delay or prevent a flight.

c) National Aviation System (NAS). This category includes "a broad set of conditions, such as non-extreme weather conditions, airport operations, heavy traffic volume, and air traffic control" BTS (2018), as well as all weather-related causes that might delay operations but do not prevent flying and therefore cannot be classified as extreme. Such conditions include convective weather, ceiling and visibility, turbulence, temperature, humidity, in-flight icing, ground de-icing, etc. (Maharjan, 2010). In 2017, weather accounted for 63.3% of NAS delays, equivalent to 25.1% of total delays in that year.

d) Late-arriving aircraft. When the same aircraft serves two consecutive flights, i.e., an inbound flight to an airport followed by an outbound flight from that airport, and the inbound flight is delayed, then the delay is propagated to the outbound flight as well.

e) Security. Security causes include "evacuation of a terminal or concourse, re-boarding of aircraft because of security breach, inoperative screening equipment and/or long lines in excess of 29 minutes at screening areas" BTS (2018).

1.1.2 Congestion

Congestion is the result of heavy traffic volumes combined with limited infrastructure capacity. The types of congestion that affect air transport are airspace congestion and airport congestion, resulting from resource scarcity in the Air Traffic Management System and in the airport infrastructure, respectively.

On the one hand, airspace congestion exists when the available airspace is occupied by more aircraft than the ones allowed by its capacity. To relieve airspace congestion, flight trajectories are shifted in time (slot re-allocation) or in space (route reallocation). For more details, the reader shall refer to Nosedal et al. (2014), who classify congestion mitigation strategies in categories, including general route-time allocation, collaborative en-route resource allocation modeling, conflict risk assessment, slot allocation problems focused on controllers' workload, multi-sector complexity planning, and others.

Airport congestion, on the other hand, exists when the number of aircraft in the airport is larger than the airport's capacity, as determined by the capacity of the airside (e.g., runways, taxiways, apron) and of the landside (e.g., terminal buildings). In certain airports, such shortages might be present only during during peak hours (morning, noon and evening), while others, like London/Heathrow, Frankfurt, Paris/de Gaulle and New York/La Guardia Gelhausen et al. (2013), operate at or near capacity for many hours along the day. Congestion is strongly related to dense spacing of arrival and departure operations in hub-and-spoke systems (Santos and Robin, 2010), where airlines tend to use a small number of hub airports where the majority of flights are concentrated. As highlighted by Baumgarten et al. (2014) and Fageda and Flores-Fillol (2015), this concentration of operations has contributed to an increase in airport congestion.

Frequently, the runway is the bottleneck of airport capacity. Most runways can handle up to 30 to 50 movements/hour, resulting in 250,000 movements/runway/year, assuming that the airport operates 18 hours/day (Roosens, 2008). However, these values are not achieved in practice, due to operational and legal constraints. As a result, a large amount of studies have focused on optimizing runway management, especially for airports with more than one runway.

1.1.3 Delay Propagation

Regardless of their cause, a key feature of delays is their ability to be propagated, thus extending the temporal and spatial impact of the original delay to the whole network. A number of studies (Abdelghany et al., 2004; Beatty et al., 1999; Kafle and Zou, 2016; Pyrgiotis et al., 2013; Schaefer and Millner, 2001; Wong and Tsai, 2012; Wu and Wong, 2007) have developed models to quantify propagation patterns and propose strategies to absorb delays in the various stages of planning procedure. Campanelli et al. (2016) modeled delay propagation in US and European networks, taking into account the differences in flight management strategies between the two regions. Their results indicated that the ATFM (Air Traffic Flow Management) system adopted in Europe results in larger delays than the first come-first served US protocol, which is better at preventing large congestion, but requires more intensive flight management procedures.

1.1.4 Impact of Delays

Schedule delays have significant direct and indirect impacts on airlines, airports and passengers. A number of studies (Ball et al., 2010; Schumer and Maloney, 2008) have provided estimates of the monetary cost of delays. Schumer and Maloney (2008) estimated that total cost of delays in US domestic flights in 2007 as $41 billion, out of which $19 billion were additional airline operating costs for fuel, maintenance and crew, $12 billion corresponded to lost passenger time resulting in lower productivity and business opportunities and $10 billion were the indirect costs to dependent industries like food service and public transportation. According to a more recent estimation by Ball et al. (2010), the direct cost of US flight delays in 2007 was \$32.9 billion, of which \$8.3 billion were the direct costs to airlines, \$16.7 billion were the costs to passengers, and \$3.9 billion were the costs from lost demand. The remaining impact on GDP was estimated as \$4 billion. More recently, Cook and Tanner (2011) estimated the direct costs of delays in Europe as €1.25 (\$1.46) billion in 2011.

Furthermore, delays have a substantial impact on schedule adherence and therefore on the level of service provided to passengers, since they often result in airport queues, long waiting times and, consequently, deterioration of the passenger experience as a whole, due to discomfort and inconvenience. In a recent study, Kim and Park (2016) used structural equation modeling to quantify the impact of service delays on passengers emotional reactions and behavior. The results indicated that delays can be a major source of negative emotions and anger for passengers. As a result, they also have a negative influence on passengers' repurchase intention and increase the probability for negative word-of-mouth, which in turn is expected to harm the reputation of the airline.

1.1.5 Dealing With Delays

As explained before, one of the major causes of delays is resource scarcity. Therefore, in theory, a potential solution would be the expansion of existing infrastructure by constructing additional runways or building new terminals. However, physical constraints might not allow for such measures, while the cost and time scale of new infrastructure planning can be excessive (Eurocontrol, 2016). In Munich Airport, for example, the costs of planning were €800 million (approximately $930 million), while the construction of one additional terminal in London/Heathrow required 14 years to build and €500 million (approximately $567 million) for planning, which accounted for 12% of the total investment cost. In addition, expansion of current infrastructure may not be a feasible or efficient option because of geographic, environmental, socio-economic and political issues (Vaze and Barnhart, 2012). Apart from the fact that the expansion of the existing infrastructure is not always possible, another factor that should be taken into account is the high degree of interdependence among the decisions that influence the use of most resources (flights, terminals, crews, baggage) Dorndorf et al. (2007a), which results in the need for interventions in different stages of planning and decision making. For this purpose, airport and airline managers collaborate to set up the base for "a complex management system for airports and airlines of any size" (Dorndorf et al., 2007a) that supports decisions about crews, disruptions, fleet, aircraft scheduling, ground operations, and allocation of flights to gates.

In this context, mitigating delays involves management strategies in two phases of decision making, i.e., the planning phase and the recovery phase.

- In the planning phase, airport and airline operations should include a certain level of robustness to deal with unexpected occurrences (Dunbar et al., 2012) by absorbing potential delays throughout the different steps of decision making.

Robust scheduling accounts not only for the planned costs but also for the recovery costs.

- Schedule recovery is the response to disruptions by modifying, delaying or canceling services in order to return to the originally planned state. Therefore, depending on the magnitude of the disruption and the type of the problem, quick and efficient real-time modification of the original schedule is required. Gate reassignment, which is the topic of this dissertation, is a typical example of a schedule recovery operation.

1.1.6 Reassigning Flights to Gates

The Gate Assignment Problem (GAP) deals with the assignment of aircraft activities, i.e., departure, arrival, or parking, to airport gates, for a given time period within the day of operations. It is an important planning step which affects the deployment of ramp personnel and equipment, as well as the available time for transferring passengers and their baggage (Gu and Chung, 1999). GAP is a complicated problem which deals with a variety of different resources like aircrafts, gates and crews (Dorndorf et al., 2007a). In practice, it has to be solved both in the planning phase, where an optimal assignment of flights to gates is determined (gate assignment), and in the real-time phase, where the original assignment needs to be updated based on data about delays and disruptions (real-time gate assignment). In this study, we will be using the term “gate reassignment” instead of “recovery of scheduled gate assignment” for simplicity.

In the planning phase, a feasible allocation of flights to gates is determined, given the arrival and departure time of each flight. To introduce a degree of robustness in GAP, every gate is scheduled to remain idle for some time between two consecutive flight assignments ("idle time"). The minimum imposed idle time is often referred to as "red zone" or "buffer time" and its purpose is to absorb potential delays to prevent a knock-on effect on other flights. However, due to the stochastic nature of delays and the factors that generate them in the first place, robust scheduling alone does not guarantee that the impact of delays is kept to a minimum. In particular, airport congestion, weather conditions, machine breakdowns, and other factors, might make the scheduled assignment infeasible or impractical. Large airports with heavy traffic are generally more prone to such disruptions.

In the context of gate assignment, schedule disruptions cause gate conflicts: A gate conflict refers to a situation where, because of some previous delay, two or more flights require the same gate with a time overlap. For example, if the departure time of an outbound flight is delayed, an inbound flight which was originally assigned to the same gate might not be able to park because the gate is blocked by the delayed outbound flight. In this case, the inbound flight should either wait in the apron (i.e., "the defined area on an airport intended to accommodate aircraft for purposes of loading or unloading passengers or cargo, refueling, parking, or maintenance" (FAA, 1996)) before the gate becomes available again, or use a different gate from the scheduled one, or both. Therefore, the assignment has to be readjusted in order to restore feasibility, minimize operation disruptions, and prevent further delays.

During the reassignment of flights to gates, airport operators use real-time data to introduce a combination of additional disturbances to the scheduled assignment, so that they generate a new, feasible schedule, in a limited amount of time. The new schedule should minimize the impact of delays and be optimal in term of the operators' objectives. Wang et al. (2013) identify three main types of disturbances that the operators may introduce:

- Apron disturbances: Allocating flights to the apron (or "tarmac").
- Gate disturbances: Allocating flights to a different gate than the one they were originally assigned to.
- Time disturbances: Changing the original departure or arrival time of a flight.

In this dissertation, we focus particularly on the impact of gate reassignment on connecting passengers. Therefore, before proceeding to the mathematical modeling part of our study, it is essential to examine the types of connecting passengers, since different passenger categories result not only in different paths within the airport, but also in different cost factors associated with the objective function of the problem.

1.2 Passenger Connections

De Neufville and Odoni (2003) classify passengers according to whether they use the airport as a starting, transfer, or ending point of their trip. The different types of passengers within an airport generate different types of flow patterns: Passengers move from the entrance of the airport to the gate of their departing flight, or from the gate of their arriving flight to the exit of the airport, or, in case of connecting passengers, between different gates.

Transfer (connecting) passengers are of particular importance to airport operations, since their needs are generally different compared to the rest of the passengers. Transfer passengers require connections that possess three main properties: Speed, efficiency, and reliability (De Neufville and Odoni, 2003):

- Speed: Passenger transfers should be fast to ensure that the airlines are capable of providing competitive services through their selected hub airport.

- Efficiency: Especially when the available connection time is tight, long and complicated connection paths are undesirable for passengers. Ideally, direct connections within the same terminal are desirable.

- Reliability: Apart from causing inconvenience to travelers, missed connections or delayed baggage have both a direct and an indirect impact on the airlines. Direct costs include expenses such as passenger compensation, possible hotel accommodation, or the cost of baggage delivery to a passenger's destination, while indirect costs arise from passenger inconvenience and reduction in willingness to repurchase, as shown in Kim and Park (2016).

All passengers, including the connecting ones, can be further divided in categories according to their origin/destination and their trip purpose. This classification is of particular importance when defining the objective function of the problem (see Chapter 3, that might target specific passenger categories, which are associated with different cost factors.

1.2.1 Passenger Types

Based on their origin/destination and their trip purpose, De Neufville and Odoni (2003) classify passengers in different categories as follows:

- According to their origin or destination, passengers can be either domestic or international. International passengers undergo passport and customs control, while domestic passengers do not. A similar distinction can be made between Schengen and Out-of-Schengen passengers for airports within the European Union.

- According to their trip purpose, passengers can be classified as business or leisure passengers. Business passengers are generally willing to pay higher fares and carry little or no checked baggage, while leisure passengers tend to travel in large groups, occasionally with their family, and usually carry checked baggage.

1.2.2 A Realistic View: Temporal - Spatial Dependence of Connection Success

Contrary to the majority of current studies, the framework proposed in this dissertation relies on a fundamental assumption about passenger connections: We assume that, whether a passenger connection will eventually be made or missed, is a function of two parameters:

1. The available connection time, i.e., the time between the arrival of the first flight and the departure of the second flight.

2. The required connection time, i.e., the estimated time that a passengers needs to move from the gate of the arriving flight to the gate of the departing flight, given the layout of the airport, the available transportation modes, and the duration of the mandatory procedures.

As will be explained in detail in Chapter 3, the relationship between these two time components will determine the success or failure of each connection.

1.3 Contributions of This Dissertation

Since gate assignment is a critical link in the long chain of airport operations, it is not surprising that GAP has attracted the attention of researchers from a variety of fields, such as engineers, analysts, and managers. As will be shown in Chapter 2, numerous optimization models have been developed to address the problem of the optimal scheduled assignment of flights to gates.

However, judging from the scarcity of relevant studies, optimal real-time assignment has received significantly less attention in the literature. In particular, there is a lack of mathematical models which incorporate passenger connections in what we consider to be a realistic way, as explained in section 1.2.2.

In addition, the basis of the optimization in gate reassignment is not quite clear. As will be shown in Chapter 2, the optimization objectives vary across different studies, while only few of them incorporate objectives that are related to passenger connections. Therefore, it is interesting to summarize and classify the currently applied objectives from a practical and methodological point of view and also examine whether it is justified to consider transfer passengers at the cost of increasing the size of the problem and therefore its computational complexity.

At the same time, the major challenge when incorporating passenger connections in the optimization procedure is the intractable increase in the size of the problem. Although researchers have developed alternative formulations for gate assignment, the mathematical properties of these formulations, and therefore the perspective to improve their computational efficiency, have not been sufficiently explored. As a result, most studies adopt assignment or network flow approaches without taking advantage of the specific features of each mathematical formulation. However, since real-time assignment requires the generation of optimal or near-optimal updated schedules in a limited amount of time, strong formulations are required to ensure that the problem in the smallest amount of time possible.

One could possibly argue that focusing on the improvement of the formulation is of rather trivial importance, since high-quality solutions can be obtained through heuristic methodologies. Inarguably, when modeling passenger transfers, exact optimal solutions to the commonly used integer programming models cannot be achieved within a few seconds or minutes. However, although MIP solvers might not yet able to handle the problem as a whole, they have evolved into extremely powerful tools in recent years. Therefore, they can be particularly useful for the development of mathematical-programming based heuristics, which combine the mathematical formulation of the problem with sophisticated search techniques that guide the search procedure towards promising areas of the solution space.

Our first task is to obtain a comprehensive view of state-of-art approaches on gate assignment by reviewing the relevant literature in Chapter 2. Then, motivated by the critical role of gate scheduling in airport recovery procedures, as well as by the scarcity of mathematical models and algorithms that deal with the optimization of passenger connections, in this dissertation we perform the following main tasks:

- In Chapter 3, we develop a novel time-index mathematical formulation which determines the success or failure of passenger connections based on the relationship between the available time between connecting flights and the required connection time as a function of the airport layout and the time allocated to mandatory procedures. We then use this formulation to compare and analyze the measures of effectiveness that are commonly used to express the objective function in gate reassignment studies and demonstrate the necessity of considering passenger connections in the optimization.

- In Chapter 4, we propose a sequence of improvements to the proposed formulations in order to improve computational efficiency. Such improvements include the addition of valid inequalities as well as complete restructuring of the formulation. Afterwards, we compare the new, improved formulation with existing, network flow-based gate assignment formulations, in both a theoretical and an experimental level. We also highlight the differences in their fundamental assumptions and determine the circumstances under which each of them outperforms the others.

- In Chapter 5, we develop a solution algorithm for the gate reassignment problem with passenger connections, which incorporates our improved formulation in a framework that uses Variable Neighborhood Search with Local Branching, a metaheuristic originally proposed by Hansen et al. (2006). Our methodology is then applied to the gate reassignment problem with consideration of failed baggage connections.

Finally, in Chapter 6 we summarize the main contributions of our research and propose directions for improvement and extension of our findings.

Chapter 2

Literature Review

Chapter Overview

In this chapter, we review the state-of-art approaches to the Gate Assignment and Gate Reassignment problems. First, we introduce the existing studies on the planned gate assignment problem. Then, we focus on the Gate Reassignment Problem, which is the main topic of this dissertation. In particular, we emphasize three elements of existing approaches: a) The objective function and constraints of the problem, b) the mathematical formulation, and c) the solution methodology. Based on our review, we identify the gaps that this dissertation attempts to fill: First of all, our review of the objective function and constraints reveals the importance of considering passenger transfers and motivates us to develop a model that handles them in what we consider to be a realistic way (Chapter 3). Second, examining the alternative mathematical formulations is a necessary step for proceeding to the formulation analysis and improvement presented in Chapter 4. Finally, studying the existing methodological approaches to the solution procedure is a key step for investigating the applicability of different model-based heuristics and for developing a new metaheuristic approach that will be presented in Chapter 5.

Keywords: *gate reassignment; literature review; objective function; constraints; formulation; solution algorithms*

2.1 Introduction

To better understand the state-of-art optimization approaches of the flight-to-gate reassignment problem, we perform a thorough literature review. Since the problem of real-time assignment is closely associated with the scheduled Gate Assignment Problem (GAP), we first examine the existing approaches of GAP, in terms of their objective function, constraints, and methodological approach (section 2.2). Then, we follow up with a detailed review of gate reassignment studies (section 2.3).

2.2 The Gate Assignment Problem: Review

As explained in the Introduction (Chapter 1), the assignment of aircraft, or, more precisely, of aircraft activities, to airport gates, is an essential decision procedure for airline and airport management. The main purpose of the Gate Assignment Problem (GAP) is to determine an assignment of flights to aircraft stands, as well as the start and completion times for processing an aircraft at the gate it has been assigned to. In this context, a gate can refer either to an aircraft stand at the terminal, or to an off-pier stand on the apron (Dorndorf et al., 2007a).

2.2.1 Model Input

To formulate and solve a gate assignment problem, information associated with the airport, the flight schedule, and operator strategies, is required. In more detail, the different types of information that are used as input for the optimization model are the following:

- Flight information: The flight schedule and the properties of each flight are the key information elements from this category. On the one hand, the flight schedule is a timetable with the arrival and departure times of all flights. On the other hand, flight properties include the type of flight (domestic Vs. international, or Schengen Vs. out-of-Schengen for European countries of the Schengen Area), as well as information about passengers (number of passengers in each flight, number of transferring passengers, number of business and first class passengers) and aircraft (aircraft type, wing span, consecutive arriving and departing flights served by the same aircraft).

- Airport information: The required airport information includes the number and location of gates, as well as the type of flights that are compatible with each gate. Each gate is generally restricted to specific flight types, based on criteria like the operating airline, as well as whether a flight is domestic or international.

- Information based on operator decisions: Different operators might use different policies according to their priorities. Examples of parameters determined by the decision maker include the length of the assignment slot, the minimum imposed idle time between consecutive flight assignments to the same gate, etc.

The main input for gate scheduling is a flight schedule with flight arrival and departure times and additional detailed flight information, including pair-wise links between successive flights served by the same aircraft, the type of aircraft, the number of passengers for each booking class and the origin or destination of a flight. The information in the flight schedule defines the time frame for processing a flight as well as the subset of gates where the flight can be assigned, based on restrictions such as airline-specific gates, aircraft size, and access to governmental inspection facilities for international flights.

2.2.2 GAP State-of-Art Overview

The scheduled gate assignment problem has been studied extensively in the literature. A comprehensive study of the approaches up to 2007 can be found in Dorndorf et al. (2007a). The authors identify three main research directions in the area: The number of slots in the model, the types of objectives, and the mathematical models used. Regarding solution algorithms, they identify two main directions: Mathematical programming techniques and rule-based expert systems.

Dorndorf et al. (2007a) make a thorough presentation of the algorithms used to solve GAP by classifying the relevant approaches in two categories, namely mathematical programming and rule-based expert systems approaches. In accordance with the scope of this dissertation, which focuses on mathematical programming approaches for gate reassignment, we will not examine rule-based expert systems approaches. Instead, our classification will rely on the following criteria:

- The objective function of the problem.
- The problem constraints.
- The methodological approach, i.e., the mathematical formulation and the solution algorithm.

2.2.2.1 Objective Function

The objective function of GAP depends on the priorities of the decision maker. Dorndorf et al. (2007a) have observed that the objectives commonly used in GAP can be divided in two main categories: passenger-oriented and airport oriented. Using this distinction, we can classify the objectives as follows:

Passenger-Oriented Objectives:

Passenger-oriented objectives emphasize the level of service provided to passengers. They include measures of discomfort (e.g., passenger waiting time) or impedance when moving to and from the gates (e.g., walking distance). Such measures include:

- Passenger walking distance (Bihr, 1990; Cheng et al., 2012; Ding et al., 2004b, 2005; Genç et al., 2012; Haghani and Chen, 1998; Marinelli et al., 2015; Yan and Chang, 1998; Yan and Huo, 2001; Yu et al., 2016).

• Passenger transit time in terminal (Kim et al., 2013).

• Total baggage distance (i.e., the distance over which the baggage is transported when moving between the terminal and the aircraft, or between aircraft for connecting flights) (Haghani and Chen, 1998).

• Proximity to airport facilities, like customs and airline lounges (Tang and Wang, 2013).

• Passenger waiting time (Yan and Huo, 2001).

Airport and Airline-Oriented Objectives:

This category includes objectives which are directly associated with the management of the airport and airline operations. Examples of such objectives are:

• Assignments to the apron. Due to the high cost of ungated flights, assigning flight to remote stands at the apron instead of contact gates is undesirable. This is expressed with a number of measures, including the number (Ding et al., 2004a,b; Marinelli et al., 2015; Tang and Wang, 2013) and duration (Genç et al., 2012) of ungated flights.

• Flight-gate preferences (Dorndorf et al., 2007b, 2008, 2012, 2017).

• Aircraft handling procedure costs. Such objectives include the number of towing activities (Dorndorf et al., 2007b, 2008, 2012, 2017; Guépet et al., 2015; Yu et al., 2016) or, similarly, the number of maximization of the number of arriving flights and subsequent departing flights assigned to the same gate, if they are served by the same aircraft (Tang and Wang, 2013). They also include aircraft taxi time (Kim et al., 2013).

- Robustness measures. These include a variety of objective functions which aim to minimize the impact of schedule disruptions by making the assignment able to absorb potential disturbances. Such objectives include various functions of the idle times and of gate blockage. Functions of idle times include the variance (Bolat, 2000) and the total semi-deviation or positive semi-deviation (Şeker and Noyan, 2012) and range (Bolat, 2000). On the other hand, typical functions of gate blockage are the expected number (Castaing et al., 2016; Şeker and Noyan, 2012; Dorndorf et al., 2017), duration (Castaing et al., 2016; Kim et al., 2013), cost (Yu et al., 2016) and worst-case number (Castaing et al., 2016) of gate conflicts. Dorndorf et al. (2012) also minimize the assignment of two flights with low buffer time to the same gate, while they also penalize small idle times. Dorndorf et al. (2017) also used the expected number of gate closure, shadow restrictions, and tow time restrictions, as measures of robustness.

- Deviation from a reference schedule.

Of course, the distinction between the two types of objectives serves purposes of analysis and classification, rather than strictly differentiating between the two parts, i.e. passengers and operators. In fact, the benefits of passengers, airlines, and airports are interdependent. For example, improving the level of service provided to passengers by assigning the flights to gates that are close to the VIP lounges increase the satisfaction levels of priority passengers and, consequently, their willingness to repurchase, thus increasing the market share of the respective airline. Since airlines tend to concentrate their activities in specific airports, the latter also benefit from the profitability of the airlines they serve. Reversely, objectives which might only seem to concern the operators, have a more important effect on passengers than it seems at first glance. For example, the uniform allocation of idle time among the gates ensures a minimum level of robustness for the schedule and therefore prevents delay propagation in case of schedule disruptions. In the opposite case, passengers would experience additional delays and an overall negative impression of the provided level of service.

2.2.2.2 Constraints

Regardless of the objective function or the methodological approach, the optimization is always subject to the following two constraints:

1. Flight constraint: Each flight is assigned to one gate (contact gate or remote gate - apron stand).

2. Gate constraint: Each gate is occupied by at most one flight at any moment.

Although a feasible schedule always satisfies the above constraints, under certain circumstances, they can both be relaxed during the optimization based on the objective function of the problem. On the one hand, the flight constraint can be relaxed when non-assignment is an option. The concept of non-assignment has various interpretations according to the decision maker. For example, not assigning a flight to any gate can automatically imply the assignment of the flight to any parking spot in the apron. On the other hand, when the gate constraint is relaxed, the final solution may involve two or more flights occupying the same gate at the same time. In this case, the objective usually includes the minimization of flight conflicts. If conflicts are included in the final solution, they can be resolved manually.

In addition, more types of constraints can be included in the model. These can be either operational constraints, which reflect tactical or practical limitations in the optimization, or methodological constraints, which depend on the mathematical formulation of the problem and are added to define logical relationships between the variables.

Operational Constraints:

Common operational constraints include the following:

- Shadow Constraints: Shadow constraints prevent flights which are both operated by aircraft with large wing span from occupying neighboring gates at the same time. A study which has considered such constraints is the one by Tang and Wang (2013).

- Minimum processing time constraints: Some studies assume that the duration of gate occupancy is not fixed for each flight, but that there exists a minimum processing time limit, as in Dorndorf et al. (2007b).

Methodological Constraints:

These constraints depend on the formulation of the problem and ensure feasibility according to the solution representation. For example, in assignment formulations, a set of constraints ensures that the main decision variables are binary, while network flow formulations include constraints for flow conservation and arc capacity.

2.2.2.3 Methodological Approach

We will now examine state-of-art GAP approaches based on the methodological approach, i.e., the type of mathematical formulation and the solution algorithm.

Mathematical Formulation

We can identify three main types of mathematical formulations used for modeling the problem:

1. Assignment models. In a typical assignment problem, one has to allocate agents to tasks, usually under specific constraints. In the context of GAP, a flight is an agent, which can be assigned to exactly one task (gate). Additional constraints guarantee the feasibility of the problem and exclude infeasible assignments based on flight-gate compatibility. Such models are generally formulated as Binary Integer Programming models with generally linear LP relaxations. However, when passenger connections are considered in the optimization (e.g., Haghani and Chen (1998)), the most common approach involves formulating the problem as a quadratic assignment

problem, which can be linearized with the addition of extra variables and constraints. The assignment formulation is the most common in the literature (Bihr, 1990; Bolat, 2000; Şeker and Noyan, 2012; Ding et al., 2004b, 2005; Guépet et al., 2015; Haghani and Chen, 1998; Kim et al., 2013; Marinelli et al., 2015; Yu et al., 2016).

2. Network flow models. Using flow networks is the second most common way to handle GAP. In this case, GAP is usually formulated as a multi-commodity network flow problem (Castaing et al., 2016; Cheng et al., 2012; Tang and Wang, 2013; Yan and Chang, 1998; Yu et al., 2016). In a multi-commodity network flow formulation, each gate corresponds to one network, where a feasible flow between the source and the sink represents the sequence of flights that are assigned to the gate throughout the planning horizon. A less common approach is the use of a time-space network (Yan and Chang, 1998).

3. Graph models. More recently, Dorndorf et al. (2008, 2012) relied on a graph representation to formulate GAP as a clique partitioning problem. In their approach, the graph consists of $n + m - 1$ vertices, divided in two sets: The first set contains n vertices corresponding to flight activities, while the second set contains $m - 1$ vertices which correspond to gates. The assignment of a flight activity to a gate is indicated by selecting the edge that connects the two respective nodes, while a set of transitivity constraints defines vertices belonging to the same clique in the final solution. The weight of an edge connecting an activity vertex with a gate vertex is equal to the preference value of the respective assignment (or to $-\infty$, if the gate

and the flight are not compatible). Finally, edges connecting two gate nodes have a weight equal to $-\infty$.

Solution Algorithms

Various solution approaches have been proposed for GAP. Some of them involve exact methods that find the optimal solution under specific assumptions and problem conditions. More recently, the large size of the problem has made researchers focus on the development of heuristic approaches, that can produce optimal or near-optimal solutions in an acceptable amount of time. Furthermore, the adaptation of metaheuristics is another research direction which has gained increasing popularity in recent years. We will examine the following categories of solution algorithms: Exact methods, heuristics approaches, and metaheuristics.

1. Exact methods: For relevantly simple problems, using an MIP solver to apply a typical branch-and-bound or branch-and cut approach can produce exact optimal solutions in a small amount of time (Haghani and Chen, 1998). Before MIP solvers evolved into the powerful tools they are today, exact solutions could be found for formulations where the optimal solution to the LP relaxation was provably integer under certain conditions (Bihr, 1990; Bolat, 2000)). Other studies have applied a Lagrangian relaxation approach aided by subgradient methods (Yan and Chang, 1998) and a column generation technique (Yan and Huo, 2001).

2. Heuristics: Haghani and Chen (1998) proposed a constructive heuristic which relies on the iterative assignment of flights to gates. Ding et al. (2004a,b) devel-

oped a greedy heuristic, where the flights were sorted by departure time which they proved to be optimal under certain conditions, as well as a neighborhood search approach. Dorndorf et al. (2007b) developed a truncated branch-and-bound method which relied on two models to ensure robustness. The first model used the concept of fault tolerant recovery paths, while the second relied on a fuzzy approach using a membership function. The clique partitioning problem of Dorndorf et al. (2008) was solved using an ejection chain algorithm. Dorndorf et al. (2012) proposed an algorithm which iteratively solves the problem for multiple periods. The authors used the ejection chain algorithm from Dorndorf et al. (2008), but they also showed that their algorithm could be easily coupled with any heuristic that is able to improve a given starting solution. More recently, a modified version of the ejection chain algorithm was used in Dorndorf et al. (2017). Genç et al. (2012) developed a "Big Bang-Big Crunch" method coupled with a greedy algorithm for minimizing the number of flights that are assigned to the apron, as well as a heuristic to maximize ground time duration. The authors used the heuristics to generate good initial solutions which they improved using stochastic search. In another study, Guépet et al. (2015) tested a number of MIP-based heuristics for the binary integer assignment formulation, namely spatial decomposition, time decomposition, a greedy algorithm, and an ejection chain algorithm. MIP heuristics were also the focus of Yu et al. (2016), who adapted a diving heuristic, RINS (Relaxation Induced Neighborhood Search), Local Branching, and VRNS (Variable Reduce Neighborhood Search), in order to generate schedules which exhibit robustness to potential disruptions.

3. Metaheuristics: The general applicability of metaheuristics has resulted in numerous adaptations for GAP in recent years. One of the first studies where a metaheuristic technique was applied in this context is the one by Bolat (2001), who adapted a genetic algorithm (GA) to solve the problem. To represent the solution, each gene corresponded to a flight, while the value of the gene corresponded to the gate where the flight was assigned. Ding et al. (2004b) adapted tabu search (TS), while Ding et al. (2004a) proposed a hybrid simulated annealing-tabu search (SA-TS) approach. In both studies, tabu search used three main search moves: "Insert", "interval exchange" and "apron exchange". Cheng et al. (2012) used real data to evaluate different objective functions by testing different metaheuristics, namely a genetic algorithm (GA), tabu search (TS), simulated annealing (SA), as well as a hybrid approach combining tabu search with simulated annealing. Tabu search was also used by Şeker and Noyan (2012) and Kim et al. (2013). Finally, Marinelli et al. (2015) used Bee Colony Optimization (BCO).

2.3 The Gate Reassignment Problem: Review

We can claim that the gate reassignment problem is, more or less, a "child" problem of GAP. Similarly to its parent problem, the purpose of gate reassignment is to determine the optimal assignment of flights to gates, as well as the starting and ending times of gate occupancy. The structure of the mathematical formulation is also the same for both problems. The elements that distinguish the two refer to the optimization objectives, the inclusion (or omission) of certain constraints, as well as the computational time requirements. We should keep in mind that gate reassignment is performed in real time, after new information on delays becomes available. Therefore, a new assignment that minimizes the impact of delays is required, and, most importantly, in a limited amount of time.

To gain an insight of existing studies and identify aspects of the problem that require further investigation, we will examine the problem of gate reassignment from three main aspects:

- The basic problem elements, i.e., objective function, constraints, methodological approach, and measures of effectiveness (section 2.3.1).
- The problem formulation (section 2.3.2).
- The state-of-art solution approaches (section 2.3.3).

2.3.1 Basic Problem Elements

In this section, we analyze the main features of the gate reassignment problem, such as the objective function (2.3.1.1), the constraints (2.3.1.2), the methodological approach (2.3.1.3), and the measures of effectiveness used to evaluate the quality of the solutions (2.3.1.4).

2.3.1.1 Objective function

Traditionally, the majority of gate reassignment studies aim to generate a schedule that is as close as possible to the original one. In that case, the objective function depends on the spatial and temporal deviation from the scheduled assignment. Under this reasoning, Gu and Chung (1999) and Maharjan and Matis (2011) minimize the walking distance of passengers affected by the reassignment. The former optimize a combination of the additional delay (i.e., delay caused exclusively by schedule adjustment) and the distance between the optimal and the reassigned gate, while the latter minimize the total walking distance of departing and connecting passengers for which the assigned gate is different from the one printed on the boarding pass. Furthermore, Yan and Tang (2007) minimize passenger waiting times, while Yan et al. (2009) minimize the number of gate changes. Tang et al. (2010) minimize two types of inconsistency: space inconsistency, for flights assigned to different gates than the original ones, and time inconsistency, for flights with altered starting time. Yan et al. (2011) use the same objective as Tang et al. (2010), but treat flights which are further away from the decision moment as stochastic. In a more recent study, Wang et al. (2013) classify each flight as certain (closer to the decision-making point of time) or uncertain, and minimize three objective function components, namely a) apron and gate disturbances for certain flights, b) apron and gate disturbances for uncertain flights and c) total penalty for the violation of gate constraints. Yu and Lau (2015) minimize the walking distance of connecting passengers and the number of passengers who miss their connection. Dorndorf et al. (2017) aim to keep the new assignment as close to the scheduled one as possible. Finally, Zhang and Klabjan (2017) minimize a weighted sum of delay, missed passenger connections and passenger transfer cost.

2.3.1.2 Constraints

As in the Gate Assignment Problem, the flight constraint (every flight is assigned to exactly one gate) and the gate constraint (every gate is occupied by at most one flight at any moment) are present in all gate reassignment approaches. In many studies (Gu and Chung, 1999; Tang et al., 2010; Yan et al., 2011; Yu and Lau, 2015), a gate adjacency constraint is also added, to prevent flights operated by aircraft with a wide wingspan from being assigned to adjacent gates at the same time. Dorndorf et al. (2017) also include "shadow restrictions" for situations where a flight blocks its neighboring gates.

Gu and Chung (1999) define a minimum ground time, as well as a minimum idle time between successive flight assignments to the same gate. Tang et al. (2010) additionally consider a maximum delay time constraint, while the models of Yan et al. (2009), Yan et al. (2011), and Maharjan and Matis (2011) can be extended to divide flights into zones to determine feasible assignments (e.g., Schengen and non-Schengen flights for European airports).

Wang et al. (2013) relax the gate constraint by allowing more than one uncertain flights to occupy the same gate concurrently. Finally, Dorndorf et al. (2017) include a towing time constraint, which ensures that the total time for towing does not exceed the available time for parking.

2.3.1.3 Methodological approach

The Gate Reassignment Problem is typically formulated as an integer programming model, usually either as a network flow problem (Yan et al., 2009; Yan and Tang, 2007) or as an assignment problem (Tang et al., 2010; Wang et al., 2013; Yan et al., 2011; Yu and Lau, 2015; Zhang and Klabjan, 2017). Depending on the formulation, different solution approaches have been proposed. Some studies (Tang et al., 2010; Yan et al., 2009, 2011) use exact solution techniques, like Branch and Bound. However, the large size of the problem, especially when passenger connections are involved, has made researchers focus on heuristic techniques (Dorndorf et al., 2017; Yu and Lau, 2015; Zhang and Klabjan, 2017) or adapt metaheuristics (Gu and Chung, 1999; Wang et al., 2013). The existing literature is summarized based on the objective function and constraints in Table 2.1, and based on the mathematical formulation and the solution methodology in Table 2.2.

2.3.1.4 Measures of Effectiveness

In the literature, the quality of flight-to-gate reassignment is evaluated based on a variety of Measures of Effectiveness (MOEs). We can classify these MOEs according to three criteria: The purpose of the reassignment, the "Flight Vs. Connection" criterion, and the weighting factors.

A. Purpose. We identify two main purpose directions in GRAP:

Table 2.1: Objective function and constraints of existing gate reassignment studies.

Paper	Objective	Constraints
Gu and Chung (1999)	Extra delayed time and walking distance	Flight, gate
Yan and Tang (2007)	Passenger waiting times	Flight, gate, zoning
Yan et al. (2009)	Number of gate changes	Flight, gate
Tang et al. (2010)	Total time and space inconsistency	Flight, gate, shadow, zoning, maximum delay
Maharjan and Matis (2011)	Total walking distance of affected passengers with boarding passes issued before reassignment	Flight, gate, zoning
Yan et al. (2011)	Total time and space inconsistency	Flight, gate, shadow, zoning
Wang et al. (2013)	Apron and gate disturbances, gate constraint violation	Flight, gate, zoning
Yu and Lau (2015)	Walking distance of transferring passengers and number of passengers who miss their connection	Flight, gate, shadow
Dorndorf et al. (2017)	Number of flights assigned to different gates	Flight, gate, shadow, towing time
Zhang and Klabjan (2017)	Total delay, number of missed connections, number of gate reassignments	Flight, gate

Table 2.2: Methodological approaches of existing gate reassignment studies.

Paper	**Formulation**	**Solution**
Gu and Chung (1999)	-	Metaheuristic (Genetic Algorithm)
Yan and Tang (2007)	Multi-commodity network flow problem	Heuristic
Yan et al. (2009)	Multi-commodity network flow problem	Exact Solution
Tang et al. (2010)	Assignment (Binary Linear)	Exact Solution
Maharjan and Matis (2011)	Assignment (Binary Quadratic)	Exact Solution
Yan et al. (2011)	Assignment (Binary Linear)	Exact Solution
Wang et al. (2013)	Assignment (Binary Linear)	Metaheuristic (Ant Colony Optimization)
Yu and Lau (2015)	Assignment (Binary Linear) and Multi-commodity network flow problem	Heuristic
Dorndorf et al. (2017)	Clique Partitioning Problem	Modified Ejection Chain
Zhang and Klabjan (2017)	Multi-commodity network flow problem	Guided diving heuristic Variable rolling horizon

- To minimize the deviation from the planned assignment. This category includes objectives such as minimizing the number of gate changes or delayed flights, expressed as a function of the number of passengers, delay duration (for temporal disruption), and the distance between scheduled and reassigned gate (for spatial disruption).
- To adjust the assignment such that passenger inconvenience and delay propagation is minimized. This category includes the minimization of walking distance for connecting passengers whose flights undergo gate changes.

B. Flight Vs. Connection. MOEs may refer to successful or failed individual flights or flight connections. A flight assignment is considered successful when, in the final solution, the gate and time window of the flight have been determined. In most cases, a solution is considered feasible when all flights have been assigned to gates. However, some approaches (e.g., Zhang and Klabjan (2017)) allow ungated flights, which are assigned to the apron. Regarding transfers, a connection is successful when there exists adequate time between the departing and the arriving flight, while it fails if the available time is less than the required time, or if either of the connecting flights is canceled. Based on this classification, we identify four types of MOEs:

(a) MOEs that depend on successful flight assignments: They express the cost of assigning a flight to a specific gate and time window.

(b) MOEs that depend on failed flight assignments: They express the cost of failing to produce a gate assignment for a flight.

(c) MOEs that depend on successful flight connections: They express the combined cost of assigning a pair of connecting flights to specific gates and time windows.

(d) MOEs that depend on failed flight connections: They express the cost of misconnections (failed connections).

This classification is important from a methodological perspective. In assignment-based formulations, types (a)-(b) and (c)-(d) correspond to the cost coefficients of the assignment variables and the quadratic terms respectively; in network flow-based formulations, types (a)-(b) and (c)-(d) correspond to the arc weights in the flight assignment and passenger connection network, respectively.

C. **Weighting Factors.** The impact of disruptions can be quantified based on (a) the number and type of passengers it affects, as well as (b) the temporal and spatial deviation from the scheduled assignment:

(a) Target passenger group: The decision maker may choose to minimize the impact of disruptions on priority passengers, i.e., business and first-class passengers although this approach is more common in planned gate assignment approaches.

(b) Time and space: Time and space express the magnitude of the difference between the scheduled assignment and the reassignment. For example, moving a flight to a gate located next to the scheduled one is less disruptive for the schedule than relocating it to a different concourse.

Table 2.3 summarizes the MOEs used in the literature, based on the relevant studies where they can be found, as well as based on their purpose, while Table 2.4 presents their classification based on the "flight Vs. connection" criterion, as well as based on their weighting factors.

Table 2.3: Classification of the Measures of Effectiveness Based on Studies and Purpose

MOE	**Study**	**Purpose**
Delay Time	Gu and Chung (1999); Zhang and Klabjan (2017)	Deviation
Distance between original and reassigned gates	Gu and Chung (1999)	Inconvenience and Propagation
Number of Gate Changes	Dorndorf et al. (2017); Wang et al. (2013); Yan et al. (2009); Zhang and Klabjan (2017)	Deviation
Total Time Inconsistency	Tang et al. (2010)	Deviation
Total Space Inconsistency	Tang et al. (2010)	Deviation
Apron Disturbances	Wang et al. (2013)	Deviation
Walking Distance of Connecting Passengers	Maharjan and Matis (2011); Yu and Lau (2015)	Inconvenience and Propagation
Number of passengers who miss their connection	Yu and Lau (2015)	Inconvenience and Propagation
Number of missed connections	Zhang and Klabjan (2017)	Inconvenience and Propagation

2.3.2 Mathematical Formulation

From a mathematical perspective, the gate reassignment problem is quite similar to its parent problem, GAP. Therefore, the alternative formulation types used for gate reassignment are the same as the ones used in GAP, which were presented in section 2.3.1.3. Overall, the gate reassignment problem can be formulated:

- As an assignment model with side constraints (Maharjan and Matis, 2011; Tang et al., 2010; Wang et al., 2013; Yan et al., 2011; Yu and Lau, 2015).
- As a network flow problem (Yu and Lau, 2015; Zhang and Klabjan, 2017).
- As a clique partitioning problem (CPP) (Dorndorf et al., 2017).

However, few studies present exact mathematical formulations for modeling passenger flows between gates. This aspect of the solution has either been overlooked in the literature or handled indirectly. The main reason is that modeling passenger connections increases significantly the size of the problem and therefore the required computational time. The problem is more evident in cases where the operator determines not only the gate of each flight, but also the time window during which a flight occupies a gate. In this dissertation, we will focus on the assignment and network flow formulations, which are the most common ones.

2.3.2.1 Gate Reassignment as an Assignment Model

Formulating the problem as an assignment model with side constraints is one of the most common approaches not only to the planned gate assignment, but to the

Table 2.4: Classification of the Measures of Effectiveness Based on "Flight Vs. Connection" and Weighting Factors

MOE	**Flight Vs. Connection**	**Weighting Factors**		
		Passengers / Distance / Time	**Target Passengers**	**Spatial / Temporal**
Delay Time	Successful Flight	Time	All	Temporal
Distance between original and reassigned gates	Successful	Flight	Distance	All Spatial
Number of Gate Changes	Successful	Flight	None	All Spatial
Total Time Inconsistency	Successful Flight	Time	All	Temporal
Total Space Inconsistency	Successful Flight	Distance	All	Spatial
Apron Disturbances	Successful Flight	None	All	Spatial
Walking Distance of Connecting Passengers	Successful Connection	Passengers	Those with wrong gate number on boarding pass	Spatial
Number of passengers who miss their connection	Passengers	All	Temporal	
Number of missed connections	Failed Connection	None	All	Temporal and Spatial

gate reassignment problem as well. Tang et al. (2010),Maharjan and Matis (2011), Yan et al. (2011), Wang et al. (2013), and Yu and Lau (2015) are typical examples of studies of this category.

However, the concept of passenger connections has not been adequately addressed in assignment formulations. The main reason behind the lack of such studies is the increase in the size of the problem due to the introduction of quadratic constraints. To explain in more detail the concept of assignment formulation combined with passenger flows, we will briefly introduce the notation and the basic variable definitions of a typical assignment problem.

In an assignment problem, one deals with (at least) two sets of elements, I and J, where $i \in I$ represent tasks that have to be completed, and $j \in J$ represent agents, employees, or machines, where the tasks have to be assigned. In a typical assignment formulation, the main decision variable is binary variable X_{ij}, such that:

$$X_{ij} = \begin{cases} 1, & \text{if } i \in I \text{ is assigned to } j \in J \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

Each potential assignment of i to j is associated with a cost, corresponding to the coefficient C_{ij} of variable X_{ij} in the objective function. The objective of the problem is to find the assignment which minimizes the total cost $\sum_{i \in I} \sum_{j \in J} C_{ij} X_{ij}$, such that all tasks are assigned to some agent ($\sum_{j \in J} X_{ij} = 1, i \in I$). The optimization is usually subject to a set of side constraints: For example, in a machine scheduling problem, such constraints prevent the use of a machine for two or more tasks at the same time.

In the gate assignment (or reassignment) problem, an assignment formulation can be built based on the representation of flights as tasks $i \in I$ and of gates as machines, or agents, $j \in J$. The flight constraint requires that every flight is assigned to one (and only one) gate, while, similar to a parallel machine scheduling problem, the gate constraint prevents the occupancy of a gate by two or more flights concurrently.

However, when we consider passenger connections, we examine flights, and their respective assignments, in pairs: In this case, we deal with a modified version of the Quadratic Assignment Problem (QAP). Introduced by Koopmans and Beckmann (1957), the objective of QAP is to minimize the total assignment cost where the assignment cost of a pair of facilities j, j' depends on the flow, but also on the distance between them. In the original form of QAP, the number of locations is equal to the number of facilities and the problem is formulated as follows:

Sets:

$N = \{1, 2, \ldots, n\}$:	Set of flights.
$S_n = \phi : N \to N$:	Set of permutations.
G:	Set of facilities.
T:	Set of passenger connections.

Parameters:

$F = (f_{ij})$:	A $n \cdot n$ matrix, where f_{ij} is the flow from facility i to facility j.
$D = (d_{ij})$:	A $n \cdot n$ matrix, where d_{ij} is the distance from facility i to facility j.

Objective Function:

$$\text{Minimize} \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{\phi(i)\phi(j)} \tag{2.2}$$

When the problem was first proposed, all candidate locations where feasible for all facilities, so every permutation represented a feasible solution. Of course, there are significantly more constraints in a gate reassignment problem than in the traditional QAP. However, the common element between the two problems is the calculation of cost as a function of both location and flows, when passenger connections are considered. De Neufville and Odoni (2003) analyze the movement of passengers using two origin-destination matrices: The impedance matrix and the flow matrix.

The impedance matrix D defines the level of difficulty in the movement of passengers between two locations in the airport. The impedance depends on the physical characteristics of the building and can be expressed as walking distance but it can also take into account the existence of transporters, people movers, or moving walkways, which in turn affect passenger walking time. The flow matrix F determines the passenger volume moving between every OD pair and therefore captures the number of transferring passengers as well.

The entrywise product of the flow and the impedance matrix is the passenger-impedance matrix, where each element $F_{ij} \cdot D_{ij}$ corresponds to the impedance of moving between an origin i and a destination j, weighted by the number of passengers who move from i to j. Among the measures of effectiveness analyzed in Table 2.3, total walking distance is such an impedance measure.

Let us now explain how the connection of passenger impedance to the quadratic assignment formulation of our problem. Using the decision variable X_{ij} defined at the beginning of this section, we also introduce the following notation:

$N_{ii'}$: Flow (number of passengers) between flights i, i'.

$D_{jj'}$: Impedance (e.g. distance or time) between gates j, j'.

The total impedance $C_{ii'}$ of connecting passengers from i to i', as a function of the main decision variables, can be expressed as:

$$C_{ii'} = N_{ii'} \sum_{j} \sum_{j'} D_{jj'} (X_{ij} \cdot X_{i'j'}) \tag{2.3}$$

and the respective total cost TC for all transfers $(i, i') \in T$, as:

$$TC = \sum_{(i,i') \in T} N_{ii'} \sum_{j} \sum_{j'} D_{jj'} (X_{ij} \cdot X_{i'j'}) \tag{2.4}$$

As can be seen, the total passenger impedance and, respectively, the total connection cost, depends on the product of decision variables X_{ij} and $X_{i'j'}$, which, from the perspective of the mathematical formulation, implies that the assignment problem is now quadratic. In the literature, a study that models passenger flows directly is the one by Maharjan and Matis (2011), who develop a quadratic model to minimize the total walking distance of originating and connecting passengers whose boarding passes were issued prior to gate changes.

Since the decision variables are binary, linearizing the quadratic problem is straightforward: We simply define the product of every pair of variables X_{ij}, $X_{i'j'}$

as a new decision variable $Z_{iji'j'}$:

$$Z_{iji'j'} = X_{ij} \cdot X_{i'j'} \tag{2.5}$$

such that

$$Z_{iji'j'} \leq X_{ij} \tag{2.6}$$

$$Z_{iji'j'} \leq X_{i'j'} \tag{2.7}$$

$$Z_{iji'j'} \geq X_{ij} + X_{i'j'} - 1 \tag{2.8}$$

Linearizing the problem allows us to use an LP solver to find the optimal solution. However, at the same time, the introduction of the new variables and the linearization constraints also makes the problem intractably large. The additional computational challenges conflict with the requirement for fast solution procedures in real-time assignment and are exaggerated for formulations that rely on multi-indexed decision variables.

In this dissertation, our decisions include not only the gate to which every flight is assigned, but also the exact time when the flight starts to occupy the gate. We therefore adopt a time-index formulation, with our main decision variable X_{ijk}, such that

$$X_{ijk} = \begin{cases} 1, & \text{if gate } i \text{ is assigned to gate } j \text{ and time window } k \\ 0, & \text{otherwise} \end{cases} \tag{2.9}$$

Therefore, the quadratic decision variables are now defined as

$$Z_{ijki'j'k} = X_{ijk} \cdot X_{i'j'k'} \tag{2.10}$$

while the respective linearizing constraints are modified accordingly

$$Z_{ijki'j'k} \leq X_{ijk} \tag{2.11}$$

$$Z_{ijki'j'k} \leq X_{i'j'k'} \tag{2.12}$$

$$Z_{ijki'j'k} \geq X_{ij} + X_{i'j'k'} - 1 \tag{2.13}$$

Inevitably, the 3-index formulation results in an even larger number of variables and respective constraints than the 2-index formulation. A detailed estimation of upper bounds on the number of variables and constraints for the time-indexed quadratic formulation, as well as for different versions of the assignment formulation, follows in Chapter 4. In summary, the multidimensional assignment formulation of the gate reassignment problem is straightforward in its reasoning, but results in a large increase in the size of the problem.

2.3.2.2 Gate Reassignment as a Network Flow Problem

The second most common formulation of the gate reassignment problem relies on the use of network flows. In a typical network flow representation, one network is created for each gate, with nodes corresponding to time windows. A feasible flow corresponds to a sequence of flights occupying the gate throughout the planning horizon: Starting from the source, the order of the incident nodes of arcs with positive flow corresponds to the sequence of flights assigned to the gate, as well as their respective time windows. The problem is then solved as a minimum cost network flow problem. Examples of studies that rely on network flows for the solution representation are the ones by Yan and Tang (2007), Yan et al. (2009), Yu and Lau (2015), and Zhang and Klabjan (2017).

However, similarly to the case of the binary assignment formulation (2.3.2.1), the multi-commodity network flow formulations also require extensions to be able to handle passenger connections. So far, Yu and Lau (2015) and Zhang and Klabjan (2017) are the only studies in the literature that consider passenger connections in multi-commodity network flow formulations.

On the one hand, Yu and Lau (2015) minimize the total assignment cost and maximize the number of passengers who miss their connecting flight. Their formulation relies on a single multi-commodity network for the whole problem. The planning horizon is divided in time windows. Each node of the network corresponds to the beginning of a time window, while every arc corresponds to a single flight. However, each flight is associated with more than one arcs, with different starting and/or ending nodes. A positive flow in the network indicates the sequence of flights occupying every gate. A set of flow conservation constraints ensures the feasibility of the assignment for each gate, while a separate set of constraints ensures that every flight is assigned to one and only one gate.

On the other hand, Zhang and Klabjan (2017) build an assignment network for each gate, and a passenger network for each connection. Each network is associated with its own set of variables and constraints, while an additional set of constraints establishes the relationship between the two networks. Among the main innovations of this study are a) the introduction of passenger connection networks, and b) the use of gate cliques, i.e., clusters of neighboring gates such that the distance between any two gates is defined uniquely by the distance of the cliques where the gates belong to.

The assignment network is very similar to the one by Yu and Lau (2015), but in this case each gate has its own network, while feasibility is guaranteed by using flow conservation constraints for each network. For passenger connections, the authors create one network for each transfer. The nodes in each transfer can be divided in two sets: The nodes on the left side of the graph refer to the arriving (inbound) flight of the connection, while the nodes on the right side of the graph refer to the departing (outbound) flight of the connection. Each node corresponds to a unique combination of clique and time window where the respective flight can be assigned to. A feasible flow in this network starts from the source, moves to a node corresponding to the arriving flight, connects this node with a node corresponding to the departing flight, and moves to the sink. A cycle arc is also added to ensure flow conservation.

2.3.2.3 Gate Reassignment as a Clique Partitioning Problem

A more recent approach by Dorndorf et al. (2016) uses a clique partitioning problem formulation. In an undirected graph, a clique is a subset of the vertices which are all pairwise adjacent. Consequently, when the graph is complete, every subset of its vertices is a clique. For a complete and weighted graph, the clique partitioning problem is to "partition the vertices into cliques such that the sum of all edge weights within all cliques is maximized" (Dorndorf et al., 2016).

In the problem representation, each node represents either a flight activity or a gate. A flight activity i is assigned to a gate k if nodes j and k belong to the same clique. If there is a flight activity that does not belong to the same clique as any gate, then the activity is assumed to be assigned to apron ("dummy gate"). The objective of the problem is to maximize the total flight-gate preference score, thus the weights of the edges are selected so that they reflect the assignment feasibility and the objective function. The weight of the edges which connect vertices that must not be in the same clique is equal to $-\infty$. There are three types of vertex pairs that cannot be assigned to the same clique:

- Pairs of gate vertices.
- Pairs of activity vertices which overlap in time.
- A flight activity vertex with any gate vertex if the flight is incompatible with the gate.

The weight of every other edge is equal to the respective weighted preference value: For a pair of activity vertices, this reflects the cost of towing operations, expected overlaps, and shadow restrictions, while for a gate and an activity pair, it reflects the activity-gate preference score.

2.3.3 Solution Approaches

Using the same criteria as in 2.3.1.3, we can classify the solution approaches to the gate reassignment problem as exact methods, heuristics approaches, and metaheuristics.

2.3.3.1 Exact Methods

The computational power of modern commercial solvers has greatly facilitated the solution of integer (IP) and mixed integer (MIP) programming models using traditional MIP techniques, such as branch and bound. The solution procedure is facilitated by a number of computational procedures which most MIP solvers execute by default. Such procedures include presolving, cutting planes generation, and heuristic solution search. These properties of MIP solvers make them powerful tools for solving large problems to optimality in a few minutes or even seconds, depending on the problem size and the strength of its mathematical formulation. However, this does not mean that they are always fast enough so as to produce solutions within only a few minutes, as is required in gate reassignment. Apart from that, the user needs to provide the MIP solver with a strong problem formulation. As will be shown in Chapter 4, the time required to solve large instances of the gate reassignment problem can exceed the acceptable time limits, while the problem formulation can make a tremendous difference in the resulting running time.

When passenger transfers are not considered, the gate reassignment problem can be solved fast with exact methods, both for the assignment formulation (Tang et al., 2010; Yan et al., 2011) and for the network flow formulation. There are also cases (Maharjan and Matis, 2011; Yan et al., 2009) that consider passenger transfers and use exact methods (which are not explicitly specified) to solve an instance of the problem to optimality within a few seconds. However, in these studies, the decision variables include only the gate where each flight is allocated, and not the respective time window.

2.3.3.2 Heuristics

For large and complicated problems, for which solutions are required within a limited amount of time (as in the case of real-time assignment), the use of heuristic techniques might be the only viable solution. As a result, many researchers have developed heuristics to generate near-optimal solutions fast and effectively.

Yan and Tang (2007) developed a heuristic method and embedded it in a solution framework to minimize the impact of flight delays under uncertainty. The framework consists of three main components: A stochastic flight-gate assignment model, which corresponds to the planning stage, a reassignment model, which corresponds to the real-time (reassignment) stage, and two penalty adjustment methods. In the planning stage, a planned assignment schedule is obtained based on a number of randomly generated delay scenarios to capture the stochastic nature of delays. In the real-time stage, a different reassignment schedule is generated for every scenario, based on a specific rule process. Therefore, the reassignment is not formulated mathematically, but is performed based on a set of flow chart rules which determine which aircraft should be held, or assigned to a different gates, so that the feasibility of the original schedule is restored. Then, a penalty adjustment method is called, which penalizes the generated recovery schedules according to constraint violation (e.g., overlapping flights in the same gate). Using the newly adjusted penalty values, the planning stage model is solved again, and the procedure continues until a stopping criterion is met.

Yu and Lau (2015) use a MIP heuristic to solve the gate reassignment problem based on passenger connections. Their main idea is to divide passenger connections in two sets of predetermined size: A hard set, which contains all transfers that are not allowed to be missed in the final solution, and a soft set, with the transfers that are allowed to be missed, under some objective function penalty. First, the transfers are sorted by deceasing number of connecting passengers. Then, a predetermined number of transfers is moved to the hard set, with the selection probability increasing with the number of passengers, and the new problem is solved to optimality using CPLEX.

Zhang and Klabjan (2017), who, like Yu and Lau (2015) also use a network flow formulation to optimize for passenger connections, developed two MIP heuristic approaches, namely a guided diving heuristic and variable rolling horizon heuristic.

In the diving heuristic, the aircraft are first sorted by increasing arrival time. Then, the linear relaxation is solved. If the value of the flow in a flight arc is equal to 1, the flight is automatically fixed to the respective gate clique. A limited number of flights can also be fixed to gate cliques even if the resulting decision variables receive fractional values, based on the highest cumulative fractional values. When the maximum number of aircraft per iteration is fixed, the LP relaxation is solved again and the procedure is repeated. After all aircraft have been fixed to cliques, the MIP solver is called to solve a restricted version of the original problem.

The variable rolling horizon algorithm uses the guided diving heuristic to solve the problem for long reassignment windows. The reassignment window is divided in several intervals so that the passenger connections are uniformly distributed among them. Then, a subproblem is solved for each interval, using the diving heuristic described before.

Dorndorf et al. (2017), who used a clique partitioning problem formulation, applied an ejection chain algorithm. The algorithm divides the set of edges into clusters. The purpose of the algorithm is to identify the sequence ("chain") of at most n moves which results in the best objective function score. A move constitutes in moving an activity vertex from one cluster to another. In every step, the move which results in a feasible solution with the greatest improvement in the objective function is chosen, while the respective vertex is marked as tabu and the clusters are updated. At the end of the procedure, the value $r \in \{1, \ldots, n\}$ corresponding to a chain of r moves is selected if it resulted in the maximum objective value found so far. The respective solutions is marked as the new, improved solution, and the procedure starts from the beginning. The recovery strategy proposed by the authors also involved the manual resolution of conflicts in cases of infeasibility.

2.3.3.3 Metaheuristics

Thanks to advantages like their "black box" applicability and their ability to perform global search by escaping local optima, metaheuristics have been adapted by researchers to solve different versions of the gate reassignment problem.

One of the earliest studies of this kind is by Gu and Chung (1999), who adapted a genetic algorithm (GA) to minimize the total spatial and temporal deviation between the scheduled assignment and the reassignment. They adopt a simple solution representation, with every solution represented as a linear chromosome, such that the value of the gene in the nth position corresponds to the gate to which flight n is assigned. The fitness value of each individual is a convex combination of two normalized function values, one for temporal deviation and one for spatial deviation. The initialization of a feasible solution is followed by the common genetic algorithm operators, i.e. selection, crossover, and mutation. In the selection procedure, individuals with higher fitness values respectively have a larger probability to survive and reproduce. The crossover procedure is linear: For a pair of selected parents, the chromosomes are split in a specific location ("crossover point") and their parts are exchanged. In mutation, the value of each gene is changed with a given probability. Infeasible individuals are discarded and replaced with feasible ones. The procedure stops when a termination criterion (number of generations or fitness threshold) is satisfied.

More recently, Wang et al. (2013) adapted Ant Colony Optimization (ACO) to solve the problem of real time gate assignment in a hub airport. Based on the fact that airport managers continuously receive new information about delays, they classify the flights as "certain" or "uncertain". A flight is "certain" if it is expected to arrive or depart in the near future, at some time close to the time of the decision making. On the contrary, if the scheduled time of the flight is further away in the future, the flight is treated as "uncertain". The decision maker has to determine the gate where each flight, either certain or uncertain, will be assigned; however, for uncertain flights, the assignment time is also a decision variable. In the ACO adaptation, each ant corresponds to a feasible solution. The ant first traverses the 2-dimensional solution space of certain flights and then the three-dimensional solution space of uncertain flights. While traversing the solution space, the ant consecutively adds nodes to the solution under construction, where each node corresponds to the assignment (i.e., a gate for certain flights, or a combination of gate and time for uncertain flights). The node selection is based on the pheromone concentration of the candidate nodes, as well as on the heuristic information of the nodes that follow. At the end of each iteration, the pheromone information is updated based on the fitness function of the new solution. The procedure is iterated until a predetermined number of iterations is completed, i.e., until all ants have traversed the solution space.

2.4 Conclusions: From state-of-art to this dissertation

As explained in the introduction (Chapter 1), there exist rather few studies on the gate reassignment problem, compared to the planned assignment "parent" problem. Even fewer of them (Maharjan and Matis, 2011; Yu and Lau, 2015; Zhang and Klabjan, 2017), focus on the optimization of passenger transfers. After reviewing the literature, it is clear that there is a lack of studies which incorporate passenger connections in a realistic way.

Although the literature on gate reassignment is not vast, we observe that studies present a large variety regarding the objective functions and constraints which are used across the different studies. To understand the problem in more depth, it is essential to study the currently applied objectives from a practical and methodological point of view and also examine how the problem is affected by the inclusion of objectives that refer to passenger connections (Chapter 3).

Another area of focus of this dissertation is the mathematical formulation of the gate reassignment problem, when passenger connections are considered. We will therefore study existing formulations, compare their performance, and incorporate state-of-art knowledge about the parent problem GAP to improve the computational efficiency of the mathematical formulation (Chapter 4).

After we have established an efficient problem formulation, we will proceed to the adaptation of heuristic and metaheuristic techniques. As can be seen, current metaheuristic approaches do not take advantage of the MIP formulation of the problem. In this study, we will explore the use of modern metaheuristic techniques, such as Variable Neighborhood Search, which combine the computational capabilities of an MIP solver with the exploration abilities of metaheuristics (Chapter 5).

Chapter 3

A Novel Multidimensional Assignment Model for Gate Reassignment With Passenger Connections

Chapter Overview

In this chapter, we introduce a multidimensional assignment model for optimizing flight-to-gate reassignment considering transfer passengers. The proposed binary model is the first multidimensional assignment model in the literature that uses gate location and the resulting required connection time to assess the success of passenger transfers. We show that the model is easily generalizable, since it can be optimized for various objective functions and can also be extended to consider apron capacity and flight cancellations. After formulating the model, we review the measures of effectiveness used in current gate reassignment approaches. Afterwards, we perform a set of preliminary experiments to demonstrate the main features of the model and verify its sensitivity to input changes. Then, we apply it in a real case study, where we examine the interaction of measures of effectiveness and investigate the impact of our proposed way of modeling missed connections on the optimal solution. Our results demonstrate the necessity of considering passenger connections in the optimization procedure since they contribute significantly to the total solution cost. We also show that, except for extreme cases (negligible disruptions or high delays within the airport), considering gate location yields different results than simply assuming that a fixed time threshold is sufficient for all connections.

Keywords: *Gate reassignment; passenger connections; measures of effectiveness; integer programming*

3.1 Introduction

As explained in Chapter 1, transfer passengers often comprise a large percentage of airport users and generally have different needs compared to passengers who start or end their trip at the airport (De Neufville and Odoni, 2003). Since failed passenger connections result in high costs for airports and airlines, passenger connections affect all stages of decision making. Gate assignment is a typical operation that affects the level of service provided to transfer passengers. In this context, Maharjan and Matis (2012) propose a GAP model that minimizes passenger discomfort for rushed connections. Also, Narciso and Piera (2015) develop a GAP model where the distances between terminals are critical for transfer connectivity when inbound flights are late.

However, the only studies that consider missed connections in the recovery phase of gate assignment are the ones by Yu and Lau (2015) and Zhang and Klabjan (2017). On the one hand, Yu and Lau (2015) assume that, as long as a predetermined time threshold exists between two connecting flights, the connection will be made, regardless of the location of gates. For the rest of this paper, we will refer to this assumption as "the simple assumption". On the other hand, the multi-commodity network flow model by Zhang and Klabjan (2017) takes gate location into account, but does not quantify the required connection time between gates and cannot be generalized for cases where the cost of allowing a connection is higher than the cost of missing it. Also, the model is solved with heuristic methods because of the problem size, while our model is solved to optimality using branch-and-cut procedures for small and medium-size cases.

3.1.1 Contributions of This Study

Motivated by the lack of studies that incorporate missed connections in gate reassignment in what we consider to be a realistic way, as well as by the variation in the MOEs used in current literature (Chapter 2), we examine the problem of flight-to-gate reassignment from two perspectives:

First, we develop the first gate reassignment model in the literature which combines both of the following properties: a) It is formulated as an assignment problem, where flights are the agents that undertake specific "tasks", i.e., combinations of a gate and a time window. b) It considers distances between gates and respective required connection times when determining whether a connection will be missed. Furthermore, the proposed model can be extended to account for apron assignment and flight cancellations. Contrary to current studies (Wang et al., 2013; Zhang and Klabjan, 2017) which assume unlimited apron capacity, our approach considers scheduled apron occupancy (section 3.3.2.2) and can be extended to include cancellation decisions (section 3.3.2.1).

Second, we use the proposed model to analyze the different optimization objectives considered in the gate reassignment literature. In particular, we examine how including passenger connections affects the quality of the optimal solution. We also show how MOEs interact with each other and test their ability to replicate actual delay cost. To quantify cost accurately, we use official guidelines and existing studies that consider cost aspects such as the piecewise linearity of the cost-delay function, or the impact of alternative flight availability on passenger compensation. In the existing literature, accurate cost estimation has either been completely ignored or only included in approximate solution approaches. Our model allows us to switch between different objectives by adjusting the cost coefficients of the objective function terms.

The generalized model and the analysis presented in this chapter have been accepted for publication in Pternea and Haghani (2019).

The rest of the chapter is structured as follows: In section 3.2, we explain the impact of passenger connections on decision making. In section 3.3, we present the mathematical formulation of the model. The design and results of our experiments are discussed in section 3.5. Finally, in section 3.6, we summarize our main conclusions and propose directions for future research.

3.2 Missed Connections

As explained in Chapter 1, our analysis is based on the idea that treating the potential success or failure of a transfer as a function of both the required and the available connection time is more realistic than simply assuming that a connection will be successful as long as the available time between the connecting flights satisfies a predetermined threshold.

3.2.1 Why Required Connection Time matters

While in planned assignment scheduled transfers satisfy a minimum connection time, the available connection time in case of schedule disruptions might not be sufficient for transfer passengers. The problem is more eminent in large hub airports, as in the case of European hubs serving connecting passengers who travel from all across Europe to a US destination.

In general, scheduled connections served by the same airline (or airlines of the same alliance) take place in the same terminal, since airline-dedicated facilities facilitate intra-line connections (de Barros et al., 2007; Phillips, 1987; Wu and Lee, 2014). However, this is not always the case: First, even airlines of the same alliance might be located in different terminals. Second, some terminals might be dedicated to specific flight categories, regardless of the operating airline (e.g., JFK's Terminal 4 serves international flights (John F. Kennedy International Airport official website, 2018)). Third, in the case of self-transfers, where the connecting flights do not belong to the same alliance, passengers have to walk between terminals. Especially when the airport consists of multiple terminals, the time required to move between them is far from negligible, as it depends on the available transportation modes and the existence of moving walkways, elevators, etc. Therefore, when an inbound flight is delayed, and the distance between its gate and the gate of the departing flight is large, transfer passengers are at risk of missing their connecting flight.

3.2.2 Estimation of Required Connection Time

Since the goal of our formulation is to incorporate the concept of passenger connections in the most realistic way possible, we need to estimate the required connection time in detail. The required connection time consists of two parts:

a) The time for moving between the two gates by walking and using the available airport transportation facilities, such as people movers and moving walkways, and

b) The time spent in mandatory procedures.

The time required for passengers to move between gates depends on the distance between them, the walking speed within the terminal, as well as on the existence of means that assist passenger mobility, like moving walkways. On the one hand, connection time depends largely on the design of the airport; De Neufville and Odoni (2003) identify the percentage of connecting passengers as the most important parameter affecting the selection of terminal layout. For example, midfield concourses, as in Chicago/O'Hare and Munich, facilitate passenger connections, in contrast to linear buildings with one airside, or decentralized buildings, as in the cases of Kansas City and Sydney Airport (De Neufville and Odoni, 2003). On the other hand, moving walkways can facilitate passenger connections. Kusumaningtyas and Lodewijks (2013) show that accelerating walkways with a minimum length of 120 meters can reduce passenger transport time significantly.

Mandatory procedures depend on the type of connecting flights as well as local regulations and include disembarking, passport control, security check, potentially HRF (high-risk flight) screening, and boarding (Kusumaningtyas and Lodewijks, 2013). The procedures themselves, and consecutively their duration, vary according to the origin and destination of flights, as well as local regulations. For example, international transfer passengers with a domestic connection flight may have to undergo visa and passport control and then present their boarding pass or even go through check-in if a boarding pass was not issued at the origin airport. For European airports, the total time required for mandatory procedures ranges from around 19 minutes (between European flights) to 30 minutes (between intercontinental flights) (Ashford, 1988; Competition Commission, 2002a,b; Horstmeier and de Haan, 2013; IATA, 2004; Kusumaningtyas and Lodewijks, 2013). Meanwhile, the required Minimum Connecting Times (MCTs) range from 40 minutes (between European flights) to 50 minutes (from/to/between intercontinental flights). The duration of mandatory procedures is extended in the case of self-connecting passengers, who might be required to go through a new check-in and security control procedure, unless the airport provides special support for self-connections, as in Milan/Malpensa and London/Gatwick (Cattaneo et al., 2017).

The success of a connection in our model depends on the comparison between the required and the available connection time. Let us assume that a pair i and i' of connecting flights are allocated to gates j and j', starting at time windows k and k', respectively.

The required connection time $t_{jj'}^{req}$ includes disembarking the aircraft serving flight i, moving from gate j to gate j', and boarding flight i'. The connection time $t_{jj'}^{req}$ can be calculated using walking speed, moving speed of walkways, and the estimated duration of mandatory procedures (it can be further differentiated for every connection, if we assume that the duration of mandatory procedures varies according to the type of flights). Meanwhile, the available time $t_{jj'}^{av}$ between the two flights is calculated as $t_{jj'}^{av} = k' - k$. As a result, a connection (i, i') will be missed if and only if

$$t_{jj'}^{av} < t_{jj'}^{req} \Leftrightarrow k' - k < t_{jj'} \tag{3.1}$$

Using this condition, we form sets $Q_{ii'}^{A}$ and $Q_{ii'}^{F}$, which contain all combinations (j, j', k, k') resulting in success and failure, respectively, of each connection (i, i'). Sets $Q_{ii'}^{A}$ and $Q_{ii'}^{F}$ are then used to define the quadratic variables in the model (section 3.3).

3.3 Problem Formulation

In this section, we present the Integer Programming model for the Gate Reassignment Problem.

3.3.1 General Case

Let us define the following notation for the mathematical formulation:

Sets:

F: Set of flights.

G: Set of gates.

T: Set of passenger connections.

W: Set of time windows.

$G_i \subset G$: Set of gates that are compatible with flight i.

$W_i \subset W$: Set of time windows that are compatible with flight i.

F_k^O: Set of flights that may occupy a gate at time window k.

F_j^G: Set of flights that can be assigned to gate j.

G_j^N: Set of gates adjacent to gate j.

F^L: Set of flights which are served by large aircraft.

H_{is}: Set of time windows such that, if flight i is assigned to them, it occupies its gate at time window s.

$Q_{ii'}^A$: Set of allowed combinations (j, j', k, k') for connection (i, i'), i.e. combinations that result in connecting passengers catching the outbound flight.

$Q_{ii'}^F$: Set of forbidden combinations (j, j', k, k') for connection (i, i'), i.e. combinations that result in connecting passengers missing the outbound flight.

Parameters:

C_k^A: Number of aircraft scheduled to occupy the apron at time window k.

C^A: Apron capacity.

g^A: Apron gate (indicates the assignment of a flight to the apron).

Costs:

C_{ijk}^{FS}: Cost of assigning flight i to gate j at time window k.

$C_{ijki'j'k'}^{CS}$: Cost of successful passenger connection (i, i') when assigning flight i to gate j at time window k and flight i' to gate j' at time window k'.

$C_{ii'}^{CF}$: Cost of failed passenger connection (i, i').

Decision Variables:

X_{ijk}: Binary, equal to 1 if flight i is assigned to gate j at time window k, 0 otherwise.

$Z_{ijki'j'k'}$: Binary, equal to 1 if flight i is assigned to gate j at time window k, and flight i' is assigned to gate j' at time window k', where $(i, i') \in T$.

The problem is formulated as follows:

Minimize:

$$\sum_{i \in F} \sum_{j \in G_i} \sum_{k \in W_i} C_{ijk}^{FS} X_{ijk} + \tag{3.2}$$

$$\sum_{(i,i') \in T} \sum_{(j,j',k,k') \in Q_{ii'}^A} C_{ijki'j'k'}^{CS} Z_{ijki'j'k'} + \tag{3.3}$$

$$\sum_{(i,i') \in T} C_{ii'}^{CF} \sum_{(j,j',k,k') \in Q_{ii'}^F} Z_{ijki'j'k'} \tag{3.4}$$

Subject to:

$$\sum_{j \in G} \sum_{k \in W_i} X_{ijk} = 1, \; i \in F \tag{3.5}$$

$$\sum_{i \in F_k^O \cap F_j^G} \sum_{k \in H_{is}} X_{ijk} \leq 1, \; j \in G, \; s \in W \tag{3.6}$$

$$Z_{ijki'j'k'} \leq X_{ijk}, \; (i, i') \in T, \; j' \in G_{i'}, \; k \in W_i, \; k' \in W_{i'} \tag{3.7}$$

$$Z_{ijki'j'k'} \leq X_{i'j'k'}, \; (i, i') \in T, \; j' \in G_{i'}, \; k \in W_i, \; k' \in W_{i'} \tag{3.8}$$

$$Z_{ijki'j'k'} \geq X_{ijk} + X_{i'j'k'} - 1, \; (i, i') \in T, \; j' \in G_{i'}, \; k \in W_i, \; k' \in W_{i'} \tag{3.9}$$

$$X_{ijk} \in 0, 1, i \in F, \; j \in G_i, \; k \in W_i \tag{3.10}$$

Expressions 3.2 - 3.4 define the objective function, which consists of three components, corresponding to MOE types a, c and d, described in section 2.3.1.4. Expression 3.2 is the cost of flight assignment, expression 3.3 is the cost of successful passenger connections, and expression 3.4 is the cost of failed passenger connections.

Equation 3.5 ("flight constraint") forces each flight to be assigned to exactly one gate and time window. Constraint 3.6 ("gate constraint") defines that every gate must be occupied by at most one flight at any given moment.

Constraints 3.7-3.9 linearize the quadratic expression

$$Z_{ijki'j'k'} = X_{ijk} \; X_{i'j'k'}, \; (i, i') \in T, \; j' \in G_{i'}, \; k \in W_i, \; k' \in W_{i'} \tag{3.11}$$

which is used when transfer passengers are involved in the optimization, i.e., when terms (3.3) and (3.4) are included in the objective function. Finally, constraint

3.10 restricts the main decision variable X_{ijk} to be binary. Note that, because of constraints 3.7-3.9, the respective constraint for $Z_{ijki'j'k'}$ need not be specified explicitly.

When modeling passenger connections, large problem instances are generally too time-consuming for real-time optimization regardless of the formulation type. For real-size case studies and a relatively short reassignment windows, e.g., 2 hours, our model can optimize exactly within a few seconds, provided that the objective function does not include both expression 3.4 (missed connection cost) and expression 3.3 (successful connection cost). In this study, we allow sufficient time for the problem to be solved exactly in order to study the interaction between the MOEs. To the best of our knowledge, this is the only study that provides exact solutions for the gate reassignment problem while including missed connections directly in the optimization model.

3.3.2 Model Extensions

Constraints 3.5 - 3.10 define the basic formulation of the gate reassignment problem. However, the problem can be further modified to accommodate additional decisions, such as flight cancellation, as well as to include additional constraints that reflect practical and operational restrictions, as will be shown in paragraphs that follow.

3.3.2.1 Embedding Unassigned Flights and Cancellation Decisions

We now modify the formulation to handle the case of unassigned flights. The interpretation of an unassigned flight i differs according to the decision maker: For example, one might assume that unassigned flights will be handled in a later step of the optimization, or will be manually assigned to the apron. Alternatively, if we assume that cancellation decisions can be made at this stage, an unassigned flight is equivalent to a cancelled flight. Of course, the decision maker may always relax some constraints, or allow for additional holding time, to generate a feasible schedule. In this study, we examine the problem from a pure modeling perspective to demonstrate how the mathematical model can be adapted to account for flight cancellations. Let C_i^{FF} be the cost of failing to assign flight i. In this case, we add term 3.12 to the objective function:

$$\sum_{i \in F} C_i^{FF} \left(1 - \sum_{j \in G_i} \sum_{k \in W_i} X_{ijk} \right) \tag{3.12}$$

which expresses the total cost of unassigned flights. We also relax flight constraint (3.5):

$$\sum_{j \in G} \sum_{k \in W_i} X_{ijk} \leq 1, \ i \in F \tag{3.13}$$

We should also take into account that, if the unassigned flight participates in connections, these connections will fail. Therefore, we define an additional binary variable $M_{ii'}^c$, for every connection $(i, i') \in T$, which is equal to 1 if the connection is missed because of cancellation of either flight i or i', and 0 otherwise. This is expressed by using the additional constraints (3.14)-(3.15):

$$M^c_{ii'} \geq 1 - \sum_{j \in G_i} \sum_{k \in W_i} X_{ijk}, \ (i,i') \in T \tag{3.14}$$

$$M^c_{ii'} \geq 1 - \sum_{j' \in G_{i'}} \sum_{k' \in W_{i'}} X_{i'j'k'}, \ (i,i') \in T \tag{3.15}$$

We then modify term (3.4) to include missed connections due to flight cancellations:

$$\sum_{(i,i') \in T} C^{CF}_{ii'} \left(M^c_{ii'} + \sum_{(j,j',k,k') \in Q^F_{ii'}} Z_{ijki'j'k'} \right) \tag{3.16}$$

3.3.2.2 Apron Capacity

We can extend our model to account for apron occupancy by adding constraint 3.17:

$$\sum_{i \in F^O_k \cap F^G_j} \sum_{k \in H_{is}} X_{ig^A k} + C^A_s \leq C^A, \ s \in W \tag{3.17}$$

which prevents the number of aircraft occupying the apron from exceeding the available apron static capacity at any given time.

3.3.2.3 Shadow Constraints

A shadow constraint is added to prevent aircraft with large wing span from occupying adjacent gates at the same time and can be mathematically expressed as follows:

$$\sum_{k \in H_{is}} X_{ijk} + \sum_{k' \in H_{i's}} X_{i'j'k'} \leq 1,\ i \in F^L,\ i' \in F^L, j \in G_i, j' \in G_j^N \cap G_{i'}, s \in W \quad (3.18)$$

3.3.2.4 Buffer Constraints

Buffer time constraints impose a minimum time limit on the required idle time of a gate between two successive flights assignments to the gate. Buffer constraints are generally used in planned assignment with the purpose to absorb flight delays. However, for reasons of completeness, we will incorporate them in the formulation of the gate reassignment problem as well. Let t_{idle} be the minimum required idle time for every gate, and dur_i the occupancy duration of flight $i \in F$, i.e., the elapsed time from the moment the flight arrives at the gate to the moment it leaves the gate. Mathematically, buffer constraints can be expressed as

$$\sum_{i \in F_j^G} X_{ijk} + \sum_{i' \in F_j^G} X_{i'jk'} \leq 1, j \in G, k, \in W, k' \in W : k' - k - dur_i < t_{idle} \quad (3.19)$$

Alternatively, we can group flights according to their gate occupancy duration dur_i and obtain a set $D = \{D_1, D_2, \dots D_n\}$ containing the different values of occupancy duration of the flights under consideration. Then we can write

$$\sum_{\substack{i \in F: \\ dur_i = D_m}} X_{ijk} + \sum_{i' \in F} \sum_{\substack{t \in [k+D_m, k+D_m+t_{idle}) \\ \cap W_{i'}}} X_{i'jt} \leq 1, j \in G, k \in W, D_m \in D \quad (3.20)$$

3.3.2.5 Flights Served by the Same Aircraft

In many cases, an aircraft serves an arriving and a subsequent departing flight. Upon landing, the aircraft is assigned to a gate so that the passengers boarding the arriving flight can disembark. To avoid towing operations, especially when the time between the arrival and the departure of the aircraft is relatively short, it is usually required that the departing flight will occupy the same gate as the arriving flight. However, when schedule disruptions happen, the decision maker has the option to assign the two flights to different gates. This is also the case for long layovers between the two flights, as in the case when the aircraft is scheduled for maintenance procedures between the two flights. In every case, we have to take into account that a pair of flights which are served by the same aircraft are not independent. Therefore, we add a constraint which prevents the temporal overlap of flight activities corresponding to the same aircraft. Let F^{PS} be the set of all flight pairs operated by the same aircraft. For every pair of flights $i \in F^A$, $i' \in F^D$, such that $(i, i') \in F^{PS}$, we create a set of time window pairs $W^{PS}(i, i')$, such that every member $(k, k') \in W^{PS}(i, i')$ indicates that flights i and i' cannot both be assigned to time windows k and k'. Mathematically, this can be written as follows:

$$\sum_{j \in G_i} X_{ijk} + \sum_{j' \in G_{i'}} X_{i'j'k'} \leq 1, (i, i') \in F^{PS}, (k, k') \in W^{PS}(i, i'). \tag{3.21}$$

An equivalent, but stronger formulation of constraint 3.21 can be written as follows:

$$\sum_{j \in G_i} X_{ijk} + \sum_{j' \in G_{i'}} \sum_{\substack{k' \in W_{i'} \\ (k,k') \in W^{PS}(i,i')}} X_{i'j'k'} \leq 1, (i,i') \in F^{PS}, k \in W_i \tag{3.22}$$

Because of the gate constraint 3.6, equations 3.22 can be furthered strengthened as:

$$\sum_{j \in G_i} X_{ijk} + \sum_{j' \in G_{i'}} \sum_{\substack{k' \in W_{i'} \\ (k,k') \in W^{PS}(i,i')}} X_{i'j'k'} + \sum_{j \in G_i} \sum_{\substack{w \in W_i: \\ w > k}} X_{ijk} \leq 1, (i,i') \in F^{PS}, k \in W_i \tag{3.23}$$

3.3.2.6 Arriving and Departing Passengers

Some measures of effectiveness, such as the total walking distance, are generally calculated for transfer passengers. However, the decision maker may also choose to minimize the walking distance for passengers who start or end their journey at the airport. In this case, the distance between the gate and the entrance/exit of the airport is used as input. Let N_{0i} (N_{i0}) the number of passengers who start (end) their trip at the airport, and d_{0j} (d_{j0}) the distance between the gate and the entrance (exit) of the airport. The respective flight assignment cost component is a a type (a) MOE (Chapter 2, section 2.3.1.4) and can be calculated as:

$$C^{FS}_{ijk} = N_{0i} d_{0j} \sum_{m \in W_i} X_{ijm}, i \in F, j \in G_i, k \in W_i \tag{3.24}$$

for departing passengers, and as

$$C_{ijk}^{FS} = N_{i0} d_{j0} \sum_{m \in W_i} X_{ijm}, i \in F, j \in G_i, k \in W_i \tag{3.25}$$

for arriving passengers.

3.4 Assumptions and Limitations

As shown in section 3.3, the model presented in this study can be generalized to accommodate a variety of objective functions and constraints, according to the priorities of the decision maker. To apply the model, the following assumptions are made:

First, the model is deterministic: It provides a one-step optimization approach, which uses the information about delays to generate the cost coefficients, as well as the problem sets, over which the decision variables are defined. Similarly, information regarding the required passenger connection time, which normally fluctuates during the day, should be fixed to an appropriate value at the time of the optimization.

Second, only decisions that are directly related to gate assignment can be made. The output of the model provides the decision maker with the appropriate course of action, which primarily consists of gate switching and flight holding. In practice, however, the eventual connection cost depends not only on gate assignment, but also on other airport operations, such as runway scheduling and apron bus scheduling.

3.5 Experimental Framework

In this section, we implement our model to a set of experimental cases of various sizes and properties.

3.5.1 Description of Experiments

Our experiments are divided into two main groups, each with a different purpose:

- Preliminary experiments on customized case studies (subsection 3.5.4), which demonstrate the importance of considering gate location and verify the ability of the model to replicate the gate reassignment procedure.

- Main experiments on a real-size case study (subsection 3.5.5), which demonstrate the relationship between the various MOEs used in the literature and actual costs, as well as the impact of considering the airport layout and connection time.

The experimental procedure was coded in Python 3, while Gurobi© solver was used for the optimization.

3.5.2 Cost Functions

Overall, we can identify four sources of costs: Delay costs, gate change costs, missed connection costs, and cancellation costs. Every objective function of the gate reassignment problem can be captured by one of the expressions 3.2 - 3.4 and 3.12, provided that the cost coefficients are properly defined and calculated. To determine the ability of each MOE to represent the actual cost of schedule disruptions, we need to express all of their aspects in monetary terms. For our experiments, we rely on official guidelines and existing literature to generate a realistic cost function that captures all aspects of solution quality.

3.5.2.1 Delay Costs

As can be seen from the literature review (Chapter 2), many studies focus on minimizing either additional delay or temporal deviation from the original schedule. Although this approach seems reasonable, it has two drawbacks. First, it isolates the gate reassignment procedure from the rest of the recovery management framework, ignoring critical components like flight connections, or the availability of alternative departing flights to which misconnected passengers can be rebooked. These components contribute to the total cost both directly (passenger compensation) and indirectly (level of service, passenger goodwill loss). Second, it fails to capture the nonlinearity of delay cost. In practice, for example, assigning one unit of delay to each of ten flights is significantly less costly that assigning ten units of delay to a single flight.

Hansen and Zou (2013) present a piecewise linear relationship between delay duration and cost. Initially, the cost of delay per passenger can be attributed to airport services, rebookings, and other operating expenses, and is relatively low. However, as delay increases, additional expenses (such as handling surcharges or provision of snacks and meals) arise and the total cost increases. For even longer delays, passenger and crew accommodation and transportation are required, resulting in further cost increase. Santos et al. (2017) assumed that a passenger experiencing 1-2 hours of delay receives a compensation of $10, while for delays longer than 8 hours, it costs $250 the airline to provide hotel accommodation and lounge access to a passenger.

3.5.2.2 Gate Change Costs

Changing the gate of a flight is a source of inconvenience for passengers. Most studies (2.3.1.4) minimize the number of gate changes or the total space inconsistency. In this study, we adopt the assumptions of Zhang and Klabjan (2017), who break down gate reassignment cost into two components:

- A fixed component for all gate changes.
- An additional component for departing flights, which depends on the time between the reassignment decision and the departure of the flight. Zhang and Klabjan (2017) used a step function of time; in this study, we use a piecewise linear function (Figure 3.1).

3.5.2.3 Missed Connection Costs

The cost of a failed connection consists of the following components:

- The cost of passenger compensation: When a passenger misses a connecting flight, he is generally booked on the next available flight to his/her destination. The

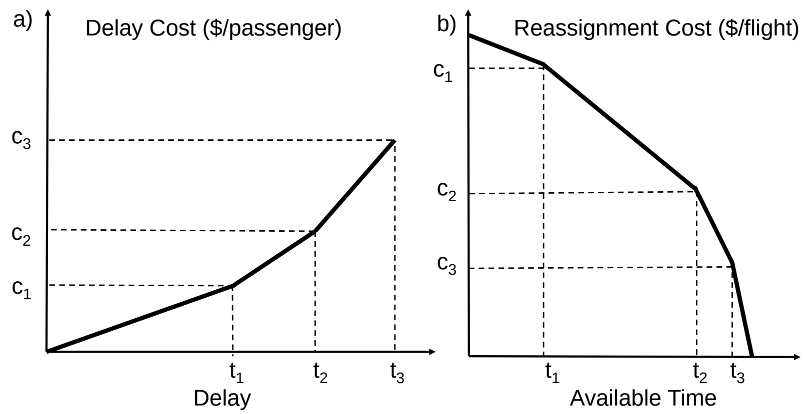

Figure 3.1: Qualitative form of a) the delay-cost function and b) the departing flight reassignment component function.

amount of compensation depends on the additional waiting time and may include hotel accommodation.

- The cost of a missed crew connection: The utilization of a reserve crew increases the total cost.

3.5.2.4 Cancellation Costs

Under extreme disruption circumstances, an airline might decide to cancel a flight to avoid delay propagation to subsequent flight legs. Cancellations result in direct costs for providing passengers with meals, accommodation, and possibly transportation to their destination, as well as opportunity cost of capital, loss of potential revenue, and additional cost when there is no available capacity for passenger rebooking (Hansen and Zou, 2013). In this study, we use the values of flight cancellation costs according to the aircraft type as provided by Eurocontrol (2013). Specifically, the recommended value is €3,700 ($4,453) for 50-seat, narrow-body aircraft, €17,300 ($20,818) for 120-seat, narrow-body aircraft, and €81,000 ($97,474) for 400-seat, wide-body aircraft. These costs include service recovery costs (passenger compensation), interline costs (rebooking revenue), and loss of future value (passenger opportunity costs), and are reduced by operational savings (fuel, crew, on-board supplies). Table 3.1 describes the calculation of cost coefficients for each of the MOEs used in the experiments.

The delay cost per passenger is a piecewise linear function of delay (section 4.2.1), ranging from $0 to $50 for 0-5 hours, from $50 to $200 for 5-7 hours, and from $200 to $250 for 7-8 hours of delay. We also used the same value to calculate passenger missed connection costs, assuming we know in advance the next available flight where passengers are redirected. Each missed connection was also penalized with $1000 for crew costs, similarly to the approach by Zhang and Klabjan (2017). For gate changes, we assumed a basic operational cost of $40 for all flights and an inconvenience cost for departing passengers which decreases from $380 to $260 for 0-1 hours notification in advance, from $260 to $120 for 1-2 hours, from $160 to $120 for 2-3 hours, and from $120 to $80 for 4-5 hours. Assignments to the apron gate were penalized with a $2000 operational cost. Flight cancellation costs were based on the values recommended by Eurocontrol (2013): €3,700 ($4,366) for 50-seat, narrow-body aircraft, €17,300 ($20,414) for 120-seat, narrow-body aircraft, and €81,000 ($95,580) for 400-seat, wide-body aircraft.

Table 3.1: Calculating Cost Coefficients for Different Objectives

Objective	Symbol	Cost Coefficient
Number of flights with gate changes	OS_F	$C_{ijk}^{FS} = \begin{cases} 1, j \neq g_i^B, \\ 0, j = g_i^B \end{cases}$
Number of passengers with gate changes	OS_P	$C_{ijk}^{FS} = \begin{cases} P_i, j \neq g_i^B, \\ 0, j = g_i^B \end{cases}$
Total distance change from the original schedule	OS_{PD}	$C_{ijk}^{FS} = \begin{cases} P_i L_{jj'}, j \neq g_i^B i, \\ 0, j = g_i^B \end{cases}$
Number of flights assigned to remote gates but originally assigned to contact gates	OR_F	$C_{ijk}^{FS} = \begin{cases} 1, j \in G^R and j \neq g_i^B, \\ 0, j =\notin G^R or j \neq g_i^B \end{cases}$
Number of passengers assigned to remote gates but originally assigned to contact gates	OR_P	$C_{ijk}^{FS} = \begin{cases} P_i, j \in G^R and j \neq g_i^B, \\ 0, j =\notin G^R or j \neq g_i^B \end{cases}$
Number of flights with time changes	OT_F	$C_{ijk}^{FS} = \begin{cases} 1, k \neq t_i^B, \\ 0, k = t_i^B \end{cases}$
Total time deviation (hours)	OT_T	$C_{ijk}^{FS} = \|k - t_i^B\|$
Total time deviation weighted by passengers (passengers * hours)	OT_{PT}	$C_{ijk}^{FS} = \|k - t_i^B\| P_i$
Number of canceled flights	OC_F	$C_i^{FF} = 1$
Number of passengers whose flight is canceled	OC_P	$C_i^{FF} = P_i$
Total walking distance of connecting passengers with a gate change	OW	$C_{ijki'j'k'}^{CS} = N_{ii'} L_{jj'}$
Number of failed connections	OM_F	$C_{ii'}^{CF} = 1$
Number of passengers who miss their connection	OM_P	$C_{ii'}^{CF} = N_{ii'}$
Monetary cost	$COST$	$C_{ijk}^{FS}, C_i^{FF}, C_{ijki'j'k'}^{CS}, C_{ii'}^{CF}$ calculated based on monetary values.
P_i: Passengers boarding flight i.		
$L_{jj'}$: Walking distance between gates j and j'.		
$N_{ii'}$: Number of connecting passengers from flight i to flight i'.		
g_i^B: Gate where flight i was originally assigned to.		
t_i^B: Time window when flight i becomes available after a disruption.		

3.5.3 Hierarchical Optimization

In experimental sets 2 (section 3.5.4.2), 3 (section 3.5.4.3), and A1 (section 3.5.5.5), we apply hierarchical optimization to optimize for multiple objective functions concurrently. Each of the n objective functions i is assigned a priority $p_i \in 1, \ldots, n$ such that $p_i \neq p_j \ \forall \ i \neq j$, with 1 being the highest priority and n the lowest. The objectives are then sorted based on decreasing priority. In each step i, we optimize for objective i such that all constraints hold, and we additionally impose the condition that none of the objectives j with a higher priority than i ($j < i$) can receive a worse value than the value it received in step j. Hierarchical optimization helps overcome the problem of selecting appropriate weighting coefficients when using a weighted sum approach.

3.5.4 Preliminary Experiments: Customized Case Studies

In this section, we demonstrate the modeling capabilities of the proposed formulation using the following experimental sets:

- Experimental Set 1: We show how the consideration of connection times affects the assessment of missed connections.
- Experimental Set 2: We adopt different measures of effectiveness in the objective function and show the impact of missed connections on total cost.
- Experimental Set 3: We test the models sensitivity to parameters controlled by the decision maker.

- Experimental Set 4: We modify the model to account for flight cancellations.

Each of the case studies used is defined by a unique combination of four components: Airport, flight schedule, disruption information, and parameters. In more detail:

Airport: Using the number of gates as input, we generate airport terminals of various shapes and layouts. The gates are divided into groups according to the type of flights they can accommodate. Except for set 1 (section 3.5.4.1), we assume a linear terminal with one airside, with gates evenly spaced every 50 meters.

Flight Schedule: The flight schedule contains information about the number and scheduled arrival/departure time of flights, individual flight properties (e.g., arriving or departing, type of aircraft, number of passengers, and gate in the planned assignment), and scheduled passenger connections.

Disruption: A disruption is defined as a change in the time when a flight becomes available for gate assignment, compared to the planned schedule. Delay patterns are generated randomly as follows: First, we distinguish between arriving and departing flights. Occasionally, an aircraft might serve both an arriving and a subsequent departing flight. In this case, since the delay of the departing flight depends on the arriving flight, we temporarily set the departing flight aside to handle it later. We determine the amount of delay of each of the arriving and remaining departing flights using a statistical distribution with given parameters. Then, we handle the departing flights which are served by the same aircraft as an arriving flight. If the arriving flight is not disrupted, the departing one is not disrupted either. However, if the arriving flight is disrupted, we calculate the available time between $t^{a,new}$ (when the arriving flight becomes available) and the planned departure time $t^{d,planned}$ of the departing flight ($Dt = t^{d,planned} - t^{a,new}$). We assume that a minimum time threshold Dt^{req} is required between the two flights. If $Dt \geq Dt^{req}$, the new availability time of the departing flight is the same as before ($t^{d,new} = t^{d,old}$). Otherwise, $t^{d,new} = t^{a,new} + Dt^{req}$. In this section, we assume a binomial distribution with disruption probability p and uniformly distributed delay duration between bounds (l_b, u_b).

Parameters: Parameters used for data generation and processing can be divided into the following categories:

- Flight features, which include the ratio of arriving to departing flights, as well as the percentage of flights per type, aircraft type and associated metrics (seat capacity, load factor, number of passenger per class). In our experiments, we use 9 aircraft types, while gate occupancy duration is equal to 30 minutes for aircraft with fewer than 150 seats, and 40 minutes for aircraft with at least 150 seats. Load factors ranged between 0.5 and 1 for each flight, with connecting passengers accounting for about 50% of total passengers.

- Tactical parameter values, which are associated with operator strategies, such as the maximum holding time (40 minutes) or the length of each time window k (10 minutes). These values are used for creating the sets for the definition of the model costs, which include all cost components described in section 3.5.2.

- External factors, which include all parameters associated with the distributions of delay generation and duration. In this section, we use $p = 0.4$, $l_b = 10$ and $u_b = 70$.

We construct each case study by defining the number of flights and gates and the planning horizon. We use random number generators to define the flight features, as well as to generate disruptions. For example, for a given ratio of arriving and departing flights, each flight is defined as arriving or departing with a fixed probability. When we develop multiple case studies with the same number of flights, gates, and hours, we specify a random number generator seed for each of them (i.e., the starting number of the random number sequence generated).

3.5.4.1 Experimental Set 1: Considering Connection Time

This set highlights the importance of considering all factors that affect connection time when assessing the success or failure of a transfer (section 3.2.2). We consider two airports, with the same number of gates but with different layouts. Airport #1 contains a linear terminal with one airside, while Airport #2 consists of 4 satellite concourses, uniformly located around a central building, from which each can be accessed using a moving walkway. We use a common flight schedule in both cases, including identical passenger connections and schedule disruptions. We perform gate assignment for two values of passenger processing time t^{PR}., 30 and 70 minutes, using a weighted sum of missed connections (multiplied by a weighting factor of 1000) and total deviation from the planned assignment as objective. To

Table 3.2: Connection Metrics for Two Airport Layouts

Airport	$\bar{t}^{PR}$ **(minutes)**	$\bar{d}^{U}$ **(minutes)**	$\bar{t}^{U}$ **(minutes)**	$\bar{t}^{W}$ **(minutes)**	**% Missed Connections**
1	30	745.9	40.7	38.9	0.0%
	70		80.7	79.7	7.8 %
2	30	1232.1	93.8	83.6	0.0%
	70		193.8	104.2	21.9%

quantify the connection distance and time, we use a set of descriptive measures, namely the static mean distance and time ($\bar{d}^{U}$ and $\bar{t}^{U}$, respectively) which only depend on the layout of the airport, as well as the passenger mean time $\bar{t}^{W}$, which is weighted by the number of passengers and therefore depends on gate allocation. We observe (Table 3.2) that high passenger processing time (70 minutes) results in 7.8% of connections being missed in Airport #1 and 21.9% in Airport #2. The static mean distance $\bar{d}^{U}$ of Airport #2 is 1.65 times greater than the respective of Airport #1. As a result, after the reassignment, passenger mean time $\bar{t}^{W}$ is about twice as large in Airport #2 than in Airport #1 for t^{PR}=30 minutes, and 1.3 times for t^{PR}=70 minutes. It is worth noting that all missed connections are counted as successful under the simple assumption.

3.5.4.2 Experimental Set 2: Using Different Objective Functions

In this set, we examine the monetary cost of the optimal solution for different objective functions. First, we optimize separately for three objectives: number of passengers with gate changes OS_F, number of delayed passengers OT_F, and total cost $COST$. Then, we calculate the resulting total cost of each solution. These experiments use 20 case studies, with 50-80 flights, 22-36 gates, and a horizon of 1-2 hours.

Our findings (Figure 3.2) demonstrate that deviation is not a representative measure of the actual cost. In fact, using temporal or spatial deviation as the objective can result in even 8 times higher total cost compared to using the total cost itself. (The dot and the line inside each box represent the mean and median, respectively. The same holds for Figures 3.3, 3.4, 3.5, and 3.6).

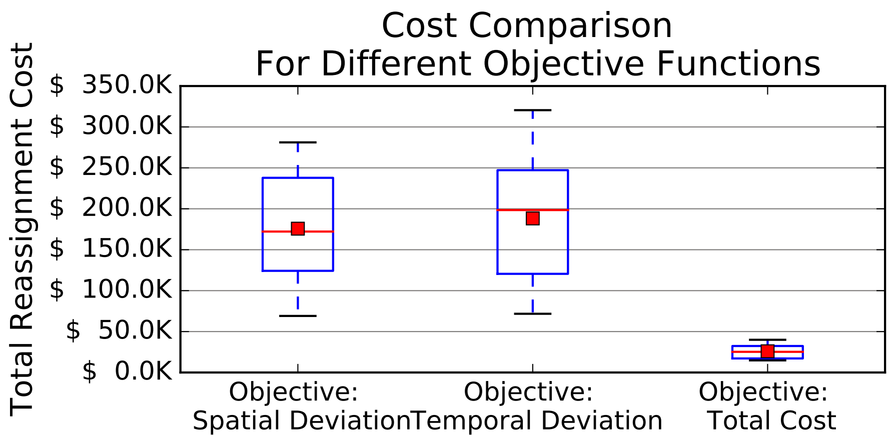

Figure 3.2: Total cost for different objective functions.

Afterwards, we perform hierarchical optimization for priority schemes 1-4 (Table 3.3), to show that the total cost can be reduced if we consider missed connections in the objective function. The priority schemes include three objectives: Spatial deviation (OS_{PD}), temporal deviation (OT_{PT}), and number of misconnected passengers (OM_P). Since we are interested in demonstrating the impact of missed connections, we do not examine all ($3! = 6$) priority permutations, but only these where missed connections are given most (1) or least (3) priority. Numbers 1-3 in Table 3.3 refer to the priority given to the MOE of the respective column in the Pri-

Table 3.3: Experimental Set 2.

	Spatial Deviation (OS_{PD})	**Temporal Deviation** (OT_{PT})	**Passengers with Missed Connections** (OM_P)
Priority Scheme 1	1	2	3
Priority Scheme 2	2	1	3
Priority Scheme 3	2	3	1
Priority Scheme 4	3	2	1

ority Scheme of the respective row. We generate 18 case studies, with 50-70 flights, 22-32 gates, and a planning horizon of 1-2 hours.

According to the results (Figure 3.3), the lowest cost is achieved for schemes 3 and 4, where we prioritize missed connections OM_P (Figure 3.3a). On the other hand, prioritizing spatial (OS_{PD}) and temporal (OT_{PT}) disruption yields poor results in terms of total costs (Figure 3.3a) and missed connections (Figure 3.3b): Schemes 1 and 2 perform on average 256% and 263% worse, respectively, than scheme 4.

3.5.4.3 Experimental Set 3: Testing the Decision Maker's Strategies

In this set, we examine cases with planned schedules of various minimum idle time requirements and holding time limits. We generate 13 case studies with 30-80 flights, 30-60 gates, and a planning horizon of 2 hours. We optimize hierarchically for missed passenger connections OM_P and total gate changes OS_P. Our results reasonably show that allowing sufficient idle time improves the performance of gate

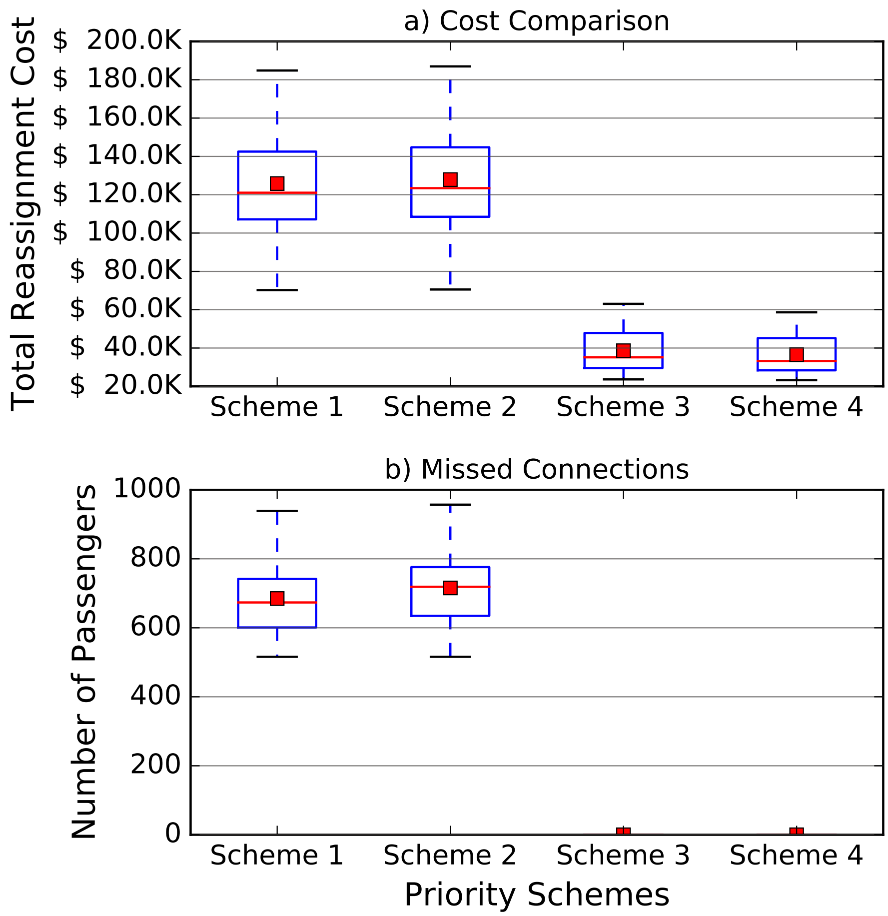

Figure 3.3: (a) Total cost ($) for different priority schemes and case studies. (b) Passengers who miss their connecting flight, for different priority schemes and case studies.

reassignment. We observe (Figure 3.4) a 32% reduction in the number of required gate changes for a buffer time of 20 minutes, compared to the case of 0 minutes.

Next, we test the impact of the maximum holding time. We generate 28 cases of 60-90 flights, 30-45 gates, and a horizon of 2 hours, and optimize hierarchically for misconnected passengers OM_P and gate changes OS_P. The results (Figure 3.5a, b) show that the primary objective OM_P is reduced as holding time increases, with this benefit inducing additional passenger delays (Figure 3.5c). In all cases, there

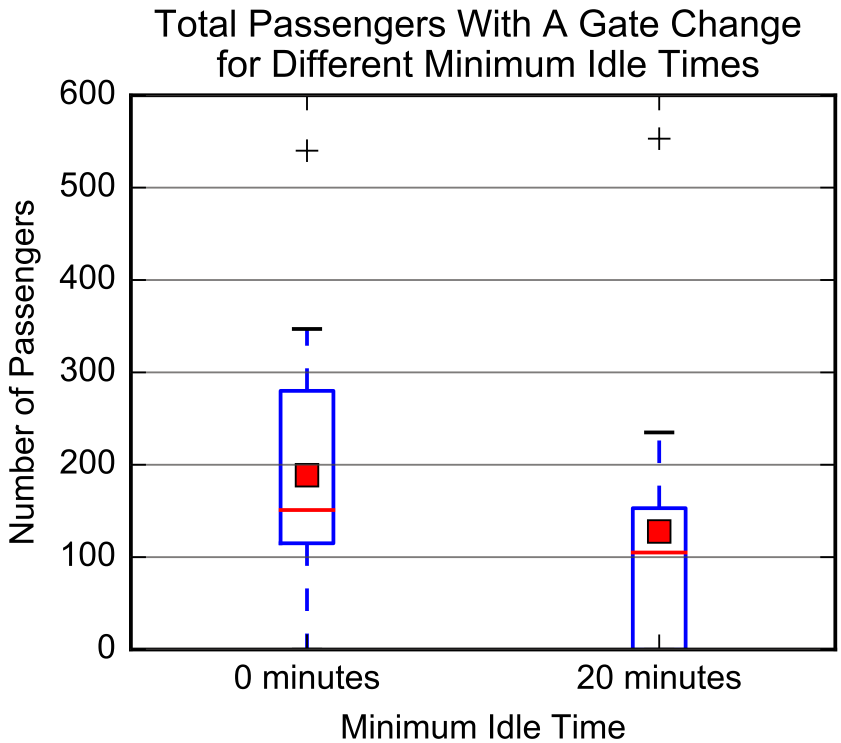

Figure 3.4: (a) Total cost ($) for different priority schemes and case studies. (b) Passengers who miss their connecting flight, for different priority schemes and case studies.

was no further improvement in the objective for maximum holding time greater than 30 minutes.

3.5.4.4 Experimental Set 4: Including Flight Cancellations

Let us consider an extreme disruption case, where due to an unexpected event (e.g., a temporary airport closure), flight departures/arrivals are collectively accumulated in specific time windows. We generate six case studies of practically the same size and different random generator seeds, each consisting of 20 gates serving 64 flights over a 2-hour horizon. In all cases, there exists no feasible solution to the reassignment problem since all possible allocation schemes violate gate constraint (3.6). However, if we assume that flight cancellations are allowed in this level of decision making, our model can always find a feasible solution. In fact, all six cases were solved to optimality, each with 2-4 cancelled flights in the optimal solution.

3.5.5 Main Experiments: Real Case Study

To test our model in a real-size problem, we approximate the layout of Athens International Airport Eleftherios Venizelos (IATA: ATH, ICAO: LGAV), which is the busiest airport in Greece and served more than 20 million passengers in 2016 (AIA, 2018).

3.5.5.1 Airport Data

The airport consists of a main and a satellite terminal. The main terminal is a linear building with one airside and two levels, i.e., upper and lower. The upper-level gates are contact gates where jet bridges are used for boarding the aircraft, while the lower-level gates are remote gates, involving the transportation of passengers to the aircraft via apron buses. Gates can further be classified based on whether they serve Schengen or Out-of-Schengen flights. To estimate the connection times, we simulate the layout of the airport by approximating the location of the gates, the entrance, the stairs connecting the two terminal levels, as well as the aircraft parking spots.

3.5.5.2 Flight Data

We use the information provided by the Airport (AIA, 2016) for a 2-hour period (12-2 pm) on a particular day of June 2016. A total of 74 flights (40 arriving and 34 departing) were scheduled to be served, including 25 domestic and 49 international flights; out of all international flights, 17 (35%) are Schengen flights, while the remaining 32 (65%) are non-Schengen flights (domestic flights are all Schengen flights as well).

3.5.5.3 Passenger Data

Due to the lack of publicly available data, the number of passengers boarding each flight was estimated as a function of a) the aircraft type and b) the region connected with Athens through each flight. Based on the origins and destinations of the flights, we assumed a total of nine different aircraft types, with capacity ranging from 46 seats (mainly for domestic connections with the Greek islands) to 300 seats (for long-haul flights to US and Canada or medium-haul flights to Qatar and the United Arab Emirates). Each flight was associated with a load factor according to its geographical area based on the classification provided by the Association of European Airlines (2016).

3.5.5.4 Disruptions

Disruptions are generated as described in section 3.5.4. In experimental sets A1 (section 3.5.5.5) and A2 (section 3.5.5.6), we use a binomial distribution (section

3.5.4). For set A3, we generate two disruption patterns that will be analyzed in section 3.5.5.7.

3.5.5.5 Experimental Set A1

This set examines the interaction among four MOEs that are generally used as objective functions in the existing literature, as well as their impact on total monetary cost:

- MOE 1: Additional temporal disruption (delay) due to the reassignment, OT_{PT}.
- MOE 2: Total spatial deviation, OS_{PD}.
- MOE 3: Number of passengers who are assigned to remote gates but were assigned to contact gates before, OR_P.
- MOE 4: Total walking distance for connecting passengers with a gate change in (at least one) flight, OW.

We generate five different experiment groups, each corresponding to a different disruption scenario. First, we use total cost minimization ($COST$) as the objective. Then, we perform hierarchical optimization, with missed connections OM_P as the objective with the highest priority, and each of the MOEs separately as the second objective. Finally, we optimize for each of the four MOEs as the only objective. Figure 3.6 shows the box-and-whisker plots for each of the hierarchical optimization schemes, while Table 3.4 shows the average value of each MOE for all experiments. Our main findings can be summarized as follows:

Table 3.4: Average MOE Values for Different Objective Functions (Hierarchical Optimization - Priority To Missed Connections.

		Measures of effectiveness			
		MOE1	**MOE2**	**MOE3**	**MOE4**
SECOND OBJECTIVE	MOE1	59670	6070749	1384	1348659
	MOE2	198812	2478	0	132274
	MOE3	192305	1859825	0	379947
	MOE4	206370	2050869	94	1161
(BEST AVERAGE - WORST AVERAGE) WORST AVERAGE		-71.09%	-99.96%	-100.00%	-99.91%

First, minimizing additional delay OT_{PT} (MOE 1) makes the other MOEs perform poorly. This incompatibility can be explained by the fact that OT_{PT} is a temporal MOE (see section 1.3.4), while the others are spatial MOEs. On the other hand, optimizing for total spatial deviation OS_{PD} (MOE 2) achieves a significantly better balance in the solution in terms of the remaining MOEs, with all of them assuming their second best value in this case.

Second, single-objective optimization results in significantly higher monetary cost (between 2 and 30 times, Appendix Table A1) than the optimal, while hierarchical optimization with priority to missed connections reduces the total cost significantly (up to 13.7 times, as shown in Appendix Table A2) compared to single-objective optimization. However, the obtained solution is still far from optimal in terms of monetary cost.

3.5.5.6 Experimental Set A2

We now investigate the tradeoffs between temporal and spatial disturbances when used as objectives. Existing literature demonstrates different approaches:

- Gu and Chung (1999) calculate a normalized weighted sum of a time anf a space component.

- Tang et al. (2010) assume that a gate change equals 30 minutes of temporal disturbance.

- Yan et al. (2011) and Wang et al. (2013) add time and space inconsistency without any weighting factor.

- Zhang and Klabjan (2017) assume a cost of \$20 /minute of delay and \$150 for reassigning a flight.

As can be seen, there is neither a common ground regarding the ideal way of adding the two types of disturbances nor a direct physical interpretation of the weighting coefficients. It was not until recently that monetary values were used in the gate reassignment problem (Zhang and Klabjan, 2017). In this set, we generate 6 different disruption scenarios and minimize the weighted sum of the total temporal disruption (passengers*minutes) and the total spatial disruption (passengers*meters), i.e.

$$OT_{PT} + w_{eq} * OS_{PD}. \tag{3.26}$$

Our experiments (Figure 3.7) indicate tradeoffs between the two measures for values of w_{eq} between at least 0.02 and at most 1.8. Any values outside these bounds result in practically the same solution.

3.5.5.7 Experimental Set A3

In this set, we compare the basic assumption of the proposed model, i.e., that the success of a connection depends on both the location of the gates and the available time between connecting flights, with the simple assumption that it only depends on the available time. We use:

a) Two disruption scenarios. In Scenario 1, delays follow a normal distribution $N(40, 10)$ for arriving flights and $N(0, 5)$ for departing flights (with the mean and standard deviation in minutes). Scenario 2 involves small disruptions, with delays following a Gamma distribution for arriving flights (as in Dorndorf et al. (2017)) $\Gamma(3, 1)$ (where 3 and 1 are the shape and scale parameters, respectively), and a Normal distribution $N(10, 10^2)$ (with a mean and standard deviation equal to 10 minutes) for departing flights.

b) Nine combinations of passenger processing time and connection speed between terminals: We tested three values of passenger processing time t^{PR} (30, 50, and 70 minutes) and three values of connection speed s^b (30, 70, and 140 meters/minute).

c) Six time thresholds. More precisely, each case was optimized seven times with the objective to minimize missed connections. For the first optimization, we use our own assumption, while for the remaining six, we use the simple assumption, each time with a different threshold. For meaningful and comparable results, the thresholds T were based on the static connection time, as a function of passenger processing time and transportation speed. Therefore, we used as threshold values T_V the mean ($\bar{t}^U$), maximum (t^U_{max}), and minimum (t^U_{min}), as well as the first (Q^{TU}_1), second (Q^{TU}_2) (median) and third (Q^{TU}_3) quantiles of static connection time.

For the sake of comparability, we only test passenger connections that are feasible for all models in the planned assignment. In the end, we calculate the percentage p_s of missed connections under the simple assumption, as well as the percentage p_p based on the proposed model. When appropriate, we also show the resulting values of passenger-weighted distance and time, i.e., mean ($\bar{d}^W$, $\bar{t}^W$), maximum (d^W_{max}, t^W_{max}), as well as the first (Q^{DW}_1, Q^{TW}_1), second (Q^{DW}_2, Q^{TW}_2) (median) and third (Q^{DW}_3, Q^{TW}_3) quantiles.

Passenger processing time is a critical component in the proposed model (Table 3.5) since an increase in processing time from 30 to 70 minutes increases the percentage of missed connections (up to 7 times, in scenario 1). Also, the impact of terminal connectivity is seen in scenario 1: When the connection speed increases, the percentage of missed connections decreases (from 13% to 12%).

The results also demonstrate a significant difference in the optimal solution between the simple assumption and our methodology (corresponding to rows with T = "-" in Tables 3.6 - 3.9). The difference between p_s and the respective p_p

ranges between -90% and 675%, and is especially significant for high thresholds (e.g. t^U_{max}), which overestimate the number of missed connections, and low thresholds (e.g. t^U_{min}), which underestimate it. We also observe (Tables 3.6, 3.7) that the proposed methodology generally yields smaller values for passenger-weighted connection metrics, compared to the simple approach. For example, the average weighted time $\bar{t}^W$ is minimum under our methodology in all cases, while quantiles Q_1^{TW}-Q_3^{TW} are always at most 1 minute higher than their minimum value. The two methods yield similar results in extreme cases, i.e., either when high connection times (low speed and high processing times) result in a large percentage of missed connections (Table 3.8), or when disruptions are negligible (Table 3.9) and consequently almost all connections are made. The only exception is the conservative approach of a high threshold t^U_{max} in the simple assumption, as in the case of $s^b = 30$ and $T = t^U_{max}$ in Table 3.9.

3.6 Summary, Conclusions, and Future Research

In this chapter, we have developed a new Binary Integer Model for flight-to-gate reassignment. It is the first multidimensional assignment model that assesses the success of passenger transfers as a function of gate location and the resulting required connection time. The model can be optimized for a variety of objectives since the objective function can be broken down into four components which capture all possible cost factors associated both with individual flights and with passenger connections. The formulation was extended to consider flight cancellations and

Table 3.5: Results of the proposed model.

Scenario	$\boldsymbol{t^{PR}}$	$\boldsymbol{s^b}$	$\boldsymbol{t^W}$	$\boldsymbol{t^W_{\max}}$	$\boldsymbol{Q_1^{TW}}$	$\boldsymbol{Q_2^{TW}}$	$\boldsymbol{Q_3^{TW}}$	$\boldsymbol{p_p}$
1	30	30	34.0	40.9	32.4	34.8	36.6	13%
		70	33.8	41.2	32.4	33.6	35.7	13%
		140	33.6	44.5	31.5	34.6	36.7	12%
	50	30	53.4	101.3	51.1	52.8	55.5	67%
		70	53.1	76.1	51.1	52.8	55.5	67%
		140	54.1	66.0	51.8	53.0	56.1	67%
	70	30	77.5	120.7	72.5	74.2	78.2	96%
		70	75.1	95.5	72.5	74.2	78.2	96%
		140	74.2	86.1	72.5	74.2	78.2	96%
2	30	30	49.7	82.7	32.6	36.0	78.2	0%
		70	41.3	56.4	32.6	35.6	52.4	0%
		140	37.5	46.9	32.6	35.6	42.9	0%
	50	30	54.1	97.6	51.8	53.3	55.0	0%
		70	57.8	75.7	51.9	54.6	58.7	0%
		140	56.6	66.9	53.1	56.4	62.5	0%
	70	30	74.9	118.0	72.1	73.2	75.1	0%
		70	78.9	96.8	72.3	75.7	91.7	0%
		140	78.4	87.0	72.3	76.4	84.9	0%

Table 3.6: Distance And Time Metrics for Scenario 1 - Low Processing Time t^{PR} (30 Minutes).

$\boldsymbol{s^b}$	$\boldsymbol{T}$	$\boldsymbol{T_V}$	$\boldsymbol{d^W}$	$\boldsymbol{t^W}$	$\boldsymbol{d^W_{\max}}$	$\boldsymbol{t^W_{\max}}$	$\boldsymbol{Q_1^{DW}}$	$\boldsymbol{Q_2^{DW}}$	$\boldsymbol{Q_3^{DW}}$	$\boldsymbol{Q_1^{TW}}$	$\boldsymbol{Q_2^{TW}}$	$\boldsymbol{Q_3^{TW}}$	$\boldsymbol{p_s}$	$\boldsymbol{p_p}$
	-	-	280	34	764	41	166	336	463	32	35	37	94%	13%
	$t^U_{\min}$	30	451	41	1706	82	169	388	567	32	36	38	3%	30%
	$t^U_{\max}$	84	366	38	1666	81	101	234	502	31	33	37	100%	66%
30	t^U	48	335	37	1512	79	129	270	447	32	34	36	49%	29%
	Q_1^{TU}	32	433	40	1616	81	134	343	567	32	35	39	19%	25%
	Q_2^{TU}	36	433	40	1616	81	134	343	567	32	35	39	19%	25%
	Q_3^{TU}	78	406	40	1678	81	174	293	567	32	34	38	99%	67%
	-	-	265	34	787	41	165	254	402	32	34	36	93%	13%
	$t^U_{\min}$	30	451	37	1706	57	169	388	567	32	36	38	3%	25%
	$t^U_{\max}$	58	418	37	1678	57	167	419	611	32	36	39	75%	37%
70	t^U	40	335	35	1512	54	129	270	447	32	34	36	49%	27%
	Q_1^{TU}	32	433	37	1616	55	134	343	567	32	35	39	19%	23%
	Q_2^{TU}	36	433	37	1616	55	134	343	567	32	35	39	19%	23%
	Q_3^{TU}	53	418	37	1678	57	167	419	611	32	36	39	75%	37%
	-	-	283	34	1527	45	105	323	467	32	35	37	93%	12%
	$t^U_{\min}$	30	451	36	1706	47	169	388	567	32	36	38	3%	22%
	$t^U_{\max}$	49	335	34	1512	44	129	270	447	32	34	36	49%	26%
140	t^U	37	433	35	1616	46	134	343	567	32	35	39	19%	22%
	Q_1^{TU}	32	433	35	1616	46	134	343	567	32	35	39	19%	22%
	Q_2^{TU}	36	433	35	1616	46	134	343	567	32	35	39	19%	22%
	Q_3^{TU}	43	335	34	1512	44	129	270	447	32	34	36	49%	26%

Table 3.7: Distance And Time Metrics for Scenario 1 - Medium Processing Time t^{PR} (50 Minutes).

s^b	T	T_V	d^W	t^W	$d^W_{\max}$	$t^W_{\max}$	Q_1^{DW}	Q_2^{DW}	Q_3^{DW}	Q_1^{TW}	Q_2^{TW}	Q_3^{TW}	p_s	p_p
	-	-	213	53	1054	101	80	195	387	51	53	56	93%	67%
	$t^U_{\min}$	50	335	57	1512	99	129	270	447	52	54	56	49%	77%
	$t^U_{\max}$	104	366	58	1666	101	101	234	502	51	53	57	100%	97%
30	t^U	68	304	56	1447	102	161	239	417	52	53	56	93%	80%
	Q_1^{TU}	52	418	60	1678	102	167	419	611	52	56	59	75%	81%
	Q_2^TU	56	418	60	1678	102	167	419	611	52	56	59	75%	81%
	Q_3^{TU}	98	366	58	1666	101	101	234	502	51	53	57	100%	97%
	-	-	213	53	1054	76	80	195	387	51	53	56	93%	67%
	$t^U_{\min}$	50	335	55	1512	74	129	270	447	52	54	56	49%	77%
	$t^U_{\max}$	78	406	56	1678	76	174	293	567	52	54	58	99%	91%
70	t^U	60	304	55	1447	77	161	239	417	52	53	56	93%	80%
	Q_1^{TU}	52	418	57	1678	77	167	419	611	52	56	59	75%	81%
	Q_2^{TU}	56	418	57	1678	77	167	419	611	52	56	59	75%	81%
	Q_3^{TU}	73	406	56	1678	76	174	293	567	52	54	58	99%	91%
	-	-	316	54	1561	66	129	212	426	52	53	56	93%	67%
	$t^U_{\min}$	50	335	54	1512	64	129	270	447	52	54	56	49%	77%
	$t^U_{\max}$	69	304	54	1447	67	161	239	417	52	53	56	93%	80%
140	t^U	57	418	56	1678	67	167	419	611	52	56	59	75%	80%
	Q_1^{TU}	52	418	56	1678	67	167	419	611	52	56	59	75%	80%
	Q_2^{TU}	56	418	56	1678	67	167	419	611	52	56	59	75%	80%
	Q_3^{TU}	63	304	54	1447	67	161	239	417	52	53	56	93%	80%

Table 3.8: Distance And Time Metrics for Scenario 1 - High Processing Time t^{PR} (70 Minutes).

$\boldsymbol{s^b}$	$\boldsymbol{T}$	$\boldsymbol{T_V}$	$\boldsymbol{d^W}$	$\boldsymbol{t^W}$	$\boldsymbol{d^W_{\max}}$	$\boldsymbol{t^W_{\max}}$	$\boldsymbol{Q_1^{DW}}$	$\boldsymbol{Q_2^{DW}}$	$\boldsymbol{Q_3^{DW}}$	$\boldsymbol{Q_1^{TW}}$	$\boldsymbol{Q_2^{TW}}$	$\boldsymbol{Q_3^{TW}}$	$\boldsymbol{p_s}$	$\boldsymbol{p_p}$
30	-	-	336	78	1537	121	175	297	577	73	74	78	93%	96%
	$t^U_{\min}$	70	304	76	1447	122	161	239	417	72	73	76	93%	99%
	$t^U_{\max}$	124	366	78	1666	121	101	234	502	71	73	77	100%	100%
	t^U	88	366	78	1666	121	101	234	502	71	73	77	100%	100%
	Q_1^{TU}	72	406	80	1678	121	174	293	567	72	74	78	99%	99%
	Q_2^TU	76	406	80	1678	121	174	293	567	72	74	78	99%	99%
	Q_3^{TU}	118	366	78	1666	121	101	234	502	71	73	77	100%	100%
70	-	-	336	75	1537	96	175	297	577	73	74	78	93%	96%
	$t^U_{\min}$	70	304	75	1447	97	161	239	417	72	73	76	93%	99%
	$t^U_{\max}$	98	366	76	1666	96	101	234	502	71	73	77	100%	100%
	t^U	80	366	76	1666	96	101	234	502	71	73	77	100%	100%
	Q_1^{TU}	72	406	76	1678	96	174	293	567	72	74	78	99%	99%
	Q_2^TU	76	406	76	1678	96	174	293	567	72	74	78	99%	99%
	Q_3^{TU}	93	366	76	1666	96	101	234	502	71	73	77	100%	100%
140	-	-	336	74	1537	86	175	297	577	73	74	78	93%	96%
	$t^U_{\min}$	70	304	74	1447	87	161	239	417	72	73	76	93%	99%
	$t^U_{\max}$	89	366	75	1666	86	101	234	502	71	73	77	100%	100%
	t^U	77	406	75	1678	87	174	293	567	72	74	78	99%	99%
	Q_1^{TU}	72	406	75	1678	87	174	293	567	72	74	78	99%	99%
	Q_3^{TU}	83	366	75	1666	86	101	234	502	71	73	77	100%	100%

Table 3.9: Percentage of Missed Connections - Scenario 2 (Low Disruption level).

s^b	T	$t_{PR} = 30$ min			$t_{PR} = 50$ min			$t_{PR} = 70$ min		
		T_V (min)	p_s	p_p	T_V (min)	p_s	p_p	T_V (min)	p_s	p_p
30	-	70	4%	0%	70	0%	0%	70	0%	0%
	$t^U_{\min}$	30	0%	3%	50	0%	10%	70	0%	14%
	$t^U_{\max}$	83.7	0%	0%	103.7	15%	1%	123.7	83%	10%
	$\bar{t}^U$	48.2	0%	3%	68.2	0%	7%	88.2	0%	18%
	Q_1^{TU}	32.1	0%	3%	52.1	0%	10%	72.1	0%	11%
	Q_2^{TU}	36.1	0%	3%	56.1	0%	10%	76.1	0%	11%
	Q_3^{TU}	78.1	0%	0%	98.1	11%	7%	118.1	39%	10%
70	-	70.0	1%	0%	70.0	0%	0%	70.0	0%	0%
	$t^U_{\min}$	30.0	0%	0%	50.0	0%	3%	70.0	0%	7%
	$t^U_{\max}$	58.5	0%	0%	78.5	0%	0%	98.5	11%	6%
	$\bar{t}^U$	40.4	0%	0%	60.4	0%	0%	80.4	0%	11%
	Q_1^{TU}	32.1	0%	0%	52.1	0%	3%	72.1	0%	3%
	Q_2^{TU}	36.1	0%	0%	56.1	0%	3%	76.1	0%	3%
	Q_3^{TU}	52.9	0%	0%	72.9	0%	0%	92.9	11%	6%
140	-	70.0	1%	0%	70.0	0%	0%	70.0	0%	0%
	$t^U_{\min}$	30	0%	0%	50.0	0%	1%	70.0	0%	4%
	$t^U_{\max}$	49.0	0%	0%	69.0	0%	0%	89.0	0%	0%
	$\bar{t}^U$	37.4	0%	0%	57.4	0%	1%	77.4	0%	0%
	Q_1^{TU}	32.1	0%	0%	52.1	0%	1%	72.1	0%	0%
	Q_2^{TU}	36.1	0%	0%	56.1	0%	1%	76.1	0%	0%
	Q_3^{TU}	43.4	0%	0%	63.4	0%	0%	83.4	0%	0%

apron capacity. We also use our model to perform a detailed review and analysis of the objective functions used in existing studies on gate reassignment.

Our experiments are embedded in a framework which allows us to switch between different objectives by adjusting the cost coefficients of each of four types of objective function terms. First, we verify our model by testing its output in a preliminary set of customized case studies with different airports, schedules, objective functions, and decision parameters. The results indicate that the model produces reasonable results when its input changes. We then perform our main experiments, where we apply the model to a large-size case study based on a real airport. We first use a hierarchical optimization framework to examine the interaction between the measures of effectiveness used in the literature and then we compare the proposed assumption on connection time with the simple assumption of fixed thresholds. Our experiments demonstrate the following findings: First, omitting passenger connections from the model results in an extreme increase in the reassignment cost. Second, prioritizing missed connections is necessary in the absence of monetary values, although the total cost of the optimal solution is still far from the optimal cost yielded when using total monetary cost as an objective. Finally, there exist significant differences between the solutions yielded when using the simple and the proposed assumption, except for extreme cases, such as negligible disruptions or high delays within the airport.

We believe that our model has the potential to be of use to both researchers and practitioners. On the one hand, it can provide researchers with an insight of the underlying relationships between solution quality indices and can be used for guidance for the development of heuristic techniques to achieve low solution time, which is the key for real-time decision making. On the other hand, practitioners can take advantage of the model's versatility and adapt it according to their own objectives, priorities, and strategies. Our research will now focus on improving the current mathematical formulation to achieve higher computational performance (see Chapter 4), as well as on developing a mathematical programming heuristic approach based on the multidimensional assignment formulation (see Chapter 5).

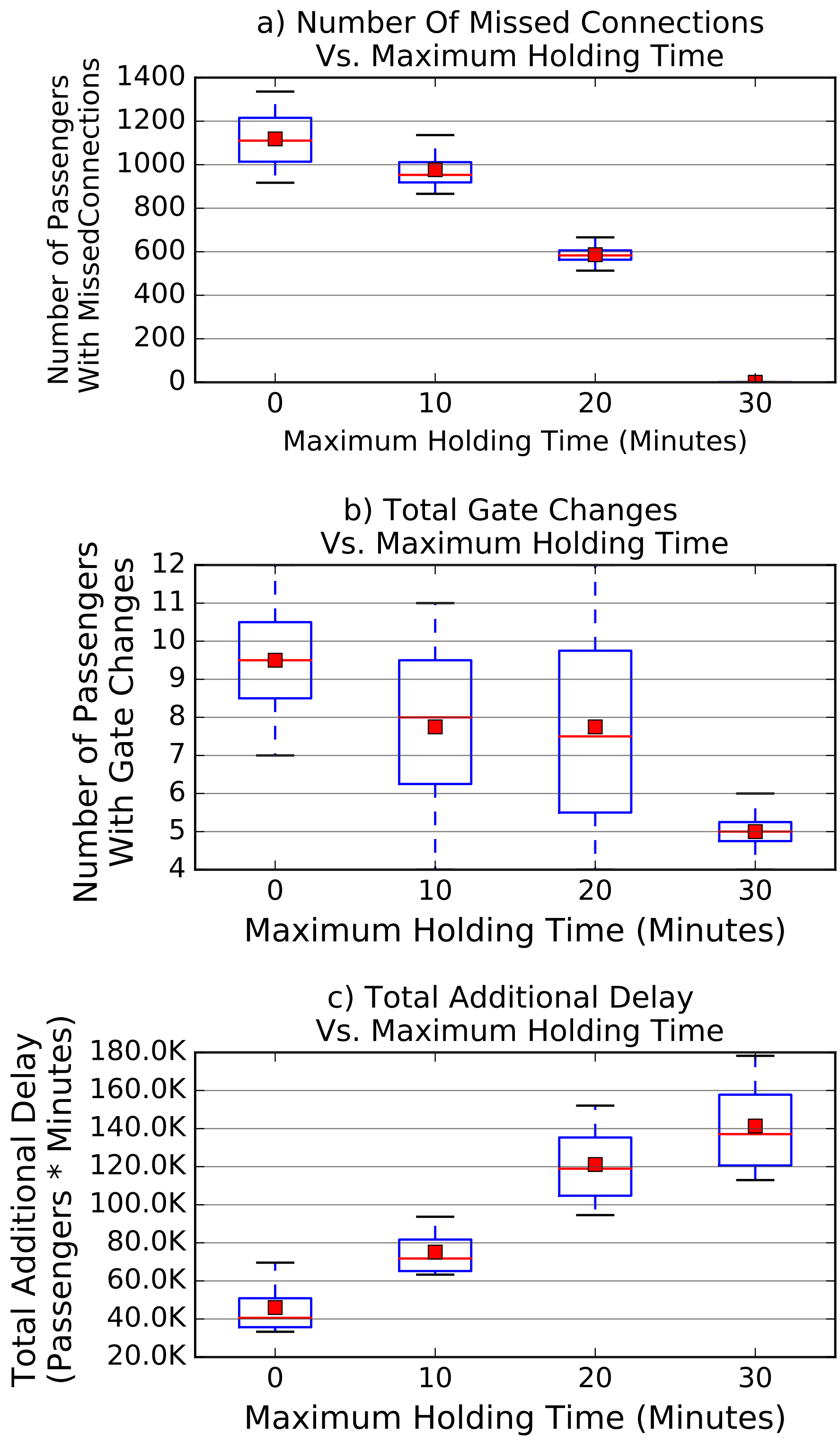

Figure 3.5: Misconnected passengers (a), gate changes (b) and total additional delay (c) Vs. Maximum holding time.

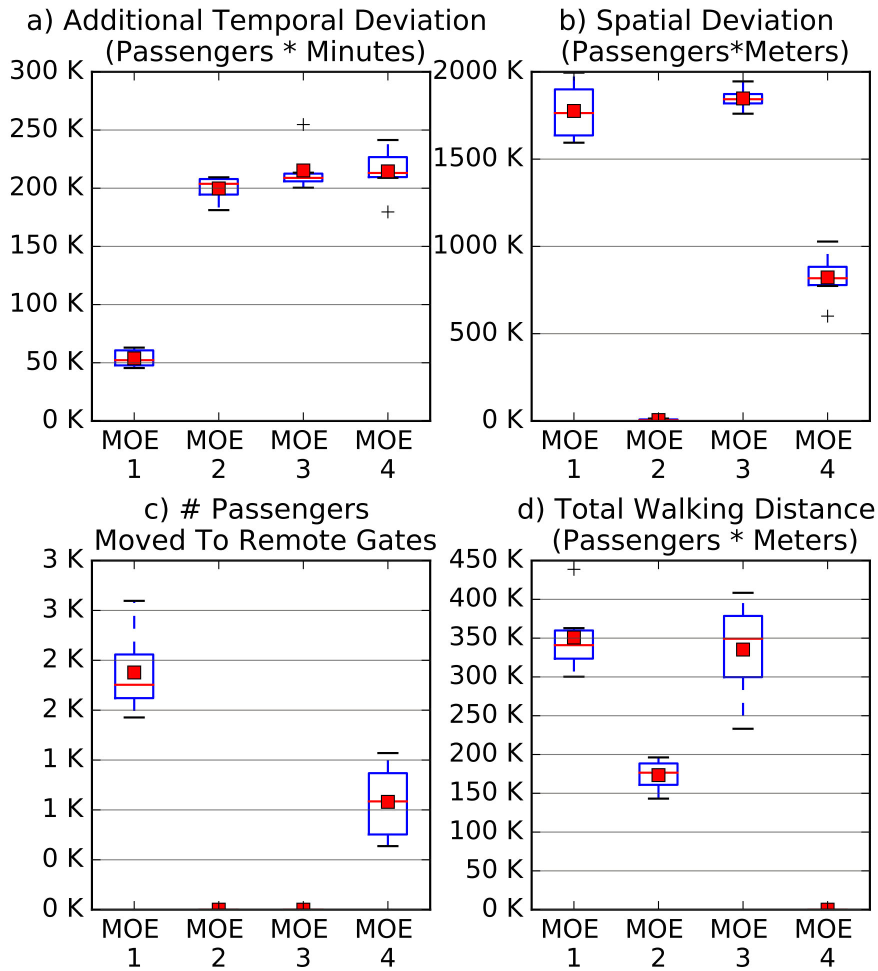

Figure 3.6: The values of four basic MOEs for different objective functions (hierarchical optimization).

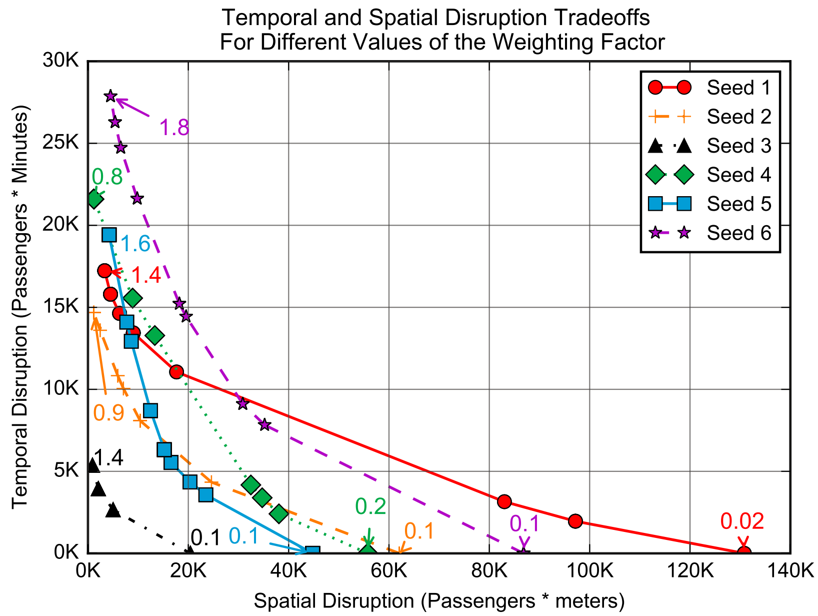

Figure 3.7: Temporal and spatial disruption for different values of the weighting factor. The data labels indicate the values of w_{eq} beyond which no change in the optimal solution is observed.

Chapter 4

Mathematical Models for Flight-to-Gate Reassignment with Passenger Flows: State-of-Art Comparative Analysis, Formulation Improvement, and a New Multidimensional Assignment Model

Chapter Overview

This chapter explores the mathematical programming formulation of flight-to gate reassignment with passenger connections. Motivated by the intractable increase in problem size when passenger flows are considered, combined with the need for low solution time, we perform three main tasks: (a) We compare and analyze both theoretically and experimentally the different types of state-of-art formulations, and identify the limitations of each one. (b) We improve the performance of existing models by modifying their formulations and introducing valid inequalities. (c) We improve the mathematical formulation proposed in Chapter 3, which accounts for passenger connections considering the layout of the airport and the available time between connecting flights. For the purpose of our experiments, we generate a number of cases of various sizes and schedule scenarios, as well as a set based on a real European airport. We then use our results to identify the most efficient formulations under different objective functions and problem assumptions. We expect that our work can provide researchers with a valuable tool for formulating efficient models that can be embedded in mathematical programming-based heuristics.

Keywords: *gate reassignment; passenger connections; mathematical programming; quadratic formulation; aggregating formulation*

4.1 Introduction

The concept of passenger flows plays a major role in the aircraft-to-gate reassignment problem, with different types of passengers generating different flow patterns: For example, passengers whose trip begins at the airport generally move from the entrance to the gate of the departing flight; passengers whose trip ends at the airport move from the gate of their arriving flight to the airport exit. Finally, connecting passengers move from the gate of the arriving flight to the gate of the departing flight of their transfer. In every case, as we showed in Chapter 3, flow patterns, as well as the resulting walking distance and time, vary according to layout of the airport (e.g. terminal location, existence of people movers and moving walkways) and the processing procedures (e.g. passport control) for passengers with different origins and destinations.

In Chapter 3, we used the required connection time to create a realistic model which determines the success of passenger transfers based on the relationship between the required and the available connection time. However, in order to linearize the quadratic model proposed in section 3.3, we added a set of constraints 3.7 - 3.9, which increase dramatically the size of the problem.

Motivated by the intractable size of mathematical models which consider passenger flows, in this chapter we study different types of state-of-art formulations, improve them by introducing valid inequalities, and propose a novel assignment model that considers the layout of the airport and the available time between connecting flights. Our experiments demonstrate that we achieve significant improvements in optimization time and indicate the conditions under which each formulation is preferable.

The work presented in this chapter has been published in Pternea and Haghani (2018).

The remaining of this chapter is structured as follows: In section 4.3, we propose a series of steps to strengthen the assignment formulation and gradually build a new formulation that significantly increases the speed of the branch-and-cut optimization procedure applied by an MIP solver. In section 4.4, we compare and analyze the alternative formulations by defining upper bounds on the number of variables and constraints, and identify the underlying assumptions and the limitations of each one. In section 4.5, we apply the different models in real-size experimental cases. Finally, in section 4.6 we draw the main conclusions of the study and identify paths for future research.

4.2 Literature Review: Mathematical Modeling of Passenger Flows

As explained in the literature review (Chapter 2), only few gate reassignment studies explore the problem from the perspective of transfer passengers, namely the ones by Maharjan and Matis (2011), Yu and Lau (2015), and Zhang and Klabjan (2017). However, the quadratic assignment model proposed by Maharjan and Matis (2011) does not include assignment time as a decision variable. We will therefore examine the remaining two studies (Yu and Lau, 2015; Zhang and Klabjan, 2017), which determine not only the gate that each flight is assigned to, but also the exact time period that the flight occupies the gate.

4.2.1 State-of-Art Overview

Yu and Lau (2015) minimize the total assignment cost and maximize the number of passengers who miss their connecting flight, while Zhang and Klabjan (2017) build an assignment network for each gate, and a passenger network for each connection.

In a typical network flow approach of GAP, one network is created for each gate, with nodes corresponding to time windows. A feasible flow corresponds to a sequence of flights occupying the gate throughout the planning horizon: Starting from the source, the order of the incident nodes of arcs with positive flow corresponds to the sequence of flights assigned to the gate, as well as their respective time windows. The problem is then solved as a minimum cost network flow problem.

However, the model requires extensions to capture passenger flows, as in Yu and Lau (2015) and Zhang and Klabjan (2017). The reader shall refer to these studies for further details. Yu and Lau (2015) minimize the total assignment cost and the number of passengers who miss their connecting flight. Their formulation relies on a single network for the whole problem with separate flow conservation constraints for each gate. To solve the problem, they iteratively divide passenger connections into a hard set and a soft set. Zhang and Klabjan (2017) build an assignment network for each gate, and a passenger network for each connection. Each network is associated with its own set of variables and constraints, while an additional set of constraints establishes the relationship between the two. The authors propose a diving and a rolling horizon heuristic. In the diving heuristic, flights are iteratively fixed in cliques, i.e. groups of neighboring gates, according to the solution of the linear relaxation, while in the rolling horizon heuristic the planning horizon is divided into smaller windows based on the number of connecting passengers. For simplicity, in the rest of this chapter, the formulations by Yu and Lau (2015) and Zhang and Klabjan (2017) will be referred to as `YL` and `ZK`, respectively.

4.2.2 Common Modeling Assumptions

In gate assignment, the decision maker (airport or airline) aims to determine the optimal allocation of flights to airport gates, subject to physical, operational and practical constraints, where optimality is defined according to the incurred monetary costs and/or the priorities of the decision maker.

Every flight can be assigned to a subset of the gates, based on criteria such as the airline, the size of the aircraft, and the type of the flight (e.g. domestic/international, or Schengen/Out-of-Schengen for international European airports). Certain studies additionally include the assignment time as a decision variable. In general, arrival and departure times are planned weeks or months in advance. However, using updated information about arrival/departure times and gate availability to modify the time when a flight begins to occupy its gate allows for more precise flight configuration. In this context, the decision maker can delay the arrival of a flight to its designated gate, or hold an outbound flight at the gate to prevent missed connections.

The planning horizon is divided into discrete time windows, each 5-10 minutes long; for every flight, the time window of planned arrival/departure is known in advance. Based on the airport/airline policies for maximum holding time, each flight is associated with the earliest and latest time windows at which it may start occupying a gate. The duration of gate occupancy depends on the type of the flight, the number of passengers, and the size of the aircraft, and is known in advance. As a result, every flight occupies a gate for a fixed number of consecutive time windows, the first of which is the assignment window, and is associated with two sets: a) The set of gates to which it can potentially be assigned, and b) The set of time windows where it can potentially be assigned.

4.2.3 Contributions of This Research

As can be seen, even when mathematical programming models are available, researchers apply heuristics to find good solutions to the gate reassignment problem. Both Yu and Lau (2015) and Zhang and Klabjan (2017) develop mathematical programming-based heuristics, in the sense that they use the model formulation to obtain solution bounds by solving restricted versions of the linear relaxation based on specific branching rules. From this perspective, the development of efficient heuristics requires an appropriate formulation that can handle the modeling assumptions and restrictions of the problem in hand fast and effectively. However, developing strong mathematical formulations is an aspect of the solution that has been neglected in the context of gate reassignment, especially when passenger connections are involved.

In addition, current research on modeling transfer passengers is occasionally based on unrealistic assumptions. Most importantly, with the exception of the study by Zhang and Klabjan (2017), the chance of passengers missing their connecting flight is generally treated as independent of the walking distance between the gates of the inbound and the outbound flight. In reality, a tight connection is more likely to be missed if the required walking time between the gates is long, given the layout and the available transportation modes, as well as the required mandatory passenger processing procedures.

In this chapter, we focus on the mathematical formulation of the gate reassignment problem with passenger flows. To tackle the issues explained above, the key contributions of this study are the following:

- We compare and analyze existing formulations of the gate reassignment problem with time as a decision variable and passenger flows in the objective function. To achieve this, we examine the mathematical models both from a theoretical (section 4.4) and a practical-experimental perspective (section 4.5). We also study the limitations and underlying assumptions of each formulation and provide guidelines for selecting a suitable formulation accordingly.

- We extend and improve existing formulations: We take advantage of the properties of the problem, such as the set partitioning constraints, to develop strong integer formulations that result in significant time savings when solved to optimality.

- We extend the time-indexed assignment formulation from Chapter 3, where the main idea is that the success of a passenger connection depends on both the physical location of the gates and the available time between the arrival of the inbound flight and the departure of the outbound flight. We then apply a number of transformations to accelerate the exact cut-and-branch solution procedure.

4.3 Improving the Assignment Formulation

In this section, we explore ways to strengthen the assignment formulation in order to accelerate the solution procedure.

4.3.1 Decision Variables

In the planned version of the gate assignment problem, the assignment formulation is the most common. As explained in Chapter 2, a binary assignment variable can be defined as the main decision variable in the literature (Maharjan and Matis, 2011; Wang et al., 2013) is binary X_{ij}, such that

$$X_{ij} = \begin{cases} 1, & \text{if flight } i \text{ is assigned to gate } j \\ 0, & \text{otherwise} \end{cases} \tag{4.1}$$

When allocation time is also a decision variable (Tang et al., 2010; Yan et al., 2011), a time-indexed formulation is adopted, where the main decision variable is X_{ijk} :

$$X_{ijk} = \begin{cases} 1, & \text{if flight } i \text{ is assigned to gate } j \text{ and time window } k \\ 0, & \text{otherwise} \end{cases} \tag{4.2}$$

In this study, we use a three-index formulation, since we assume that the decision maker can adjust the time at which a flight occupies a gate, within some predefined range. We will rely on two approaches exhibited in the literature for handling connections in an assignment model:

- A quadratic approach that is linearized with the introduction of appropriate inequalities.

- An aggregating approach that calculates the total connection cost for every arriving flight, based on a study by Yu et al. (2016).

4.3.2 Notation

In the following paragraphs, we will be using the following general notation for the basic problem elements:

Sets:

F^A:	Set of arriving flights.
F^D:	Set of departing flights.
F:	Set of flights, $F = F^A \cup F^D$.
G:	Set of gates.
T:	Set of passenger connections (i, i') ($i \in F^A$ and $i' \in F^D$).
W:	Set of time windows.
$G_i \subset G$:	Set of gates that are compatible with flight i.
$W_i \subset W$:	Set of time windows that are compatible with flight i.
H_{is}:	Set of time windows such that, if flight i is assigned to them, it occupies its gate at time window s.
$Q^A_{ii'}$:	Set of allowed combinations (j, j', k, k') for connection (i, i'), i.e. combinations that result in connecting passengers catching the outbound flight.
$Q^F_{ii'}$:	Set of forbidden combinations (j, j', k, k') for connection (i, i'), i.e. combinations that result in connecting passengers missing the outbound flight.

Costs:

C_{ijk}: Cost of assigning flight i to gate j at time window k.

$C^S_{ijki'j'k'}$: Successful connection cost due to the assignment of flight i to gate j and time window k, and of flight i' to gate j' and time window k', where $(ijki'j'k') \in Q^A_{ii'}$.

$C^F_{ii'}$: Cost of failed connection (i, i').

The costs vary according to the objectives of the problem, which are generally oriented towards restoring the original schedule and minimizing the impact of the disruptions on both passengers and operators. More details on the definition of cost have been provided in Chapter 3. In this context, C_{ijk} represents the delay costs, as well as the deviation from the original schedule. For example, it can be 0 if flight i was originally assigned to gate j and time k, and 1 otherwise. Alternatively, it can measure the distance between the planned and the reassigned gate, the additional imposed delay, or a combination of the two. Regarding connections, $C^S_{ijki'j'k'}$ expresses the cost of connection (i, i') with regards to the assignment of the flights. For example, it may be equal to the walking distance of passengers whose departing flight is assigned to a different gate. Finally, $C^F_{ii'}$ is the cost of a failed connection (i, i') and generally depends on the number of connecting passengers, the amount of compensation per passenger, and possible expenses for hotel accommodation.

4.3.3 The Quadratic Approach for Passenger Flows

The combination of individual connecting flight assignments is defined as the product of the respective binary assignment variables (4.3):

$$Z_{ijki'j'k'} = X_{ijk} \cdot X_{i'j'k'}, (i,i') \in T, j \in G_i, j' \in G_{i'}, k \in W_i, k' \in W_{i'} \tag{4.3}$$

which is equal to 1 if arriving flight i is assigned to gate j and departing flight i' is assigned to gate j'. The quadratic terms are linearized by adding inequalities (4.4)-(4.6):

$$Z_{ijki'j'k'} \leq X_{ijk}, (i,i') \in T, j \in G_i, j' \in G_{i'}, k \in W_i, k' \in W_{i'} \tag{4.4}$$

$$Z_{ijki'j'k'} \leq X_{i'j'k'}, (i,i') \in T, j \in G_i, j' \in G_{i'}, k \in W_i, k' \in W_{i'} \tag{4.5}$$

$$-Z_{ijki'j'k'} \geq X_{ijk} + X_{i'j'k'} - 1, (i,i') \in T, j \in G_i, j' \in G_{i'}, k \in W_i, k' \in W_{i'} \tag{4.6}$$

In this study, we assume that whether a connection will be made or missed depends on the location of the gates where the connecting flights are assigned, as well as the available time for passengers to walk between the gates. As a result, if inbound flight i is assigned to gate j and time window k, and outbound flight i' is assigned to gate j' and time window k', the connection will be made if and only if the available time $k' - k$ is at greater than or equal to the time required by passengers for moving between gates j and j', given the layout of the airport as well as the time required for passenger processing. We therefore classify each potential assignment combination of gates and time windows as "allowed", if it results in a successful connection, or "forbidden", if it results in a missed connection. Based on the measures of effectiveness that are used as objectives in the literature and in practice, we break down the objective function into three distinct components:

- Assignment cost $\sum_{i\in I}\sum_{j\in G_i}\sum_{k\in W_i} C_{ijk}X_{ijk}$. It captures all costs that depend on individual assignments (e.g., flight delays, gate changes, undesirable assignments to remote gates, etc.).

- Successful connection cost $\sum_{(i,i')\in T}\sum_{(j,j',k,k')\in Q^A_{ii'}} C^S_{ijki'j'k'}Z_{ijki'j'k'}$. It includes all connection costs, provided that the connection is successful (e.g., total connecting passenger walking distance).

- Missed connection cost $\sum_{(i,i')\in T} C^F_{ii'}\sum_{(j,j',k,k')\in Q^F_{ii'}} Z_{ijki'j'k'}$. It captures all costs resulting from failed connections (e.g., number of passengers who miss their connection, total compensation for connecting passengers, etc.).

Using the above components, and putting together the constraints that are normally used in the optimization, we can wrap up the basic three-index assignment-based formulation as follows:

Formulation Q-A (Quadratic Using All Connections)

Minimize:

$$\begin{aligned}\sum_{i\in I}\sum_{j\in G_i}\sum_{k\in W_i} C_{ijk}X_{ijk} \quad &+ \\ \sum_{(i,i')\in T}\sum_{(j,j',k,k')\in Q^A_{ii'}} C^S_{ijki'j'k'}Z_{ijki'j'k'} \quad &+ \\ \sum_{(i,i')\in T} C^F_{ii'}\sum_{(j,j',k,k')\in Q^F_{ii'}} Z_{ijki'j'k'} &\end{aligned} \tag{4.7}$$

Subject to:

Equations 8-11.

$$\sum_{j \in G_i} \sum_{k \in W_i} X_{ijk} = 1, \ i \in F \tag{4.8}$$

$$\sum_{i \in F} \sum_{k \in H_{is} \cap W_i} X_{ijk} \leq 1, \ j \in G_i, s \in W \tag{4.9}$$

$$X_{ijk} \in \{0,1\}, \ i \in F, j \in G_i, k \in W_i \tag{4.10}$$

$$Z_{ijki'j'k'} \in \{0,1\}, (i,i') \in T, j \in G_i, j' \in G_{i'}, k \in W_i, k' \in W_{i'} \tag{4.11}$$

Objective function (4.7) is the minimization of the sum of assignment and connection costs. Constraint (4.8) is the flight constraint, which enforces that each flight will be assigned to exactly one gate and time window. Constraint (4.9) is the gate constraint, stipulating that every gate can be occupied by at most one flight at any moment. Finally, constraints (4.10) and (4.11) enforce that all decision variables are binary.

4.3.3.1 First Modification: Defining Quadratic Variables over Allowed Combinations

Introducing passenger flow variables $Z_{ijki'j'k'}$ increases dramatically the number of variables and constraints of the problem. A detailed estimation of upper bounds on the number of variables and constraints follows in section 4.4. In this section, we investigate ways to create an equivalent formulation with a reduced number of variables and constraints so that the required computational time is reduced.

The main idea is based on the following implied valid equality:

$$\sum_{j \in G_i} \sum_{j' \in G_{i'}} \sum_{k \in W_i} \sum_{k' \in W_{i'}} Z_{ijki'j'k'} = 1, \ (i, i') \in T \tag{4.12}$$

which holds because of the set partitioning flight constraints (4.8) and the linearization constraints (4.4)-(4.6).

The following observation allows us to reduce the number of variables: While the cost of a successful connection depends on the location and the time of the individual flights, the cost of a missed connection is independent of walking distance or time and depends on flight properties, such as the number of connecting passengers, the availability of other outbound flights with the same destination for passenger rebooking, etc.

Therefore, if flights (i, i') are assigned to a forbidden combination (j, j', k, k') , we are not interested in the combination itself, but only in the fact that it belongs to $Q^F_{ii'}$. As a result, we only need to define Z variables over allowed combinations belonging to $Q^A_{ii'}$. We rename them to Z^A, where A stands for "allowed combinations", and obtain Formulation `Q-S`:

Formulation `Q-S` (Quadratic Using Successful Connections)

Minimize:

$$\begin{aligned} \textstyle\sum_{i \in F} \sum_{j \in G_i} \sum_{k \in W_i} C_{ijk} X_{ijk} \quad + \\ \textstyle\sum_{(i,i') \in T} \sum_{\substack{(j,j',k,k') \\ \in Q^A_{ii'}}} C^S_{ijki'j'k'} Z^A_{ijki'j'k'} \quad + \\ \textstyle\sum_{(i,i') \in T} C^F_{ii'} (1 - \sum_{(j,j',k,k') \in Q^A_{ii'}} Z^A_{ijki'j'k'}) \end{aligned} \tag{4.13}$$

Subject to:

Constraints (4.8) - (4.10)

$$Z^A_{ijki'j'k'} \leq X_{ijk},\ (i,i') \in T, (j,j',k,k') \in Q^A_{ii'} \tag{4.14}$$

$$Z^A_{ijki'j'k'} \leq X_{i'j'k'},\ (i,i') \in T, (j,j',k,k') \in Q^A_{ii'} \tag{4.15}$$

$$Z^A_{ijki'j'k'} \geq X_{ijk} + X_{i'j'k'} - 1, (i,i') \in T, (j,j',k,k') \in Q^A_{ii'} \tag{4.16}$$

$$Z^A_{ijki'j'k'} \in \{0,1\},\ (i,i') \in T,\ (j,j',k,k') \in Q^A_{ii'} \tag{4.17}$$

Observe that, since $\sum_{(i,i')\in T} C^F_{ii'}$ is a constant, the objective function (4.13) can be written equivalently as

Minimize:

$$\begin{aligned} \textstyle\sum_{i\in F}\sum_{j\in G_i}\sum_{k\in W_i} C_{ijk}X_{ijk} \quad + \\ \textstyle\sum_{(i,i')\in T}\sum_{(j,j',k,k')\in Q^A_{ii'}} (C^S_{ijki'j'k'} - C^F_{ii'})Z^A_{ijki'j'k'} \end{aligned} \tag{4.18}$$

Compared to Formulation `Q-A`, Formulation `Q-S` contains significantly fewer variables and constraints, but also introduces an obstacle to the optimization procedure: In all likelihood, the objective function coefficients of the quadratic variables $Z^A_{ijki'j'k'}$ are negative, since the cost $C^F_{ii'}$ of a missed connection is usually larger than the cost $C^S_{ijki'j'k'}$ of a successful connection. This increases computational time by preventing the solver from finding a feasible solution (i.e. an upper bound), as will be shown experimentally in section 4.5. To facilitate the solution procedure, we add a technically redundant, yet effective auxiliary constraint, i.e. a valid inequality that speeds up significantly the solution procedure:

$$\sum_{(j,j',k,k') \in Q^A_{ii'}} Z^A_{ijki'j'k'} \leq 1, \; (i,i') \in T \tag{4.19}$$

Detailed experiments to evaluate this approach are presented in section 4.5.2. We will distinguish between the formulations `Q-S` with and without the auxiliary constraint by referring to them as `Q-S2` and `Q-S1`, respectively.

4.3.3.2 Second Modification: Defining Quadratic Variables over Forbidden Combinations

In real time assignment, it is quite common that the cost of successful connections is not considered. In this case, we can define Z variables only over combinations $(j, j', k, k') \in Q^F_{ii'}$. Renaming the quadratic variables Z to Z^F, the problem now becomes:

Formulation `Q-F` (Quadratic Using Failed Connections)

Minimize:

$$\begin{aligned} \textstyle\sum_{i \in F} \sum_{j \in G_i} \sum_{k \in W_i} C_{ijk} X_{ijk} \quad + \\ \textstyle\sum_{(i,i') \in T} C^F_{ii'} \sum_{(j,j',k,k') \in Q^F_{ii'}} Z^F_{ijki'j'k'} \end{aligned} \tag{4.20}$$

Subject to:

Constraints (4.8) - (4.10)

$$Z^F_{ijki'j'k'} \leq X_{ijk},\ (i,i') \in T, (j,j',k,k') \in Q^F_{ii'} \tag{4.21}$$

$$Z^F_{ijki'j'k'} \leq X_{i'j'k'},\ (i,i') \in T, (j,j',k,k') \in Q^F_{ii'} \tag{4.22}$$

$$Z^F_{ijki'j'k'} \geq X_{ijk} + X_{i'j'k'} - 1, (i,i') \in T, (j,j',k,k') \in Q^F_{ii'} \tag{4.23}$$

$$Z^F_{ijki'j'k'} \in \{0,1\},\ (i,i') \in T,\ (j,j',k,k') \in Q^F_{ii'} \tag{4.24}$$

4.3.3.3 Third Modification: Defining Quadratic Variables over Aggregated Forbidden Combinations

We now test whether we can further improve Formulation `Q-F` by taking advantage of the set partitioning flight constraints (4.8), combined with the fact that cost coefficients $C^F_{ii'}$ only depend on the connection itself. We propose an equivalent, yet more concise formulation of the problem, by summing the assignment variables X_{ijk} of either the arriving or the departing flight corresponding to a connection. We will demonstrate this approach by summing the assignment variables for the departing flight of each pair (i,i') . Now the quadratic variables are defined as:

$$Z^G_{ii'jk} = X_{ijk} \cdot \sum_{(j,j',k,k') \in Q^F_{ii'}} X_{i'j'k'}, (i,i') \in T, j \in G_i, k \in W_i \tag{4.25}$$

and the problem is formulated as follows:

Formulation $\texttt{Q-FA}$ (Quadratic Using Aggregated Failed Connections)

Minimize:

$$\sum_{i \in F} \sum_{j \in G_i} \sum_{k \in W_i} C_{ijk} X_{ijk} + \sum_{(i,i') \in T} C^F_{ii'} \sum_{j \in G_i} \sum_{k \in W_i} Z^G_{ii'jk} \tag{4.26}$$

Subject to:

Constraints (4.8) - (4.10)

$$Z^G_{ii'jk} \leq X_{ijk}, \ (i,i') \in T, j \in G_i, k \in W_i \tag{4.27}$$

$$Z^G_{ii'jk} \leq \sum_{(j,j',k,k') \in Q^F_{ii'}} X_{i'j'k'}, (i,i') \in T, j \in G_i, k \in W_i \tag{4.28}$$

$$Z^G_{ii'jk} \geq X_{ijk} + \sum_{(j,j',k,k') \in Q^F_{ii'}} X_{i'j'k'} - 1 (i,i') \in T, j \in G_i, k \in W_i \tag{4.29}$$

$$Z^G_{ii'jk} \in \{0,1\}, \ (i,i') \in T, \ j \in G_i, \ k \in W_i \tag{4.30}$$

This approach allows us to remove the dimensions corresponding to indices j' and k' and therefore further reduce the size of the problem.

4.3.3.4 Summary of the Quadratic Approach Q

In summary, the Quadratic formulation Q can be further categorized based on the definition of the connection variables. In this context, formulation Q might use:

- All connections: Q-A, or
- Only the successful connections (Q-S), which might be formulated without (Q-S1) or with (Q-S2) an auxiliary constraint, or

- Only the failed connections (`Q-F`), or
- Only the failed connections, aggregated (`Q-FA`).

4.3.4 The Aggregating Approach for Passenger Connections

Yu et al. (2016) applied an alternative formulation for the planned Gate Assignment Problem, using the fact that the contribution of each arriving flight to the objective function is equal to the sum of the connection costs of all transfers where the flight participates. Adapting the notation to match the one used in this study, the passenger connection cost becomes $\sum_{i \in F^A} \sum_{j \in G_i} \xi_{ij}$.

For every decision variable ξ_{ij}, the authors add constraints (4.31)-(4.32):

$$\xi_{ij} \geq \sum_{i' \in F^D : (i,i') \in T} \sum_{j' \in G_{i'}} C^{\xi}_{ii'jj'} X_{i'j'} - UB_{ij}(1 - X_{ij}), i \in F^A, j \in G_i \tag{4.31}$$

$$\xi_{ij} \geq 0, \ i \in F^A, \ j \in G_i \tag{4.32}$$

where $C^{\xi}_{ii'jj'}$ is the cost of connection (i, i') when flights i, i' occupy gates j and j', respectively, assuming that connection cost is independent of assignment time, while UB_{ij} denotes an upper bound of the value of ξ_{ij}. In practice, constraints (4.31)-(4.32) can be interpreted as

$$\xi_{ijk} = \begin{cases} \sum_{i' \in F^D : (i,i') \in T} \sum_{j' \in G_{i'}} C^{\xi}_{ii'jj'} X_{i'j'}, & \text{if } X_{ij} = 1 \\ 0, & \text{if } X_{ij} = 0 \end{cases} \tag{4.33}$$

We will refer to all forms of the aggregating formulation as "Formulation A". The basic 2-index aggregating formulation is as follows:

Basic 2-Index Aggregating Formulation

Minimize:

$$\begin{aligned}\textstyle\sum_{i\in F}\sum_{j\in G_i} C^{\xi}_{ii'jj'}\sum_{k\in W_i} X_{ijk} \quad &+ \\ \textstyle\sum_{i\in F^A}\sum_{j\in G_i}\xi_{ij} \quad &+ \qquad\qquad\qquad (4.34) \\ \textstyle\sum_{(i,i')\in T} C^F_{ii'}\sum_{(j,j',k,k')\in Q^F_{ii'}} Z^F_{ijki'j'k'} &\end{aligned}$$

Subject to:

Constraints (4.8) - (4.10) and (4.21)-(4.24).

$$\xi_{ij} \geq \sum_{\substack{i'\in F^D:\\(i,i')\in T}}\sum_{j'\in G_{i'}} C^{\xi}_{ii'jj'}\sum_{k'\in W_{i'}} X_{i'j'k'} \; - UB_{ij}\left(1-\sum_{k\in W_i} X_{ijk}\right),\; i\in F^A,\; j\in G_i \tag{4.35}$$

$$\xi_{ij} \geq 0,\; i\in F^A,\; j\in G_i \tag{4.36}$$

where $\sum_{i\in F}$ represents the total assignment cost, $\sum_{i\in F^A}\sum_{j\in G_i}\xi_{ij}$ the total successful connection cost, and $\sum_{(i,i')\in T} C^F_{ii'}\sum_{(j,j',k,k')\in Q^F_{ii'}} Z^F_{ijki'j'k'}$ the total missed connection cost.

In order for this formulation to be valid, we assume that connection costs are independent of the assignment times k, k' of the connecting flights i, i' respectively. This is a reasonable assumption when connection cost depends only on flight (e.g. number of connecting passengers) and gate properties, as in the case of total walking distance. Based on the idea of Yu et al. (2016), we first propose a new three-dimensional assignment formulation that considers time as a decision variable and then we develop alternative versions of this formulation to determine the best one in terms of computational efficiency.

4.3.4.1 Two-Index Vs. Three-Index Aggregating Formulation

In some cases, connection cost can be time-dependent: For example, Maharjan and Matis (2012) proposed a penalty cost factor $C_{jj'}^{kk'}$ to quantify passenger discomfort due to long walking distances combined with tight connections, for planned gate assignment. To accommodate this case, we redefine the ξ variables to capture the temporal dimension of the decision. As a result, the cost of a successful connection becomes $\sum_{i\in F^A}\sum_{j\in G_i}\sum_{k\in W_i}\xi_{ijk}$. Building on the formulation proposed by Yu et al. (2016), we create a time-index formulation as follows:

Basic 3-Index Aggregating Formulation

Minimize:

$$\begin{aligned}\textstyle\sum_{i\in F}\sum_{j\in G_i}\sum_{k\in W_i} C_{ijk}X_{ijk} \quad &+ \\ \textstyle\sum_{i\in F^A}\sum_{j\in G_i}\sum_{k\in W_i}\xi_{ijk} \quad &+ \\ \textstyle\sum_{(i,i')\in T} C^F_{ii'}\sum_{(j,j',k,k')\in Q^F_{ii'}} Z^F_{ijki'j'k'}\end{aligned} \tag{4.37}$$

Subject to:

Constraints (4.8)-(4.10) and (4.21)-(4.24)

$$\xi_{ijk} \geq \sum_{\substack{i'\in F^D:\\(i,i')\in T}}\sum_{j'\in G_{i'}}\sum_{k'\in W_{i'}} C^S_{ijki'j'k'}X_{i'j'k'} - UB_{ijk}(1-X_{ijk}), i\in F^A, j\in G_i, k\in W_i \tag{4.38}$$

$$\xi_{ijk} \geq 0, \ i \in F^A, \ j \in G_i, \ k \in W_i \tag{4.39}$$

4.3.5 ξ For Successful Connections Vs. ξ For All Connections

As in Yu et al. (2016), ξ expresses the cost of successful connections only, while the cost of missed connections is calculated using the quadratic formulation. In order to extend the aggregating formulation to handle missed connections, we modify the upper bound UB^{ξ}_{ij} (2-index formulation) or UB^{ξ}_{ijk} (3-index formulation). For successful connections only, an upper bound can be estimated as follows: Assume that flight i will be assigned to gate j and time window k. Then, we assume that each flight i' connected with i is assigned to gate j' and time window k' such that the highest cost $C^S_{ijki'j'k'}$ is achieved:

$$UB_{ijk} = \sum_{\substack{i' \in F^D: \\ (i,i') \in T}} \max_{\substack{(j,j',k,k') \\ \in Q^A_{ii'}}} C^S_{ijki'j'k'} \tag{4.40}$$

On the other hand, if ξ_{ijk} are defined over successful and missed connections, the calculation of UB_{ijk} is modified accordingly:

$$UB_{ijk} = \sum_{\substack{i' \in F^D: \\ (i,i') \in T}} \max \left(C^F_{ii'}, \max_{\substack{(j,j',k,k') \\ \in Q^A_{ii'}}} C^S_{ijki'j'k'} \right) \tag{4.41}$$

For a 2-index formulation, $UB_{ij} = \max_k UB_{ijk}$.

Therefore, the Basic 2-index Aggregating Formulation is modified as follows:

Minimize:

$$\begin{gathered}\textstyle\sum_{i\in F}\sum_{j\in G_i}\sum_{k\in W_i} C_{ijk}X_{ijk} \quad + \\ \textstyle\sum_{i\in F^A:(i,i')\in T}\sum_{j\in G_i}\xi_{ij}\end{gathered} \tag{4.42}$$

Subject to:

Constraints (4.8)-(4.10)

$$\begin{aligned}\xi_{ij} \geq \quad & \textstyle\sum_{\substack{i'\in F^D:\\(i,i')\in T}}\sum_{j'\in G_{i'}} C^{\xi}_{ii'jj'}\sum_{k'\in W_{i'}} X_{i'j'k'} + \\ & \textstyle\sum_{\substack{i'\in F^D:\\(i,i')\in T}} C^F_{ii'}\sum_{\substack{(j',k'):\\(j,j',k,k')\in Q^A_{ii'}}} X_{i'j'k'} - \\ & UB_{ij}\left(1-\textstyle\sum_{k\in W_i} X_{ijk}\right),\ i\in F^A,\ j\in G_i\end{aligned} \tag{4.43}$$

$$\xi_{ij} \geq 0,\ i\in F^A,\ j\in G_i \tag{4.44}$$

The Basic 3-Index Aggregating Formulation is modified as follows:

Minimize:

$$\sum_{i\in F}\sum_{j\in G_i}\sum_{k\in W_i} C_{ijk}X_{ijk} + \sum_{\substack{i\in F^A:\\(i,i')\in T}}\sum_{j\in G_i}\sum_{k\in W_i}\xi_{ijk} \tag{4.45}$$

Subject to:

Constraints (4.8)-(4.10)

$$\begin{aligned}\xi_{ijk} \geq \quad & \sum_{\substack{i' \in F^D: \\ (i,i') \in T}} \sum_{j' \in G_{i'}} \sum_{k' \in W_{i'}} C^S_{ijki'j'k'} X_{i'j'k'} + \\ & \sum_{\substack{i' \in F^D: \\ (i,i') \in T}} C^F_{ii'} \sum_{\substack{(j',k'): \\ (j,j',k,k') \in Q^A_{ii'}}} X_{i'j'k'} - \\ & UB_{ijk} \left(1 - X_{ijk}\right), \; i \in F^A, \; j \in G_i, \; k \in W_i \end{aligned} \tag{4.46}$$

$$\xi_{ijk} \geq 0, \; i \in F^A, \; j \in G_i, \; k \in W_i \tag{4.47}$$

4.3.5.1 Defining ξ For Arriving Flights Vs. Defining ξ For Departing Flights

Yu et al. (2016) define one ξ_{ij} variable for each combination of arriving flight i and potential gate j. Equivalently, we can define the variables over the set of departing flights, instead of the arriving flights. In this case, the Basic 2-Index Aggregating Formulation is modified as follows:

Minimize:

$$\begin{aligned} & \sum_{i \in F} \sum_{j \in G_i} \sum_{k \in W_i} C_{ijk} X_{ijk} \quad + \\ & \sum_{i' \in F^D : (i,i') \in T} \sum_{j' \in G_{i'}} \xi_{i'j'} \quad + \\ & \sum_{(i,i') \in T} C^F_{ii'} \sum_{(j,j',k,k') \in Q^F_{ii'}} Z^F_{ijki'j'k'} \end{aligned} \tag{4.48}$$

Subject to:

Constraints (4.8)-(4.10) and (4.21)-(4.24)

$$\begin{aligned}\xi_{i'j'} \geq \sum_{\substack{i \in F^A: \\ (i,i') \in T}} \sum_{j \in G_i} C^{\xi}_{ii'jj'} \sum_{k \in W_i} X_{ijk} \, - \\ UB_{i'j'} \left(1 - \sum_{k \in W_i} X_{ijk}\right), \; i' \in F^D, j' \in G_{i'}\end{aligned} \tag{4.49}$$

$$\xi_{i'j'} \geq 0, \; i' \in F^D, \; j' \in G_{i'} \tag{4.50}$$

Similarly, the 3-index formulation is written as follows:

Minimize:

$$\begin{aligned}\textstyle\sum_{i \in F} \sum_{j \in G_i} \sum_{k \in W_i} C_{ijk} X_{ijk} \quad + \\ \textstyle\sum_{i' \in F^D} \sum_{j' \in G_{i'}} \sum_{k' \in W_{i'}} \xi_{i'j'k'}\end{aligned} \tag{4.51}$$

Subject to:

Constraints (4.8)-(4.10) and (4.21)-(4.24)

$$\begin{aligned}\xi_{i'j'k'} \geq \sum_{\substack{i \in F^A: \\ (i,i') \in T}} \sum_{j \in G_i} \sum_{k \in W_i} C^{\xi}_{ii'jj'kk'} X_{ijk} - \\ UB_{i'j'k'} \left(1 - X_{i'j'k'}\right), \; i' \in F^D, \; j' \in G_{i'}, \; k' \in W_{i'}\end{aligned} \tag{4.52}$$

$$\xi_{i'j'k'} \geq 0, \; i' \in F^D, \; j' \in G_{i'}, \; k' \in W_{i'} \tag{4.53}$$

To improve the efficiency of the Aggregating Formulation A, we use a minimum cardinality criterion: If $|F^A| \geq |F^D|$, we define ξ over departing flights, while if

$|F^A| < |F^D|$, we define ξ over arriving flights. Since every connection is included in the formulation exactly once, the two formulations are equivalent, but one of them should require fewer variables and constraints (assuming that the average number of potential gate and time windows does not vary significantly among the flights). In practice, $|F^A|$ can differ significantly from $|F^D|$ within the reassignment window in case of grouped coordinated arrivals and departures.

By expressing the total connection cost considering either the arriving or the departing flights, we make sure that the cost of each transfer is counted exactly once in the cost function. Selecting between $|F^A|$ and $|F^D|$ using the minimum cardinality criterion helps reduce the required number of continuous ξ variables and respective constraints (assuming that the average number of compatible gates and assignment windows does not vary significantly across flights). We can further reduce the required number of variables and constraints using a bipartite graph to represent transfers: The nodes on the left side correspond to arriving flights, while the nodes on the right side correspond to departing flights. An arc connecting an arriving to a departing flight represents a transfer. To consider all transfers, we essentially have to cover all arcs. An arc is covered if at least one of its adjacent nodes is considered.

For example, consider the case of Figure 4.1a, with four arriving and four departing flights ($|F^A| = |F^D| = 4$). Instead of using either F^A or F^D to define ξ variables, we can cover all transfers (Figure 4.1b) by assigning transfers A1-D1, A1-D2, A1-D3, and A1-D4 to flight A1, transfer A2-D2 to flight D2, and transfers A2-D3, A3-D3 and A4-D3 to flight D3, and therefore use only 3 flights to define ξ.

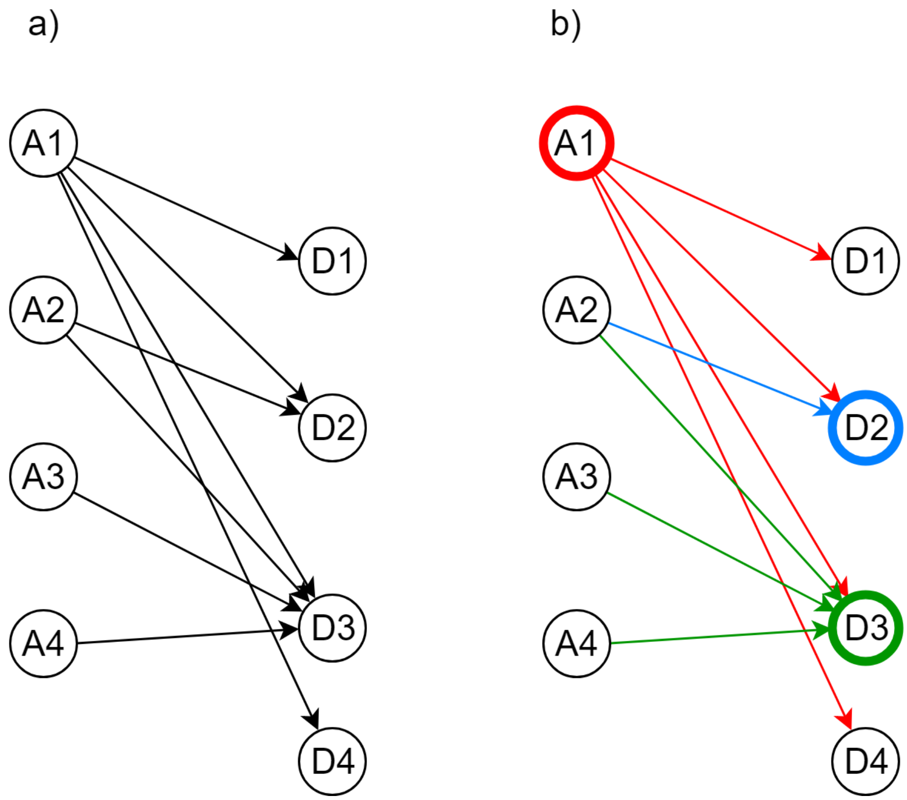

Figure 4.1: Representing transfers using a bipartite graph.

To minimize the number of nodes required, we can solve a minimum cover problem. Transfers corresponding to arcs that are covered by both their starting and their ending node, such as A1-D2 and A1-D3 in the example, can be randomly assigned to one of the two nodes. For the rest of this chapter, however, we will simply use the cardinality criterion.

4.3.5.2 Summary of the Aggregating Formulation $\mathtt{A}$

In summary, different versions of Formulation $\mathtt{A}$ can be created by modifying the definition of ξ variables, i.e.:

a) Whether ξ variables express the cost of successful (S) only, or of all (A) connections.

b) Whether between the sets of arriving and departing flights we choose the one that has the smallest (S) or the largest (L) cardinality.

c) Whether ξ variables are 2-index (ξ_{ij}) or 3-index (ξ_{ijk}).

The combinations of (a), (b), and (c), result in $2^3 = 8$ formulations, summarized in Table 4.1.

Table 4.1: Different Versions of Aggregating Formulation A

	All Vs. Successful Connections		**Smallest Vs. Largest Cardinality**		**2 - Index Vs. 3-Index**	
Form.	**All**	**Successful**	**Smallest**	**Largest**	**2-Index**	**3-Index**
`A-SS2`		✓	✓		✓	
`A-SS3`		✓	✓			✓
`A-SL2`		✓		✓	✓	
`A-SL3`		✓		✓		✓
`A-AS2`	✓		✓		✓	
`A-AS3`	✓		✓			✓
`A-AL2`	✓			✓	✓	
`A-AL3`	✓			✓		✓

4.4 Theoretical Analysis of Formulations

Before we implement the mathematical formulations in practice, we analyze them based on the size of the resulting problem, as well as on their underlying assumptions.

4.4.1 Number of Variables and Constraints

In this section, we perform a worst-case analysis of the alternative formulations based on the number of variables and constraints. We introduce the following additional notation:

C:	Set of gate cliques, as defined in Zhang and Klabjan (2017).
F :	Upper bound on the number of flights which are compatible with a gate.
$\bar{G}$:	Upper bound on the number of gates that are compatible with a flight.
$\bar{C}$:	Upper bound on the number of cliques that are compatible with a flight.
$\bar{W}$:	Upper bound on the number of potential time windows for a flight.
$F^{A,C}$:	Set of arriving flights participating in connections.
$F^{D,C}$:	Set of departing flights participating in connections.

Both `YL` and `ZK` formulations include shadow constraints, while `YL` also includes constraints for flights using the same aircraft. In this section, we present the number of variables and constraints for the basic formulation only. Tables 4.2 and 4.3 present the variables and constraints for the assignment formulation, while Tables 4.4 and 4.5 are the respective ones for network flow formulations.

A summary of the total number of variables and constraints for each formulation can be found in Table A3 of the Appendix.

4.4.2 Comparison of Assumptions - Limitations

The above formulations are generally equivalent in terms of the problem representation and optimal solution. However, under certain conditions, some of the

Table 4.2: Number of Variables for Assignment Formulations

Formulation	Assignment Variables	Connection Variables
`Q-A`	$\lvert F\rvert\bar{G}\bar{W}$	$\lvert T\rvert\bar{G}^2\bar{W}^2$
`Q-S`		
`Q-F`		
`Q-FA`		$\lvert T\rvert\bar{G}\bar{W}$
`A-SS2, A-SL2, A-AS2, A-AL2`		$\min(\lvert F^{A,C}\rvert, \lvert F^{D,C}\rvert)\bar{G}$
`A-SS3, A-SL3, A-AS3, A-AL3`		$\min(\lvert F^{A,C}\rvert, \lvert F^{D,C}\rvert)\bar{G}\bar{W}$

Table 4.3: Number of Constraints for Assignment Formulations

Formulation	Flight	Gate	Linearization	Variable Definition	Variable Type
`Q-A`	$\lvert F\rvert$	$\lvert T\rvert$	$3\lvert T\rvert\bar{G}^2\bar{W}^2$. `Q-S2`: Extra $\lvert T\rvert\bar{G}^2\bar{W}^2$	$\lvert F\rvert\bar{G}\bar{W}+$ $\lvert T\rvert\bar{G}^2\bar{W}^2$	$\lvert F\rvert\bar{G}\bar{W}+$ $\lvert T\rvert\bar{G}^2\bar{W}^2$
`Q-S`					
`Q-F`					
`Q-FA`			$3\lvert T\rvert\bar{G}\bar{W}$	$\lvert F\rvert\bar{G}\bar{W}+$ $\lvert T\rvert\bar{G}\bar{W}$	$\lvert F\rvert\bar{G}\bar{W}+$ $\lvert T\rvert\bar{G}\bar{W}$
`A-SS2, A-SL2, A-AS2, A-AL2`			0	$2\min(\lvert F^{A,C}\rvert, \lvert F^{D,C}\rvert)\bar{G}$	$\lvert F\rvert\bar{G}\bar{W}$
`A-SS3, A-SL3, A-AS3, A-AL3`			0	$\min(\lvert F^{A,C}\rvert, \lvert F^{D,C}\rvert)\bar{G}$	$\lvert F\rvert\bar{G}\bar{W}$

Table 4.4: Number of Variables for Network Flow Formulations

Formulation	Assignment Variables	Ground Variables	Flow Connection Variables
`YL`	$\lvert F\rvert\bar{G}\bar{W}$	$\lvert G\rvert(\lvert W\rvert+1)$	$\lvert T\rvert\bar{W}^2$
`ZK`			$\lvert T\rvert+\lvert T\rvert(2\bar{C}\bar{W}+\bar{C}^2\bar{W}^2+1)$

Table 4.5: Number of Constraints for Network Flow Formulations

Formulation	**Flight**	**Linearization**	**Flow Conservation**	**Connection Constraints**	**Variable Type**
YL	—F—	$3\|T\|\bar{W}^2$	$\|G\|(\|W\|+1)$	$\|T\|+2\|T\|WC$	$\|F\|\bar{W}\bar{G}+\|T\|\bar{W}^2$
ZK		-	$\|G\|(\|W\|+2)+\|T\|(2\bar{C}\bar{W}+2)$	$2\|T\|$	$\bar{F}\bar{W}+(W+1)+\|T\|$

formulations are not adequate and therefore cannot be applied. These conditions include:

- The way a successful or failed connection is determined.
- The type of objective functions that can be accommodated.
- The value of connection costs.
- The variation in flight duration.

4.4.3 Determining Success/Failure of Connections

All formulations presented assume that whether a connection will be made depends on the assignment of flights that comprise it. However, the YL formulation assumes for simplicity that the location of gates does not affect the success or failure of a connection. In other words, passengers will always catch the departing flight if its departure time is later than the arrival time of the arriving flight, regardless of the distance between the gates and the required connection time. On the other hand, both the proposed assignment approach and the ZK assume that the success of a connection depends also on the required connection time. While the latter approaches can be easily adapted to capture the simplifying assumption of YL, the reverse does not hold.

4.4.4 Objective Function Types

The objective function generally consists of three cost components, namely cost of flight assignment, missed connections, and successful connections. However, formulation YL does not associate the individual assignments with the incurred connection costs and therefore cannot be used with objectives like total passenger walking distance.

4.4.5 Value of Connection Costs

In formulation ZK, the relationship between gate and connection networks is established using the inequality

$$\begin{aligned} &\textstyle\sum_{e \in E_{(f,fd,k)}} x_e - \sum_{\omega \in E^p_{(\tau,f,fd,k)}} z_\omega \geq 0, \\ &\tau \in T, f \in F, fd \in FD_f, k \in K \end{aligned} \tag{4.54}$$

which stipulates that variable z_ω of a passenger arc ω whose starting node corresponds to clique c and time window k, and ending node corresponds to clique c' and time window k' of a connection $t = (i, i')$, cannot be positive unless the x_e variable corresponding to flight i, gate $j \in C$, and time window k, and the x_e variable corresponding to flight i', gate $j' \in C'$, and time window k', are both positive. The inverse relationship, however, is not explicitly enforced; in other words, the z_ω variable is allowed to remain 0 even when both x_e variables are equal to 1. Nevertheless, if z_ω was equal to 0, then flow conservation constraints, combined with constraint

$$z_{e_\tau}^{cycle} + y_\tau = 1, \ \tau \in T \tag{4.55}$$

would make y_τ equal to 1. This would contribute to the term of the objective function dealing with missed connections, i.e. $\sum_{\tau \in T} cancel_t y_\tau$. Under the reasonable assumption that the costs of missing a connection is always larger than the cost of the same connection if it succeeds, and given that the problem is a minimization problem, the optimization will not allow the value of y_τ to become 1 unless it is "forced to", i.e. when the respective $z_{e_\tau^{cycle}}$ is equal to 0. But, due to flow conservation constraints, this can only happen when there is zero flow in the network of the connection, i.e. when the connection fails.

However, it might occasionally be more beneficial to allow for a connection to be missed, as in the case where intentionally holding an outbound flight to facilitate passenger connections from inbound flights would result in delay propagation across the whole network. In that case, there exist combinations $(j, j', k, k') \in Q_{ii'}^A$ for a connection (i, i') which correspond to cost coefficients $C_{ijki'j'k'}^S$ that are greater than the respective $C_{ii'}^F$ for missing a connection. This case cannot be handled by Formulation `ZK`, since the model might result in an "expensive" successful connection having all its z_ω variable values equal to 0 and therefore falsely count the connection as missed.

4.4.6 Variation in Flight Duration

In most approaches, the duration of gate occupancy for each flight is fixed and known beforehand. Therefore, the starting time of a gate assignment determines uniquely the ending time. However, the duration of gate occupancy for a given flight can change in practice. This means that the same starting time for a flight can correspond to more than one potential ending times. In the time-index formulation, this case cannot be covered without further modification. On the contrary, network flow formulations can easily handle this case by defining suitable flight arcs.

4.5 Implementation

To compare the existing formulations and test the modified formulations we developed, we carry out a number of experiments for a variety of custom generated cases, as well as for cases based on the layout of Athens International Airport in Greece (AIA, 2016). The optimization is performed using Gurobi© solver, which applies a branch-and-cut procedure, assisted by a number of presolving and heuristic techniques. All experiments were ran on a single host equipped with a quad core Intel i7 2860QM processor (2.50 GHz) and 32GB of RAM, running Windows 10.

4.5.1 Design of Experiments

Since we are interested in the performance of the alternative formulations, we focus on the effect of the problem size, which depends on three factors: a) The number of flights, which is associated with the number of passengers, b) the size of the airport, i.e. the number of gates, and c) the planning horizon, i.e. the number of time windows. To ensure that the case studies produced are reasonable, we generate linear concourses with gates on one or both sides, and different scenarios for the hourly rate of inbound and outbound flights. Details on the size and properties of the case studies for each set can be found in the Appendix. All flight properties, like scheduled arrival time, aircraft size, number of passengers, feasible set of gates, etc., are determined using a random number generator, according to predefined values of distribution parameters. Table 4.6 summarizes the values of the basic parameters used in the experiments.

4.5.2 Set 1: Improving the Assignment Formulation

In this set of experiments (Appendix Table A4), we compare the performance of the different versions of quadratic formulation `Q`, the different versions of aggregating `Formulation A`, and also compare `Q` and `A` to each other. The results (Table 4.7) can be summarized as follows:

- The solution speed increases significantly from Formulation `Q-A` to `Q-S` and `Q-F`.

Table 4.6: Basic Parameter Values in Sensitivity Analysis Experiments

	Parameter	**Value**
Flights	Aircraft load factor limits	(0.5, 1)
	Percentage of connecting passengers	50%
	Gate occupancy duration (minute)	30' for $\leq$150 seats, 40' for $>$ 150 seats
	Disruption Probability	0.4
	Number of aircraft types	9
Airport	Shadow Constraint limit (m)	50
	Concourse Layout	Linear
Operator Decisions	Additional holding time range (min)	(-20,50)
	Step size (minutes)	10
Assumptions	Random Delay Range (minutes)	(-10,70)
	Passenger Walking Speed (meters/minute)	70

- Without the auxiliary constraint (4.19), Formulation `Q-S` cannot produce a feasible solution within half an hour.
- The fastest version of formulation `A` is `A-AS3`, which is 3-index, is defined over all connections, and uses the set of arriving or departing flights with the minimum cardinality.
- Models with formulation `A` are faster than problems with formulation `Q-S` in most cases.

First, we compare the performance and model properties between the different quadratic formulations `Q-A`, `Q-S`, `Q-F`, `Q-FA`, where we optimize for a linear combination of missed passenger connections and passengers with gate changes. The objective function does not include cost components that depend on the success of connections, since formulations `Q-F` and `Q-FA` cannot handle them, as explained in section 4.2.2. On average (Figure 4.2), formulation `Q-F` is the fastest one, with average running time of 0.2 seconds; despite the fact that we expected formulation `Q-FA` to result in further improvement, this did not happen, since formulation `Q-F` is on average 3.9 times faster. However, both formulations result in low running times and they both present significant speedup compared to formulations `Q-A` and `Q-S`. Specifically, formulation `Q-F` is more than 500 times faster than formulation `Q-A` (average running time of 103 seconds) and almost 100 times faster than formulation `Q-S` (average running time of 21 seconds).

We observe there does not exist a single case where Formulation `Q-S1` yields a single feasible solution within the predefined time limit (1800 seconds), since the

Table 4.7: Comparison between Formulations Q-A, Q-S1, and Q-S2

Combination	Flow Approach	Time	Gap
1	Q-A	-	100%
	Q-S1	-	32545%
	Q-S2	63.9	0%
2	Q-A	-	100%
	Q-S1	-	13392%
	Q-S2	23.9	0%
3	Q-A	-	100%
	Q-S1	-	8730%
	Q-S2	12.8	0%
4	Q-A	-	100%
	Q-S1	-	1E+100%
	Q-S2	23.3	0%
5	Q-A	-	97%
	Q-S1	-	11417%
	Q-S2	48.3	0%

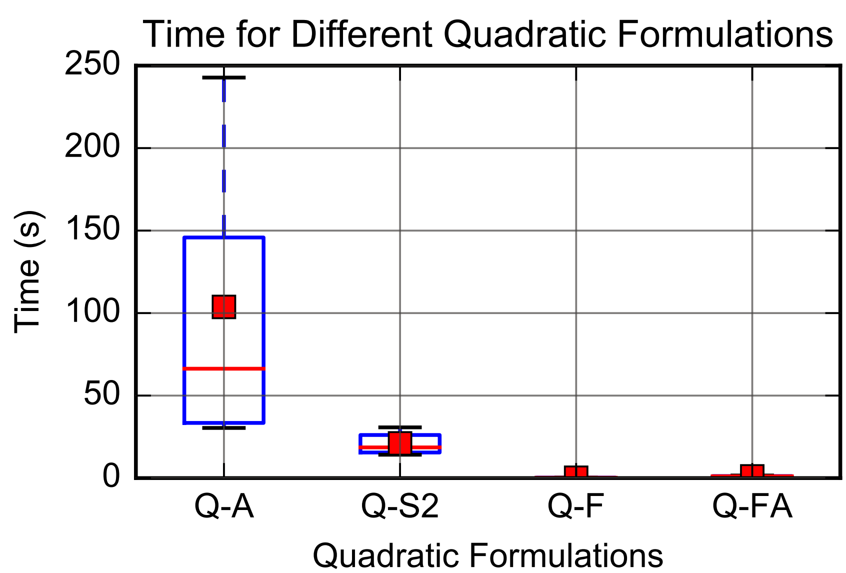

Figure 4.2: Running time for different quadratic formulations.

solver cannot find a feasible solution in the root of the branch-and-bound tree without the addition of auxiliary constraint set (4.19).

Next (Appendix Table A5), we compare formulations `Q-A` and `Q-S` assuming the objective is to minimize a weighted sum of missed connections, passenger walking distance, and gate changes. The results present a similar trend to the previous case, with Formulation `Q-S1` consistently failing to produce feasible solutions, while version `Q-S2` produced feasible solutions relatively quickly (64 seconds in the worst case, for an airport of 50 gates and a planning horizon of 4 hours, with 30 flight arrivals and departures per hour). The improvement in running time between formulations `Q-A` and `Q-D` can be attributed not only to the reduction in problem size (approximately 29 fewer variables and constraints) but also to the use of the auxiliary constraints.

Moving on to the aggregating approach, we test (Table A6) all alternative versions of formulation A. The Box-and-whisker plots for the running times are shown in Figure 4.3. On average, formulation `A-AS3` is the fastest one, yielding the lowest times in 5 out of the 12 cases and the second lowest in 3 out of 12, where it is outperformed only by formulation `A-SL2`. As can be seen in Figure 4.4, these are the two formulations with the smallest number of variables and constraints. However, reducing the number of variables and constraints does not necessarily improve computational performance: This is obvious in the case of formulation `A-AL2`, which fails to find an optimal solution in 11 out of the 12 cases, despite the fact that, for every case, it produces the third smallest model, following only formulations `A-SL2` and `A-AS3`.

In addition, it is interesting to observe the importance of selecting an appropriate flight set over which the ξ variables are defined. Out of all the models that fail to find the optimal solution within the time limit of 1/2 hour, 97% used formulations `A-SL2`, `A-SL3`, `A-AL2`, and `A-AL3`, which define ξ over the set of flights (arriving or departing) with the largest cardinality. In total, out of all models where the variables are defined over the largest set of flights, 67% eventually run out of time.

It is also worth noting (Figure 4.3) that for every pair of formulations, a 3-index formulation always outperforms its respective 2-index formulation.

We then compare the aggregating approach with the quadratic one (case studies in Table A7. For each approach, we use the formulation that has yielded the best results so far, i.e. `Q-S2` and `A-AS3`. Clearly (Figure 4.5), `A-AS3` outperforms `Q-S2`, which runs out of time in 33% of the cases and has an optimality gap that varies between 4% and 65%.

4.5.3 Set 2: Comparing the Assignment Formulation with Network Flow Formulations

In this set of experiments, we compare our best aggregating formulation, i.e., `A-AS3`, with the network flow formulations `YL` and `ZK`, proposed by Yu and Lau (2015) and Zhang and Klabjan (2017), respectively. The findings of this experimental set can be summarized as follows:

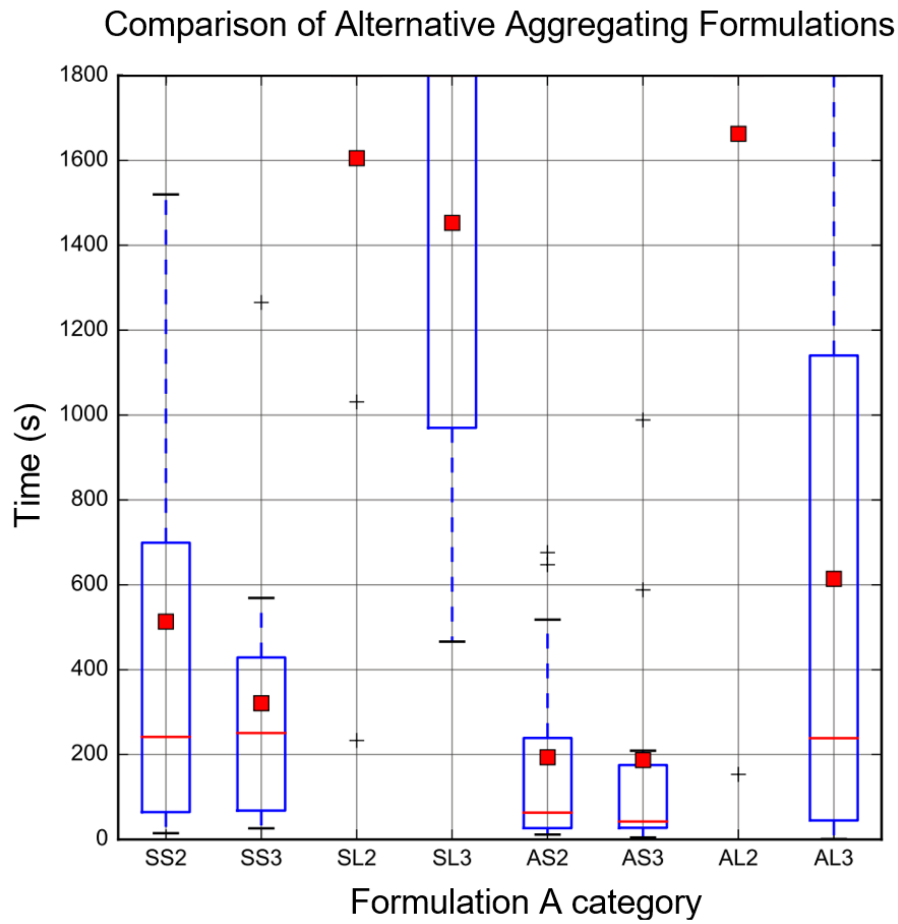

Figure 4.3: Running time for all aggregating formulations. Reaching the time limit is represented with a value of 1800.

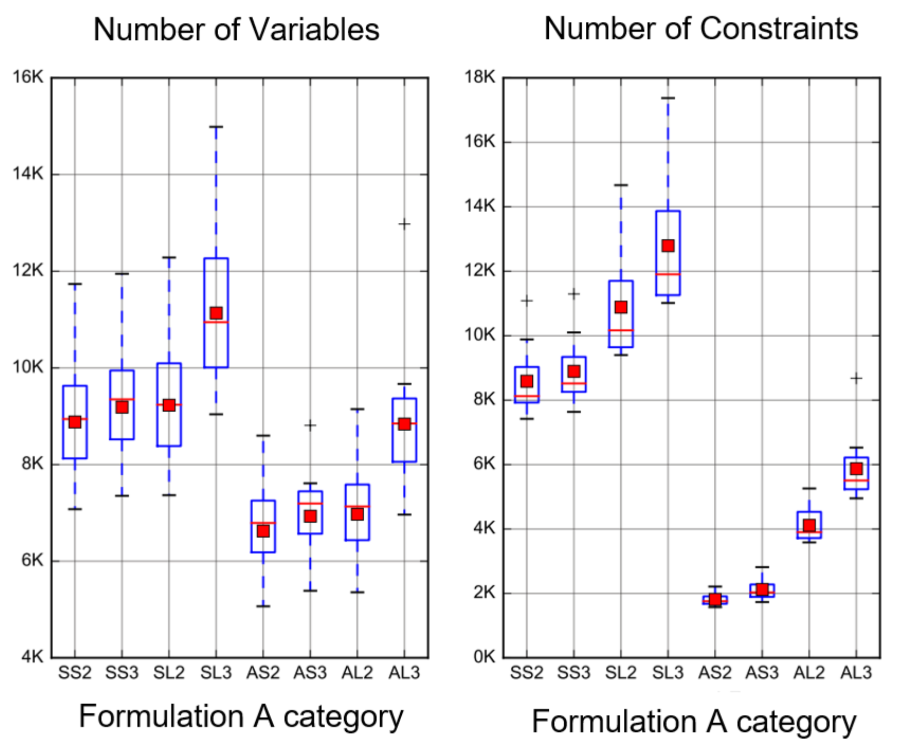

Figure 4.4: Number of variables and constraints for different versions of the aggregating formulation.

- The proposed formulation `A-AS3` always outperforms `YL` approach, but is less efficient than `ZK` approach when the cost of successful connections is considered.

- However, when the only objective associated with transfers is the number of missed connections, `A-AS3` outperforms `ZK`, with the difference in the required time increasing with the number of gate cliques.

4.5.3.1 Our Formulation Vs. Yu and Lau's Formulation

To compare the proposed formulation with Yu and Lau's (2015), fwe adapt the models based on each others simplifying assumptions. Therefore, in `A-AS3`, we a) simplify the sets $Q^A_{ii'}$ and $Q^F_{ii'}$ so that they only depend on the available time (section

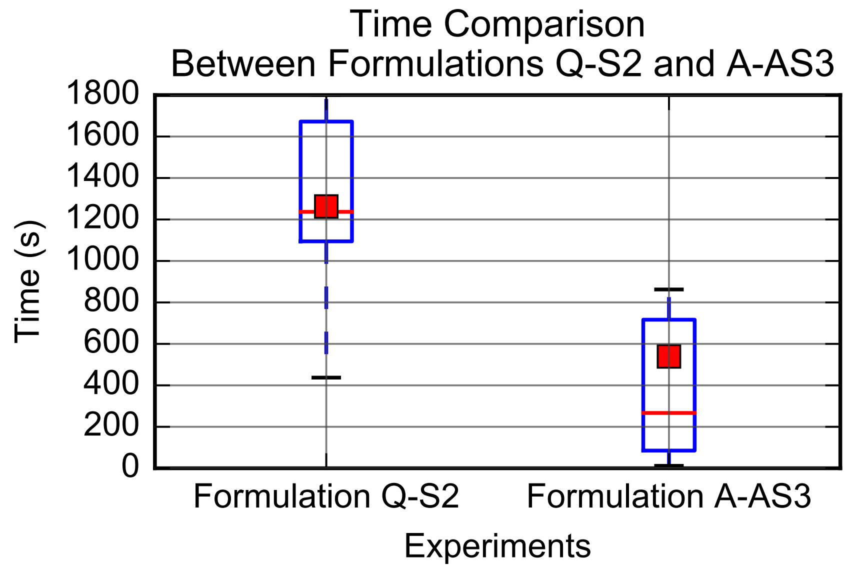

Figure 4.5: Running time for Formulations `Q-S2` and `A-AS3`.

4.4.3) and b) omit any objective terms referring to the cost of successful connections (section 4.4.4). On the other hand, to account for fixed occupancy duration (section 4.4.6), `YL` formulation is simplified by only including arcs of specific length for each flight. In all cases (Appendix Table A8), Gurobi can find the optimal solution within seconds, not exceeding 1.6 seconds under Formulation `A-AS3`, and 13.1 seconds in Yu and Laus model. Also in all cases, our model outperformed Yu and Lau's, requiring between 46% and 88% less time. Figure 4.6 depicts the optimization time (a), number of variables (b), number of constraints (c), and number of non-zeros (d) for the two formulations.

4.5.3.2 Our Formulation Vs. Zhang and Klabjan's Formulation

To compare our model with Zhang and Klabjan's, the assignment formulation `A-AS3` is modified to account for gate cliques, while formulation `ZK` is simplified

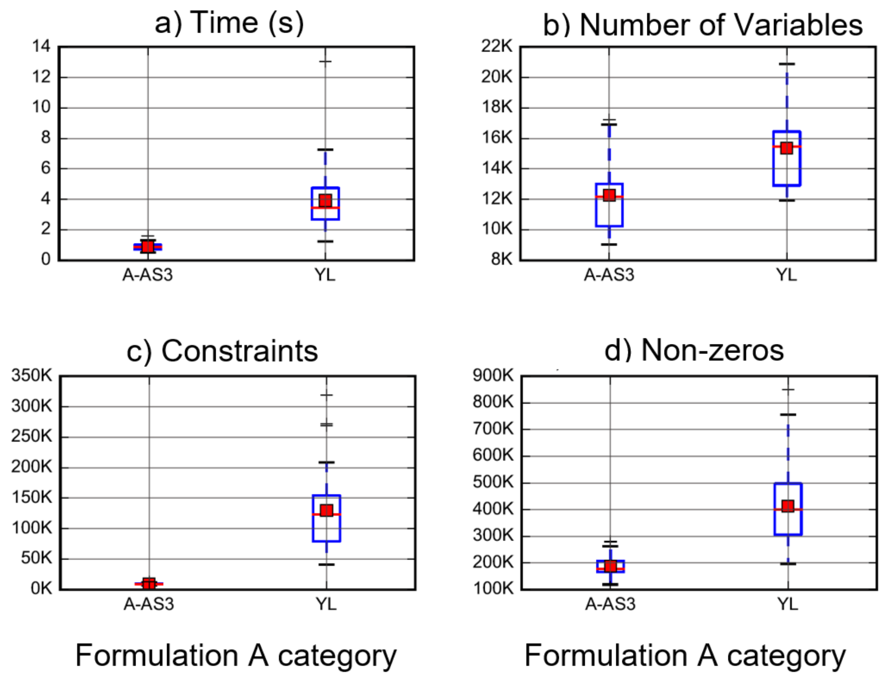

Figure 4.6: Proposed Formulation Vs. YL Formulation

by assuming fixed flight duration (section 4.4.6). We also assume that the cost of missing a connection is always greater than the cost of a successful connection (section 4.4.5). To divide the gates into cliques we adapt out formulation accordingly and introduce the following notation:

Sets:

C: Set of gate cliques (indexed as c and c').

$C_i \subset C$: Set of gate cliques compatible with flight i.

Costs:

$C^{\xi,S}_{(i,i',c,c',k,k')}$: Aggregating cost coefficients.

$Q^A_{ii'}C$: Set of allowed combinations (c, c', k, k') for connection (i, i') .

$Q^F_{ii'}C$: Set of forbidden combinations (c, c', k, k') for connection (i, i') .

The aggregating variables ξ are defined similarly to before, but since they are affected by the clique assignment instead of the specific gate assignment, constraints (47)-(48) are modified accordingly:

$$\xi_{ick} \geq \sum_{\substack{i' \in F^D: \\ (i,i') \in T}} \sum_{c' \in C_{i'} \cap C'} \sum_{k' \in W_{i'}} C^{\xi,S}_{ii'cc'kk'} X_{i'j'k'} - UB^S_{ick}, \tag{4.56}$$
$$i \in F^A,\ c \in C_i,\ k \in W_i$$

$$\xi_{ick} \geq 0, i \in F^A, c \in C_i, k \in W_i \tag{4.57}$$

To cluster the gates, we use a Multidimensional Scaling (MDS) approach. MDS was first proposed by Torgerson (1952) and is used for visualizing the level of similarity between different points of a dataset, based on the value of a distance function. For m objects, we develop an $m \times m$ dissimilarity matrix Δ, where the element δ_{ij} is the value of the distance function for objects i and j. MDS then maps the objects to the N-dimensional space by calculating m co-ordinate vectors $\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m \in \mathbb{R}^N$ (one for each object), each of length N, such that the distance between every two objects i and j, based on the new coordinates, is as close as possible to the dissimilarity value δ_{ij}:

$$\|\hat{x}_i - \hat{x}_j\| \approx \delta_{i,j},\ i, j \in 1, \ldots, I. \tag{4.58}$$

In this case, we use passenger connection times as a measure of dissimilarity. When examining gate connections, we are interested in the required connection time between the gates, which depends not only on the distance between gates, but also on the layout of the airport, the available transportation modes, etc. As a result, Euclidean distance is not sufficiently representative of the required connection time, especially in airports with multiple terminals. We map the gates on the N-dimensional space (for simplicity we choose $N = m$) and perform k-means clustering based on the new coordinate vectors. Each case is tested for k=5, 6, and 7 cliques, as well us without the use of cliques.

When walking distance is included in the objective, all cases (Table A9) are solved to optimality within 30 minutes for both formulations (Table 4.8). However, in spite of the fact that the assignment models are smaller (they require up to 96% fewer variables and up to 98% fewer constraints), the formulation of Zhang and Klabjan (2017) is faster in 33 of the total 48 models generated. We also observe that, whenever the assignment formulation is faster, it is when no cliques have been used and all gates are treated individually. When the network formulation is faster, it can be faster by around 10% up to 98%; similarly, when the assignment formulation is faster, it can be faster by less than 8%, up to 93%. No patterns regarding direct correlation between the model performance and the size of the problem were observed.

However, when we remove the cost of successful connections from the objective function (Table A10), a different trend is observed: the assignment formulation outperforms the network flow formulation in all cases, regardless of clustering or problem size (Table 4.9). The performance difference is now consistently significant, with the proposed formulation being 9 to almost 30 times faster. The speedup (ratio of ZK time to A−AS3 time) increases with the number of clusters (Figure 4.7), while without clustering, the proposed formulation is between 11 and 100 times faster.

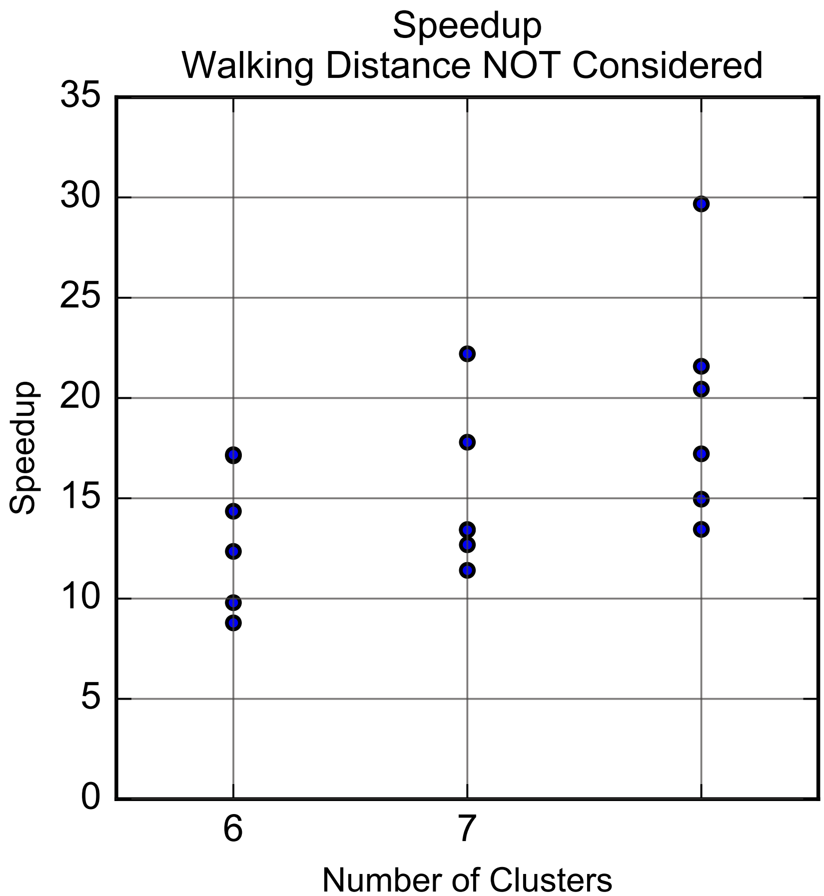

Figure 4.7: Speedup of A−AS3 Vs. ZK formulation. Successful connection cost is not taken into account

Table 4.8: Time Comparison Between Formulations A–AS3 and ZK, Walking Distance Considered

Case	**Clusters**											
	5			**6**			**7**			**None**		
	A–AS3	**ZK**	**Diff.**	**A–AS3**	**ZK**	**Diff.**	**A–AS3**	**ZK**	**Diff.**	**A–AS3**	**ZK**	**Diff.**
1	4.7	2.4	95.4%	17.2	4.5	283.9%	41.5	4.2	895.6%	15.4	58.1	-73.4%
2	5.6	4.5	26.3%	62.7	12.5	400.1%	33.9	5.4	527.5%	43.7	20	118.0%
3	24.5	3.4	619.1%	19.7	5.2	278.4%	19.1	4.1	367.1%	47.1	34.9	35.0%
4	10.8	5.8	87.1%	103.5	7.4	1302.1%	113.5	12.5	807.3%	170.5	88.7	92.2%
5	8.4	11.5	-27.1%	32.7	8.8	269.4%	59.5	21.7	174.8%	81.9	97.5	-16.1%
6	0.7	6.4	-89.7%	0.6	6.1	-90.7%	0.6	6.3	-90.1%	0.6	8.3	-92.5%
7	4.7	3.9	19.7%	5.8	7	-16.9%	2.6	7	-63.2%	6.6	53.1	-87.6%
8	3.1	6.2	-50.5%	16.8	11.8	42.7%	16.4	28.8	-43.1%	21.4	95.8	-77.7%
9	107.2	4.2	2432.3%	153.8	13.6	1032.7%	636.8	13.5	4613.9%	422	130.4	223.6%
10	28.7	5.9	387.6%	467.4	8	5774.3%	221.6	8.9	2396.2%	1241.2	157.6	687.8%
11	5.5	7.7	-29.0%	30.2	19.6	53.8%	58.4	28	108.6%	82.7	27.7	197.9%
12	8	5.8	37.1%	18.1	19.5	-7.0%	33.4	18.8	78.0%	71.1	32.6	118.0%

Table 4.9: Time Comparison Between Formulations `A-AS3` and `ZK`, Walking Distance Not Considered

Case	**Clusters**											
	5			**6**			**7**			**None**		
	A-AS3	**ZK**	**Diff.**	**A-AS3**	**ZK**	**Diff.**	**A-AS3**	**ZK**	**Diff.**	**A-AS3**	**ZK**	**Diff.**
1	0.5	5.8	-91.9%	0.5	6.6	-92.6%	0.4	8.9	-95.1%	1	99.2	-99.0%
2	0.5	8.8	-94.2%	0.8	9.8	-92.1%	1.1	16.7	-93.3%	2.6	120.4	-97.8%
3	0.6	5.9	-89.8%	0.5	6.2	-91.2%	0.5	7.1	-92.6%	0.5	6.2	-91.2%
4	0.4	6.5	-94.2%	0.6	8	-92.5%	0.6	12.1	-95.4%	1.4	182.4	-99.2%
5	0.4	3.2	-88.6%	0.4	7	-94.4%	0.4	6	-94.2%	0.6	37	-98.5%
6	0.6	9.3	-93.0%	0.9	20.1	-95.5%	0.8	22.3	-96.6%	0.8	259.5	-99.7%

4.5.4 Set 3: Large Case Study

In the final set of experiments, we verify our results in a real-sized airport. We borrow the layout of Athens International Airport in Greece, which consists of a main and a satellite terminal.

We perform the reassignment for 3, 6, and 8 hour-long reassignment windows, assuming hourly flow rate equal to 35 and 40 flights per hour, 52 total gates, and a different random number generator seeds for each case, resulting in a total of 12 cases (Table 4.10).

Table 4.10: Case Study Description

Case	**Flights / Hour**	**Schedule (Hours)**	**Reassignment Window (Hours)**
1	35	3	1
2	35	3	1
3	40	6	4
4	40	6	4
5	35	8	4
6	35	8	4
7	40	3	1
8	40	3	1
9	35	6	4
10	35	6	4
11	40	8	4
12	40	8	4

For each case, we run three subsets of experiments:

- Subset A1: We test the aggregating formulation A-AS3 for three different objective function combinations: a) When walking distance is involved (`WALK`), b) When walking distance is not involved (`NO_WALK`), and c) When walking distance is not involved, and we also assume that the success of a connection depends only on the available time between the flights (NO_WALK_SIMPLE) (section 4.4.3). No additional constraints were considered.

- Subset A2: We compare the aggregating formulation A-AS3 with YL formulation for the NO_WALK_SIMPLE objective function combination. The common assumptions of both formulations are adopted. We also consider additional constraints included in the study by Yu and Lau (2015), namely shadow constraints and constraints regarding flights operated by the same aircraft.

- Subset A3: We compare the aggregating formulation A-AS3 with ZK formulation for the WALK and NO_WALK objective function combinations. Shadow constraints, as in the study by Zhang and Klabjan (2017), are also included.

The basic model properties (number of variables, constraints, and non-zeros) are shown in Tables 4.11 - 4.13, while Tables 4.14 - 4.16 shows the optimization results (running time, resulting gap, and nodes explored in the branching procedure). Subset A1 clearly demonstrates that, even under the same constraints, i.e. the same feasible region of the problem, the objective function is a critical component determining the running time of the cut-and-branch procedure: When total walking distance is not included in the objective, i.e. in the NO_WALK and SIMPLE cases, an exact solution is always found within less than 3.5 seconds (Table 4.14). However, the WALK case is much more difficult to solve, with the solution time always higher than 41 seconds and exceeding the limit of 15 minutes in large case studies. In addition, the default cut generation procedures applied by Gurobi can solve all

Table 4.11: Number of Variables, Constraints, and Non-Zeros for Experimental Set A1

Case	**Variables**	**Constraints**	**Non-Zeros**		
			W	**NW**	**S**
1	8976	1773	246001	104946	104946
2	7074	1563	193270	113450	113450
3	16020	2735	484220	142719	142719
4	16004	2603	408439	141250	141250
5	17162	2721	661029	183445	183445
6	17209	2449	359468	86234	86234
7	9126	1178	214436	69831	69831
8	9329	1296	173469	73798	73798
9	20342	2602	591817	179286	179286
10	21621	2873	768324	260138	260138
11	22233	2698	550537	145534	145534
12	22274	2469	537952	120725	120725

NO_WALK and SIMPLE cases at the root; however, all WALK cases require further branching, as can be seen from the number of nodes explored.

Subset A2 confirms the results of section 4.5.3.1, where formulation A−AS3 was shown to be more efficient compared to YL, as can be seen from the running times and the number of nodes explored in the cut and branch tree. We also observe that the additional constraints occasionally make the problem infeasible (Table 4.12).

In subset A3, we compare formulation A−AS3 with ZK, assuming that each terminal corresponds to a gate clique. The results confirm that formulation A−AS3 is consistently faster than ZK when walking distance is not included in the objective function. However, A−AS3 is also shown to outperform ZK in 7 out of 12 WALK cases, while both approaches run out of time in 2 cases.

Table 4.12: Number of Variables, Constraints, and Non-Zeros for Experimental Set A2

Case	**Variables**		**Constraints**		**Non-Zeros**	
	A-AS3	**YL**	**A-AS3**	**YL**	**A-AS3**	**YL**
1	8827	10358	7459	218077	233774	553413
2	8492	9028	7653	193963	158470	482834
3	18806	21359	15143	843152	377439	1932884
4	19214	21295	16000	567109	424874	1375949
5	22170	23358	19718	580755	371965	1404367
6	19629	21718	16881	587342	394498	1423561
7	11884	12583	9129	297673	255203	726276
8	10992	11472	8479	263844	253544	656668
9	24667	26505	18034	804322	485695	1914903
10	24679	26598	18300	855093	441432	2014969
11	27375	28825	20952	711869	437681	1748297
12	28077	29444	21821	1055492	496998	2448461

Table 4.13: Number of Variables, Constraints, and Non-Zeros for Experimental Set A3

Case	**Time**			**Gap**			**Nodes**		
	W	NW	S	W	NW	S	W	NW	S
1	32	25	1	12	0%	0%	0%	0%	17117
2	9	7	0	6	0%	0%	0%	0%	1088
3	52	437	1	130	0%	0%	0%	0%	1254
4	22	51	1	49	0%	0%	0%	0%	1101
5	100	227	3	70	0%	0%	0%	0%	1159
6	27	183	1	64	0%	0%	0%	0%	1091
7	14	47	0	18	0%	0%	0%	0%	1031
8	51	929	2	1800	0%	0%	0%	1%	6179
9	-	-	6	-	1%	1E+100	0%	11%	4494
10	-	-	9	-	2%	9%	0%	13%	1790
11	171	1802	6	645	0%	0%	0%	0%	1219
12	433	-	15	-	0%	1E+100	0%	10%	1212

Table 4.14: Running Time, Gap, and Number of Nodes for Experimental Set A1

Case	**Time (s)**			**Gap**			**Nodes**		
	W	**NW**	**S**	**W**	**NW**	**S**	**W**	**NW**	**S**
1	552	0	0	0%	0	0	34785	0	0
2	393	0	0	0%	0	0	27912	0	0
3	348	1	1	0%	0	0	17692	0	0
4	88	1	1	0%	0	0	1200	0	0
5	900	2	2	40%	0	0	20837	0	0
6	132	0	0	0%	0	0	1277	0	0
7	42	0	0	0%	0	0	1156	0	0
8	57	0	0	0%	0	0	2117	0	0
9	725	2	2	0%	0	0	10685	0	0
10	900	3	3	50%	0	0	10709	0	0
11	734	2	2	0%	0	0	5174	0	0
12	900	0	0	29%	0	0	4072	0	0

Table 4.15: Running Time, Gap, and Number of Nodes for Experimental Set A2

Case	**Time (s)**		**Gap**		**Nodes**	
	`A-AS3`	**`YL`**	**`A-AS3`**	**`YL`**	**`A-AS3`**	**`YL`**
1	2	7	0	0	0	0
2	Inf	Inf	1E+100	0	0	0
3	4	628	0	0	0	12129
4	12	121	0	0	1735	1120
5	Inf	Inf	1E+100	0	0	0
6	4	90	0	0	0	6123
7	2	32	0	0	0	1114
8	3	TL	0	0	0	173478
9	16	TL	0	0	0	8733
10	7	TL	0	0	0	1693
11	Inf	Inf	1E+100	0	0	0
12	Inf	Inf	1E+100	0	0	0

Inf: Infeasible,
TL: Time Limit

Table 4.16: Running Time, Gap, and Number of Nodes for Experimental Set A3

Case	**Time (s)**				**Gap**				**Nodes**			
	W		**NW**		**W**		**NW**		**W**		**NW**	
	A-AS3	**ZK**	**A-AS3**	**ZK**	**A-AS3**	**ZK**	**A-AS3**	**ZK**	**A-AS3**	**ZK**	**A-AS3**	**ZK**
1	32	25	1	12	0%	0%	0%	0%	17117	0	0	0
2	9	7	0	6	0%	0%	0%	0%	1088	0	0	0
3	52	437	1	130	0%	0%	0%	0%	1254	3337	0	2181
4	22	51	1	49	0%	0%	0%	0%	1101	0	0	0
5	100	227	3	70	0%	0%	0%	0%	1159	931	0	0
6	27	183	1	64	0%	0%	0%	0%	1091	2366	0	0
7	14	47	0	18	0%	0%	0%	0%	1031	1521	0	489
8	51	929	2	1800	0%	0%	0%	1%	6179	46295	0	51558
9	-	-	6	-	1%	1E+100%	0%	11%	4494	1466	0	4566
10	-	-	9	-	2%	9%	0%	13%	1790	2724	0	2666
11	171	1802	6	645	0%	0%	0%	0%	1219	21062	0	12895
12	433	-	15	-	0%	1E+100%	0%	10%	1212	1801	0	2290

4.5.5 Discussion and Implications

The results of sections 4.5.2-4.5.4 can facilitate airport authorities in real-time decision making, where the most suitable formulation depends on a) the priorities of the decision maker, and b) the problem limitations.

On the one hand, the experiments showed that assignment formulations always outperform network flow formulations when walking distance is not considered. When walking distance is optimized, their behavior is comparable network flow formulation ZK, occasionally outperforming them and sometimes performing rather poorly. Meanwhile, priorities vary among airports and airlines and can also change according to the circumstances. For example, in cases of large scale disruptions (e.g. due to temporary airport closures), the airport authority would focus on reducing missed connections rather than improving the level of passenger service; in this case, an aggregating assignment formulation would always be more suitable than a network flow formulation. However, other circumstances, such as the existence of flights with a high percentage of priority (first class and business) passengers, might require to optimize the walking distance. In this case, either a network flow or an assignment formulation could work. Finally, the choice of formulation depends on the available time for optimization and can change in airports where demand presents high variability: In days of relatively low traffic, when the available time is higher, a multi-objective approach that includes level of service can be used. However, when high demand increases both the problem size and the number of potential gate conflicts, an assignment model which does not consider walking distance might be preferred.

On the other hand, certain problem circumstances might require specific formulations. For example, when disruptions are so intense that flight cancellations are preferable to additional delays, only an assignment formulation can be used, as explained in section 4.4.5. Finally, assuming fixed flight duration, the proposed assignment formulation is the most easily adaptable to different assumptions and objective functions.

4.6 Summary, Conclusions, and Future Research

In this study, we have analyzed the formulation of the flight-to-gate reassignment problem with passenger connections. Our research has focused on three main directions: First, we have analyzed the two dominant formulations in the literature, i.e. the multidimensional assignment and the network flow-based formulations. Second, we have adapted and strengthened the existing assignment-based formulations by reformulating the constraints and introducing valid inequalities that facilitate the cut-and-branch procedure. Finally, we have developed a new assignment formulation that considers the layout of the airport and the available connection time to determine whether a connection will eventually be made or missed.

We have analyzed the existing and the new formulations from both a theoretical and an experimental perspective: First, we have estimated upper bounds on the number of variables and constraints of each formulation. We have also pinpointed the differences in the underlying assumptions of each approach and, consequently, in the applicability of each one under different types of objective functions, cost coefficients, and modeling assumptions. To evaluate each formulation in practice, we generated experimental sets of various sizes with different airports and flight schedules. Our results indicate that the aggregating assignment formulations outperform the quadratic formulations, while assignment formulations are consistently more efficient than network flow-based approaches when the cost of successful connections is not included in the objective function.

Having developed a strong and efficient mathematical formulation, the next step is to embed it in a model-based metaheuristic framework to further accelerate the solution procedure and produce near-optimal results in real time.

Chapter 5

A Modified Variable Neighborhood Search With Local Branching Approach for the Flight-To-Gate Reassignment Problem

Chapter Overview

In this chapter, we develop a metaheuristic framework to assist airport operators in reassigning aircraft to gates in a quick and efficient way. Since we have already explored and significantly improved the mathematical formulation of the gate reassignment problem in Chapter 4, we embed the mathematical programing model in a metaheuristic framework that relies on Variable Neighborhood Search with Local Branching. We explore alternative ways to define the key notion of a solution neighborhood, given that transfer passengers are the main consideration of the problem. We calibrate the algorithm to determine the optimal parameter combinations and test it in a separate set of experiments to verify its applicability. The results indicate that within 10 minutes the algorithm can reach a provably optimal solution for which an MIP solver requires between 1/2 and 3 hours. We then use the algorithm in a set of sensitivity analysis experiments and verify its performance under variations in external and operational parameters. In the end, we propose a set of extensions to the algorithm to improve the quality of the final solution and the progress of the optimization. In this chapter, we also introduce a new measure of effectiveness concerning transfer passengers, i.e., the number of baggage pieces ("misconnected baggage") that fail to make the connection from the inbound to the outbound flight.

Keywords: *gate reassignment; MIP heuristics; metaheuristics; Local Branching; Variable Neighborhood Search with Local Branching*

5.1 Introduction

By this point, we have examined the Gate Reassignment Problem with passenger connections from a practical (Chapter 3) and a modeling (Chapters 3 and 4) perspective. First (Chapter 3), we examined the measures of effectiveness and the way to incorporate them in the objective function cost coefficients. Based on our observations, we proceeded to the development of a time-index assignment model that handles passenger transfers in a realistic way. Then (Chapter 4), we examined alternative formulations of the gate reassignment problem with passenger connections and identified the assumptions under which each formulation is more suitable. We also used state-of-art knowledge combined with our own research to improve the proposed assignment model.

As it naturally follows, the next step is to use the findings described in the previous sections to develop an algorithm that is capable of solving the gate reassignment problem with passenger connections in an acceptable amount of time. In this section, we will examine how we can embed the formulation in a solution framework that combines the advantages of our improved mathematical formulation with the efficiency and exploration capability of modern metaheuristics. A recently developed metaheuristic which combines the above properties is Variable Neighborhood Search with Local Branching (VNS-LB), which will be implemented in this Chapter. For comparison, but primarily because it shares the same basic properties with VNS-LB, we will also adapt simple Local Branching, which is an MIP heuristic that has already been tested in a similar context (Yu and Lau, 2015).

Before proceeding to our application, we will first explain the three main methodological concepts of this chapter, i.e., MIP heuristics, metaheuristics, and MIP ("model-based") metaheuristics. We will then introduce a new measure of effectiveness, i.e. missed baggage connections, which we will use in our objective function in some of the experiments that follow.

5.1.1 MIP Heuristics

MIP heuristics use the mathematical formulation of an optimization problem to generate a near-optimal or optimal solution in a limited amount of time. Contemporary MIP solvers embed MIP heuristics in the Branch-And-Bound process, along with presolve and cut generation, to accelerate the solution procedure. A comprehensive review of MIP heuristics can be found in Berthold (2006), who identify two main categories of MIP heuristics, namely start heuristics and improvement heuristics.

1. Start heuristics: Start heuristics aim at finding relatively good feasible solutions in the early stages of the Branch-and-Bound process (Berthold, 2006).

2. Improvement heuristics: Improvement heuristics use some "provided information" to generate new feasible solutions which are better than one or more given solutions (Berthold, 2006).

5.1.2 Metaheuristics

Metaheuristic techniques are a major subfield of stochastic optimization which includes a variety of general algorithms that can be applied to a very wide range of problems (Luke, 2013). Glover and Kochenberger (2003), define them as "solution methods that orchestrate an interaction between local improvement procedures and higher level strategies to create a process capable of escaping from local optima and performing a robust search of a solution space". In other words, metaheuristics combine the ability to explore different areas of the solution space ("diversification") while also focusing on discovering the best solution in every area ("intensification"). At each stage of the solution procedure, the metaheuristic uses the available information to determine whether the search should be guided towards discovering new solutions or into a deeper local search of the current solution area ("exploration Vs. exploitation").

Contrary to common heuristics, a metaheuristic generally does not take into account the specific structure or properties of the problem in hand and treats the objective function of the problem as a “black box”. Consequently, metaheuristics are generally problem-independent, a fact which contributes to their wide applicability to a broad range of combinatorial optimization problems.

Simulated annealing, tabu search, and genetic algorithms are only a few examples of metaheuristics. Recent research in the field focuses on the development of so-called “hybrid metaheuristics” or “metaheuristic hybrids”. A detailed survey on this special category of algorithms can be found in Raidl et al. (2010). Metaheuristic hybrids may combine features of different metaheuristics, or combine metaheuristics with other optimization techniques, such as branch-and-bound or mathematical programming (Raidl et al., 2010).

5.1.3 MIP Metaheuristics

As explained, a certain category of hybrid metaheuristics are the result of combining classical metaheuristics with mathematical programming. The algorithms of this class are sometimes referred to as “model-based metaheuristics” or “matheuristics”. In this dissertation, we will use the term “model-based metaheuristics” or the equivalent “MIP-Metaheuristics”.

There are two main types of model-based metaheuristics:

1. Algorithms which use the MIP model as a subroutine for known metaheuristics, such as genetic algorithms, tabu search, and Variable Neighborhood Search.

2. Algorithms which use the MIP as a paradigm for new metaheuristics. These algorithms could not exist without the use of the MIP model. Examples from this category are Local Branching (Fischetti and Lodi, 2003), Dynasearch (Congram et al., 2002), and Very Large Neighborhood Search (Ahuja et al., 2002).

In both cases, the algorithm handles the MIP formulation in different ways, such as for generating columns of a set partitioning formulation, or for generating relaxed versions of the problem. Therefore, a restricted version (which is easier to solve) of the original MIP problem is used in the procedure.

5.2 Local Branching and Variable Neighborhood Search with Local Branching: Overview

In this chapter, we use a model-based metaheuristic, namely Variable Neighborhood Search with Local Branching, to solve large instances of the gate reassignment problem. Since variable Neighborhood Search with Local Branching shares many sub-procedures with simple Local Branching, we will also apply the Local Branching heuristic in the same context. Before applying these techniques, we will introduce their main features and describe the general version of each procedure. To the best of our knowledge, this is the first application of Variable Neighborhood Search with Local Branching for solving the planned or real-time gate assignment problem.

5.2.1 Local Branching

Local branching (LB) is an MIP technique introduced by Fischetti and Lodi (2003) which relies on the iterative solution of properly defined sub-problems using the mathematical formulation of the problem. Although in principle designed as an exact method, LB can be adapted and used as a heuristic method to solve large and complicated problems (Hansen et al., 2006) by adding one or more linear constraints which restrict the solution space during the search procedure. These constraints push the search procedure to explore solutions that are “close” or “further away” from the current solution. For a general Binary IP problem, the local branching procedure is as follows:

Let X represent the total solution space. After a feasible solution $\tilde{x}_1$ is obtained, a new constraint is added, so that the candidate solution space is reduced to solution space X_1, which includes only solutions that are within a given Hamming distance from $\tilde{x}_1$, i.e.:

$$N_k(\tilde{x}_1) = \{x \mid d(x, \tilde{x}_1) \leq k\} \tag{5.1}$$

Using the MIP formulation, $N_k(\tilde{x}_1)$ can be represented by a local branching constraint:

$$\Delta(x, \tilde{x}_1) = \sum_{j \in \bar{S}} (1 - x_j) + \sum_{j \in B \setminus \bar{S}} x_j \leq k \tag{5.2}$$

where $\bar{S} = \{j \in B \mid \tilde{x}_1 = 1\}$. The restricted problem is ran for a given time limit and has three possible outcomes:

- If a better solution $\tilde{x}_2$ is found, a new branching constraint is created by replacing the "$\leq$" of constraint 5.2 with "$>$":

$$\sum_{j \in \bar{S}} (1 - x_j) + \sum_{j \in B \setminus \bar{S}} x_j > k \tag{5.3}$$

By adding constraint 5.3 to the problem, the search is moved "far" from $\tilde{x}_1$ and is now centered around the new solution $\tilde{x}_2$. Therefore, the feasible region X_2 is now given as

$$X_2 = X \cap (N_k(\tilde{x}_2) \setminus N_k(\tilde{x}_1)) = (X \setminus X_1) \cup N_k(\tilde{x}_2) \tag{5.4}$$

The new restricted MIP is solved and the procedure continues as long as the incumbent solution is improved. In general, after l steps, l new branching constraints have been added, and the feasible region X_l is found as

$$X_l = (X \setminus X_1 \setminus X_2 \cdots \setminus X_l) \cap N_k(\tilde{x}_{l-1}) \tag{5.5}$$

- If a new feasible solution is found within a predetermined time limit t_{node}, but is not better than the incumbent, the neighborhood X_l is reduced by replacing k with $k/2$. This procedure is called intensification:

$$\sum_{j \in \bar{S}} (1 - x_j) + \sum_{j \in B \setminus \bar{S}} x_j \leq k/2 \tag{5.6}$$

- If the problem is proven to be infeasible, or no feasible solution at all is found within t_{node} time, a diversification procedure is followed. First, the right-hand side of the branching constraint is increased by $k/2$:

$$\sum_{j \in \bar{S}} (1 - x_j) + \sum_{j \in B \setminus \bar{S}} x_j \leq k + k/2 \tag{5.7}$$

Then, a new constraint is added, to force the exploration of neighborhoods that are further away from the current solution x_l:

$$\Delta(x, x_l) > 1 \tag{5.8}$$

Finally, all previously added branching constraints are deleted. The new MIP problem is solved within a given time limit. The maximum allowed number of diversifications is given by a parameter $dv_{\max}$.

The solution procedure continues when the total time limit is reached, or when the maximum number of diversifications, $dv_{\max}$ has been made. Algorithm 1 shows in detail the steps of a general local branching algorithm, as was originally introduced by Fischetti and Lodi (2003).

Algorithm 1 General Local Branching Procedure (Fischetti and Lodi, 2003)

procedure LocalBranching(k, T^{TL}, N^{TL}, $dv_{\max}$, x^{opt})
    $rhs \leftarrow \infty, UB^{Best} \leftarrow \infty, UB \leftarrow \infty, TL \leftarrow \infty$
    $x^{\text{opt}} \leftarrow None$
    $\text{opt} \leftarrow \text{True}, \text{First} \leftarrow \text{True}$
    $d_v \leftarrow 0$
    $\text{Diversify} \leftarrow \text{False}$
    repeat
        if $rhs < \infty$ **then**
            Add $\Delta(x, x_{new}) \leq rhs$
        end if
        $TL \leftarrow \min T^{TL}, T^{TL} - \texttt{elapsed\_time}$
        $\text{stat} \leftarrow \text{MIPSolve}(TL, UB, \text{First}, x_{new})$
        $TL \leftarrow N^{TL}$

if stat = `optimal_solution_found` **then**
  if $c^T x_{new} < UB^{Best}$ **then**
    $UB^{Best} \leftarrow c^T x_{new}$
    $x^{\text{opt}} \leftarrow c$
  end if
  if $rhs \geq \infty$ **then return** opt
  end if
  Reverse the last local branching constraint into $\Delta(x, x_{cur}) \geq rhs + 1$
  Diversify $\leftarrow$ False, First $\leftarrow$ False
else if stat = `proven_infeasible` **then**
  if $rhs \geq \infty$ **then return** (opt)
    Reverse the last local branching constraint into $\Delta(x, x_{cur}) \geq rhs + 1$
    if Diversify **then**
      $UB \leftarrow \infty, TL \leftarrow \infty$; $dv \leftarrow dv + 1$; First $\leftarrow$ True
    end if
    $rhs \leftarrow rhs + k/2$; Diversify $\leftarrow$ True
  end if
else if stat = `feasible_solution_found` **then**
  if $rhs \leq \infty$ **then**
    if First = True **then**
      Delete the last local branching constraint $\Delta(x, x_{cur}) \leq rhs$
    else
      Replace the last local branching constraint $\Delta(x, x_{cur}) \leq rhs$ by $\Delta(x, x_{cur}) \geq 1$
    end if
  end if
  Refine(x_{new})
  if $c^T x_{new} \leq UB^{Best}$ **then**
    $UB^{Best} \leftarrow c^T x_{new}$;$x^{\text{opt}} \leftarrow_{new}$
  end if
  First $\leftarrow$ False, Diversify $\leftarrow$ False, $x_{new} \leftarrow x_{new}, UB \leftarrow c^T x_{new}, rhs \leftarrow k$
else if stat = `no_feasible_solution_found` **then**
  if Diversify **then**
    Replace the last local branching constraint $\Delta(x, x_{cur}) \leq rhs$ by $\Delta(x, x_{cur}) \geq 1$
    $UB \leftarrow \infty, TL \leftarrow \infty$; $dv \leftarrow dv + 1$; $rhs \leftarrow rhs + k/2$; First $\leftarrow$ True
  else
    Delete the last local branching constraint $\Delta(x, x_{cur}) \leq rhs$
    $rhs \leftarrow rhs - k/2$
  end if
  $Diversify \leftarrow$ True
end if
until `elapsed_time` $> T^{TL}$) or ($dv > dv_{\max}$

$TL \leftarrow T^{TL} - \texttt{elapsed\_time}$; First $\leftarrow$ False; stat $\leftarrow$ MIP SOLVE $(TL, UB^{Best}, \text{First}, x^{\text{opt}})$; opt $\leftarrow$ (stat $=$ optimal solution found) or (stat $=$ $\texttt{proven\_infeasible}$) **return** (opt)
end procedure

5.2.2 Variable Neighborhood Search with Local Branching

Variable Neighborhood Search (VNS) is a metaheuristic technique originally introduced by Mladenović and Hansen (1997). The basic idea of VNS is "a systematic change of neighborhood both within a descent phase to find a local optimum and in a perturbation phase to get out of the corresponding valley".

In its original form, VNS was not proposed as a model-based metaheuristic. However, Hansen et al. (2006) developed an alternative version of VNS, called Variable Neighborhood Search with Local Branching (VNS-LB), which takes advantage of the MIP formulation by using local branching constraints to define the neighborhood that is used in every step of the procedure.

In order to understand VNS-LB, we will first define the fundamental concepts of classic VNS, such as the neighborhood and the k-neighborhood (paragraph 5.2.2.1), and describe the steps of the algorithm (paragraph 5.2.2.2).

5.2.2.1 Defining a Neighborhood in VNS

A key concept of Neighborhood Search is the definition of the neighborhood of a solution. In general, the neighborhood of a solution p refers to the set of solutions that are "close" to p. The term "close" means that they can be easily computed from

p or that they have a common structure with p (Ambite, 2001). For example, in an assignment problem, one could define the neighborhood of a solution by exchanging the assigned tasks between two agents.

The definition of "neighborhood" is directly associated with the notion of "distance" between two solutions. The distance between two solutions x_1, x_2 measures the degree of difference between them, regardless of their objective function value, and is defined according to the specific problem it refers to. For example, for a Traveling Salesman Problem, the distance can be calculated as the number of different edges between the two tours. For a facility location problem, it can be measured as the number of facilities assigned to different locations. For a general binary integer program, it can be equal to the number of variables that receive a different value in the two solutions.

Sometimes, the definition of a neighborhood depends not only on the neighborhood structure, but also on the value of some distance measure as expressed through a parameter k; in this case we refer to the k-neighborhood of a solution x. Assume, for example, a Traveling Salesman Problem defined for a set of nodes $1, \ldots, N$, where a feasible solution x is a permutation of the nodes. Then, for $k = 2$, the 2-neighborhood of x can be defined as the set of solutions that share all but 2 common edges with x. In the assignment problem described before, the k-neighborhood of a solution may include all assignments for which exactly k agents have been assigned the same task.

In this context, the neighborhood of a solution x_1 is the set of all feasible solutions which belong within a certain distance from x_1. Setting the distance limit equal to k, we get the definition of the k-neighborhood of a solution, i.e., the set of solutions that are within at most k distance units from the solution. Therefore, in a Traveling Salesman Problem, the k-neighborhood is the set of all tours with most k different edges compared to the tour corresponding to $\tilde{x}$, while for the facility location problem, the k-neighborhood is the set of solutions with at most k facilities assigned to a different location compared to their assignment in $\tilde{x}$.

5.2.2.2 VNS Procedure

In VNS, a set of neighborhood structures is defined and sorted from 1 to k. The main idea of the search is to explore the neighborhood that is supposed to be the most promising ($k = 1$) and move to the next neighborhood only if no improved solution is found.

A typical VNS procedure can be roughly described as follows: First, a starting solution is found and set as current. Then, the neighborhood of the current solution is searched, and a new solution is found. If the solution is better than the current, it replaces the current solution and the search is centered around the new current solution. In the opposite case, the search continues to the next neighborhood of the current solution and a new solution is yielded. Similarly to before, if the solution is better than the current one, it replaces the current one and the search is centered around the new solution. Otherwise, the search again continues to the next neighborhood of the current solution. The procedure continued until all neighborhoods of a solution have been explored without improvement, or until a time limit is reached.

The main body of VNS involves the iterative execution of three basic subprocedures: The Shake function (Shake), the Variable Neighborhood Decent function (VND), and the Neighborhood Change function (NeighborhoodChange).

The NeighborhoodChange function is called when a new solution x' is found. If the new solution is better than the current x, then x' becomes the current solution and the procedure restarts centered around x'. If not, then the next neighborhood of x is searched. The procedure is described in Algorithm 2.

Algorithm 2 The Neighborhood Change Function, as shown in Hansen et al. (2010)

procedure NeighborhoodChange(x, x', k)
  if $f(x') < f(x)$ **then**
    $x \leftarrow x'$ ▷ Make a move
    $k \leftarrow 1$ ▷ Initial Neighborhood
  else
    $k \leftarrow k + 1$ ▷ Next Neighborhood
  end if
  return (x, k)
end procedure

The Shake Function generates a random point x' within the k-neighborhood of x. The pseudocode of the procedure follows in Algorithm 3:

Algorithm 3 The Shake Function, as shown in Hansen et al. (2010)

procedure Shaking(x, k)
  $w \leftarrow [1 + \text{rand}(0,1) \times \mid N_k(x) \mid]$
  $x' \leftarrow x^w$
  return x'
end procedure

Finally, VND is used instead of simple local search to find the best solution in the neighborhood under consideration, within a specified amount of time. VND itself is a simplified form of VNS, since it involves the systematic exploration of neighborhoods in a predetermined order and calls the NeighborhoodChange function when a new candidate solution is obtained. The steps of VND are described in Algorithm 4:

Algorithm 4 The Variable Neighborhood Descent Function, as shown in Hansen et al. (2010)

procedure VND(x, $K_{\max}$)
  $k \leftarrow 1$
  repeat
    $x' \leftarrow \arg\min_{y \in N_k(x)} f(y)$ ▷ Find the best neighborhood in $N_k(x)$
    $x, k \leftarrow$ NeighborhoodChange(x, x', k) ▷ Change neighborhood
  until $k = k_{\max}$
  return (x)
end procedure

The main VNS function iteratively calls all three sub-procedures, i.e., Shake, VND, and NeighborhoodChange. The procedure is shown in Algorithm 5.

Algorithm 5 The General Variable Neighborhood Search Procedure

procedure GVNS$(X, l_{\max}, k_{\max}, t_{\max})$
  repeat
    $k \leftarrow 1$
    repeat
      $x' \leftarrow$ Shake(x, k)
      $x'' \leftarrow$ VND$(x', l_{\max})$
      $x, k \leftarrow$ NeighborhoodChange(x, x'', k)
    until $k = k_{\max}$
  until $t > t_{\max}$
  return x
end procedure

A number of potential modifications Hansen et al. (2010) allow us to obtain simpler versions of VNS: By replacing the VND procedure with a simple local search heuristic, like best improvement (steepest descent) or first improvement, the Basic Variable Neighborhood Search (BVNS) is obtained; if the descent phase is completely eliminated, the so-called Reduced Variable Neighborhood Search (RVNS) is obtained. However, in this study, we use the general form of Variable Neighborhood Search, which includes all of the sub-procedures described before.

5.2.2.3 Variable Neighborhood Search With Local Branching

In VNS-LB, local branching constraints are used to define the neighborhood of the current solution $\tilde{x}$. More specifically, the k neighborhood of $\tilde{x}$ is the set of feasible solutions with a Hamming distance from $\tilde{x}$ (constraint 5.2) that is no greater than k. This defines a set of neighborhood structures $\{N_1, N_2, \ldots, N_k\}$, which are used both in the inner VND loop of Algorithm 5, and in the outer loop, when the SHAKE function is called. The value of k changes from 1 to $k_{\max}$ in the outer loop and from 1 to $l_{\max}$ in the inner loop. In each step of the outer loop, k is increased by a predetermined value k_{step}. The total procedure runs for at most $t_{\max}$ time, while t_{node} is the time limit of every MIP subproblem.

To reduce the number of required parameters, Hansen et al. (2006) allowed unlimited increase in the size of the neighborhood both in the outer and in the inner loop by not specifying a value for $k_{\max}$ or $l_{\max}$. For simplicity, they also set $k_{\min} = k_{\text{step}}$. Algorithm 6 presents in detail the steps of the procedure. Hansen et al. (2006) use the following parameters:

UB: Input parameter for the MIP solver. It defines the objective function cutoff, i.e., the worst acceptable objective function value.

First: Input for the MIP solver. If True, the solver returns the first feasible solution found. If False, the solver continues until the termination conditions (e.g., node time limit) are satisfied.

TL: Total time limit, used as input for the MIP solver. It is equal to the time limit allocated to the optimization procedure.

rhs: The right-hand side of the local branching constraint for the inner (VND) loop.

Cont: If True, the inner loop continues; if False, it breaks.

x_{opt}, f_{opt}: Incumbent (best so far) solution and corresponding objective function value, respectively.

$x_{\text{cur}}, f_{\text{cur}}$: Current solution and corresponding objective function value, respectively.

k_{cur}: The neighborhood from where VND starts.

$x_{\text{next}}, f_{\text{next}}$: Solution and corresponding objective function value in inner loop.

5.3 Adaptation of LB and VNS-LB to the Gate Reassignment Problem

In this section, we will describe in detail how Variable Neighborhood Search with Local Branching is appropriately modified so that it is adapted to the problem

Algorithm 6 The Variable Neighborhood Search With Local Branching Procedure (Hansen et al., 2006)

procedure VNSBRANCHING(T^{TL}, N^{TL}, k_{step}, x^{opt})
    $TL \leftarrow T^{TL}$; $UB \leftarrow \infty$; First $\leftarrow$ $textrmTrue$
    stat $\leftarrow$ MIPSOLVE(TL, UB, First, $x_{\text{opt}}, f_{\text{opt}}$)
    $x_{\text{cur}} \leftarrow x_{\text{opt}}$; $f_{\text{cur}} \leftarrow f_{\text{opt}}$
    while `elapsed_time` $< T^{TL}$ **do**
        cont $\leftarrow$ True; $rhs \leftarrow 1$; First $\leftarrow$ False
        while cont or `elapsed_time` $< T^{TL}$ **do**
            $TL \leftarrow \min(N^{TL}, T^{TL} -$ `elapsed_time`$)$
            Add local branching constraint $\Delta(x, x_{\text{cur}}) \leq rhs$; $UB \leftarrow f_{\text{cur}}$
            stat $\leftarrow$ MIPSOLVE(TL, UB, First, $x_{\text{next}}, f_{\text{next}}$)
            if stat = `optimal_solution_found` **then**
                Reverse the last local branching constraint into $\Delta(x, x_{\text{cur}}) \geq rhs + 1$
                $x_{\text{cur}} \leftarrow x_{\text{next}}$; $f_{\text{cur}} \leftarrow f_{\text{next}}$; $rhs \leftarrow 1$
            else if stat = `feasible_solution_found` **then**
                Reverse the last local branching constraint into $\Delta(x, x_{cur}) \geq 1$
                $x_{\text{cur}} \leftarrow x_{\text{next}}$; $f_{\text{cur}} \leftarrow f_{\text{next}}$; $rhs \leftarrow 1$
            else if stat = `proven_infeasible` **then**
                Remove last local branching constraint; $rhs \leftarrow rhs + 1$
            else if *stat* = `no_feasible_solution_found` **then**
                cont $\leftarrow$ False
            end if
        end while
        if $f_{\text{cur}} < f_{\text{opt}}$ **then**
            $x_{\text{opt}} \leftarrow x_{\text{cur}}$; $f_{\text{opt}} \leftarrow f_{\text{cur}}$; $k_{\text{cur}} \leftarrow k_{\text{step}}$
        else
            $k_{\text{cur}} \leftarrow k_{\text{cur}} + k_{step}$
        end if
        Remove all added constraints; cont $\leftarrow$ True
        while cont and `elapsed_time` $< T^{TL}$ **do**
            Add constraints $k_{\text{cur}} \leq \Delta(x, x_{\text{opt}}) < rhs + 1$
            $TL \leftarrow T^{TL} -$ `elapsed_time`; $UB \leftarrow \infty$; First $\leftarrow$ True
            stat $\leftarrow$ MIPSOLVE(TL, UB, First, $x_{\text{cur}}, f_{\text{cur}}$)
            Remove last two added constraints; cont $\leftarrow$ False
            if stat $\leftarrow$ `proven_infeasible` or `no_feasible_solution_found` **then**
                cont $\leftarrow$ True; $k_{\text{cur}} \leftarrow k_{\text{cur}} + k_{step}$
            end if
        end while
    end while
end procedure

of gate reassignment. In addition, we implement the Local Branching heuristic, since both techniques are centered around the concept of a solution neighborhood.

Therefore, before we implement these techniques, we have to determine a proper way to define the "neighborhood" of a solution. In particular, we are interested in two neighborhood properties, i.e., the structure and the size limits. Afterwards, we proceed by identifying the main parameters which define the search procedure in each algorithm. We will calibrate the algorithm by determining the most suitable values of each of these parameters before we implement it for solving large-size problems (section 5.5).

5.3.1 Prior to Implementing

As explained in sections 5.2.1 and 5.2.2, to embed the notion of neighborhood within an MIP formulation, we use a local branching constraint which limits the Hamming distance between each solution of the neighborhood and the current solution, within some bound k. In the original version of LB Fischetti and Lodi (2003), the value of k is fixed and selected according to the problem. In VNS-LB, the values of k define the order in which the neighborhoods of a specific solution are explored, with the search procedure starting from $k = 1$ and increasing to $k = 2$, $k = 3, \ldots$, etc., if no neighborhood change has been performed.

The proper selection of k is one of the major implementation issues investigated in this study. Therefore, prior to applying LB and VNS-LB, we have to answer the following questions, from the perspective of the problem in hand:

1. What is the neighborhood of a solution?

2. For each neighborhood definition, how should we define the limits of the parameters that control the size of the neighborhood?

3. Based on our answers to the previous two questions, what are the alternative ways to formulate the local branching constraints?

4. Which combinations of parameters give the best results?

Questions 1-3 can be answered by examining the properties of the problem combined with the formulation of the local branching constraints, as will be shown in section 5.3.1.1. For question 4, a systematic calibration and validation of the algorithms using different parameter combinations is required. Details on the procedure of parameter combinations follow in section 5.5

5.3.1.1 Defining a Neighborhood and Its Size Limits

To better adapt the VNS and VNS-LB algorithms to the gate reassignment problem, we experiment with alternative neighborhood definitions by using our knowledge about a) the practical aspects of the problem, and b) the properties of the MIP time-index assignment formulation.

From a practical perspective, the key features of the problem are incorporated in the objective function, which is a weighted sum of missed and successful connection cost, as well as of the assignment cost, with the latter expressed as a function of the temporal and spatial deviation from the planned schedule. Therefore, apart from using the number of binary variables with different values, we additionally propose the following four alternative ways to define the distance between two gate assignment solutions:

1. Number of binary variables with a different value between the solutions.

2. Number of flights with a gate change.

3. Number of flights with a time change.

4. Number of flights with either a gate or a time change.

5. Number of connections with a different outcome, i.e., connections which are successful in one solution but fail in the other, and vice versa.

While it is simple to formulate the left-hand side of the local branching constraints when using neighborhood definitions 1-4, using the transfer-based definition 5 is less straightforward. Simply put, we want to use the decision variables of the problem to create an expression that is translated as "the number of connections which have a different outcome in the two solutions". Based on the results from Chapter 4, we use the aggregating formulation `A-AS3`, which expresses the total connection cost of each flight i assigned to gate j in time window k as the value of a continuous variable ξ_{ijk}. However, formulation `A-AS3` does not explicitly include information on the outcome (success or failure) of a transfer (i, i'), while the success or failure of a connections cannot be directly assessed by evaluating the values of ξ_{ijk} variables.

To tackle this issue, we use the linearized quadratic mathematical formulation `Q-FA` (Chapter 4) and define a new set of variables, Z^G. It is reminded that Z^G was defined as follows:

$$Z^G_{ii'jk} = \begin{cases} 1, & \text{if connection } (i,i') \text{ is missed, and } X_{ijk} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{5.9}$$

We also specify the following sets: Let T_S be the set of successful transfers in the current solution x_1, and T_F the set of failed transfers in x_1. The local branching constraint which imposes that the number of transfers with a different outcome than the one in solution x_1 should not exceed k can be formulated as follows:

$$\sum_{(i,i') \in T_S} \sum_{j \in G_i} \sum_{k \in W_i} Z^G_{ii'jk} + \sum_{(i,i') \in T_F} \left(1 - \sum_{j \in G_i} \sum_{k \in W_i} Z^G_{ii'jk} \right) \leq k \tag{5.10}$$

To ensure the validity of the formulation, we additionally include the linearization constraints:

$$Z^G_{ii'jk} \leq X_{ijk}, (i,i') \in T, j \in G_i, k \in W_i \tag{5.11}$$

$$Z^G_{ii'jk} \leq \sum_{(j,j',k,k') \in Q^F_{ii'}} X_{i'j'k'}, (i,i') \in T, j \in G_i, k \in W_i \tag{5.12}$$

$$Z^G_{ii'jk} \geq X_{ijk} + \sum_{(j,j',k,k') \in Q^F_{ii'}} X_{i'j'k'} - 1, (i,i') \in T, j \in G_i, k \in W_i \tag{5.13}$$

Obviously, every definition of the distance between solutions corresponds to a neighborhood structure: For example, if the distance is defined as the number of flights with a gate change, then the k-neighborhood of solution x contains all solutions where at most k flights are assigned to a different gate than they were assigned to in x. Respectively, each neighborhood structure is associated with the minimum and maximum value of k in the right-hand side of the local branching constraint. In theory, the maximum value could be set to infinity, or at least equal to the number of binary variables in the formulation. However, we can limit the value of k to prevent redundant neighborhood exploration by using the set partitioning flight constraints:

$$\sum_{j \in G} \sum_{k \in W_i} X_{ijk} = 1, \ i \in F \tag{5.14}$$

Because of the flight constraints, we know that, out of all variables X_{ijk} which are associated with a flight i, exactly one is equal to 1, while all of the others are 0. When either the assigned gate or the time of flight i are changed, the the values of exactly two variables change: The variable X_{ijk} that was previously equal to 1 drops to 0, while some other variable, which was previously equal to 0 and corresponds to the new gate-time combination, receives a value of 1 in the new solution. All other X_{ijk} variables associated with flight i remain zero. Therefore, two solutions cannot differ in more than two variables for each flight. Based on that, we can make the following propositions regarding the value of k:

1. Two different solutions have at least two different variables.

2. Two different solutions have at most $2\times \mid F \mid$ different variables, where $\mid F \mid$ represents the total number of flights considered.

Using a similar reasoning, we can define the respective bounds for the remaining neighborhood definitions, as shown in Table 5.1. We should highlight here that the values provided for the minimum difference between two solutions, according to the definition of the neighborhood, are defined in the context of the algorithm with the purpose to guide the optimization procedure using different neighborhood definitions. The fact that two solutions are different from each other does not necessarily imply that the distance between them is necessarily greater than or equal to the minimum values calculated based on Table 5.1 for all distance metrics. For example, if we change the gate of a flight and keep everything else the same, then the distance between the previous and the new solution measured in terms of time changes, for example, will be equal to 0 (not 1).

Table 5.1: Minimum and Maximum Distance Between Solutions ($\mid F \mid$ = Total Number of Flights, $\mid T \mid$ = Total Number of Transfers).

Neighborhood	**Minimum Distance**	**Maximum Distance**
Variables	2	$2\mid F \mid$
Gates	1	$\mid F \mid$
Time	1	$\mid F \mid$
Gate or Time	1	$2\mid F \mid$
Transfers	1	$\mid T \mid$

5.3.2 Additional Parameters

As can be seen in Algorithm 6, the user-defined parameters of VNS-LB are the total time limit T^{TL}, the time limit allocated to each sub-problem, N^{TL}, and the incremental change in the neighborhood size in every iteration of the `SHAKE` function, k_{step}. For our application, we also introduce a number of additional parameters, which are not calibrated in the original version of the algorithm either because they have a predetermined value or because they are not considered at all.

The first set of parameters we introduce is based on our observations about the neighborhood size and the maximum and minimum possible values of distance presented in section 5.3.1.1 (Table 5.1).

NeighborhoodType:	The neighborhood structure chosen (1 - 5).
k^p_{min}:	Minimum possible value (percentage) of the right-hand side constant k in the VND loop.
k^p_{step}:	Increase (percentage) of the right-hand side constant k in the VND loop.
k^p_{max}:	Maximum possible value (percentage) of the right-hand side constant k in the VND loop.
rhs^p_{min}:	Minimum possible value (percentage) of the right-hand side constant rhs in the Shake function.
rhs^p_{step}:	Increase (percentage) of the right-hand side constant rhs in the Shake function.
rhs^p_{max}:	Maximum possible value (percentage) of the right-hand side constant rhs in the Shake function.

To account for problems of different sizes, with different numbers of flights and transfers, we do not calibrate directly the absolute maximum, minimum, and step values of the right-hand side of the local branching constraints. Instead, we calibrate the percentage of flights, transfers, or gate/time changes, and multiply them with the total number of flights, transfers, or gate/time changes, respectively, according to the value of NeighborhoodType.

The second set of parameters we introduce is based on our observations about the performance of VNS-LB throughout a set of preliminary experiments. In the original version of the algorithm (6), both in the initialization phase (line 3) and in the SHAKE phase (line 32), the solver returns the first feasible solution. To take advantage of the speed and efficiency of the solver, we allow it to continue the branch-and-cut procedure for a few more seconds, instead of terminating after finding a random solution. At the same time, to maintain the required level of randomness, we impose a time limit on the sub-problem solution, which is different from the time limit defined for the VND loop. The goal is to strike a balance between the solver's ability to converge to local optima, and the randomness needed for diversification and effective exploration of the solution space. The parameters we introduce based on these observations are the following:

- $\mathrm{First}_{\mathrm{I}}$: Input for the MIP solver in the initialization phase. If True, the solver returns the first feasible solution found. If False, the solver continues until the time limit T_I^{TL} is reached.
- $\mathrm{First}_{\mathrm{S}}$: Input for the MIP solver in the SHAKE phase. If True, the solver returns the first feasible solution found. If False, the solver continues until the time limit T_S^{TL} is reached.
- T_I^{TL}: Solver time limit in the initialization phase. Used if $\mathrm{First}_{\mathrm{I}} = 0$.
- T_S^{TL}: Solver time limit in the SHAKE phase. Used if $\mathrm{First}_{\mathrm{S}} = 0$

As will be shown in section 5.5.1.3, parameters $k^p_{\min}$, $k^p_{\max}$, and k^p_{step} are introduced in the implementation of the Local Branching algorithm as well.

5.4 MIP Formulation for Baggage Connections

The proposed solution algorithms will be applied in a set of experimental cases with the objective to minimize the total reassignment cost. For details on the various cost components and measures of effectiveness used in gate reassignment problems, the reader shall refer to Chapter 3. In this section, we will also introduce a new measure of effectiveness, i.e., the cost of failed baggage connections.

5.4.1 Delayed Baggage as a Measure of Effectiveness

As explained in Chapter 3, scheduled transfers satisfy a minimum connection time (MCT) between the arrival of the inbound flight and the departure of the outbound flight. However, in case of schedule disruptions, such as a delay in the inbound flight, the actual available connection time may not be sufficient for passengers.

However, apart from transporting passengers, airlines are also responsible for the transportation of baggage. The safe and timely delivery of baggage to the destination airport is one of the most important handling procedures since it directly affects the provided level of service. In fact, the Office of Aviation Enforcement and Proceedings of the US Department of Transportation ranked baggage mishandling as the third most common passenger complaint in 2017, following only flight problems and high fares (US DOT, 2018).

To better understand how the problem of missed baggage connections is associated with the gate assignment problem, we have to explain the general structure of baggage handling for connecting flights, as described in Barth (2013): Upon the arrival of the inbound flight, baggage is unloaded from the aircraft and transported with vehicles to the so-called infeed area, where the bag tags are scanned. Then, according to the type of the outbound flight, security screening might follow, and the bags are transported through the baggage handling system to the respective handling facilities of the outbound flights, where they are loaded to containers and carried to the aircraft of the departing flight. The whole procedure must be finished before a given time threshold in advance to the departure time. In the case of self-transfers, passengers have to collect their baggage themselves and check them in at the departing flight.

However, when the available time between the connecting flights is significantly short, the baggage might not be transported and loaded to the aircraft of the departing flight on time. This is a result of long required baggage processing time and large transportation distances and can further be associated with the service period of the baggage station serving the departing flight. In practice, the service period for an outbound flight, during which the baggage may be processed before it is loaded in the aircraft, ends t_b minutes before the departure of the flight. After the end of the service period, baggage for that flight cannot be processed.

As will be shown in section 5.4.2, these restrictions can be modeled similarly to passenger connections in Chapter 3.

Current research on baggage handling systems examines the optimal assignment of sorting stations to gates or gate piers: For outbound flights, Abdelghany et al. (2006) developed an algorithm to determine the assignment of flights to airport piers so that the utilization of piers is optimized, while Frey et al. (2017) minimized workload peaks using a decomposition method to handle efficiently the symmetries of the problem. Huang et al. (2016) used a Stochastic Vector Assignment Problem to find the optimal assignment of unloading zoned (chutes) to outgoing flights. Regarding transfer baggage, Barth (2013) developed a MIP model to minimize the number of missed bags, while Clausen and Pisinger (2010) developed a binary integer model for the optimal assignment of baggage to sorting stations so that the number of undelivered baggage pieces in short transfers is minimized, considering the capacity of baggage delivery vehicles.

All of the aforementioned studies examine the planning phase of the problem and do not consider alternative strategies to handle schedule disruptions. Therefore, in this chapter, we bridge the gap between real-time gate assignment and baggage handling optimization, by modifying our gate reassignment model so that it considers not only the cost of missed passenger connections, but also the cost of missed baggage connections. To achieve this, the model takes into account the time required for baggage processing and transportation, given the available vehicles and the distance between the gates of every pair of connecting flights. The main idea of the model is that, due to the difference between passenger and baggage transfer procedures, a successful passenger connection does not imply a successful baggage connection as well.

5.4.2 Modeling Delayed Baggage in an Assignment-Based Formulation

To model baggage connections, we follow a procedure very similar to the one described in Chapter 3 for passenger connections. We will using the indices “P” and “B” for measures referring to passengers and baggage, respectively.

The required time $t_{jj}^{B,req}$ for transferring baggage between gates j and j' includes the time allocated to the above procedures and depends on the distance between the infeed area of gate j and the sorting station allocated to gate j', as well as the time required to screen and transport the baggage under the baggage handling system. On the other hand, the available time $t_{ii'}^{B,av}$ depends on the time between the arrival and departure of the inbound and outbound flights, but also on the time threshold t^b which determines the minimum buffer time between the end of the service period for flight i' and the departure time of flight i'. Therefore, we can define a similar condition to 3.1 and assume that if flight i is assigned to gate j and time window k, and flight i' is connected to gate j' and time window k', the transfer baggage of connection (i, i') will not make the connection if and only if

$$t_{jj'}^{B,req} > k' - t_b - k \tag{5.15}$$

Based on condition 5.15, we form sets $A^B(i, i')$ and $F^B(i, i')$, which contain all combinations (j, j', k, k') resulting in success and failure, respectively, of each baggage connection (i, i').

In this study, we assume that $F^B(i,i') \subset F^A(i,i')$, which means that all elements (j,j',k,k') of $F^B(i,i')$ correspond to successful passenger connections, to reflect the situation where a passenger boards the departing flight on time but their baggage is left behind. We assume that the reverse situation i.e., successful baggage but failed passenger connection, is not possible, due to the PPBM (Positive Passenger Bag Matching) regulation which prevents the baggage from being transported without their owner boarding the plane. In addition, when both the passenger and the baggage are left behind, only the cost of the missed passenger connection is added to the total cost. The case of failed baggage connections will also be referred to as "missed baggage connections" or "misconnected baggage".

5.4.3 Model Formulation

Sets:

F: Set of flights.

F^A: Set of arriving flights.

G: Set of gates.

T: Set of passenger connections.

W: Set of time windows.

$W_i \subset W$: Set of time windows that are compatible with flight i.

$G_i \subset G$: Set of gates that are compatible with flight i.

F_k^O: Set of flights that may occupy a gate at time window k.

F_j^G: Set of flights that can be assigned to gate j.

G_j^N: Set of gates adjacent to gate j.

G^L: Set of large gates.

H_{is}: Set of time windows such that, if flight i is assigned to them, it occupies its gate at time window s.

Costs:

C_{ijk}^{FS}: Cost of assigning flight i to gate j at time window k.

$C_{ijki'j'k'}^{\xi}$: Connection cost of transfer $(i, i') \in T$, given that flight i is assigned to gate j and time window k, and flight i' is assigned to gate j' and time window k'.

Parameters:

UB_{ijk}: The estimated upper bound on the total connection cost for all connections of flight $i \in F^A$, if it is assigned to gate j at time window k.

Decision Variables:

X_{ijk}: Binary, equal to 1 if flight i is assigned to gate j at time window k, 0 otherwise.

ξ_{ijk}: Continuous, equal to the total connection cost of all connections with flight i as the arriving flight, given that flight i is assigned to gate j and time window k.

The problem is formulated using the aggregating formulation `A-AS3` (Chapter 4) as follows:

Minimize:

$$\sum_{i \in F} \sum_{j \in G_i} \sum_{k \in W_i} C^{FS}_{ijk} X_{ijk} + \qquad (5.16)$$

$$\sum_{i \in F} \sum_{j \in G_i} \sum_{k \in W_i} \xi_{ijk} \qquad (5.17)$$

Subject to:

$$\sum_{j \in G} \sum_{k \in W_i} X_{ijk} = 1,\ i \in F \tag{5.18}$$

$$\sum_{i \in F_k^O \cap F_j^G} \sum_{k \in H_{is}} X_{ijk} \leq 1,\ j \in G,\ s \in W \tag{5.19}$$

$$\sum_{k \in H_{is}} X_{ijk} + \sum_{k' \in H_{i's}} X_{i'j'k'} \leq 1, j \in G^L, j' \in G_j^N, i \in G_i, i' \in G_{i'}, s \in W \tag{5.20}$$

$$\xi_{ijk} \geq \sum_{\substack{i' \in F^D: \\ (i,i') \in T}} \sum_{j' \in G_{i'}} \sum_{k' \in W_{i'}} C^S_{ijki'j'k'} X_{i'j'k'} - UB_{ijk}(1 - X_{ijk}), i \in F^A, j \in G_i, k \in W_i \tag{5.21}$$

$$\xi_{ijk} \geq 0,\ i \in F^A,\ j \in G_i,\ k \in W_i \tag{5.22}$$

$$X_{ijk} \in 0, 1, i \in F, j \in G_i, k \in W_i \tag{5.23}$$

The objective function of the problem is to minimize the total cost, which includes the sum of the individual assignment costs 5.16 and connection costs 5.17. The assignment cost is a function of additional flight delays and gate changes, while the connection cost accounts for missed passenger and baggage connections.

Constraint 5.18 is the flight constraint, which ensures that every flight is assigned to one gate and time window, while constraint 5.19 is the gate constraint, which ensures that a gate cannot be occupied concurrently by more than one aircraft. Constraint 5.20 is the shadow constraint, which prevents the concurrent occupation of adjacent gates by aircraft with a long wingspan. Constraints 5.21 and 5.22 define the aggregating variable ξ_{ijk} as a function of the respective binary variables X_{ijk} , while constraint 5.23 defines the main binary decision variable ξ_{ijk} . Constraints 5.21 and 5.22 define the continuous ξ variables, as in Chapter 4.

To calculate the cost coefficients C^{FS}_{ijk}, $C^{\xi}_{ijki'j'k'}$, we need the following parameters and sets:

Sets:

t_i^A: Arrival / departure time of flight i according to the updated delay information.

g_i^B: Gate where flight i was assigned to in the planned schedule.

P_i: Number of passengers in flight i.

$N_{ii'}$: Number of passengers transferring from flight i to flight i'.

$B_{ii'}$: Number of baggage pieces to be transferred from flight i to flight i'.

C^B: Cost of failed baggage connection ($/piece).

C^M: Cost of missed passenger connection ($/passenger).

C^G: Operational cost for a gate change ($/flight).

C_t^{GD}: Passenger inconvenience cost for a gate change of a departing flight, if the departure time is t minutes after the start of the reassignment window ($/passenger).

C^T: Delay cost ($/passenger/minute).

t_{step}: The duration of the elementary time windows in which we divide the planning horizon.

$F^P(i,i')$: Set of combinations (j,j',k,k') of gate and time assignments for connecting flights i and i', that result in the failure of passenger connection (i,i').

$F^B(i,i')$: Set of combinations (j,j',k,k') of gate and time assignments for connecting flights i and i', that result in the success of passenger connection (i,i'), but in the failure of the respective baggage connection.

Therefore, the cost coefficients in 5.16 and 5.17 are calculated as follows:

Delay cost:

$$C_{ijk}^{T} = C_{ijk}^{T}\ t_{step} \max\{0, k - t_i^A\}, i \in F, j \in G_i, k \in W_i \tag{5.24}$$

Gate change cost:

$$C_{ijk}^{G} = \begin{cases} 0, \text{ if } j = g_i^b \\ \\ C^G, \text{ if } i \in F^A \text{ and } j \neq g_i^b \\ \\ C^G + C_k^{GD}, \text{ otherwise} \end{cases} \tag{5.25}$$

Missed passenger connection cost:

$$C_{ijki'j'k'}^{P} = \begin{cases} C^M N_{ii'} \text{ if } (j, j', k, k') \in F^P(i, i') \\ \\ 0, \text{ otherwise} \end{cases} \tag{5.26}$$

Missed baggage connection cost:

$$C_{ijki'j'k'}^{B} = \begin{cases} C^B B_{ii'} \text{ if } (j, j', k, k') \in F^B(i, i') \\ \\ 0, \text{ otherwise} \end{cases} \tag{5.27}$$

Consequently, the assignment coefficients in 5.16 are equal to

$$C_{ijk}^{FS} = C_{ijk}^{T} + C_{ijk}^{G},\ i \in F,\ j \in G_i,\ k \in W_i \tag{5.28}$$

while the connection cost coefficients $C^{\xi}_{ijki'j'k'}$ in 5.17 are calculated as

$$C^{\xi}_{ijki'j'k'} = \begin{cases} C^{P}_{ijki'j'k'}, \text{ if } (j, j', k, k') \in F^{P}(i, i') \\ C^{B}_{ijki'j'k'}, \text{ if } (j, j', k, k') \in F^{B}(i, i') \\ 0, \text{ otherwise} \end{cases} \tag{5.29}$$

5.5 Numerical Experiments

In this section, we will review the procedure followed for the implementation of LB and VNS-LB and for the evaluation of the solutions produced. In summary, every set of experiments consists of three main steps: a) Parameter Calibration, b) Parameter Validation (Testing), and c) Sensitivity Analysis.

5.5.1 Calibration and Validation

5.5.1.1 Procedure Overview

For both Local Branching and Variable Neighborhood Search with Local Branching, we follow a similar fine-tuning procedure to determine the combination of parameters that yields the best results and is therefore more likely to produce near-optimal solutions. The procedure consists of two main steps: Calibration and validation (testing).

We first generate two discrete sets of experimental cases, such that the MIP solver (in this case, Gurobi) requires a relatively long time to produce a provably optimal solution; the term “relatively long” refers to cases that are solved in an amount of time that is too long for real-time schedule updates, but short enough so that we know the final optimal solution within the time frame dedicated to our experiments. In this context, case studies with running time between 30 and 180 minutes were selected. In practice, the case studies were of medium size, i.e., for an airport with 50 gates, hourly flight rate (inbound plus outbound) ranging between 30 and 50 flights/hour, and a planning horizon from 4 to 8 hours. The case studies produced are then divided in two sets, one for calibration (models MC1-MC5) and one for testing (models MT1-MT5). Details on the features of each case study are shown in Table 5.2.

Table 5.2: Case Studies Used For Calibration and Testing

Model Name	**Flights**	**Gates**	**Time Windows**	**Variables**	**Constraints**
MC1	200	50	25	11115	5890
MC2	250	50	39	14835	9700
MC3	300	50	40	16935	9260
MC4	275	50	34	15495	8395
MC5	360	45	51	17940	10663
MT1	200	50	30	11895	7700
MT2	200	50	33	11865	7850
MT3	330	60	41	22986	13446
MT4	392	52	50	18636	12400
MT5	416	52	53	19764	13156

Model Name	**Binary Variables**	**Optimal Objective**	**Solver Time**	**Solver Gap**	**Best Heuristic Bound**
MC1	8895	1399090	7201.26	7%	1300131
MC2	11085	1436460	2857.53	0%	1436460
MC3	13455	1579840	5341.37	0%	1579840
MC4	12285	443890	10800.64	0%	443369
MC5	13936	34200	10800.34	2%	33633
MT1	8895	1151590	5147.5	0	1151590
MT2	8865	1157720	2341.85	0	1157720
MT3	17658	38520	10800.32	1%	38324
MT4	13932	52280	7559.69	0%	52276
MT5	14772	43750	1995.26	0%	43746

In the calibration procedure, we test each case study of the calibration set for every parameter combination, so that we select the combinations that return the best solutions. To quantify the quality of the best solution found by the algorithm (VNS or VNS-LB), we calculate the optimality gap as follows: Let $x^{\star}$ be the optimal solution as determined by the MIP solver, and x^{U} the best solution yielded by the algorithm within the time limit selected. The optimality gap is calculated as follows:

$$gap\ (\%) = \frac{x^{U} - x^{\star}}{x^{\star}} \times 100 \tag{5.30}$$

For every parameter combination, we then calculate the average gap across all case studies.

In addition, to ensure that the selected parameter combinations perform well compared to the rest of the candidate combinations, we order the combinations for each case study in increasing order based on the *gap* value and determine the relevant position of each combination. Then, the average ranking for each combination across all experiments is calculated.

We then sort the candidate combinations based on the values of their average gap and average ranking, and select the combinations that do well in both rankings, and that also consistently give a small gap (e.g., less than 5%) across all experiments.

In the testing procedure, we apply the selected parameter combinations to the test set to verify that the algorithm produces results that satisfy a predetermined optimality gap in the test set as well.

The performance of the algorithms and the selected parameter combinations is further analyzed through a series of sensitivity analysis experiments, which verify the quality of the solution with regard to the expected results, based on various changes in the input of the model.

5.5.1.2 Calibration and Testing for Variable Neighborhood Search with Local Branching

As shown in Algorithm 6, in its original form proposed by Mladenović and Hansen (1997), VNS-LB requires three parameters: Total time limit T^{TL}, node time limit N^{TL}, and step parameter k_{step}. In the modified version we propose, we also include (section 5.3.2) the structure of the neighborhood NEIGHBORHOODTYPE, as well as minimum and maximum values for the right-hand side of the local branching constraint, $k^p_{\min}$, $k^p_{\max}$, $rhs^p_{\min}$, $rhs^p_{\max}$. We also specify the values of $\mathrm{First_I}$ and $\mathrm{First_S}$ to impose a termination criterion in the solution of the subproblems, well as T_I^{TL}, T_S^{TL} to determine their time limit.

Since trying all combinations of the above parameters would require a time consuming exhaustive search, we choose 48 of them to participate in the experiments. First, a set of preliminary experiments is performed to further narrow down the number of combinations. The final ten combinations are shown in Table 5.3. Whenever the value of is shown as "Default" or simply"D", it is implied that the minimum and maximum possible values were used, based on Table 5.1.

In all of the ten best combinations, a variable-based definition is used. These are the combinations that participate in the calibration procedure (paragraph 5.5.1.1). The results are shown in detail in Table A11 of the Appendix, while the mean optimality gap and the mean ranking of each combination are shown in Table 5.4. Figure 5.1 shows the box-and-whisker plots for each of the ten parameter combinations. Based on the results, we proceed for the rest of this study with parameter combination 33, which presents the smallest average gap (0.7%) and consistently outperforms the other combinations (average ranking 2.8, followed immediately by combination 37 following with 3.8). The parameter values for combination 33 are $k^p_{\text{step}} = 0.25, N^{TL} = N^{TL}_I = N^{TL}_S = 20$ seconds, $\text{First}_I = \text{First}_S =$ False, $k^p_{\text{min}} = k^p_{\text{max}} = rhs^p_{\text{min}} = rhs^p_{\text{max}} =$ "Default".

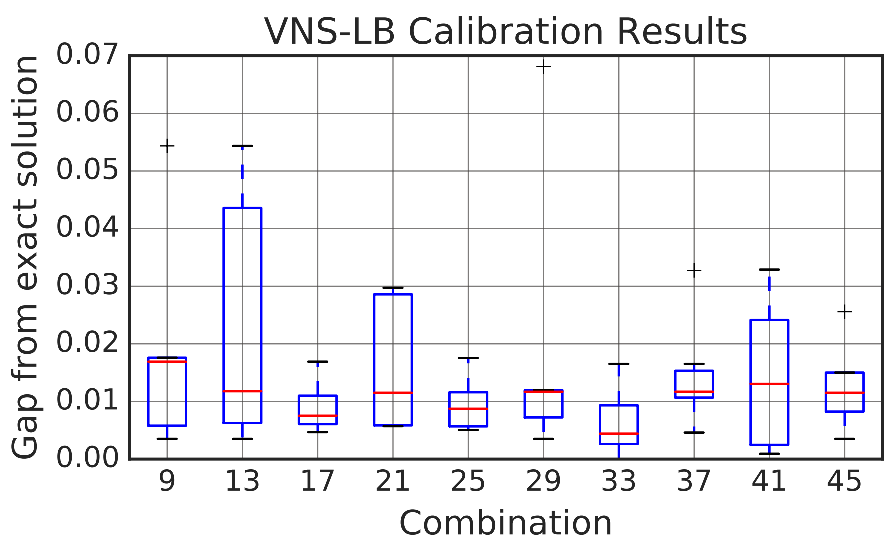

Figure 5.1: Box-and-Whisker plots for the calibration experiments of VNS-LB.

Table 5.3: Parameter Combinations For Calibration of Variable Neighborhood Search With Local Branching

Name	$\boldsymbol{k^{given}_{\mathbf{min}}}$	$\boldsymbol{k^{given}_{\text{step}}}$	$\boldsymbol{k^{given}_{\mathbf{max}}}$	$\boldsymbol{rhs^{given}_{\mathbf{min}}}$	$\boldsymbol{rhs^{given}_{\mathbf{step}}}$	$\boldsymbol{rhs^{given}_{\mathbf{max}}}$	$\boldsymbol{N^{TL}}$	$\boldsymbol{N^{TL}_I}$	**First**$_{\mathbf{I}}$	**Neighb.** Type	**First**$_S$	$\boldsymbol{N^{TL}_S}$
9	D	0.02	D	D	D	D	20	20	FALSE	Variables	FALSE	20
13	D	0.02	D	D	D	D	40	20	FALSE	Variables	FALSE	20
17	D	0.05	D	D	D	D	20	20	FALSE	Variables	FALSE	20
21	D	0.05	D	D	D	D	40	20	FALSE	Variables	FALSE	20
25	D	0.1	D	D	D	D	20	20	FALSE	Variables	FALSE	20
29	D	0.1	D	D	D	D	40	20	FALSE	Variables	FALSE	20
33	D	0.25	D	D	D	D	20	20	FALSE	Variables	FALSE	20
37	D	0.25	D	D	D	D	40	20	FALSE	Variables	FALSE	20
41	D	0.4	D	D	D	D	20	20	FALSE	Variables	FALSE	20
45	D	0.4	D	D	D	D	40	20	FALSE	Variables	FALSE	20

Table 5.4: Average Gap and Ranking For VNS-LB Combinations

Combination	**Average Gap**	**Average Ranking**
9	2.0%	5.6
13	2.4%	6.8
17	0.9%	3.8
21	1.6%	5
25	1.0%	4.2
29	2.0%	5.8
33	0.7%	2.8
37	1.5%	4.8
41	1.7%	5.2
45	1.3%	4.4

The selected combination is then tested in a separate set of five experiments to make sure that it performs well in previously unseen experimental cases. The testing procedure (Table 5.5) confirms the suitability of combination 33, which yields a consistently small (less than 5%) gap from the optimal solution for all cases examined.

5.5.1.3 Calibration and Testing for Local Branching

As shown in Algorithm 1, the parameters that have to be determined for the implementation of Local Branching according to Fischetti and Lodi (2003) are the total time limit T^{TL}, the node time limit N^{TL}, the right-hand side value k, and the maximum number of diversifications $dv_{\max}$. As explained in section 5.3.2, we also consider the neighborhood type NeighborhoodType, as well as the minimum and maximum right-hand side values $k^p_{\min}$, $k^p_{\max}$.

Table 5.5: Testing Results for Selected VNS-LB Combination 33

Model	**Gap**
MT1	0.5%
MT2	0.3%
MT3	2.7%
MT4	4.7%
MT5	3.0%

Similarly to the procedure followed for the calibration of VNS, we initially generate 48 different combinations of the above parameters. The next step is to carry out a set of preliminary experiments to find the most promising combinations which advance to the calibration phase. Using the preliminary experiments, we limit the number of parameter combinations to 10. The selected combinations are shown in Table 5.6.

We then apply the calibration procedure 5.5.1.1 to the selected ten combinations. The box-and-whisker plot for each combination is shown in Figure 5.2, while Table 5.7 shows the mean optimality gap and the mean ranking of each combination. The results are shown in detail in Table A12 of the Appendix.

For the rest of our experiments, we select combination 45, since the difference from combination 35 is negligible with regards to the average gap, but more significant in terms of the average ranking. For combination 45, the parameter values are $\textsc{NeighborhoodType}$ = 'Variables', $k^p_{\min} = k^p_{\max}$ = Default, $dv_{\max} = 10$, and $N^{TL} = 60$ seconds.

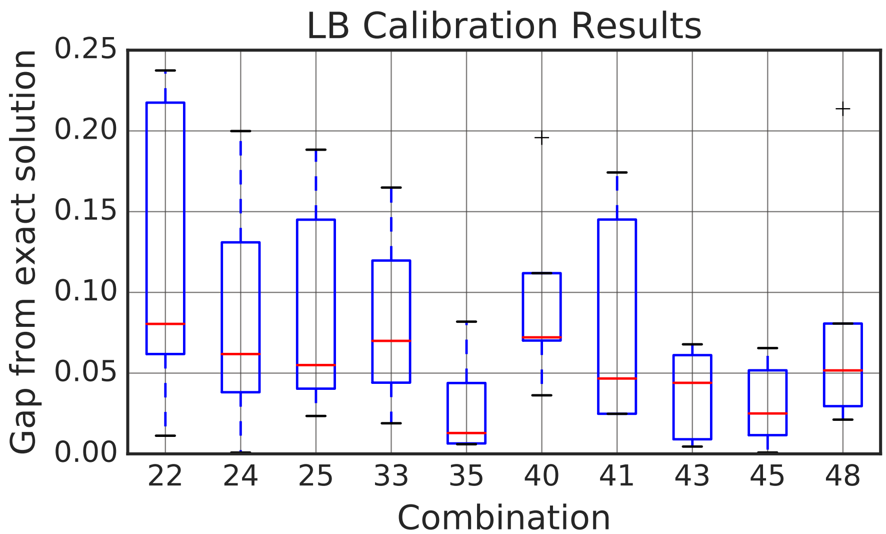

Figure 5.2: Box-and-Whisker plots for the calibration experiments of LB.

In the testing procedure (Table 5.8), the selected combination results in a relatively small optimality gap for most case studies of the testing set - although the gap is larger than 12% for the two last case studies.

5.5.1.4 Comparison of VNS-LB with LB in the Calibration Phase

In terms of the characteristic properties of the combinations that perform best, the selected ten combinations of LB present larger variability compared to the selected combinations of VNS-LB, with six of them (combinations 25, 33, 35, 41, 43, 45) having a variable-based neighborhood definition and four of them (combinations 22, 24, 40, and 48) a transfer-based definition. Overall, we also observe that connections with transfer-based neighborhood definition perform well for relatively high node time limit (three out of four require 2 minutes allocated to each node), while variable-based neighborhood definitions perform better for a lower time limit, i.e., 20-60 seconds.

Table 5.6: Parameter Combinations For Calibration of Local Branching

Name	$\boldsymbol{k^{given}}$	$\boldsymbol{k^{given}_{\min}}$	$\boldsymbol{k^{given}_{\max}}$	**Node Time Limit**	**Neighb. Type**	$\boldsymbol{dv_{\max}}$
22	0.05	Default	Default	60	Transfers	10
24	0.05	Default	Default	120	Transfers	10
25	0.1	Default	Default	20	Variables	10
33	0.25	Default	Default	20	Variables	10
35	0.25	Default	Default	40	Variables	10
40	0.25	Default	Default	120	Transfers	10
41	0.4	Default	Default	20	Variables	10
43	0.4	Default	Default	40	Variables	10
45	0.4	Default	Default	60	Variables	10
48	0.4	Default	Default	120	Transfers	10

Table 5.7: Average Gap and Ranking For LB Combinations

Combination	**Average Gap**	**Average Ranking**
22	12.2%	8.2
24	8.6%	6.2
25	9.0%	6.8
33	8.4%	6
35	3.0%	2.6
40	9.7%	7.4
41	8.3%	5.8
43	3.7%	3.2
45	3.1%	2.8
48	7.9%	5.4

Table 5.8: Testing Results for Selected LB Combination 45

Model	**Gap**
MT1	1.8%
MT2	0.4%
MT3	4.0%
MT4	11.8%
MT5	12.4%

Furthermore, we can identify significant differences in the performance of the two algorithms. Contrary to VNS-LB, there was no parameter combination for Local Branching that resulted in a consistently small optimality gap (less than 5%). In fact, the most successful combinations were 35 and 45, with a *gap* value less than 7% in the worst case. These combinations present the smallest average optimality gap as well, equal to 3.0% and 3.1% respectively, while they also perform consistently well compared to the other combinations, with average rankings equal to 2.8 and 2.6, respectively. Overall, regarding the selected best combinations (33 for VNS-LB and 45 for VNS), VNS-LB performs better than LB, since it achieves a smaller optimality gap for a time limit of 10 minutes, in 5 out of the 6 calibration experiments (Figure 5.3a) and in all of the testing experiments (Figure 5.3b).

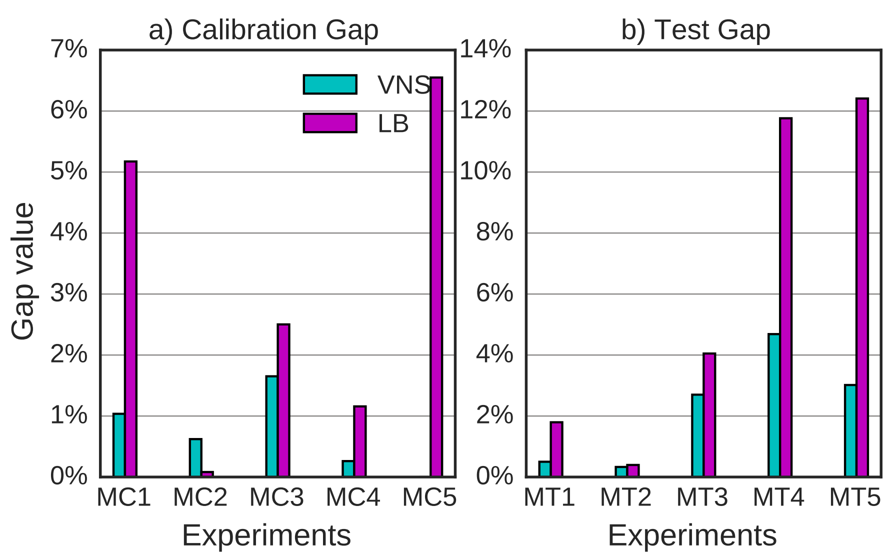

Figure 5.3: Optimality gap for VNS-LB and LB in the calibration (left) and testing (right) phase.

5.5.2 Sensitivity Analysis

The sensitivity analysis procedure consists of a set of experiments which investigate the impact of changes in external parameters ans assumptions, which are used as input to the model, on the optimal solution. We use sensitivity analysis to verify that VNS-LB produces reasonable results in a variety of "what-if" experiments. We also implement local branching to compare the results between the two methodologies.

5.5.2.1 Procedure Overview

In the gate reassignment problem, there exists a variety of parameters that can be used for sensitivity analysis, such as cost components (e.g., the unit cost of misconnected baggage), operating conditions (e.g., the layout of the airport, or the properties of the baggage handling system), or external factors (e.g., the distribution of flight delays). The case studies (Table 5.9) that we generate for the sensitivity analysis procedure are of significantly larger size, compared to the ones that we use for calibration and testing. For the purpose of this study, we construct the following experimental sets to examine the impact of changes to the following conditions:

- S-1: We change the unit cost C^B of failed baggage connections.
- S-2: We change the properties of the baggage handling system BHS, namely the service period and the baggage transportation speed.

Table 5.9: Experimental Cases For Sensitivity Analysis

Set	**Changing Parameter**	**Flights**	**Gates**	**Hours**
S1	Unit cost of misconnected baggage	612	52	12
S2	BHS Properties	600	52	12
S3	Delay Distribution	600	52	12
S4	Airport Layout	400	50	10

- S-3: We examine different delay distributions in terms of a) the probability of disruption occurrence for each flight, and b) the delay duration.

- S-4: We experiment with different terminal layouts.

Similarly to Chapters 3 and 4, for experimental sets S-1, S-2, and S-3, we use the layout of Athens International Airport (AIA, 2018), which includes one main and one smaller, satellite terminal. For experimental set S-4, we test the impact of different layouts, as will be shown in section 5.5.2.6. The values of the basic parameters used in the sensitivity analysis experiments are summarized in Table 5.10.

5.5.2.2 Comparison of VNS-LB with LB in the Sensitivity Analysis Phase

Table 5.11 summarizes the best upper bound of the objective function yielded by Local Branching an by Variable Neighborhood Search with Local Branching for all sensitivity analysis experiments. With the exception of set S-4, VNS-LB consis-

Table 5.10: Basic Parameter Values in Sensitivity Analysis Experiments

	Parameter	Value
Flights	Aircraft load factor limits	(0.5, 1)
	Percentage of connecting passengers	50%
	Gate occupancy duration (minute)	30' for $\leq$150 seats, 40' for $>$ 150 seats
	Number of aircraft types	9
	Number of baggage pieces per passenger	1
Costs	Operational cost for a gate change C^G($/flight)	40
	Passenger inconvenience cost for gate change of departing flight, C_t^{GD} (4$/flight)	Piecewise linear function of available time t, from 260 for $t = 1h$ to 0 for $t = 6h$
	Delay cost, C^T ($/flight/minute).	20
	Cost of failed passenger connection, C^M ($/passenger)	200
	Cost of failed baggage connection, C^B ($/baggage piece)	50
Operator Decisions	Additional holding time range (min)	(-20,50)
	Step size (minutes)	10
Other Values	Delay distribution	Γ (shape = 3, scale =1) for arriving flights, Normal ($\mu = \sigma = 10$) for departing flights
	Passenger walking speed (meters/minute)	70
	Baggage transportation speed / Passenger walking speed	1.4
	Service time threshold for baggage (minutes)	20

Table 5.11: Best Upper Bounds Found By LB and VNS-LB In Sensitivity Analysis

Experimental Set	**Name**	**VNS Best Objective**	**LB Best Objective**	**% Difference**
S-1	S1-1	59367.5	78385	-24%
	S1-2	59795	74410	-20%
	S1-3	62030	72997.5	-15%
	S1-4	64250	79662	-19%
S-2	S2-1	42120	70040	-40%
	S2-2	42120	71760	-41%
	S2-3	47330	63020	-25%
S-3	S3-1	6790	17250	-61%
	S3-2	80130	109930	-27%
	S3-3	284520	323817	-12%
	S3-4	417030	481363	-13%
S-4	S4-1	1342745	1342825	0%
	S4-2	1235554	1236442	0%
	S4-3	1469646	1470840	0%

tently finds better solution than LB, with the difference between them ranging from 12% up to even 61%. This means that LB has failed to reach the optimal solution at least in 3 out of the 4 experimental sets. Nevertheless, for the sake of completeness, we have presented the sensitivity analysis results for both methodologies.

5.5.2.3 Set S-1: Changes in the unit cost C^B of failed baggage connections

In the base case, we assume that, when a connecting passenger's baggage fails to make the transfer, the total objective function cost in increased by \$50 per piece, a value which accounts for the monetary compensation that the passenger is entitled to, as well as for indirect costs due to the decline in the provided level of service. In this set, we experiment with the value of unit delayed baggage cost C^B, changing it from 25% to 150% of the base case value.

Both algorithms present reasonable results (Figure 5.4), with the number of misconnected baggage decreasing with the increase in C^B. For VNS, changing the value of C_B from \$50 to \$12.5 (corresponding to a decrease of 75%) results in a 104% increase in the number of misconnected baggage pieces, while changing the value of C_B to \$75 (a 50% increase) decreases the value of baggage pieces by 14%. For Local Branching, the respective values are 42% and 32%. We also notice that the value of missed baggage is consistently higher in the solutions found by Local Branching, compared to the solutions found by VNS; this indicates that, as was already known from Table 5.11, Local Branching has not reached an optimal solution.

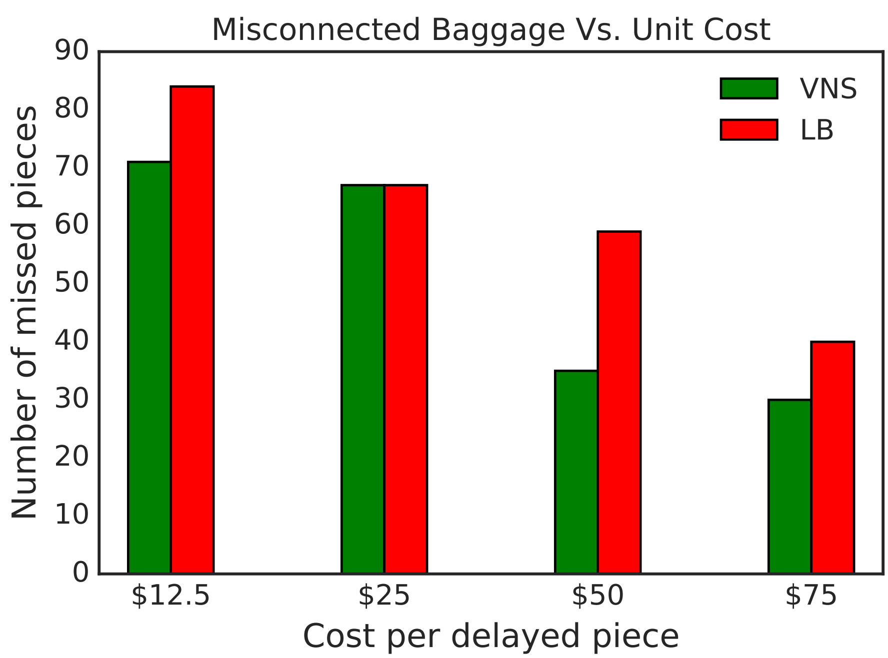

Figure 5.4: Sensitivity analysis: Changes in missed baggage cost.

5.5.2.4 Set S-2: Changes in the features of the Baggage Handling System

In this set, we experiment with the operational features of the baggage handling system. More specifically, we focus on two properties that affect the required baggage transportation time and define the time windows within which the transportation of baggage between gates will be successful: The average speed of the baggage handling system, and the service period (see section 5.4.2). To calculate the average baggage transportation speed, we assume different values of the ratio $r^{PB} = \frac{\text{Passenger speed}}{\text{Baggage speed}}$. For the service period, we assume that, for a flight that departs at time t, the baggage handling station cannot process baggage after $t - t^b$, and we experiment with different values of t^b.

Keeping everything else the same, we produce three experimental cases. In case S2-1, we assume a fast baggage handling system with $r^{PB} = 2$, which remains open until the departure of the outbound flight ($t^b = 0$). In practice, this condition can be used to capture the case where, in case of delayed inbound flights which result in tight connections, the baggage is directly transported and loaded in the aircraft. In case S2-2, we assume a slower BHS with stricter service period thresholds, with $r^{PB} = 1$ and $t^b = 20$ minutes, while in case S2-3 we use an even less efficient BHS, with with $r^{PB} = 0.8$ and $t^b = 40$ minutes.

The results for both algorithms are plotted in Figure 5.5. On the one hand, the solutions of VNS-LB demonstrate an equal cost (\$42,120) between S2-1 and S2-2, and a 12% cost increase (\$47,330) in case S2-3. On the other hand, Local

Branching in this case produces counter-intuitive results, with the total objective cost decreasing from case S2-2 to case S2-3.

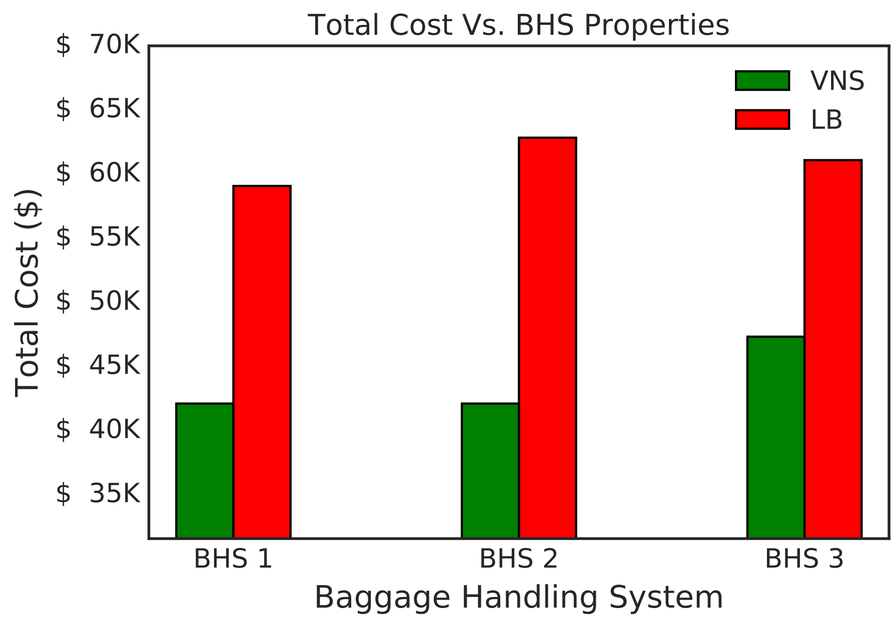

Figure 5.5: Sensitivity analysis: Changes in the baggage handling system.

5.5.2.5 Set S-3: Changes in delay patterns

For our experiments, we assume that the delays encountered by arriving flight follow a Γ distribution, with shape = 3 and scale = 1, while the delays of departing flights are normally distributed with a mean and standard deviation equal to 10 minutes (see Table 5.10. In this set, to quantify the magnitude of schedule disruptions, we use a Bernoulli trial with probability p that a flight will be delayed, and a uniform distribution $\mathcal{U}(l_b, u_b)$ to predict the delay duration, as in section 3.5.4 of Chapter 3. We therefore generate 4 delay scenarios as follows:

- S3-1: Smooth daily operations with delays lower than usual with $p = 0.1$, $l_b = 0$ minutes, $u_b = 20$ minutes.

- S3-2: Normal daily operations with usual delays with $p = 0.3$, $l_b = 10$ minutes, $u_b = 30$ minutes.

- S3-3: Delays longer than usual with $p = 0.5$, $l_b = 20$ minutes, $u_b = 100$ minutes.

- S3-4: Severe schedule disruption with $p = 0.6$, $l_b = 20$ minutes, $u_b = 120$ minutes.

We expect an increase in the total cost as we move from scenario S3-1 to S3-4. Indeed, both algorithms demonstrate an increasing cost trend, as can be seen in Figure 5.6. Compared to the base-case ("normal") scenario, VNS-LB finds a solution with a 92% lower cost in case S3-1 ("Lower delays than normal") and a 420% more expensive solution for the S3-4 scenario ("Extreme level of delays"). The respective values for Local Branching are 84% and 338%. However, as in the previous experimental sets, the solutions found by local branching are of higher cost compared to the ones of Variable Neighborhood Search with Local Branching.

5.5.2.6 Set S-4: Changes in the layout of the airport

In this set, we examine how the objective function changes for various types of airport terminals, and therefore different spatial configuration of the gates. For set S4, we examine three different terminal layouts, as follows:

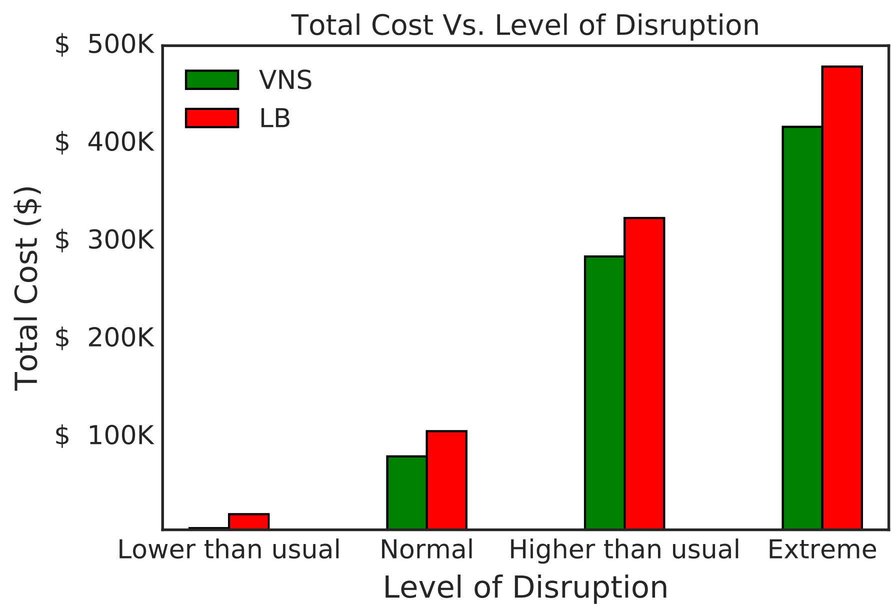

Figure 5.6: Sensitivity analysis: Changes in delay patterns.

- S4-1: Layout 1, i.e., simple linear terminal with one airside.
- S4-2: Layout 2, i.e., two parallel linear terminals, each with one airside.
- S4-3: Layout 3, i.e., one main concourse with multiple satellite teminals.

Exactly as we do in all of the experimental sets, we assume that all instances S4-1, S4-2, S4-3, have identical planned schedules and, consequently, the same planned passenger transfers, based on the original arrival/departure times and planned gate assignment. To maintain consistency in this set as well, after we generate the planned assignment for every case, we eventually consider only the transfers that were feasible according to the planned assignment across all sets.

Out of all sets (Figure 5.4), experimental set S4 is the only one where both algorithms produce the same results, with the total reassignment cost 8% lower in

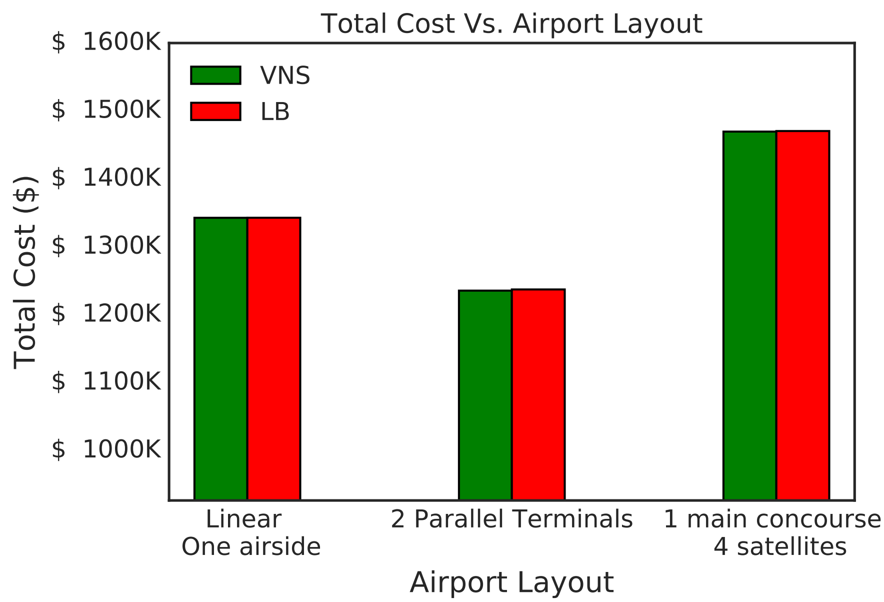

Figure 5.7: Sensitivity analysis: Changes in the layout of the airport.

the case of two parallel terminals compared to the simple Layout 1, and almost 10% higher for multiple satellite terminals located around a main concourse.

5.6 Further Extensions of VNS-LB

So far, the following main contributions of this study have been presented in this chapter:

- We have adapted a metaheuristic technique, i.e. Variable Neighborhood Search with Local Branching, to solve the gate reassignment problem in a limited amount of time.

- We have experimented with different definitions of the "solution neighborhood" and have used problem-specific properties to optimize the implementation of the algorithms.

- We have displayed the functionality of our approach by showing that it can produce near-optimal solutions within 10 minutes, and have supported our conclusion by demonstrating plausible sensitivity analysis results.

However, we have not analyzed so far the progress of the solution procedure throughout the running time. In this section, we will examine the behavior of Variable Neighborhood Search with Local Branching within the 10 minutes allocated to the solution procedure. More specifically, we will observe the best upper bound of the objective function for the VNS-LB and for the MIP solver throughout the optimization procedure, and compare the values of the two.

5.6.1 Neighborhood Change Threshold in Variable Neighborhood Descent

In this section, we examine the performance of VNS-LB throughout the optimization time and observe the progress of the search procedure from the beginning until the time limit T^{TL} is reached (10 minutes in this case). In particular, we compare the best solution found by VNS-LB with the incumbent solution found by the MIP solver throughout the 10 minute period allocated to the optimization. This information is valuable in the following aspects:

- Under certain circumstances, we might require to produce near-optimal solution in a smaller amount of time than the time limit for which we have calibrated the solution algorithm. Therefore, if we know that the algorithm converges fast to the optimal solution, we can be confident that good solutions can be found at the earlier stages of the optimization as well.

- Regardless of the required solution time, by observing the progress of the algorithm to the final solution, we can identify techniques to further accelerate the optimization procedure. These techniques may potentially extend beyond the scope of the gate assignment problem.

One would argue that, for different values of the total running time, a new calibration procedure is required to determine new parameter values. However, the purpose of this section is not to perform an exhaustive search on the different parameter combinations with respect to the possible time limits, but to suggest possible modifications to the algorithm that can accelerate the convergence of the algorithm to a good solution.

In the plots that follow, the green line corresponds to the best known bound found by VNS-LB, while the blue line is the respective incumbent solution value found by Gurobi.

Figure 5.8 demonstrates two cases where using VNS-LB essentially provides no advantage compared to using an MIP solver and terminating the solution procedure at 10 minutes. On the left plot (Figure 5.8a), Gurobi performs consistently better than VNS, while on the right plot (Figure 5.8b), VNS eventually produces a better final result that the solver, but is consistently outperformed for the first 500 seconds.

The stepwise form of the green line in Figure 5.8 implies the existence of multiple marginal improvements in the optimal solution. In other words, during the

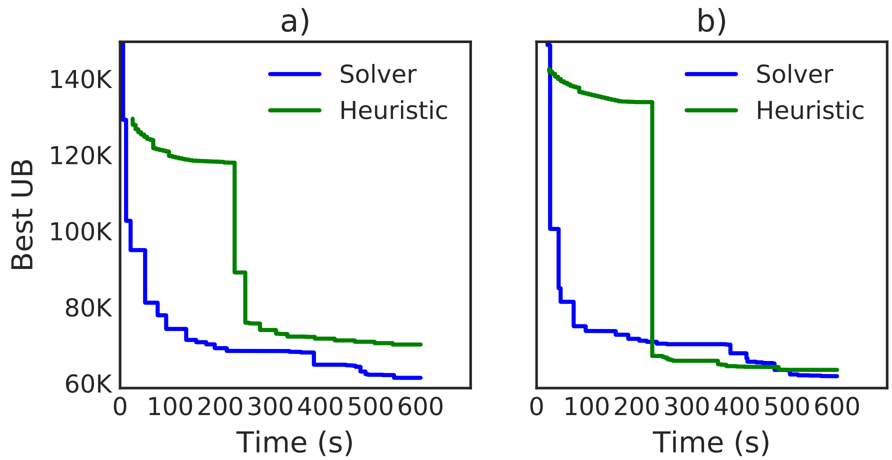

Figure 5.8: Cases where the solver produces consistently better results than VNS-LB during all (left) or almost all (right) the optimization procedure.

Variable Neighborhood Descent phase, the heuristic finds an improved solution x' within the neighborhood of the current solution $\tilde{x}$, centers the search around the new solution, and so on. The "dive" that takes place around 400 seconds after the start of the heuristic corresponds to the end of the current descent phase and the beginning of the `Shake` function. Until 250 seconds, the solver spends significant time in the descent phase, moving from a solution to nearby solutions within a small distance (low values of k) which are only marginally better compared to the current solution.

To remedy this situation, we impose a stricter threshold in the Neighborhood Change function applied within the VND loop. Specifically, we require the new solution x' to be considerably better than the current solution $\tilde{x}$ in order to replace it. To achieve this, we modify the neighborhood change condition from

$$\text{if } f(x') < f(\tilde{x}) \tag{5.31}$$

to

$$\text{if}\, f(x') \leq \alpha\, f(\tilde{x}) \tag{5.32}$$

where α is a coefficient between 0 and 1.

A suitable value for α satisfies the following criteria:

1. It is small enough, so that the current solution $\tilde{x}$ is not replaced by the new solution x', unless x' is significantly improved compared to $\tilde{x}$.

2. It is large enough, to prevent the termination of VND loop because of infeasibility.

Our next step is to determine which values of α improve the performance of VNS-LB. Our preliminary experiments indicate that, in general, values lower than $\alpha = 0.8$ force an early termination of the VND loop due to infeasibility. Therefore, we experiment with different values of α, namely 0.8, 0.9, 0.95, 0.97, and 0.99. For the purpose of our experiments, we select VNS parameter combinations 25 and 33, which were the two combinations with the smallest optimality gap in the calibration and testing experiments. However, even for these combinations there still exist large experimental cases where they converge slower compared to the MIP solver, as was shown before in Figure 5.8.

The modified VNS-LB algorithm is tested in a set of eight new experimental cases of dimensions similar to the ones used for sensitivity analysis in paragraph 5.5, and compare the plots of the best incumbent solutions found by the solver and by the heuristic during the ten minutes. Details on the number of flights, gates, and time horizon of these studies can be found in the Appendix (Table A13). We then compare the incumbent solution found by the MIP solver with the best upper bound that had been found by VNS-LB for selected time points $t \in \mathcal{T} = \{100, 200, 300, 400, 500\}$ during the optimization, namely for time equal to 100, 200, 300, 400, and 500 seconds.

Table 5.12 summarizes the total number of cases (out of the eight, i.e., the number of experimental cases) for which every modified combination outperformed the MIP solver in terms of the best solution yielded so far, for all of the selected time points. For parameter combination 25, we can find 3 cases where the heuristic finds a better solution at the end of the procedure (600 seconds), but we can increase this number to 4 for lower values of α. However, this is not the case for combination 33, which outperforms the solver at $t = 600$ seconds in only 1 out of the 8 examples. For $\alpha = 0.8$, a second model that eventually converges to a better solution can be found.

Table 5.12: Number of Cases Where VNS-LB Has Found A Better Solution Than The Current Solver Incumbent

Comb.	$\boldsymbol{\alpha}$	**Time (s)**					
		100	**200**	**300**	**400**	**500**	**600**
25	0.8	5	6	7	6	5	4
	0.9	5	7	7	6	5	4
	0.95	5	7	7	6	5	4
	0.97	4	6	6	6	4	3
	0.99	3	4	5	4	4	3
	1	2	2	4	4	3	3
33	0.8	5	5	5	5	3	2
	0.9	5	5	4	4	2	0
	0.95	5	5	5	4	1	0
	0.97	5	6	5	4	2	1
	0.99	4	5	5	5	2	1
	1	2	2	4	4	2	1

To evaluate the progress of the solution prior to termination, we examine the results produced for $t = 100, 200, \ldots 500$ seconds. For both combinations 25 and 33, using $\alpha < 1$ increases the percentage of models where the metaheuristic outperforms the solver at the earlier stages of the optimization. For example, for parameter combination 33, we observe that only 2 of the 8 total models can find a better solution than the solver by the time $t = 200$ seconds. However, this number is increased to 5 for $\alpha = 0.8, 0.9, 0.95, 0.99$, and to 6 for $\alpha = 0.97$. A similar general pattern can be observed for both combinations and all values of α and $t < 600$ seconds.

To get a more compact idea of the performance of modified VNS compared to the solver, we plot the best upper bounds found by the solver and by VNS against the running time. With simple VNS ($\alpha = 1$), models which perform worse or only slightly better than the solver at the end of the 10-minute time limit, benefit significantly or at least slightly from the introduction of the coefficient $a < 1$ (Figure 5.9 a, b) However, we also observe certain cases where, at the end of the 10-minute time limit, VNS eventually finds a worse solution than the MIP solver, although the use of the coefficient α can improve the quality of the solutions found by VNS at the early stages of the optimization (Figure 5.9c, d).

Overall, the introduction of coefficient a in the neighborhood change approach seems a plausible idea that yields promising preliminary results. Further research is required to identify more clear patterns and relationships between the progress of the solution procedure, the VNS-LB parameters, and the value of a, and therefore to determine the conditions under which a consistent improvement in the performance

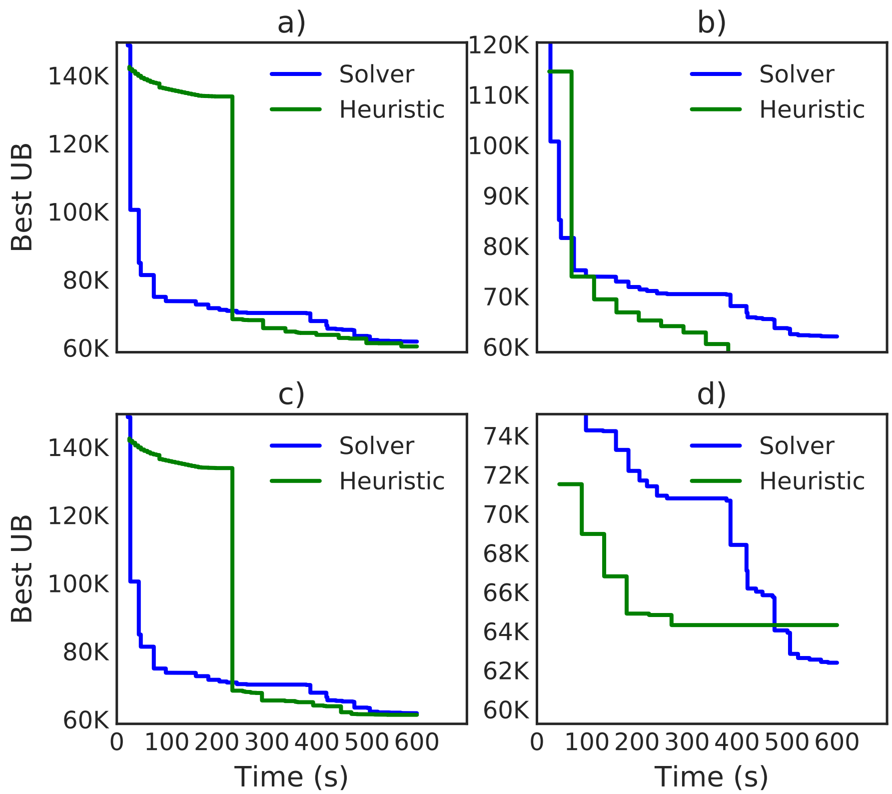

Figure 5.9: Top: A case where α (b) improves the final result compared to simple VNS-LB (a). Bottom: A case where α (d) does not improve the final result compared to simple VNS-LB (c), although it accelerates convergence at the earlier stages of the procedure.

of the method can be achieved. In the next paragraph, we will introduce an alternative approach for the neighborhood change function that relies on the gradual decrease of the value of α.

5.6.2 Increasing a During The Procedure

The plots of the solution progress as a function of time indicate that the MIP solver is able to find good solutions at the beginning of the optimization, while, as the procedure progresses, the improvement rate decreases. For example, let us consider the case shown in Figure 5.10a. Until approximately $t = 80$ seconds, multiple small improvements are observed, which correspond to consecutive iterations of the VND loop inside the same iteration of the outer loop. Around 110 seconds, a new solution is found, which corresponds in to the drop observed in the plot. The same pattern of multiple small improvements is observed again between 140 and 190 seconds, although it now lasts significantly less. Gradually, the rate of improvement decreases and the algorithm seems to converge to a solution, without further improvement after about 320 seconds. Eventually, the solution found by the MIP solver is better than the solution found by VNS-LB at the end of the 10-minute time period.

To take advantage of the fast convergence of the solver to solutions of good quality at the early stages of the algorithm, we experiment with a variable α value in the Variable Neighborhood Descent phase. The main idea is to start the VNS-LB with a low a (e.g., 0.8) and increase a in every iteration of the outer VNS-LB loop. By starting with relatively low a, we require large improvement in the objective function before we center the search around a new solution. Therefore, we do not spend time moving between solutions of marginally different objective value. On the other hand, by increasing a as the algorithm progresses, we essentially require a less strict improvement in the objective function to allow the new solution to replace the current one. The maximum value of a is equal to 1, which is equivalent to using the original neighborhood change condition 5.31. As soon as a becomes equal to 1, it remains 1 for all VNS iterations until the algorithm terminates.

The values of a that we use in the experiments belong to a predetermined set $\mathcal{A}^{\mathcal{I}}$, which we define as follows:

$$\mathcal{A}^{\mathcal{I}} = \{0.8,\ 0.9,\ 0.95,\ 0.99,\ 0.9999,\ 1\} \tag{5.33}$$

We test the modified VNS-LB algorithm with increasing a value for the same cases examined in the previous paragraph (5.6.1).

Table 5.13 contains the ratio of the best solution found by the algorithm to the best solution found by the solver, for the simple and the modified VNS-LB versions, for all values of α. In some cases (MM5 and MM8 for combination 33), using an increasing a coefficient not only returns a better solution than all other values of a (including a fixed $a = 1$), but is also the only way to produce a final solution that has a lower objective value than the solver solution.

The results indicate that, whenever the basic ($a = 1$) version of VNS-LB fails to find a better solution than the MIP solver solution (Models MM4, MM5, MM6, and MM8 for combination 33, and models MM4 and MM8 for combination 25, as shown in Table 5.13), increasing the value of a in every iteration of the outer loop results in a strictly better final solution compared to the solution produced by the solver (in 6 out of 7 cases, with the exception of combination 33 for model MM4).

Plotting the solution progress of VNS-LB and of the MIP solver as a function of time allows us to observe the relative performance of the two throughout the optimization procedure. As can be seen, using an increasing a value (Figure 5.10c) can help the metaheuristic not only reach a better final solution than the solver, but also perform better than both the solver and VND-LB with a fixed coefficient a (Figure 5.10b) during the optimization.

The above experiments allow us to test the proposed modifications and understand their potential to improve the convergence speed as well as the quality of progress towards the final solution, and of the final solution itself. Certain positive results indicate that the modified version of VNS-LB can accelerate the convergence of the procedure towards a good solution, and can provide better solutions than the ones that an MIP solver would provide if it was forced to terminate branch-and-bound after a time limit was reached. However, the number of experiments presented in the last section is limited, while we have not been able to identify yet a set of conditions under which we expect the modified version of VNS-LB to perform well. Since the experiments represent work that is currently in progress, further research is required to produce more systematic results and reach more solid conclusions in terms of the performance of the modified algorithm under different conditions.

5.7 Conclusions

In this chapter, we embedded the time-index formulation that was first presented in Chapter 3 and further improved in Chapter 4 in an MIP-based meta-heuristic framework to develop a methodology that produces near-optimal results for the gate reassignment problem in a small amount of time. In particular, we implemented Variable Neighborhood Search with Local Branching (VNS-LB), orig-

Table 5.13: Ratio Best VNS Solution / Best Gurobi Solution (“var” = increasing vaue of α

Comb.	$\boldsymbol{\alpha}$	**MM1**	**MM2**	**MM3**	**MM4**
25	**0.8**	1.03	1.02	1.07	1
	0.9	1.03	1.02	1.09	0.99
	0.95	1.03	1.03	1.09	0.99
	0.97	1.02	1.01	1.06	1.01
	0.99	1.23	1	1.05	1.05
	1	1	1	1	1.03
	var	1.04	1	1.01	0.99
33	**0.8**	1.26	1.24	1.39	1.02
	0.9	1.23	1.24	1.2	1.11
	0.95	1.23	1.23	1.31	1.08
	0.97	1.16	1.14	1.24	1.09
	0.99	1.23	1.17	1.1	1.07
	1	1	1	1	1.05
	var	1.22	1.11	1.14	1.03
		MM5	**MM6**	**MM7**	**MM8**
0.25	**0.8**	0.97	1	0.95	0.99
	0.9	0.97	1.01	0.96	0.95
	0.95	0.97	1	0.94	0.95
	0.97	1	1	0.95	0.95
	0.99	1	1.01	0.93	0.99
	1	0.97	1	0.98	1.01
	var	0.97	1.01	0.97	0.98
0.33	**0.8**	0.99	1.01	1	1.05
	0.9	1.05	1.02	1.03	1.01
	0.95	1	1.02	1.03	1.05
	0.97	0.99	1.03	1.03	1.01
	0.99	1.02	1.05	0.99	1.05
	1	1.02	1.02	0.99	1.04
	var	0.98	1	1	0.99

inally proposed by Mladenović and Hansen (1997). VNS-LB combines the ability of classic Variable Neighborhood Search for concurrent exploration and exploitation of the solution space with the strong MIP formulation that was developed in Chapter 4.

One of the key implementation issues explored in this study was the definition of a solution neighborhood. Therefore, a number of alternative definitions in the context of gate assignment were explored, and the mathematical formulation of the problem was modified accordingly when required. After fine-tuning the algorithm, we used the selected parameter combinations for sensitivity analysis, where we tested the output of the model for changes in operational and external factors. The sensitivity analysis results verified the applicability of Variable Neighborhood Search with Local Branching for the gate reassignment problem. In addition, we implemented a Local Branching heuristic approach, which shares some of its basic concepts with VNS-LB. However, the calibration for both algorithms shows that, while a transfer-based neighborhood definition works well for Local Branching, this is not the case for VNS-LB, where all of the best parameter combinations had a variable-based neighborhood definition. Overall, VNS-LB was shown to produce better results than LB, consistently returning a solution with at most 5% gap from optimality during the calibration and testing procedures.

To further improve the performance of the algorithm in large case studies, we proposed a modified version by introducing a coefficient a in the Neighborhood Change function to impose a requirement for large improvement in the objective function before centering the search procedure around a new solution. The experiments indicated that this approach is useful for improving the quality of the final solution and the progress of the optimization procedure as a whole. In addition, making a value decrease in each loop was shown to further improve the performance of the algorithm in some models.

These preliminary experiments on the modified algorithm can guide future research towards exploring in more depth the conditions under which VNS-LB can be successfully applied in cases of large airports with heavy flows and long scheduling horizons.

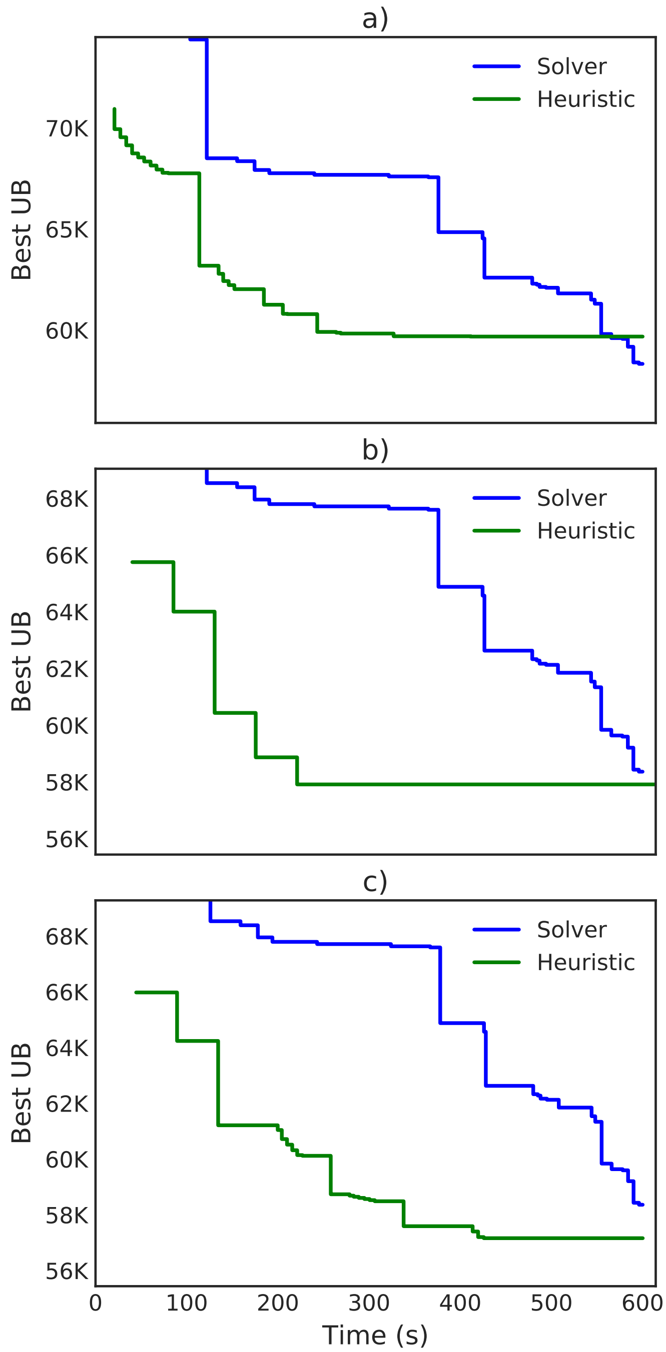

Figure 5.10: A model without the coefficient a (a), with fixed $a = 0.97$ (b), and with increasing $a \in \mathcal{A}^{\mathcal{I}}$.

Chapter 6

Summary, Conclusions, and Future Research

Chapter Overview

In the final chapter, we first summarize the dissertation and highlight the methodologies and techniques that have been presented in the previous chapters; second, we use the findings of the dissertation to draw more general conclusions; and finally, based on the research potentials identified in this study, we propose directions that seem promising for future research.

6.1 Summary

This dissertation has focused on developing a modeling framework for optimizing flight-to-gate assignment in schedule recovery procedures with a focus on transfer passengers. To a large extent, the need for schedule recovery strategies results from congestion caused by the increase in passenger demand observed over the last years, combined with tight scheduling and limited infrastructure capacity. From the perspective of gate scheduling, delayed flights may render the planned assignment infeasible, due to gate blockages, or impractical, at risk of causing further delay propagation. In this context, we proposed a methodology that allows airport and airline operators to generate a revised assignment of aircraft to airport gates, so that the effect of schedule disruptions is minimized. In particular, the proposed framework takes into account transfer passengers, who comprise a large percentage of airport users, especially in major hubs. From a modeling perspective, formulating passenger transfers is a computationally challenging task which has not been sufficiently addressed in the existing literature. Therefore, the critical role of gate scheduling in airport recovery procedures, combined with the scarcity of mathematical models and algorithms that deal with the optimization of passenger connections, has been the main motivation for the research presented in this dissertation.

Starting with a thorough review of the state-of-art approaches in Chapter 2, we introduce the existing studies on the planned gate assignment problem, which is essentially the "parent" problem of gate reassignment. Then, we emphasize three key elements of existing gate reassignment approaches, i.e., the objective function and constraints considered, the mathematical formulation, and the solution approach. Reviewing current literature helps us identify the gaps that this dissertation attempts to fill, which correspond to our three main contributions: First, we develop a model that handles passenger connections in what we consider to be a realistic way. Second, we analyze existing mathematical formulations and propose a set of sequential improvements to model. Finally, we investigate the applicability of different model-based heuristics in the context of gate reassignment and develop a metaheuristic framework which relies on our own, improved mathematical formulation.

In Chapter 3, we developed a gate reassignment framework that considers passenger transfers. Our formulation relies on a Binary Integer model which is the first multidimensional assignment model that uses gate location and the resulting required connection time to assess the success of passenger transfers. The model is generalized, in the sense that we can express every possible objective based on the observation that any measure of effectiveness corresponds to one of four discrete objective function terms categories. The cost coefficient of each of the four categories is properly adjusted so that the objective function of the problem is accurately expressed. Also, the model is easy to extend, so that it accounts for limited apron capacity, flight cancellations, and flights which are operated by the same aircraft. Its adaptability to a wide range of objectives and constraints allows us to use numerous measures of effectiveness as objective function components and evaluate the quality of the final solution in terms of its individual components under different objective functions.

In Chapter 4, we explore in depth the mathematical programming formulation of the problem. Our first step is to compare and analyze both theoretically and experimentally the two primary types of formulations in the current literature, i.e. the multidimensional assignment and the network flow-based formulations. In the context of this comparison, we estimate upper bounds on the number of variables and constraints of each formulation, and also identify the differences in the underlying assumptions of each approach and, consequently, in the applicability under different types of objective functions, cost coefficients, and modeling assumptions. Then, we strengthen the time-index assignment formulation of Chapter 3 by reformulating the constraints and by introducing valid inequalities that facilitate the cut-and-branch procedure. We also explore an alternative formulation to express the cost of passenger connections.

In Chapter 5, we embed the time-index formulation first presented in Chapter 3 and improved in Chapter 4 in an MIP-based metaheuristic framework with the goal to develop a methodology that produces near-optimal results for the gate reassignment problem in a low amount of time. In particular, we choose to implement Variable Neighborhood Search with Local Branching (VNS-LB), which combines the concurrent exploration and exploitation properties of Variable Neighborhood Search with the strong MIP formulation developed in Chapter 4. One of the key implementation issues explored is the definition of a solution neighborhood in the context of gate assignment, given that transfer passengers are the main consideration of the problem. Especially for definitions which are based on passenger transfers, the mathematical formulation of the problem has to be modified accordingly. We also apply a Local Branching heuristic, which shares some of its basic concepts with VNS-LB, to compare the results generated by the two methodologies. The VNS-LB algorithm is used to minimize the total cost, in which we also include a new measure of effectiveness concerning transfer passengers, i.e., the number of baggage pieces that fail to make the connection between the inbound and the outbound flight. At the end, we propose a set of extensions to the algorithm, based on the observed progress of the algorithm towards the final solution. These extensions concern the introduction of a tolerance coefficient in the neighborhood change function which is included in the Variable Neighborhood Descent loop of the algorithm.

6.2 Conclusions

In Chapter 3, we verify the validity of the assumption that the success of a transfer is a function of both the available time between the connecting flights, and of the required connection time, which in turn depends on numerous factors, including the layout of the airport and the duration of passenger processing procedures. As demonstrated by our experimental results, with the exception of extremely high or unusually low delays, the detailed assumption of considering gate location yields different results than simply assuming that a fixed time threshold is sufficient for all connections. Using the proposed model, we also conclude that the consideration of passenger connections in the optimization procedure is of utmost importance, since connecting passengers contribute significantly to the total solution cost. To prove this, we apply a hierarchical optimization framework to examine the interaction between the measures of effectiveness used in the literature and compare the results between the proposed assumption on connection time and the simple assumption of fixed thresholds. Our experiments demonstrate that prioritizing missed connections is necessary in the absence of monetary values, although the cost of the optimal solution is still far from the optimal cost yielded when using total monetary cost as an objective.

In Chapter 4, we evaluate the existing network flow formulations, as well as the proposed formulation, by generating experimental sets of various sizes with different airports and flight schedules. Our results indicate that aggregating assignment formulations outperform the respective linearized quadratic ones. Between the aggregating time-index assignment formulation and the network flow formulations, the experiments show that the former is consistently more efficient when the cost of successful connections is not included in the objective function, while the latter are faster whenever the objective function includes components like walking distance, which depend on successful passenger connections.

Finally, in Chapter 5, our Variable Neighborhood Search with Local Branching approach exhibits a consistently solid performance. Using medium-sized experimental cases, we calibrate the algorithm to determine the appropriate combination of parameters for the optimization. It is shown that the VNS-LB algorithm is capable of producing near-optimal results within only 10 minutes of running time, whereas the branch-and-cut procedure applied by the MIP solver would require between 1/2 and 3 hours for the same cases. We then use the algorithm in a set of sensitivity analysis experiments and verify its performance under variations in external and operational parameters. The sensitivity analysis results show that the proposed framework is capable of producing reasonable results and can answer various "what-if" questions concerning changes in operating parameters and external conditions. Furthermore, the comparison of VNS-LB with the simple Local Branching application indicates that VNS-LB outperforms Local Branching in terms of the quality of the best solution found by each of the two procedures. The last set of experiments is a preliminary evaluation of the proposed modified version of the VNS-LB metaheuristic, where a tolerance coefficient determines the threshold beyond which an improved solution replaces the current solution in the Variable Neighborhood Descent sub-process. The experiments indicated that this modification is useful not only for improving the quality of the final solution, but also the progress of the optimization procedure as a whole. A modified version of the algorithm with a decreasing tolerance coefficient was also tested, and showed promising results regarding the progress of the optimization and the quality of the final solution.

6.3 Future Research

The work presented in this dissertation can direct interesting avenues of future research. The recommendations included in this section focus primarily on the applicability of the model, as well as its methodological aspects.

6.3.1 Model Application

- **Comparison with Airline Practice:** The model provides airlines with a recommended course of action by incorporating the majority of the potential decisions into the model output. Future research can compare the optimal result of the model with the exact course of action and the empirical rules followed by airlines to quantify the magnitude of the benefits provided by using the proposed model.

- **Airport Operations:** The model treats gate reassignment as a stand-alone problem. Future research can focus on an integrated model that combines interdependent airport operations, including runway scheduling and apron bus scheduling.

- **Stochasticity:** The fluctuation of components that affect the model, such as the required connection time, is handled by appropriately adjusting their values prior to the optimization and solving the model as deterministic. In this context, a stochastic model would be able to better capture the uncertainty of delays, connection times, and other factors that affect decision making.

- **Additional Decisions:** The output of the model provides the decision maker with a set of actions that include gate switching and aircraft holding. The model can be further extended to accommodate additional decisions, such as holding a flight at the gate and consequently increasing the occupancy duration.

- **Air cargo:** The objectives that have been used so far only consider passenger transportation. An extension of the model to handle the movement of cargo is another direction for future research.

- **Delay propagation:** In the proposed model, decision making is limited to a specific airport. Future research can investigate ways to prevent the propagation of delays to the rest of the network.

6.3.2 Methodological Approach

- **Mathematical Formulation:** A significant part of this research has focused on the development and improvement of the MIP model formulation, while the set partitioning constraints have been proven to be particularly useful for strengthening the formulation in the linearized quadratic model. An alternative approach that has been shown to be quite successful in other applications is column generation. The potential of column generation lies in its application to models with complicating constraints, i.e., constraints that tie variables together. The valid inequalities introduced in Chapter 4 can be used for dividing the problem into smaller sub-problems based on passenger connections. Additional ideas from strengthening the formulation can be borrowed from problems of similar structure, such as the machine scheduling problem, for which Sousa and Wolsey (1992) proposed a number of valid cuts.

- **Variable Neighborhood Search With Local Branching:** The preliminary experiments of Chapter 5 yielded promising results for the modified version of VNS-Lb with tolerance. Therefore, research in the future can focus on identifying the values of the tolerance coefficient that result in consistently good performance of the algorithm. In addition, many concepts of pure Variable Neighborhood Search can be explored in the context of VNS-LB. For example, skewed VNS, which allows the search to be centered around a worse solution than the current one, provided that the distance between the two exceeds a specific threshold, is a concept worth examining. Furthermore, the Variable neighborhood Search with Formulation Space Search is another modification of Variable Neighborhood Search, which alternates between two different, yet equivalent, formulations, of the same problem. Given that we have already explored the different MIP formulations, combining them within a VNS-LB framework is another interesting direction for future studies.

- **Dynamic Framework:** The model that we have proposed in this study is meant to satisfy the requirement for real-time solutions. However, it is a one-step optimization model, which assumes that the operator has all the information about flight delays for the whole planning horizon and uses it to optimize the assignment of aircraft to gates. In an extension of the model, delays can be simulated as random events and the model can be optimized iteratively as soon as a new piece of information about a flight delay is obtained.

This dissertation has two main aspirations: First, to develop a versatile tool that can be adapted according to the objectives, priorities, and strategies of air transportation practitioners. Second, to provide air transportation researchers with an insight of how the features of a solution in practice are reflected in the abstract modeling aspects of the mathematical formulation. Every idea that relies on these two principles should be a promising path for future research.

Appendix

Additional Tables

Table A1: Increase in total cost when using each MOE as the objective value compared to the optimal cost.

MOE	**Case**	**Cost Increase**	**Case**	**Cost Increase**
OT_{PT}	1	1942%	4	1997%
OS_{PD}		1359%		1636%
OR_P		1905%		2149%
OW		2805%		3112%
OM_P		207%		252%
OR_F		1780%		1922%
OS_F		1654%		1916%
OT_F		2056%		2168%
OT_T		1942%		1997%
OS_P		2046%		2075%
OT_{PT}	2	1751%	5	2090%
OS_{PD}		1377%		1424%
OR_P		2079%		1549%
OW		3178%		3328%
OM_P		279%		261%
OR_F		2114%		1505%
OS_F		1579%		1248%
OT_F		2024%		2232%
OT_T		1751%		2090%
OS_P		1835%		2179%
OT_{PT}	3	2191%	6	2242%
OS_{PD}		1928%		1986%
OR_P		2080%		2177%
OW		3402%		3384%
OM_P		246%		237%
OR_F		1734%		2218%
OS_F		1749%		1525%
OT_F		2355%		2382%
OT_T		2191%		2242%
OS_P		2221%		2413%

Table A2: Ratio of single objective optimization cost to hierarchical optimization cost (with missed connections as the objective with the highest priority) for different MOEs as objectives.

MOE	**Case**	**Ratio**	**Case**	**Ratio**
OT_{PT}	1	12.4	4	11.6
OS_{PD}		5		5.7
OR_P		6.2		6.3
OW		9.7		9.1
OM_P		1		1
OR_F		5.8		5.6
OS_F		5.6		5.4
OT_F		8.2		8
OT_T		7.7		7.4
OS_P		13		12
OT_{PT}	2	9.5	5	11.7
OS_{PD}		4		4.3
OR_P		*		4.3
OW		8.1		8.3
OM_P		1		1
OR_F		5.5		4.2
OS_F		4.3		3.4
OT_F		6.3		6.8
OT_T		6.1		7.2
OS_P		10		12.1
OT_{PT}	3	12.4	6	12.8
OS_{PD}		5.6		6.1
OR_P		5		5.7
OW		10.7		10.1
OM_P		1		1
OR_F		4.5		5.6
OS_F		4.6		4.1
OT_F		7.9		8.8
OT_T		7.9		8.9
OS_P		12.5		13.7

Table A3: Summary of Number of Variables and Constraints For All Formulations

Formulation	Number of Variables	Number of Constraints
Q-A Q-S Q-F	$\lvert F\rvert\bar{G}\bar{W} + \lvert T\rvert\bar{G}^2\bar{W}^2$	$\lvert F\rvert + \lvert T\rvert + 4\lvert T\rvert\bar{G}^2\bar{W}^2 + 2\lvert F\rvert\bar{G}\bar{W} + \lvert T\rvert G^2W^2$ Additional $\bar{G}^2\bar{W}^2$ for Q-S2
Q-FA	$(\lvert F\rvert + \lvert T\rvert)\bar{G}\bar{W}$	$\lvert F\rvert + \lvert T\rvert + 4\lvert T\rvert\bar{G}\bar{W}$ $+2\lvert F\rvert\bar{G}\bar{W} + \lvert T\rvert\bar{G}$
A-SS2, A-SL2, A-AS2, A-AL2	$(\lvert F\rvert\bar{W} + \min(\lvert F^{A,C}\rvert, \lvert F^{D,C}\rvert))\bar{G}$	$\lvert F\rvert + \lvert T\rvert + \bar{G}(2\min(\lvert F^{A,C}\rvert, \lvert F^{D,C}\rvert) + \lvert F\rvert\bar{W}$
A-SS3, A-SL3, A-AS3, A-AL3	$\bar{G}\bar{W}(\lvert F\rvert + \min(\lvert F^{A,C}\rvert, \lvert F^{D,C}\rvert))$	$\lvert F\rvert + \lvert T\rvert + \bar{G}\bar{W} \times 2\min(\lvert F^{A,C}\rvert, \lvert F^{D,C}\rvert) + \lvert F\rvert)$
YL	$\lvert F\rvert\bar{G}\bar{W}+$ $\lvert G\rvert(\lvert W\rvert + 1) + \lvert T\rvert\bar{W}^2$	$\lvert F\rvert + 3(\lvert T\rvert\bar{W}^2) + \lvert G\rvert(\lvert W\rvert + 1)+$ $\lvert T\rvert + 2\lvert T\rvert WC + \lvert F\rvert\bar{W}\bar{G} + \lvert T\rvert\bar{W}^2$
ZK	$\lvert F\rvert\bar{G}\bar{W} + \lvert G\rvert(\lvert W\rvert + 1) + \lvert T\rvert$ $+\lvert T\rvert(2\bar{C}\bar{W} + \bar{C}^2\bar{W}^2 + 1)$	$\lvert F\rvert + \lvert G\rvert(\lvert W\rvert + 2) + \lvert T\rvert(2\bar{C}\bar{W} + 2)+$ $2\lvert T\rvert + (\bar{F}\bar{W}) + (W + 1) + \lvert T\rvert$

Table A4: Comparison of Quadratic Assignment Formulations

Case	Flights	Gates	Hours
1	200	40	5
2	200	40	5
3	200	40	5
4	200	40	5
5	200	40	5

Table A5: Comparison of Quadratic Assignment Formulations `Q-A` and `Q-S`

Case	Flights	Gates	Hours
1	120	50	4
2	120	50	4
3	120	50	4
4	120	50	4
5	120	50	4

Table A6: Comparison of Aggregating Formulations

Case	Flights	Gates	Hours
1	75	50	3
2	90	50	3
3	75	50	3
4	100	50	4
5	105	50	3
6	75	50	3
7	90	50	3
8	120	50	4
9	100	50	4
10	105	50	3
11	90	50	3
12	140	50	4

Table A7: Comparison of Quadratic and Aggregating Formulation

Case	**Flights**	**Gates**	**Hours**
1	120	50	4
2	120	50	4
3	120	50	4
4	120	50	4
5	120	50	4
6	120	50	4

Table A8: Aggregating Formulation (`A-AS3`) Vs. Yu and Lau's (`YL`) Formulation (Comb. = Combination)

Comb.	**Flights**	**Gates**	**Known**	**Comb.**	**Flights**	**Gates**	**Known**
1	120	50	4	25	200	50	5
2	120	50	4	26	200	50	5
3	120	50	4	27	200	50	5
4	150	50	5	28	160	50	4
5	150	50	5	29	160	50	4
6	150	50	5	30	160	50	4
7	120	50	4	31	160	50	4
8	120	50	4	32	160	50	4
9	120	50	4	33	160	50	4
10	120	50	4	34	150	50	5
11	120	50	4	35	150	50	5
12	120	50	4	36	150	50	5
13	160	50	4	37	200	50	5
14	160	50	4	38	200	50	5
15	160	50	4	39	200	50	5
16	150	50	5	40	200	50	5
17	150	50	5	41	200	50	5
18	150	50	5	42	200	50	5
19	150	50	5	43	160	50	4
20	150	50	5	44	160	50	4
21	150	50	5	45	160	50	4
22	120	50	4	46	200	50	5
23	120	50	4	47	200	50	5
24	120	50	4	48	200	50	5

Table A9: Aggregating Formulation (`A-AS3`) Vs. Zhang and Klabjan's (`ZK`) Formulation - Successful Connection Cost Included

Case	**Flights**	**Gates**	**Hours**
1	175	40	5
2	175	40	5
3	175	40	5
4	200	40	5
5	200	40	5
6	200	40	5
7	210	40	6
8	210	40	6
9	210	40	6
10	240	40	6
11	240	40	6
12	240	40	6

Table A10: Aggregating Formulation `A-AS3` Vs. `ZK` Formulation - Successful Connection Cost Not Included

Case	**Flights**	**Gates**	**Hours**
1	200	40	5
2	200	40	5
3	200	40	5
4	200	40	5
5	200	40	5
6	200	40	5

Table A11: Calibration Results For Variable Neighborhood Search With Local Branching (Time = Time of Optimal Heuristic Value)

Model Name	**Comb.**	**Heuristic Value**	**Optimal Iteration**	**Time**	**Heuristic / Optimal**	**Gap From Optimal**
MC1	22	1485570	2	61.7	1.06	6%
	24	1485570	2	120.8	1.06	6%
	25	1455590	31	600.2	1.04	4%
	33	1496990	11	433.8	1.07	7%
	35	1460470	12	526.8	1.04	4%
	40	1500040	2	121.8	1.07	7%
	41	1464380	8	143.1	1.05	5%
	43	1484620	7	322.9	1.06	6%
	45	1471470	7	484	1.05	5%
	48	1440410	2	121.1	1.03	3%
MC2	22	1452610	2	61.3	1.01	1%
	24	1437600	2	121.5	1.00	0%
	25	1515440	31	592.1	1.05	5%
	33	1463670	16	286.7	1.02	2%
	35	1445040	15	525.6	1.01	1%
	40	1488550	5	363.5	1.04	4%
	41	1472090	10	163.3	1.02	2%
	43	1442950	10	325.2	1.00	0%
	45	1437640	9	484.5	1.00	0%
	48	1510700	2	484.2	1.05	5%
MC3	22	479615	2	63.1	1.08	8%
	24	460830	2	122.4	1.04	4%
	25	454303	20	402.6	1.02	2%
	33	463464	7	147.2	1.04	4%
	35	446760	8	351.9	1.01	1%
	40	493560	2	122.9	1.11	11%
	41	454910	5	111.4	1.02	2%
	43	447900	6	231.4	1.01	1%
	45	449030	6	202.4	1.01	1%
	48	453300	2	122.6	1.02	2%

MC4	22	42320	2	63.5	1.24	24%
	24	38680	2	123.5	1.13	13%
	25	39160	19	277.4	1.15	15%
	33	39840	8	100.3	1.16	16%
	35	37000	9	296.3	1.08	8%
	40	36600	2	123.3	1.07	7%
	41	40160	5	182.3	1.17	17%
	43	36520	6	216.4	1.07	7%
	45	36440	6	167.4	1.07	7%
	48	36960	2	123.3	1.08	8%
MC5	22	1923470	2	183.2	1.22	22%
	24	1895650	2	122.4	1.20	20%
	25	1877430	27	575.7	1.19	19%
	33	1768990	11	245.8	1.12	12%
	35	1600220	13	567.5	1.01	1%
	40	1889230	3	243.1	1.20	20%
	41	1809090	7	123.3	1.15	15%
	43	1649330	7	323.5	1.04	4%
	45	1619370	7	484.7	1.03	3%
	48	1917510	2	122	1.21	21%

Table A12: Calibration Results For Local Branching (Time = Time of Optimal Heuristic Value)

Model Name	Comb.	Heuristic Value	Optimal Iteration	Time	Heuristic / Optimal	Gap From Optimal
MC1	22	1485570	2	61.7	1.06	6%
	24	1485570	2	120.8	1.06	6%
	25	1455590	31	600.2	1.04	4%
	33	1496990	11	433.8	1.07	7%
	35	1460470	12	526.8	1.04	4%
	40	1500040	2	121.8	1.07	7%
	41	1464380	8	143.1	1.05	5%
	43	1484620	7	322.9	1.06	6%
	45	1471470	7	484	1.05	5%
	48	1440410	2	121.1	1.03	3%
MC2	22	1452610	2	61.3	1.01	1%
	24	1437600	2	121.5	1.00	0%
	25	1515440	31	592.1	1.05	5%
	33	1463670	16	286.7	1.02	2%
	35	1445040	15	525.6	1.01	1%
	40	1488550	5	363.5	1.04	4%
	41	1472090	10	163.3	1.02	2%
	43	1442950	10	325.2	1.00	0%
	45	1437640	9	484.5	1.00	0%
	48	1510700	2	484.2	1.05	5%
MC3	22	479615	2	63.1	1.08	8%
	24	460830	2	122.4	1.04	4%
	25	454303	20	402.6	1.02	2%
	33	463464	7	147.2	1.04	4%
	35	446760	8	351.9	1.01	1%
	40	493560	2	122.9	1.11	11%
	41	454910	5	111.4	1.02	2%
	43	447900	6	231.4	1.01	1%
	45	449030	6	202.4	1.01	1%
	48	453300	2	122.6	1.02	2%

MC4	22	42320	2	63.5	1.24	24%
	24	38680	2	123.5	1.13	13%
	25	39160	19	277.4	1.15	15%
	33	39840	8	100.3	1.16	16%
	35	37000	9	296.3	1.08	8%
	40	36600	2	123.3	1.07	7%
	41	40160	5	182.3	1.17	17%
	43	36520	6	216.4	1.07	7%
	45	36440	6	167.4	1.07	7%
	48	36960	2	123.3	1.08	8%
MC5	22	1923470	2	183.2	1.22	22%
	24	1895650	2	122.4	1.20	20%
	25	1877430	27	575.7	1.19	19%
	33	1768990	11	245.8	1.12	12%
	35	1600220	13	567.5	1.01	1%
	40	1889230	3	243.1	1.20	20%
	41	1809090	7	123.3	1.15	15%
	43	1649330	7	323.5	1.04	4%
	45	1619370	7	484.7	1.03	3%
	48	1917510	2	122	1.21	21%

Table A13: Experimental Cases For The Modified VNS-LB Algorithm

MODEL NAME	**FLIGHTS/HOUR**	**GATES**	**HOURS**
MM1	50	52	12
MM2	50	52	12
MM3	51	52	12
MM4	51	52	12
MM5	51	52	12
MM6	51	52	12
MM7	51	52	12

Bibliography

Abdelghany, A., Abdelghany, K., and Narasimhan, R. (2006). Scheduling baggage-handling facilities in congested airports. Journal of Air Transport Management, 12(2):76 – 81.

Abdelghany, K. F., Shah, S. S., Raina, S., and Abdelghany, A. F. (2004). A model for projecting flight delays during irregular operation conditions. Journal of Air Transport Management, 10(6):385 – 394.

Ahuja, R. K., zlem Ergun, Orlin, J. B., and Punnen, A. P. (2002). A survey of very large-scale neighborhood search techniques. Discrete Applied Mathematics, 123(1):75 – 102.

AIA (2016). Athens International Airport Eleftherios Venizelos. `https://www.aia.gr/traveler/flight-info/rtfi/`. Accessed: 2016-06-15.

AIA (2018). Athens International Airport Eleftherios Venizelos. `https://www.aia.gr/userfiles/675393df-ab1a-4b77-826c-f3096a3d7f12/PaxUS_final-2015-12.pdf`. Accessed: 2018-08-30.

Ambite, J.-L. (2001). Local search in combinatorial optimization. `http://www.cs.cmu.edu/afs/cs/project/jair/pub/volume15/ambite01a-html/node9.html`. Accessed: 2018-08-30.

Ashford, N. (1988). Level of service design concept for airport passenger terminalsa european view. Transportation Planning and Technology, 12(1):5–21.

Association of European Airlines (2016). Traffic and capacity data. `http://www.aea.be/statistics.html`. Accessed: 2016-11-30.

Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A., Peterson, E., Sherry, L., Trani, A., and Zou, B. (2010). Total Delay Impact Study: A Comprehensive Assessment of the Costs and Impacts of Flight Delay in the United

States. Technical report, NEXTOR, National Center for Excellence in Aviation Operations Research.

Barth, T. (2013). Optimal assignment of incoming flights to baggage carousels at airports, DTU Management Engineering Report No. 4. Technical report, Department of Management Engineering, Technical University of Denmark. http://orbit.dtu.dk/files/53702586/Optimal_assignment.pdf. Accessed: 07/20/2018.

Baumgarten, P., Malina, R., and Lange, A. (2014). The impact of hubbing concentration on flight delays within airline networks: An empirical analysis of the us domestic market. Transportation Research Part E: Logistics and Transportation Review, 66:103 – 114.

Beatty, R., Hsu, R., Berry, L., and Rome, J. (1999). Preliminary evaluation of flight delay propagation through an airline schedule. Air Traffic Control Quarterly, 7(4):259–270.

Bendinelli, W. E., Bettini, H. F., and Oliveira, A. V. (2016). Airline delays, congestion internalization and non-price spillover effects of low cost carrier entry. Transportation Research Part A: Policy and Practice, 85:39 – 52.

Berthold, T. (2006). Primal Heuristics for Mixed Integer Programs. PhD thesis, Fachbereich Mathematik der Technischen Universitat Berlin, Berlin.

Bihr, R. A. (1990). A conceptual solution to the aircraft gate assignment problem using 0, 1 linear programming. Computers & Industrial Engineering, 19(1):280 – 284.

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. Journal of the Operational Research Society, 52(10):1107–1120.

BTS (2018). Airline On-Time Statistics and Delay Causes. http://www.rita.dot.gov/bts/help/aviation/html/understanding.html#. Accessed: 2018-08-30.

Campanelli, B., Fleurquin, P., Arranz, A., Etxebarria, I., Ciruelos, C., Eguluz, V. M., and Ramasco, J. J. (2016). Comparing the modeling of delay propagation in the us and european air traffic networks. Journal of Air Transport Management, 56:12 – 18. Long-term and Innovative Research in ATM.

Castaing, J., Mukherjee, I., Cohn, A., Hurwitz, L., Nguyen, A., and Mller, J. J. (2016). Reducing airport gate blockage in passenger aviation: Models and analysis. Computers & Operations Research, 65:189 – 199.

Cattaneo, M., Malighetti, P., Paleari, S., and Redondi, R. (2017). Evolution of the european network and implications for self-connection. Journal of Air Transport Management, 65:18 – 28.

Cheng, C.-H., Ho, S. C., and Kwan, C.-L. (2012). The use of meta-heuristics for airport gate assignment. Expert Systems with Applications, 39(16):12430 – 12437.

Clausen, T. and Pisinger, D. (2010). Dynamic Routing of Short Transfer Baggage . Technical report, Technical University of Denmark (Kgs. Lyngby: DTU Management), No. 15. http://orbit.dtu.dk/files/53702586/Optimal_assignment.pdf. Accessed: 07/20/2018.

Competition Commission (2002a). BAA Plc: A Report on the Economic Regulation of the London Airports Companies (Heathrow Airport Ltd, Gatwick Airport Ltd and Stansted Airport Ltd). Technical report, Competition Commission.

Competition Commission (2002b). Manchester Airport PLC: A Report on the Economic Regulation of Manchester Airport PLC. Technical report, Competition Commission.

Congram, R. K., Potts, C. N., and van de Velde, S. L. (2002). An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. INFORMS Journal on Computing, 14(1):52–67.

Cook, A. and Tanner, G. (2011). Modelling the airline costs of delay propagation. Presented at: AGIFORS Airline Operations Conference, London, United Kingdom, May 16-19, 2011.

Şeker, M. and Noyan, N. (2012). Stochastic optimization models for the airport gate assignment problem. Transportation Research Part E: Logistics and Transportation Review, 48(2):438 – 459.

de Barros, A. G., Somasundaraswaran, A., and Wirasinghe, S. (2007). Evaluation of level of service for transfer passengers at airports. Journal of Air Transport Management, 13(5):293 – 298. The Air Transport Research Society's 10th year Anniversary, Nagoya Conference, 2006.

De Neufville, R. and Odoni, A. (2003). Airport Systems: Planning, Design, and Management. Aviation Week Books. McGraw-Hill.

Ding, H., Lim, A., Rodrigues, B., and Zhu, Y. (2004a). Aircraft and gate scheduling optimization at airports. In 37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of the, pages 8 pp.–.

Ding, H., Lim, A., Rodrigues, B., and Zhu, Y. (2004b). New heuristics for overconstrained flight to gate assignments. Journal of the Operational Research Society, 55(7):760–768.

Ding, H., Lim, A., Rodrigues, B., and Zhu, Y. (2005). The over-constrained airport gate assignment problem. Computers & Operations Research, 32(7):1867 – 1880.

Dorndorf, U., Drexl, A., Nikulin, Y., and Pesch, E. (2007a). Flight gate scheduling: State-of-the-art and recent developments. Omega, 35(3):326 – 334.

Dorndorf, U., Jaehn, F., Lin, C., Ma, H., and Pesch, E. (2007b). Disruption management in flight gate scheduling. Statistica Neerlandica, 61(1):92–114.

Dorndorf, U., Jaehn, F., and Pesch, E. (2008). Modelling robust flight-gate scheduling as a clique partitioning problem. Transportation Science, 42(3):263–404.

Dorndorf, U., Jaehn, F., and Pesch, E. (2012). Flight gate scheduling with respect to a reference schedule. Annals of Operations Research, 194(1):177–187.

Dorndorf, U., Jaehn, F., and Pesch, E. (2016). Flight gate assignment and recovery strategies with stochastic arrival and departure times. OR Spectrum, pages 1–29.

Dorndorf, U., Jaehn, F., and Pesch, E. (2017). Flight gate assignment and recovery strategies with stochastic arrival and departure times. OR Spectrum, 39(1):65 – 93.

Dunbar, M., Froyland, G., and Wu, C.-L. (2012). Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. Transportation Science, 46(2):204–216.

Eurocontrol (2013). Standard Inputs for EUROCONTROL Cost-Benefit Analyses. Technical report, European Organisation for the Safety of Air Navigation (EUROCONTROL).

Eurocontrol (2016). Challenges of Air Transport 2030 Survey of Experts Views. Technical report, European Organisation for the Safety of Air Navigation (EUROCONTROL).

Eurocontrol (2018). CODA DIGEST 2017 - All-Causes Delay and Cancellations to Air Transport in Europe. Technical report, European Organisation for the Safety of Air Navigation (EUROCONTROL).

FAA (1996). Advisory circular: Surface and movement guidance and control system. Technical report, US Department of Transportation, Federal Aviation Administration.

Fageda, X. and Flores-Fillol, R. (2015). A note on optimal airline networks under airport congestion. Economics Letters, 128:90 – 94.

Fischetti, M. and Lodi, A. (2003). Local branching. Mathematical Programming, 98(1):23–47.

Frey, M., Kolisch, R., and Artigues, C. (2017). Column generation for outbound baggage handling at airports. Transportation Science, 51(4):1226–1241.

Gelhausen, M. C., Berster, P., and Wilken, D. (2013). Do airport capacity constraints have a serious impact on the future development of air traffic? Journal of Air Transport Management, 28:3 – 13. Selected papers from the 15th Air Transport Research Society Conference, Sydney, 2011.

Genç, H. M., Erol, O. K., Eksin, I., Berber, M. F., and Güleryüz, B. O. (2012). A stochastic neighborhood search approach for airport gate assignment problem. Expert Systems with Applications, 39(1):316 – 327.

Glover, F. and Kochenberger, G. (2003). Handbook of Metaheuristics. International Series in Operations Research & Management Science. Springer US.

Gu, Y. and Chung, C. A. (1999). Genetic algorithm approach to aircraft gate reassignment problem. Journal of Transportation Engineering, 125(5):384–389.

Guépet, J., Acuna-Agost, R., Briant, O., and Gayon, J. (2015). Exact and heuristic approaches to the airport stand allocation problem. European Journal of Operational Research, 246(2):597 – 608.

Haghani, A. and Chen, M.-C. (1998). Optimizing gate assignments at airport terminals. Transportation Research Part A: Policy and Practice, 32(6):437–454.

Hansen, M. and Zou, B. (2013). Airport operational performance and its impact on airline cost. In Zografos, K., Andreatta, G., and Odoni., A., editors, Modelling and managing Airport Performance, chapter 5, page 119144. John Wiley & Sons Inc, Hoboken.

Hansen, P., Mladenović, N., Brimberg, J., and Pérez, J. A. M. (2010). Variable neighborhood search. In Gendreau, M. and Potvin, J.-Y., editors, Handbook of Metaheuristics, chapter 3, pages 61–86. Springer, Boston, MA.

Hansen, P., Mladenovi, N., and Uroevi, D. (2006). Variable neighborhood search and local branching. Computers & Operations Research, 33(10):3034 – 3045. Part Special Issue: Constraint Programming.

Horstmeier, T. and de Haan, F. (2013). Influence of ground handling on turn round time of new large aircraft. Aircraft Engineering and Aerospace Technology, 73(3):266–271.

Huang, E., Mital, P., Goetschalckx, M., and Wu, K. (2016). Optimal assignment of airport baggage unloading zones to outgoing flights. Transportation Research Part E: Logistics and Transportation Review, 94:110 – 122.

IATA (2004). Airport Development Reference Manual. 9th edition. Technical report, Montreal, QC: International Air Transport Association.

IATA (2018). 2036 forecast reveals air passengers will nearly double to 7.8 billion. IATA (International Air Transport Association) official website, `https://www.iata.org/pressroom/pr/Pages/2017-10-24-01.aspx`. Accessed: 2018-09-30.

ICAO (2018). Continued passenger traffic growth and robust air cargo demand in 2017. ICAO (International Civil Aviation Organization) official website, https://www.icao.int/Newsroom/Pages/Continued-passenger-traffic-growth-and-robust-air-cargo-demand-in-2017.aspx. Accessed: 2018-09-30.

John F. Kennedy International Airport official website (2018). Arrivals/departures. http://www.jfkairport.com, Accessed: 2018-02-20.

Kafle, N. and Zou, B. (2016). Modeling flight delay propagation: A new analytical-econometric approach. Transportation Research, Series B: Methodological, 93:520–542.

Kim, N.-Y. and Park, J.-W. (2016). A study on the impact of airline service delays on emotional reactions and customer behavior. Journal of Air Transport Management, 57:19 – 25.

Kim, S. H., Feron, E., Clarke, J.-P., Marzuoli, A., and Delahaye, D. (2013). Airport gate scheduling for passengers, aircraft, and operations. Tenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2013).

Koopmans, T. C. and Beckmann, M. (1957). Assignment problems and the location of economic activities. Econometrica, 25(1):53–76.

Kusumaningtyas, I. and Lodewijks, G. (2013). On the application of accelerating moving walkways to support passenger processes in amsterdam airport schiphol. Transportation Planning and Technology, 36(7):617–635.

Luke, S. (2013). Essentials of Metaheuristics. Lulu, second edition. Available for free at http://cs.gmu.edu/$\sim$sean/book/metaheuristics/.

Maharjan, B. (2010). Flight Gate Assignment and Proactive Flight Gate Reassignment Optimization for Hub and Spoke Airline Operations. PhD thesis, Texas Tech University.

Maharjan, B. and Matis, T. I. (2011). An optimization model for gate reassignment in response to flight delays. Journal of Air Transport Management, 17(4):256 – 261.

Maharjan, B. and Matis, T. I. (2012). Multi-commodity flow network model of the flight gate assignment problem. Computers & Industrial Engineering, 63(4):1135 – 1144.

Marinelli, M., Dell'Orco, M., and Sassanelli, D. (2015). A metaheuristic approach to solve the flight gate assignment problem. Transportation Research Procedia, 5:211 – 220.

Mladenović, N. and Hansen, P. (1997). Variable neighborhood search. Comput. Oper. Res., 24(11):1097–1100.

Narciso, M. E. and Piera, M. A. (2015). Robust gate assignment procedures from an airport management perspective. Omega, 50:82 – 95.

Nosedal, J., Piera, M. A., Ruiz, S., and Nosedal, A. (2014). An efficient algorithm for smoothing airspace congestion by fine-tuning take-off times. Transportation Research Part C: Emerging Technologies, 44:171 – 184.

Phillips, L. T. (1987). Air carrier activity at major hub airports and changing interline practices in the united states' airline industry. Transportation Research Part A: General, 21(3):215 – 221.

Pternea, M. and Haghani, A. (2018). Mathematical models for flight-to-gate reassignment with passenger flows: State-of-the-art comparative analysis, formulation improvement, and a new multidimensional assignment model. Computers & Industrial Engineering, 123:103 – 118.

Pternea, M. and Haghani, A. (2019). An aircraft-to-gate reassignment framework for dealing with schedule disruptions. Journal of Air Transport Management, ATRS World Conference 2017 Special Issue, In Press.

Pyrgiotis, N., Malone, K. M., and Odoni, A. (2013). Modelling delay propagation within an airport network. Transportation Research Part C: Emerging Technologies, 27:60 – 75. Selected papers from the Seventh Triennial Symposium on Transportation Analysis (TRISTAN VII).

Raidl, G. R., Puchinger, J., and Blum, C. (2010). Metaheuristic Hybrids, pages 469–496. Springer US, Boston, MA.

Roosens, P. (2008). Congestion and air transport: a challenging phenomenon.

Santos, B. F., Wormer, M. M., Achola, T. A., and Curran, R. (2017). Airline delay management problem with airport capacity constraints and priority decisions. Journal of Air Transport Management, 63:34 – 44.

Santos, G. and Robin, M. (2010). Determinants of delays at european airports. Transportation Research Part B: Methodological, 44(3):392 – 403. Economic Analysis of Airport Congestion.

Schaefer, L. and Millner, D. (2001). Flight delay propagation analysis with the detailed policy assessment tool. In 2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236), volume 2, pages 1299–1303 vol.2.

Schumer, C. and Maloney, C. (2008). Your flight has been delayed again: flight delays cost passengers, airlines, and the US economy billions. Technical report, The US Senate Joint Economic Committee.

Sousa, J. P. and Wolsey, L. A. (1992). A time indexed formulation of non-preemptive single machine scheduling problems. Mathematical Programming, 54(1):353–367.

Tang, C.-H. and Wang, W.-C. (2013). Airport gate assignments for airline-specific gates. Journal of Air Transport Management, 30:10 – 16.

Tang, C.-H., Yan, S., and Hou, Y.-Z. (2010). A gate reassignment framework for real time flight delays. 4OR, 8(3):299–318.

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. Psychometrika, 17(4):401–419.

US DOT (2018). Air Travel Consumer Report. Technical report, US Department of Transportation,Office of Aviation Enforcement and Proceedings, Aviation Consumer Protection Division, Washington DC. https://www.transportation.gov/sites/dot.gov/files/docs/resources/individuals/aviation-consumer-protection/310946/2018-may-atcr.pdf. Accessed: 07/20/2018.

Vaze, V. and Barnhart, C. (2012). Modeling airline frequency competition for airport congestion mitigation. Transportation Science, 46(4):512–535.

Wang, H., Luo, Y., and Shi, Z. (2013). Real-time gate reassignment based on flight delay feature in hub airport. Mathematical Problems in Engineering, 2013.

Wong, J.-T. and Tsai, S.-C. (2012). A survival model for flight delay propagation. Journal of Air Transport Management, 23:5 – 11.

World Bank (2016). Air transport, passengers carried. http://data.worldbank.org/indicator/IS.AIR.PSGR?end=2015&start=1970. Accessed: 2016-08-30.

Wu, C. and Wong, J. (2007). Delay propagation modeling and the implications in robust airline scheduling. In 2007 ATRS World Conference, San Francisco.

Wu, C.-L. and Lee, A. (2014). The impact of airline alliance terminal co-location on airport operations and terminal development. Journal of Air Transport Management, 36:69 – 77.

Yan, S. and Chang, C.-M. (1998). A network model for gate assignment. Journal of Advanced Transportation, 32(2):176–189.

Yan, S., Chen, C.-Y., and Tang, C.-H. (2009). Airport gate reassignment following temporary airport closures. Transportmetrica, 5(1):25–41.

Yan, S. and Huo, C.-M. (2001). Optimization of multiple objective gate assignments. Transportation Research Part A: Policy and Practice, 35(5):413 – 432.

Yan, S. and Tang, C.-H. (2007). A heuristic approach for airport gate assignments for stochastic flight delays. European Journal of Operational Research, 180(2):547 – 567.

Yan, S., Tang, C.-H., and Hou, Y.-Z. (2011). Airport gate reassignments considering deterministic and stochastic flight departure/arrival times. Journal of Advanced Transportation, 45(4):304–320.

Yu, C. and Lau, H. Y. K. (2015). Airport gate reassignment based on the optimization of transfer passenger connections. Journal of Traffic and Logistics Engineering, 3(1):25–30.

Yu, C., Zhang, D., and Lau, H. (2016). Mip-based heuristics for solving robust gate assignment problems. Computers & Industrial Engineering, 93:171 – 191.

Zhang, D. and Klabjan, D. (2017). Optimization for gate re-assignment. Transportation Research Part B: Methodological, 95:260 – 284.