DTU Library

Allocation of Ground Handling Resources at Copenhagen Airport

Justesen, Tor Fog

Publication date:
2014

Document Version
Peer reviewed version

Allocation of Ground Handling Resources at Copenhagen Airport

Tor Fog Justesen

Technical University of Denmark
Kgs. Lyngby, January 2014

PhD Thesis, 2014

Tor Fog Justesen
Allocation of Ground Handling Resources at Copenhagen Airport
(Allokering af ground handling ressourcer i Københavns Lufthavn)

Department of Management Engineering
Technical University of Denmark
Produktionstorvet, Building 424
DK-2800 Kgs. Lyngby, Denmark
Phone: +45 45 25 48 00, Fax: +45 45 25 48 05
phd@man.dtu.dk
www.man.dtu.dk

Print: Rosendahls - Schultz Grafisk, Albertslund, Denmark

To Marie, Magnus & Laura

Summary

Operating an airport is a very complex task involving many stakeholders. The primary role of airport management is to ensure that the airport provides sufficient capacity in all operational areas and that all the companies carrying out business at the airport have the best possible working conditions. Moreover, management must ensure that the airport stays competitive and that its business goals are met to the greatest possible extent.

The European Organization for the Safety of Air Navigation (EUROCONTROL) expects demand for air services in Europe to double by 2030 and identifies airport capacity as a potential bottleneck that may slow this growth. Many European airports are already operating at the limit of their capacity; moreover, they are under competitive pressure from both nearby airports and fast-growing mega-hubs in the Middle East. Providing efficient and reliable airport operations is imperative for the viability and continued development of both individual airports and the air transportation industry in general.

This thesis gives a general introduction to the management of airport operations. It describes the main airport processes and optimization problems that these processes give rise to. The primary focus is on *ground handling resource allocation problems*, it looks in detail at the following problems: the check-in counter allocation problem, the baggage make-up position problem, the tactical stand and gate allocation problem, the operational stand and gate allocation problem, and the taxiway route allocation problem. Although these problems arise from different airport processes and involve different stakeholders, they share some characteristics and can be formulated as variants of the same mathematical model.

Many real-world aspects must be taken into consideration when solving airport optimization problems; the models and solutions that are developed must be able to meet the needs of airlines to the greatest possible extent. They must be easy to configure and efficient to solve. For three of the problems considered here, real-world restrictions reduce the number of possible variables to such an extent that the problem can be efficiently solved to optimality with modern, state-of-the-art MIP solvers. For the remaining problems, an LP based heuristic was developed. The method iteratively solves a restricted LP relaxed version of the problem and exploits expert knowledge to generate appropriate initial variables, enabling the heuristic to efficiently find near-optimal and operationally valid solutions.

The work described in this thesis was carried out in the context of an Industrial PhD project at Copenhagen Airport in collaboration with the Technical University of Denmark. It contributes to both the introduction and definition of various ground handling resource allocation problems, and proposes a mathematical formulation of the problems. These contributions are presented in four scientific papers and one technical report, which are included. All the models and solution methods described here are currently implemented and used in various settings at Copenhagen Airport. These include weekly operational planning of check-in counter allocation and long-term capacity/demand analyses of the airport's stands and gates.

Resumé (Danish Summary)

Lufthavnsdrift er en meget kompleks opgave, der involverer mange interessenter, heriblandt luftfartsselskaber, ground handlere, passagerer og lufthavnen selv. En lufthavns ledelses primære rolle er at sikre, at lufthavnen har tilstrækkelig kapacitet på alle operationelle områder samt at de mange virksomheder, der opererer i lufthavnen har de bedst mulige arbejdsvilkår. Derudover skal ledelsen sikre, at lufthavnen forbliver konkurrencedygtig, og at dens forretningsmæssige mål opfyldes i videst muligt omfang.

Den europæiske organisation for luftfartens sikkerhed, EUROCONTROL forventer, at efterspørgslen efter flytrafik i Europa vil blive fordoblet i forhold til i dag inden 2030 og identificerer kapaciteten af de enkelte lufthavne som en potentiel flaskehals, der kan begrænse denne vækst. Mange lufthavne opererer allerede på grænsen af deres kapacitet og er samtidig under pres fra konkurrerende lufthavne; både nærliggende, og hurtigt voksende lufthavne i Mellemøsten. En effektive og pålidelige lufthavnsdrift er således en bydende nødvendighed for den fortsatte udvikling af den enkelte lufthavn og luftfartsindustrien generelt.

Denne ph.d.-afhandling giver en generel introduktion til en lufthavns driftsstyring og beskriver de vigtigste processer samt optimeringsproblemer, som disse processer giver anledning til. Det primære fokus er på allokering af ground handling ressourcer, og følgende problemer beskrives: allokering af check-in-pulte, allokering af bagage-udsorterings-bokse, taktisk standpladsdisponering, operationel standpladsdisponering, og rutning af fly gennem taxiway-netværket. Selvom disse problemer hver især udspringer fra vidt forskellige lufthavnsprocesser og involverer forskellige interessenter, har de nogle fællestræk, der muliggør, at de kan formuleres som varianter af den samme matematiske model.

Ved løsning af optimeringsproblemer i en lufthavn, er det nødvendigt at tage mange operationelle aspekter fra den virkelige verden med i betragtning; de udviklede modeller og løsningsmetoder skal i videst muligt omfang kunne tilfredsstille de præferencer, som luftfartsselskaberne har givet udtryk for, de skal være lette at konfigurere, og de skal være effektive at løse. For tre af de beskrevne problemer, reducerer disse hensyn antallet af mulige løsninger i en sådan grad, at en optimal løsning kan findes effektivt blot ved brug moderne, state-of-the-art MIP-løsere. De resterende problemer løses ved hjælp af en effektiv LP-baseret heuristik; metoden løser iterativt en begrænset LP-version af problemet og udnytter praktisk viden til at generere nye variable. Som resultaterne viser, finder metoden nær-optimale, operationelt gyldige løsninger. Alle de beskrevne modeller og løsningsmetoder er implementeret og anvendes i forskellige sammenhænge i Københavns Lufthavn, heriblandt den ugentlige operationelle planlægning af check-in-pult-allokeringen samt langsigtede kapacitetsanalyser for standpladser og gates.

Det underliggende arbejde for afhandlingen er blevet gennemført i form af et Erhvervs-Ph.d.-projekt hos Københavns Lufthavne A/S i samarbejde med Danmarks Tekniske Universitet.

Preface

This thesis has been submitted to the Department of Management Engineering, Technical University of Denmark in partial fulfillment of the requirements for acquiring the PhD degree. The work has been supervised by Professor Jesper Larsen, Department of Management Engineering, Technical University of Denmark and Anders Høeg Dohn, PhD, Head of Security Resource Management, Copenhagen Airports A/S.

The project is an Industrial PhD in collaboration with Copenhagen Airports A/S and financial support from the Danish Ministry of Science, Innovation and Higher Education under the Industrial PhD programme. An Industrial PhD is a three-year industrially focused PhD project in which the student is hired by a company and at the same time enrolled at a university. The main purpose of an Industrial PhD is to bridge the theoretical world of academia with the more practical world of the industry. A successful Industrial PhD demonstrates that theoretical, state-of-the-art methods can be applied to real-world problems and directly or indirectly affect the profit or performance of the company.

The thesis is made up of two parts. The first part forms an introduction to the topics that have been investigated throughout the PhD project. The second part is a compilation of four scientific papers and one technical reports that have been written during the PhD project. The PhD project has been carried out from October 2010 to January 2014.

Kgs. Lyngby, January 2014

Tor Fog Justesen

Dissemination and other work

Scientific papers

- **Paper:** Tor Fog Justesen, Esben Kolind, and Jesper Larsen (2014). "Check-in Counter Allocation At Copenhagen Airport". Submitted to *Journal of Airport Management*.
- **Paper:** Tor Fog Justesen, Jesper Larsen, David Ryan, Richard Lusby, and Anders Høeg Dohn (2014). "Aircraft Stand Allocation with Associated Resource Scheduling". Submitted to *Computers and Operations Research, Special issue on Airport Operations Management*.
- **Paper:** Tor Fog Justesen, Simon Thoustrup, and Jesper Larsen (2014). "Operational Stand and Gate Allocation". Submitted to *Transportation Research Part E*.
- **Paper:** Tor Fog Justesen, Torben Barth, and Frauke Böckmann (2013). "Making OR a success - applications to airport operations management". Submitted to *Interfaces*.

Unfinished work

- **Technical report:** Tor Fog Justesen, Claudia Munk Billing, and Jesper Larsen (2014). "Taxiway Route Allocation at Copenhagen Airport". Available on demand.

Conference abstracts

- **Conference abstract:** Tor Fog Justesen, David Ryan, Jesper Larsen, and Richard Lusby (2012). "Aircraft Stand Allocation with Associated Resource Scheduling". 1st International Conference on Airport Operations Management.
- **Conference abstract:** Tor Fog Justesen, Simon Thoustrup, and Jesper Larsen (2013). "Operational Stand and Gate Allocation". INFORMS Annual Meeting 2013.

Talks

- **Conference:** Tor Fog Justesen, Simon Thoustrup, and Jesper Larsen (2013). "Operational Stand and Gate Allocation". INFORMS Annual Meeting 2013, Minneapolis.
- **External lecture:** Tor Fog Justesen, Simon Thoustrup, and Jesper Larsen (2013). "Operational Stand and Gate Allocation". Course: 13400 Simulation in Freight Transportation and Logistics, Technical University of Denmark.
- **Seminar:** Tor Fog Justesen (2013). "Operational Stand and Gate Allocation". INFORMS Annual Meeting, Minneapolis.
- **Workshop:** Tor Fog Justesen, Simon Thoustrup (2013). "Application of Optimization at Copenhagen Airport". Optimization Workshop, Amadeus.

- **Workshop:** Tor Fog Justesen, Simon Thoustrup (2013). "The Stand and Gate Allocation Problem". Optimization Workshop, Amadeus.
- **Seminar:** Tor Fog Justesen, Simon Thoustrup, and Jesper Larsen (2013). "Operational Stand and Gate Allocation". OR seminar, Technical University of Denmark.
- **Seminar:** Tor Fog Justesen (2013). "The aircraft stand allocation problem on a strategical, tactical, and operational level". Internal seminar, CPH.
- **External lecture:** Tor Fog Justesen, David Ryan, Jesper Larsen, and Richard Lusby (2012). "Aircraft Stand Allocation with Associated Resource Scheduling". Course: 13400 Simulation in Freight Transportation and Logistics, Technical University of Denmark.
- **External lecture:** Tor Fog Justesen (2012). "Application of Optimization at Copenhagen Airport". Course: 13400 Simulation in Freight Transportation and Logistics, Technical University of Denmark.
- **Conference:** Tor Fog Justesen, David Ryan, Jesper Larsen, and Richard Lusby (2012). "Aircraft Stand Allocation with Associated Resource Scheduling". 1st International Conference on Airport Operations Management, Technical University of Munich.
- **Seminar:** Tor Fog Justesen (2012). "Application of Optimization at Copenhagen Airport". Visit to CPH from Aarhus University.
- **Seminar:** Tor Fog Justesen, David Ryan, Jesper Larsen, and Richard Lusby (2012). "Aircraft Stand Allocation with Associated Resource Scheduling". Visit to CPH from Aarhus University.
- **Seminar:** Tor Fog Justesen, David Ryan, Jesper Larsen, and Richard Lusby (2012). "Aircraft Stand Allocation with Associated Resource Scheduling". OR seminar, Department of Engineering Science, University of Auckland.
- **Seminar:** Tor Fog Justesen (2011). "The Stand and Gate Allocation Problem". Internal seminar, CPH.
- **External lecture:** Tor Fog Justesen (2011). "The Stand and Gate Allocation Problem". Course: 13400 Simulation in Freight Transportation and Logistics, Technical University of Denmark.
- **Seminar:** Tor Fog Justesen (2011). "The Stand and Gate Allocation Problem". OR seminar, Technical University of Denmark.
- **Conference:** Tor Fog Justesen (2010). "Applications of OR Optimization Methods in Copenhagen Airport", OR day, Aarhus School of Business.

Projects co-supervised

- **Master's Thesis:** Claudia Munk Billing (2013). "The aircraft taxiway route allocation problem". Department of Management Engineering, Technical University of Denmark.
- **Master's Thesis:** Nikolaj Axelsen (2013). "Baggage chute allocation in CPH". Department of Management Engineering, Technical University of Denmark.
- **Master's Thesis:** Michael Bossen Møller and Jesper Bæch Sørensen (2011). "Resource Allocation in Baggage Handling at Copenhagen Airport". Department of Management Engineering, Technical University of Denmark.

- **Bachelor's Project:** Sarah Babette Schadegg (2011). "Allocation of check-in counters in Copenhagen Airport". Department of Management Engineering, Technical University of Denmark.
- **Bachelor's Project:** Christian Lous (2011). "Modelling and optimization of allocation of check-in counters in an airport". Department of Management Engineering, Technical University of Denmark.
- **Special course:** Maria Grønnegaard Nielsen (2013). "Check-in counter allocation in Copenhagen Airport". Department of Management Engineering, Technical University of Denmark.
- **Special course:** Jakob Dirksen (2010). "Counter Allocation for Copenhagen Airports". Department of Management Engineering, Technical University of Denmark.

Courses

- Developing key performance indicators for airports (Special course, 5 ECTS)
- How to write a scientific paper (Special course, 2.5 ECTS)
- 42142 Recent Research Results in Operations Research (2.5 ECTS)
- 42790 Business Course for Industrial PhD students (7.5 ECTS)
- 42142 (P42) Recent Research Results in Operations Research (2.5 ECTS)
- 42702 Research and PhD-studies at DTU Management (2.5 ECTS)
- Stochastic Programming with Applications in Logistics (Aarhus University, 5 ECTS)

Acknowledgements

This thesis could not have been written without the support of a large number of people. Firstly, I would like to thank my two main supervisors, Professor Jesper Larsen, DTU Management Engineering and Anders Høeg Dohn, Head of Security Resource Management, Copenhagen Airports A/S, for their guidance and support throughout the project period. I thank Jesper for establishing the contact to Copenhagen Airport (CPH) and the former COO Michael Løve, for his at times naive belief in the project and for constantly encouraging me to continue the work! Especially, I am thankful for his extensive and patient support and for ensuring the required academic approach, qualifying my work for a PhD. I thank Anders for his never-ending enthusiasm and for teaching me what a PhD project is all about. In spite of the fact that time was always scarce and our meetings repeatedly re-scheduled, I always left our supervision meetings with a firm belief in the project and a positive motivation to continue the hard work.

The operations research environment at DTU Management Engineering is probably the strongest in Denmark and I thank everybody in the section for three enjoyable years.

I thank Professor David Ryan for inviting me to New Zealand and for his exceptional hospitality during the stay. The mathematical modeling approach used throughout the project was developed in collaboration with David, and I am truly thankful to him for teaching me how to approach practical problems. In general, everybody we met in Auckland were extremely helpful and kind, making our stay in New Zealand an unforgettable experience.

I am thankful to my fantastic colleagues in Airport Optimization at CPH. I enjoy the great work atmosphere where working and good fun goes hand in hand. Especially I would like to thank Simon, Esben, Karina and Claudia for being such great friends and truly amazing colleagues and my two managers, Kasper and Thomas, for letting me work (almost) undisturbed throughout the period and for accepting my wish of completing the project.

I am thankful to all my co-authors for their contributions to the individual papers: I thank Richard Lusby for numerous interesting discusions during the work on the stand and gate allocation problem and for being an OR genius in general, a good friend and a truly helpful guy. I thank Torben Barth for sharing his ideas and for iniating the work on the paper on how to make the use of OR a success in airport operations management and Frauke Böckmann for critical reviews of the paper. I thank Esben Kolind for proofreading and discussing much of my work in general and for the help on the check-in counter allocation paper in particular. Finally, I thank Simon Thoustrup for the many hours of coding and developing ideas for the stand and gate allocation problem. I owe most of my knowledge about the stand and gate allocation problem to Simon. His extraordinary ability to understand the link between the practical operation of an airport and the theoretical possibilities of operations research continues to inspire me.

At the personal level, the PhD has been somewhat of a turbulent journey. I would like to thank my friends and family for their inexhaustible support, and in particular, I would like to thank Signe Justesen and Matias Sevel Rasmussen for extensive proofreading, and Simon Spoorendonk, for helping me get back on track and plan the last three months of the project.

Above all, I am very thankful to my family; to my children, Magnus and Laura, for being so wonderfully ignorant of operations research and the stressful life as a PhD student and for continuously reminding me of the truly important matters in life; to Marie for her unlimited support and care and for being the best thing that has ever happened to me! I could not have done this without you!

Contents

Part I

Airport Operations Management

Chapter 1

Introduction

The three main components of the air transportation system are airlines, airports, and air traffic management. According to the medium term forecast by the European Organization for the Safety of Air Navigation (EUROCONTROL), the number of flights in Europe is expected to grow to 11.3 million in 2018 (see [EUROCONTROL, 2012]); however, it is estimated that lack of airport capacity will limit this growth by 1.2% (134,000 flights). The growth in traffic will put massive pressure on airports, forcing them to both increase capacity and provide more efficient operations. Moreover, this pressure is likely to create congestion in the air transportation system that could potentially have an impact on airlines' ability to maintain their schedules, and, in the worst case scenario pose a threat to the safety, efficiency and competitiveness of all actors involved in the air transport supply chain (see [Marian, 2012]). Despite the fact that many airports are expanding, constructing new airport facilities is slow and expensive. Therefore, short- and medium-term growth in the aviation sector depends on the airport's ability to operate more efficiently and make better use of existing capacity.

More air traffic in Europe will mean bigger airports. According to the long term forecast by EUROCONTROL, in 2030, 13-34 European airports will be needed that can handle more than 150,000 departures per year - a number which is currently achieved by only seven airports in Europe (cf. ([EUROCONTROL, 2010]). Some Eastern European airports are growing fast, but hubs in the Middle East are also expected to attract large numbers of passengers traveling to the Middle East, Far East and Southern Africa. Therefore, in the future European airports will be under pressure not only from a general growth in traffic and competition from other European airports, but also airports outside Europe, in particular the Middle East.

To maintain and expand their share of the international air traffic market, airports will have to increase capacity and improve quality, limit charges, and increase the range of services provided to airlines and passengers. Efficient and reliable airport operations, together with optimized take-off and landing procedures are thus imperative for the viability and continued development of individual airports and the air transportation industry in general.

Operating an airport is a very complex task involving many stakeholders. The role of airport managers is to ensure that the airport has sufficient capacity in all operational areas and provides optimal working conditions for all companies operating there, including—apart from the airport itself—airlines, handling companies, shops, and restaurants. Moreover, managers must ensure that the airport stays competitive and that its business goals are met to the greatest possible extent.

The operational processes of an airport give rise to a wide variety of optimization problems. Many of these can be modeled and solved using operations research techniques; consequently *airport operations management* is an emerging field within operations research. This thesis addresses the problem of the allocation of ground handling resources at Copenhagen Airport (CPH) with a particular focus on the allocation of check-in counters, baggage make-up positions, aircraft stands and gates, and taxiways. All of these ground-handling resources are scarce and very expensive. Moreover, increasing the resource supply involves a time-consuming and often very costly re-design of both the apron and terminal buildings which is usually not feasible in the short term. It is therefore of great economic importance for CPH to use their existing resources as efficiently as possible.

The allocation problems addressed here can all be formulated as variants of the set-packing problem. In this scenario, individual ground-handling resources are represented over a series of time intervals and the solution is constrained by a set of generalized upper bound constraints. Furthermore, the number of variables can be significantly reduced by the introduction of various practical restrictions. These restrictions encapsulate preferences specified by airlines, handling companies and the airport itself; often the primary objective is to meet these preferences as far as possible.

The allocation problems described here are those experienced by CPH; however, as standardization is widespread in the aviation industry, many of these processes and optimization problems are also found in other airports. The models and solution that are developed are thus also applicable to equivalent problems arising at other airports.

1.1 Background

The work presented in this thesis was carried out in the context of an Industrial PhD project for Copenhagen Airports A/S (CAAS) in cooperation with the Technical University of Denmark (DTU). One of the purposes of an Industrial PhD is to bridge the theoretical world of academia with the more practical world of the industry. Consequently, the focus of the project is to ensure that the models and solution that were developed were applicable in real life. This not only means that the method is capable of solving real-life problems using real-life data; it must also be compatible with current work processes and take into account whether the problem needs to be addressed at the strategic, tactical, or operational level.

During my time at CPH, I have learned that doing research in an operational world is a difficult task. The primary focus of the entire organization is to ensure stable and reliable airport operations, while providing a high level of service to all customers, including airlines and handling companies. Often decisions need to be made very quickly and there seems to be a general consensus that if a near-optimal solution can be found more efficiently than an optimal solution, then it is preferred, as "things will change anyway".

I was part of the Airport Optimization department at CPH, which was established in 2010. The idea behind the creation of the department was to centralize scarce analytical resources into one department, and try to develop intellectual synergies. Before 2010, the number of employees with an academic background working in operations was very limited; a maximum of around 10 out of 1,400 full-time employees. The new Airport Optimization team was built around competencies drawn from operations research. The introduction of these new skills into CAAS has without doubt contributed to a dramatically increased level of fact-based, rather than gut-based decision making. Additionally, the establishment of the department has proven to be the start of an academic revolution within CAAS. Compared to 2010, the number of employees with an academic background working in operations is now above 50, leading to not only a significantly reduced dependency on external consultants, but also an opportunity to become world knowledge leaders within the airport industry.

In 2011, CPH established the *European Forum of Operations Analysts* (EFOA) network, together with representatives of airports in Frankfurt (FRA), London Heathrow (LHR), Zurich (ZHR), Dublin (DUB), and Prague (PRG). The purpose of EFOA is to establish a professional network consisting of the major European airports, in order to increase knowledge sharing about best practice in airport operations management. From discussions with other airports that participate in the EFOA network, it has become apparent that for many operational optimization problems, the current approach used at CPH is unique in the world. The introduction of operations research combined with extensive peer networking and cherry picking of knowledge, has enabled CPH to develop state-of-the art methodologies for strategic/tactical capacity planning and tools for operational planning. As a consequence, market leaders in software supply have discovered the benefits of working together with the operations analysts of CPH and future software suites for the airport industry are being built based on the methodology and approaches developed by CPH.

There is no doubt that operations research will play a central role in the future development of large international airports. A key issue when designing and dimensioning the airport of the future is to know current demands on capacity. With the optimization tools that have been developed and implemented, CPH now has full and clear understanding of current capacity utilization; from check-in counters, to baggage make-up positions, to stands. This allows the airport to forecast future demand by taking into account forecasted growth rates, technological developments, and legislation. With the introduction of operations research into daily operations, CPH has increased the probability that future expansion of capacity will be correctly sized and timed.

1.2 Main contributions

The main contributions of the thesis can be summarized as follows:

- A **holistic and comprehensive** description of an airport and the concepts of airport operations management, including airport processes in an optimization context
- The **identification and modeling** of a range of different **ground handling resource allocation problems** with similar characteristics
- The establishment of **a common set-packing model formulation** for the ground handling resource allocation problems that are identified
- The development of an **LP based heuristic** that efficiently finds near-optimal, operationally applicable solutions to the allocation problems in question
- The **deployment** of the developed models and solutions to operations at CPH

1.3 Thesis structure

The remainder of this thesis is structured as follows. Part I gives a general introduction to airport operations management.

Chapter 2 gives a very brief introduction to CPH. Chapter 3 gives an overall introduction to airport operations management and describes the six main processes that are part of the handling of commercial passenger aircraft. For each process the main subprocesses are described and for each subprocess, the primary planning and optimization problems arising from the process are introduced. Chapter 4 describes the ground handling resource allocation problems addressed in this thesis and introduces the basic resource allocation problem. Chapter 5 describes how the basic resource allocation problem can be formulated as a set-packing problem and shows how the ground handling resource allocation problems in question can all be formulated as variants of this model. Chapter 6 describes the solutions that were developed, and finally we draw some conclusions in Chapter 7.

Part II consists of four papers submitted to scientific journals. These papers constitute the major part of the work in this thesis. They are included in the most recent version. They have, however, been adjusted to the layout of this thesis and consequently may be slightly different to the journal versions.

The paper presented in Chapter 8 addresses the check-in counter allocation problem. It demonstrates that while the implementation is viable for the current setup at CPH, it is also the case that the problem can be solved without too big a loss in solution quality using a simple and very efficient greedy heuristic.

The paper presented in Chapter 9 addresses the *aircraft stand and gate allocation problem with associated resource scheduling*. The modeling approach and solution described in this paper have not been previously covered by the literature, and the results show that the method generates high-quality, feasible solutions within a reasonable time.

The paper presented in Chapter 10 addresses the *operational stand and gate allocation problem* and evaluates the potential of the modeling approach and solution presented in Chapter 9 in an operational context where the re-allocation of turn-rounds is needed.

The paper presented in Chapter 11 summarizes the lessons learned during the work on the check-in counter allocation problem presented in Chapter 8 and a broad range of decision support tools for baggage handling at Frankfurt Airport. It gives an overview of the drivers and challenges that need to be addressed in order for a new application or system to be accepted. The paper presents a general discussion of the most important operational aspects to consider when developing models and solutions for airport optimization problems. It proposes that a new step is added to the development process, where these aspects are considered. The idea here is that this new step mimics the existence of an operational environment and thus increases the probability that the developed application will have an impact on operations and be accepted by stakeholders.

Part III consists of an appendix describing further unfinished work that was undertaken during the PhD project but is not yet sufficiently mature to be published. Appendix A is a technical report describing initial work on the tactical taxiway route allocation problem. The problem is formulated as a ground handling resource allocation problem and the report describes a case study from CPH, in which the model is used to analyze the operational consequences of changing the push-back options for selected stands. The case study illustrates a potential use of the model; however, the solution is not fully tested and still needs development.

Chapter 2

Copenhagen Airport

Founded in 1925, Copenhagen Airport (CPH) was one of the first civil airports in the world. Located eight kilometers southeast of Copenhagen city center and 24 kilometers west of the center of the Swedish city of Malmö, CPH is a key part of the infrastructure in both Denmark and southern Sweden. It is Scandinavia's main transfer airport.

The airport is one of the main hubs used by Scandinavian Airlines and is an operating base for Thomas Cook Airlines, Scandinavia and Norwegian Air Shuttles, JetTime, and EasyJet. In 2012, CPH serviced 23,336,187 passengers and handled 242,992 aircraft movements (arrivals and departures) making it the busiest airport in Scandinavia. Unlike other Scandinavian airports, a considerable share of the airport's passengers is international; the domestic share of annual passengers is under 10%.

Copenhagen Airports A/S is a listed company; it owns and operates the airports at Copenhagen and Roskilde. The second largest shareholder is currently (January 2014) the Canadian pension fund Ontario Teachers' Pension Plan Board (OTPP); together with the Macquarie European Infrastructure Fund III (MEIF3) they own 57.7% of all shares via the holding company Copenhagen Airports Denmark (CAD). The Danish Government owns 39.2% of shares and the remaining shares are owned by private investors.

CPH makes its infrastructure, buildings and service facilities available to the companies that have business operations at the airport. Furthermore, CPH has responsibility for a number of processes that are essential to the operation of the airport, including check-in counter allocation, security check of passengers, bird control, busing of passengers, landside and airside safety, maintenance of all facilities, and stand and gate allocation.

Copenhagen Airports A/S employs approximately 1,900 staff. However, in total there are more than 20,000 workers employed by more than 500 different companies operating at CPH alone, making the airport the largest workplace in Denmark.

Despite the global financial crisis that began in 2008 (which led to a global recession and contributed to the European sovereign-debt crisis from 2008 to 2012), the number of passengers using Copenhagen Airport (CPH) increased by 29% in the same period (see Figure 2.1). CPH strategy is to offer highly integrated services for passengers, airlines and the many companies that have their business operations at CPH, while minimizing costs.

The two primary prerequisites for future growth at CPH are sufficient financial resilience and the capacity to match demand. To strengthen its competitive position, in 2012 CPH began the deployment of an ambitious growth strategy entitled *World Class Hub*. Under this strategy, CPH intends to intensify the development and optimization of the airport's infrastructure to prepare it to handle up to 40 million passengers per year. As part of the World Class Hub strategy, CPH changed its vision to be:

> *The Gateway of Northern Europe: Where you come to move on and we make you wish to stay*

The goal of CPH is thus to become the preferred northern European gateway for all its customers:

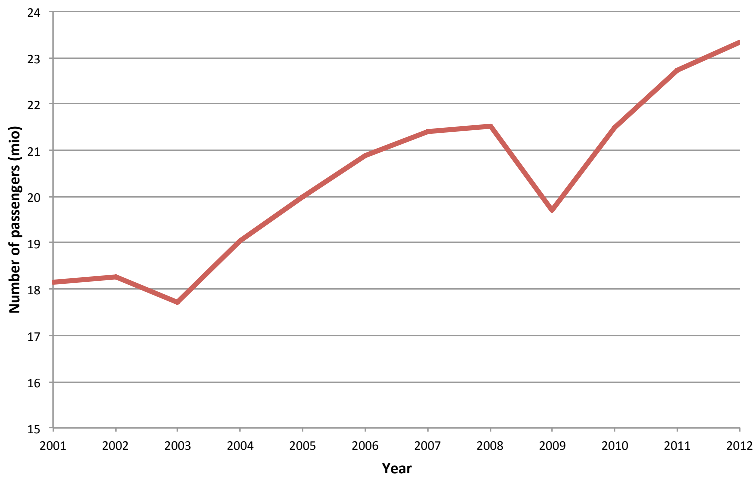

Figure 2.1: *Total annual number of passengers passing through Copenhagen Airport in the period 2001—2012. Source: [CPH, 2013b].*

passengers, airlines, cargo companies, and other business partners. To support the vision, the World Class Hub strategy focuses on the following three primary areas:

- **Extraordinary customer experiences:** CPH wants to give its guests more than they expect, by continuously providing a high level of service and by attracting international brands and trademarks in order to meet the demands of its customers.
- **Competitiveness:** By providing optimal infrastructure and sufficient capacity in all operational areas, CPH will remain competitive compared to its nearest rivals (Amsterdam (AMS), the new airport at Berlin (BER), Hamburg (HAM), and Brussels (BRU)).
- **Efficient operations:** By providing efficient operations and a high level of service, CPH will ensure that passengers, airlines and other customers choose CPH as their preferred gateway. The essential feature of efficient operations is good planning and the ability to create predictability.

The airport covers an area of 12.4 square km and the runway system consists of two parallel runways (headings 22/04) and one transverse runway (headings 12/30). The two parallel runways are the main runways servicing 98% of all arrivals and departures on average per year. The maximum capacity of the runway system is 83 movements (arrivals and departures) per hour. See Figure 2.2 for an illustration of CPH.

Figure 2.2: *Copenhagen Airport (IATA code: CPH, ICAO code: EKCH).*

Chapter 3

Airport Operations Management

The three main components of the air transportation system are airlines, airports, and air traffic management. An *airline* is a company that provides air transport services for traveling passengers and cargo. Airlines vary from those with a single aircraft carrying mail or cargo, through to full-service international airlines operating hundreds of aircraft. The largest airlines in the world carry more than a 100 million passengers per year, and in general, the international airline industry has been an integral part of the creation of a global economy. An *airport* is a location on the ground where aircraft can take off and land and where the handling of passengers, baggage, and cargo can take place. Air traffic management (ATM) encompasses all systems that assist aircraft to depart from an airport, transit airspace, and land at a destination airport. As part of the ATM, all airports must provide an *air traffic control* (ATC) service to control the landing and take-off of aircraft, together with all ground movements occurring within the airport's maneuvering area. The primary purpose of ATC is to prevent collisions and organize and expedite the flow of traffic at the airport.

Until recently, many European airports and airlines were state-owned, which led to a natural division of responsibility for ground handling operations between the airport and the airline. More recently, deregulation and increased privatization of airports in general, has led to the entry of other service providers. Today, most ground handling processes are performed by *ground handling companies* that are subsidized to handle the operations of one or more airlines at the airport. Each ground handling company may offer a specific range of services; this means that each airline contracts several ground handling companies that service their flights and aircraft in various ways. For example, one company may handle check-in, while another company handles refueling. Service operations have economies of scale, so ground handling companies will often provide several ground handling services to stay competitive. This may either be several types of services at the same airport, the same type of service at different airports, or both. To fully achieve the benefits of combined operations, it is essential to utilize cross-operational planning, particularly with respect to similar operations within the same airport.

The handling of commercial passenger aircraft consists of the following processes: landing, arrival, aircraft turn-round, transfer, departure, and take-off. Each process consists of a set of subprocesses involving the handling of cargo, baggage, passengers, and aircraft. For example, the departure process includes the following passenger handling subprocesses: *check-in*, *security check*, *immigration*, and *boarding*. Each subprocess consists of a set of activities and each activity describes a set of tasks that must be completed. For example, for the check-in subprocess, the handling of a given departure represents an activity, and the corresponding tasks include the passenger identity check, registration of bags, seating, and issuing boarding passes. See Figure 3.1 for an illustration of the main processes and their primary subprocesses. The main purpose of *airport operations management* is to ensure the stable and reliable operation of the airport. First of all, the airport must ensure that sufficient capacity is available to accommodate demand in all operational areas. Second, it must continuously monitor and improve the performance of the different processes and their activities.

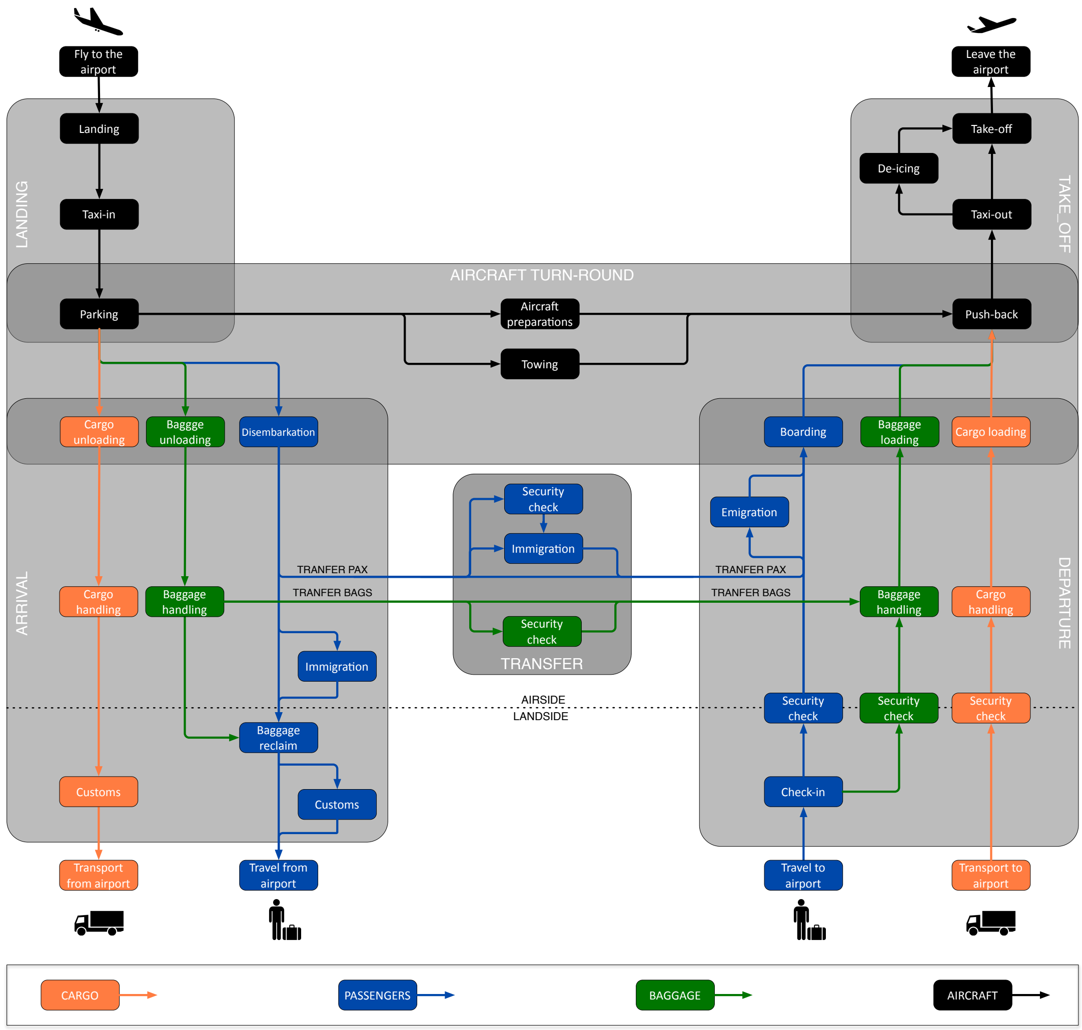

Figure 3.1: *Illustration of the main processes and their primary subprocesses involved in the handling of commercial passenger aircraft at an airport.*

Airport operations management appears in different forms at each of the *strategic*, *tactical*, and *operational* levels. At the strategic level, processes are considered from a long-term planning horizon (e.g. from six months to twenty years) and the primary focus is on capacity assessment in relation to forecast resource demand. In planning for the future, airports must evaluate the infrastructure capacity of not only existing ground handling resources but also modifications to resources they are intending to make.

At the tactical level, processes are considered in a short-/medium-term planning horizon (e.g. from one day to six months). The emphasis is on finding optimal or near-optimal feasible solutions to the different planning problems that arise from the processes. In principle, the tactical level can be seen as the link between strategic decisions made by the airport management and actual operations. In many cases, the tactical solutions to planning problems are disseminated to stakeholders and are even directly deployed in operations. An example of tactical airport operations management is the planning of maintenance. Here the management must plan maintenance such that it has a minimal impact on daily operations, and is completed as fast as possible.

Operational airport operations management refers to the daily monitoring and control of the different airport processes and activities. Conventionally, operational problems are defined as those that occur on a day-to-day basis when predetermined plans become infeasible due to unforeseen disruptions and need to be adjusted. Problems occurring at the operational level often need almost immediate resolution and the aim is to return to feasibility, with a minimum amount of changes to the original plan.

In brief, the strategic level is where the preconditions for an optimal operation are analyzed and decided upon, the tactical level is where the optimal operation is planned, and the operational level is where optimality is maintained as feasibility is restored.

Many airport processes give rise to various planning and optimization problems. In general, problems can be grouped into three main categories: manpower planning, resource allocation, and process optimization.

Manpower planning is an area of constantly increasing importance in an industrialized and knowledge intensive society. In 2013, CPH spent more than DKK 1,000 million on labour cost (see [CPH, 2013a]), i.e. there is a large potential gain in optimizing the usage of manpower. Manpower planning basically consists of two very similar problems: *rostering*, which is the problem of assigning shifts to employees, and *task scheduling*, which is the problem of assigning tasks to employees within shifts. In an airport, tasks occur around the clock and the rostering problem in turn becomes non-trivial, as one needs to consider a range of laws and union rules on rest hours between shifts, maximum number of consecutive working days, etc. Moreover, many tasks are fixed in time or in other ways restricted in time, making the task scheduling problem non-trivial. Rostering and task scheduling are in many aspects similar problems; however, in rostering, shifts of several hours and a planning horizon (the roster period) of a month or more are typically considered, whereas in task scheduling problems, the scheduling horizon is typically 24 hours or less (see [Dohn, 2010]). The rostering problem consists of distributing the shifts between the employees over the days of the roster period, and usually, at most one shift is allocated per employee per day. In task scheduling, each employee is already assigned to a specific shift, and the problem is to allocate a number of tasks to employees such that all tasks are covered and the available manpower is used as efficiently as possible.

As the name indicates, resource allocation is the problem problem of allocating (scarce) resources among alternative activities and tasks. Many different objectives and constraints may be applied when solving resource allocation problems, including maximizing preferences for some resources, minimizing the distance between certain activities, various restrictions to the set of possible resources for the different activities, etc. Examples of resource allocation problems include the allocation of classrooms and lecture halls to classes and lectures at universities and the allocation of jobs to machines in production planning.

Process optimization is a more general term referring to the aspects of the airport operations management where the focus is more on optimizing a single process rather than planning a set of activities.

In the following sections we give a brief introduction to the basic airport terminology and describe each of the subprocesses illustrated in Figure 3.1 in more detail. For each subprocess we describe the primary planning and optimization problems arising from the process. The handling of cargo resembles the handling of baggage, but has not been considered in this thesis.

3.1 Airport terminology

Despite different appearances, most airports follow the same conceptual layout. On a general level, an airport is divided into a *landside* and an *airside* area. The landside area is accessible from the street and includes parking lots, access to public transportation, train stations, and access roads. The airside area includes all areas accessible to aircraft, including maneuvering areas and aprons. Access from landside areas to airside areas is tightly controlled at most airports and everything crossing the border from landside to airside (passengers, baggage, cargo, vehicles, etc.) must pass a security check. The purpose of the security check is to ensure that no items are brought airside that could potentially constitute a risk for the security of flights arriving at, and departing from the airport.

A *maneuvering area* is a part of the airport dedicated to the take-off, landing, and taxiing of aircraft; it includes *runways* and *taxiways*. An *apron* is a designated area for the disembarkation and boarding of passengers, loading and unloading of cargo and baggage, fueling, parking, and maintenance. Demarcated aircraft parking positions on the apron are referred to as *aircraft stands*. An aircraft stand can be approved for many different aircraft types, but can only hold one aircraft at a time. The approval of a stand for a given aircraft type depends on the layout of the stand, the available equipment (Docking Guidance System (DGS), power supply, air supply, fuel pit, etc.), and its accessibility.

An airport may have one or more *terminal buildings*. The landside parts of terminal building typically hold check-in facilities, service desks, and ticket offices, while the airside parts hold *transit areas* with shops, waiting areas, passenger lounges, bars, and restaurants available to departing passengers. Airside areas also include the piers and gates through which passengers can access the aircraft servicing their flights. A *gate* is the passageway through which passengers proceed in order to get from the terminal building to the aircraft (and vice versa). If passengers can walk between the gate and the aircraft (e.g. via a jet bridge), they are said to be *pier-serviced*. If not, they are typically transported by bus, in which case they are said to be *bused*.

Passengers are typically divided into *local arriving passengers*, *transferring passengers*, and *local departing passengers*. For local arriving passengers, the airport is the final airport of their itinerary, whereas transferring passengers use the airport to connect from one flight to another. Local departing passengers use the airport as the initial airport in their itinerary.

Airports with international flights are subject to numerous regulatory requirements. Although many countries have adopted an equal level of security for both international and domestic travel, some countries still require a higher level of e.g. border control. The term *status* refers to the regulatory requirements that a given flight is subject to. With a few exceptions, the status of a given flight and its passengers is determined by the origin/destination airport of the flight. At CPH, the status of a flight can be either *domestic*, *Schengen*, *non-Schengen* (EU, but not part of the Schengen Agreement), or *non-EU*:

Domestic If the country of the origin/destination airport is Denmark, the inbound/outbound status of the aircraft is said to be *domestic*.

Schengen If the country of the origin/destination airport is part of the Schengen Agreement (see Figure 3.2), the inbound/outbound status of the aircraft is said to be *Schengen*.

Non-Schengen If the country of the origin/destination airport is within the EU, but not the Schengen Area, the inbound/outbound status of the aircraft is said to be *non-Schengen*.

Non-EU If the country of the origin/destination airport is not part of the EU and not part of the Schengen agreement, the status of the aircraft is said to be *non-EU*. The only exception to this rule is that passengers arriving from the United States, are considered to have non-Schengen status.

Figure 3.3 illustrates the different regulatory requirements that domestic, Schengen, non-Schengen, and non-EU passengers are subject to at CPH. Depending on its facilities (lounge, security checkpoint, etc.) and placement within the airport (before or after passport control, the ability to separate passengers from each other, etc.), a gate can handle passengers with one or more statuses. Passengers must always either board or disembark the aircraft through a gate that can handle their status. If it is possible to pier-service passengers between a given gate and a given stand, the stand and gate are said to be *linked* for the statuses that can be handled at the gate. For instance, if a given gate, G1, can handle Schengen and non-Schengen passengers, and it is possible to pier-service passengers for aircraft parked at stands S1 and S2, then G1 is linked to S1 and S2. If a stand is linked to one or more gates, it is more commonly referred to as a *contact stand*. Equivalently, stands that are not linked to any gates, i.e. all passengers must be bused regardless of status, are often referred to as *remote stands*. If an aircraft is parked at a contact stand for which it is not possible to pier-service passengers (e.g. domestic passengers on a stand

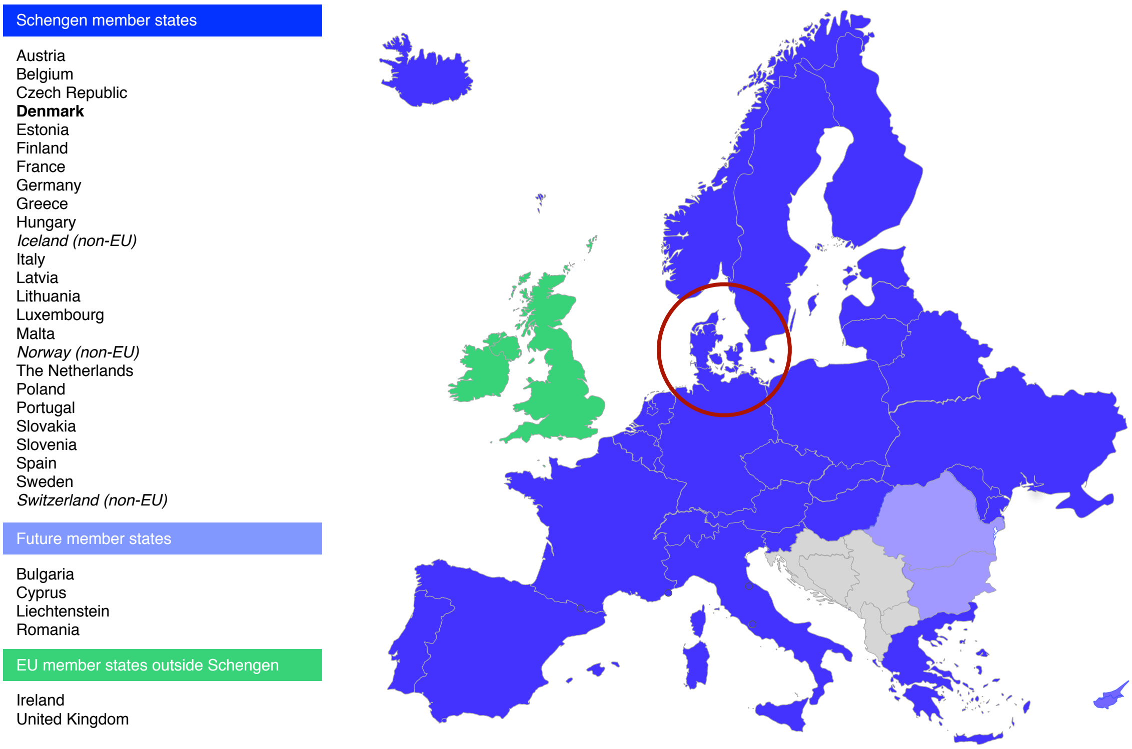

Figure 3.2: *Member countries of the Schengen Agreement (January, 2014). Iceland and Norway are not part of the EU but are part of the Nordic Passport Union and are officially classified as states that are associated with the Schengen activities of the EU. Switzerland was allowed to participate in the same manner in 2008. De facto, the Schengen Agreement also comprises several micro-states which maintain an open or semi-open border with Schengen countries. Greenland and the Faroe Islands are part of the Danish Kingdom, but they are neither members of the EU, nor members of the Schengen Agreement. However they do have an agreement with the EU to ensure that EU citizens can travel to Greenland and the Faroe Islands without showing a passport at the border, i.e. in general, airports in Greenland and the Faroe Islands are considered as Schengen airports. France excludes five overseas departments and all other overseas territories; Spain excludes Ceuta and Melilla; The Netherlands excludes Aruba, Curaçao, Saint Maarten, and the BES Islands (Bonaire, Saint Eustatius, and Saba); and Norway excludes Svalbard.*

that is only linked to Schengen and non-Schengen gates), passengers must also be bused. To avoid using contact stands for buses, many airports have a set of designated *bus gates*, where space for the bus and loading/unloading of the passengers is allocated in front of the gate.

3.2 The landing process

The landing process includes all subprocesses needed to assist aircraft landing at the airport. It includes information given by the ATC to guide the aircraft to a runway, the taxi-in process where the aircraft taxis from the runway to its arrival stand, and the parking process, where the aircraft is finally parked at a stand. The landing process only involves aircraft handling.

3.2.1 Landing

Landing is the last part of a flight, where the aircraft returns to the ground. The aircraft lands by slowing down and descending to the runway. The airport is responsible for ensuring that the runways are always ready for landing aircraft and during winter periods, this includes snow

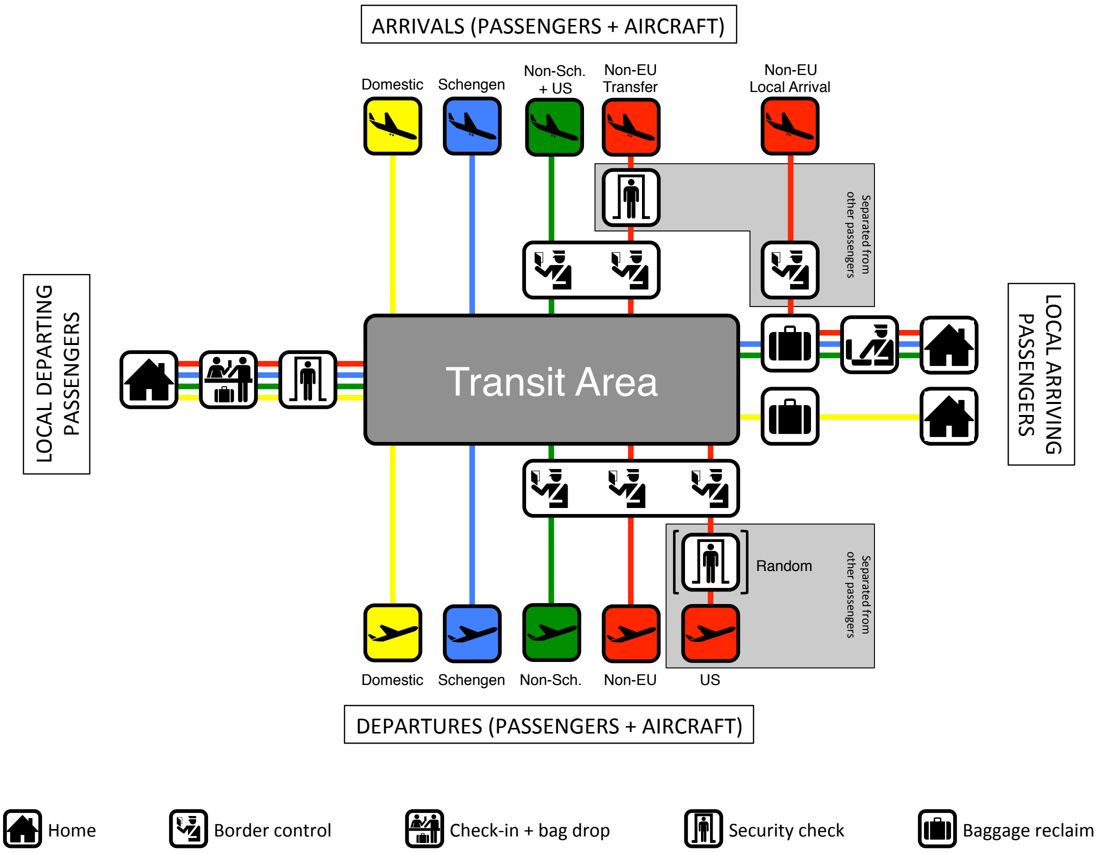

Figure 3.3: *Immigration/emigration and security rules for local arriving, transferring, and local departing passengers at CPH. Local arriving passengers from non-EU airports must be separated from other passengers in the airport until they are landside, and transferring passengers from non-EU airports must be separated from other passengers until they have passed a security check. Moreover, it must be possible to perform a random security check of any passenger departing to the United States. After the security check this passenger must either embark onto the aircraft directly or at least be separated from other passengers in the airport, e.g. in a departure lounge.*

removal and de-icing. For safety reasons, strict separation requirements have to be respected when ATC decides upon the sequence of landing aircraft. The separation is set to avoid the dangers of wake-vortex effects and to control airspace congestion. The length of the minimum required separation depends on aircraft operations (departures or arrivals), the size of the aircraft, and sequencing decisions. For example, a heavy aircraft requires a long separation time before other aircraft can land; on the other hand, a small aircraft generates little air turbulence and, therefore, such an aircraft necessitates only a short separation time.

The primary planning problems of the parking process and the main responsibilities for strategic, tactical, and operational levels are shown in Table 3.1.

Problem	Optimization	Strategic	Tactical	Operational
Runway allocation	Resource	CPH	ATC	ATC
Aircraft landing sequencing	Process	ATC	ATC	ATC
Rostering and task scheduling for ground staff	Manpower	CPH	CPH	CPH

Table 3.1: *Primary planning problems of the landing process.*

3.2.2 Taxi-in

Once the aircraft has landed and left the runway, it must taxi to its arrival stand. In most airports there are multiple taxiways, and these are collectively referred to as the *taxiway network*. The taxiway network is used for both arriving and departing aircraft, and in the ideal situation, all aircraft are routed through the taxiway network in such a way that no aircraft has to wait for other aircraft at any time. However, for many airports, the taxiway network is almost completely saturated during peak hours, and meeting the requirement that each aircraft should be able to taxi through the network without any delays would severely limit the capacity of the network. It is therefore generally accepted that aircraft are assigned intersecting routes and that some aircraft will have to wait for others to pass. The time that elapses from when the aircraft has landed until it has parked at its arrival stand is referred to as the *taxi time*. In general, the goal is to reduce the taxi time as much as possible for all aircraft.

The primary planning problems of the taxi-in process and the main responsibilities for strategic, tactical, and operational levels are shown in Table 3.2.

Problem	Optimization	Strategic	Tactical	Operational
Taxiway route allocation	Resource	CPH	ATC	ATC

Table 3.2: *Primary planning problems of the taxi-in process.*

3.2.3 Parking

Handling an aircraft turn-round is complex; from the time the aircraft lands until it departs, many different ground handling resources are required at different times. During arrival and departure, a stand, gates and gate facilities as well as equipment for unloading and loading baggage and cargo are required. During a push-back, both a push-back tractor and a part of the taxiway network are required. The stand and gate allocation problem is essentially the problem of allocating ground handling resources to turn-rounds, and a solution to problem determines where the aircraft should be parked. At CPH, the airport is responsible for solving the stand and gate allocation problem.

The primary planning problems of the parking process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.3.

Problem	Optimization	Strategic	Tactical	Operational
Stand and gate allocation	Resource	CPH	CPH	CPH

Table 3.3: *Primary planning problems of the parking process.*

3.3 The arrival process

Once an aircraft has landed, it taxis in to a pre-allocated arrival stand, where it then parks. The arrival process starts when the aircraft has parked at its arrival stand. For passengers, the primary subprocesses include disembarkation, immigration control, baggage reclaim, and customs control. For baggage, the primary subprocesses include unloading from the aircraft and the subsequent baggage handling, where arriving bags are sorted into local arriving and transfer bags.

3.3.1 Disembarkation

Once an aircraft has parked at its arrival stand, passengers must *disembark* the aircraft. If the aircraft is parked on a contact stand allowing pier-serviced handling passengers walk from the aircraft to the arrival gate, e.g. via a jet-bridge. At CPH, the jet-bridge is attached to the aircraft by a *gate agent* provided by the handling company taking care of arrivals for the airline. If the aircraft is parked on a remote or on a contact stand not allowing pier-serviced handling, passengers are instead transported by bus to an appropriate arrival bus gate. If there are passengers with reduced mobility (PRM) they are assisted by a PRM handling agent.

The primary planning problems of the disembarkation process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.4.

Problem	Optimization	Strategic	Tactical	Operational
Rostering and task scheduling for gate agents	Manpower	Handlers	Handlers	Handlers
Rostering and task scheduling for bus drivers	Manpower	CPH	CPH	CPH
Rostering and task scheduling for PRM handling agents	Manpower	PRM handlers	PRM handlers	PRM handlers

Table 3.4: *Primary planning problems of the disembarkation process.*

3.3.2 Immigration control

As shown in Figure 3.3, passengers arriving from non-Schengen and non-EU airports must pass an immigration control before entering the baggage reclaim or transit areas. At the immigration control, border police check that the arriving passenger has the appropriate documentation to enter the country and in some cases can stop passengers who are the subject of international arrest warrants. At CPH, the national police are responsible for immigration control, and in principle CPH very limited influence on the process, except at the strategic level, where the job of CPH is to ensure that enough immigration control positions are made available to the police, such that the overall waiting times of passengers are minimized. As the tactical and operational levels, the border police are fully responsible for planning and dispatching police officers to immigration control positions; however, CPH provides them with forecasts of passenger numbers that assist their planning.

The primary planning problems of the immigration control process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.5.

Problem	Optimization	Strategic	Tactical	Operational
Number of immigration control positions	Process	CPH	Police	Police
Rostering and task scheduling for immigration control police officers	Manpower	Police	Police	Police

Table 3.5: *Primary planning problems of immigration control process.*

3.3.3 Baggage claim

Passengers pick up their bags at baggage claim. At CPH, the baggage claim area consists of several racetracks, which are conveyor belt systems that deliver bags to passengers. The baggage claim area also contains airline customer service counters where passengers can report damaged or missing baggage. For international arrivals, the baggage claim area is located in a restricted zone, after immigration control and before clearing customs, so that all baggage can be inspected by customs authorities.

The primary planning problems of the baggage reclaim process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.6.

Problem	Optimization	Strategic	Tactical	Operational
Racetrack allocation	Resource	CPH	Handlers	Handlers

Table 3.6: *Primary planning problems of the baggage reclaim process.*

3.3.4 Customs control

At customs control, customs authorities (CA) check that undeclared goods including animals, personal effects and hazardous item are not brought into the country. Depending on local legislation and regulations, the import or export of some goods may be restricted or forbidden, and the CA enforces these rules. In Denmark, customs controls are implemented through government agreements and international laws. As in many other countries, customs procedures for passengers arriving at CPH are separated into Red and Green Channels; passengers with goods to declare (carrying items above the permitted customs limits and/or carrying prohibited items) should go through the Red Channel, while passengers with nothing to declare (carrying goods within customs limits and not carrying prohibited items) can go through the Green Channel. Passengers going through the Green Channel are only subject to spot checks and save time. However, if a passenger going through the Green Channel is found to have goods above the customs limits or is carrying prohibited items, they may be prosecuted for making a false declaration.

The primary planning problems of the customs control process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.7.

Problem	Optimization	Strategic	Tactical	Operational
Rostering and task scheduling for customs officers	Manpower	CA	CA	CA

Table 3.7: *Primary planning problems of the customs control process.*

3.3.5 Baggage unloading

Once the aircraft has parked at its arrival stand, the bags of passengers are unloaded from the aircraft and transported to the baggage handling facilities. Depending on the aircraft type, the bags are either unloaded individually or in containers.

The primary planning problems of the baggage unloading process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.8.

Problem	Optimization	Strategic	Tactical	Operational
Allocation of equipment	Resource	Handlers	Handlers	Handlers
Rostering and task scheduling for baggage handlers	Manpower	Handlers	Handlers	Handlers

Table 3.8: *Primary planning problems of the baggage unloading process.*

3.3.6 Inbound baggage handling

At the baggage handling facilities, inbound baggage is sorted into local arriving bags and transfer bags. Local arriving bags are loaded onto conveyor belts that take them to racetracks in the baggage claim areas at so-called infeed stations. At CPH, there is a unique infeed station for each racetrack, i.e. the choice of infeed station is determined by the racetrack allocation. Transferring bags are put back into the baggage handling system and from that point on they are considered as local departing bags. In some cases, the connection time between two flights may be so short that it is necessary to handle the transfer bags separately.

The primary planning problems of the inbound baggage handling process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.9.

Problem	Optimization	Strategic	Tactical	Operational
Rostering and task scheduling for baggage handlers	Manpower	Handlers	Handlers	Handlers

Table 3.9: *Primary planning problems of the inbound baggage handling process.*

3.4 The departure process

The departure process comprises all processes required to prepare the departure of a given flight. For passengers, the primary subprocesses include check-in, security checks, emigration control, and boarding.

3.4.1 Check-in

Check-in is usually the first process encountered by passengers when traveling by air. By checking in, the passengers confirm to the airline that they actually have the intention to board the flight for which they have booked a ticket. Moreover, at check-in, passengers have the option to choose, buy or change a seat, register bags, etc. Today check-in can be performed in various ways; online, via common use self-service kiosks (CUSS) at the airport, and via traditional check-in desks, where passengers are assisted by representatives of the airlines (see Figure 3.4). . In most cases, check-in

Figure 3.4: *A typical check-in counter at CPH.*

is performed before passengers reach the security check. At CPH, it is the responsibility of the airport to provide check-in counters, whereas the airlines must provide representatives, usually in the form of handling agents. For an illustration of the typical flow for departing passengers, see Figure 3.5. It is widely acknowledged that the less stressed passengers feel when they reach the

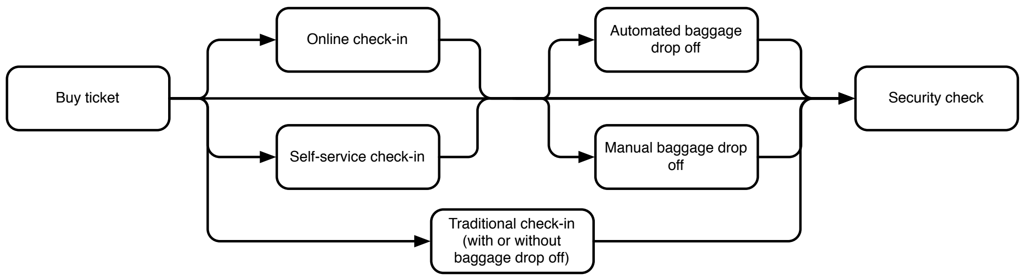

Figure 3.5: *The different possibilities for checking in and dropping off bags when traveling by air. The* traditional check-in *and* manual baggage drop off *processes require check-in counters and staff.*

transit areas of the airport, the more willing they are to spend money in shops and restaurants. Many airports therefore have a strong focus on ensuring that passengers get the best possible experience when checking in. Both the overall waiting time and general flow in terminals are important factors that can positively influence the experience of passengers. Minimizing the waiting time is a question of providing enough check-in counters, but obviously this may lead to excessive operational costs for airlines. As a compromise, CPH therefore accepts that a small proportion of passengers will have to wait in line before they can check-in and it collaborates with airlines to find the best possible counter opening profiles for each check-in. Passenger flow through the terminal is strongly influenced by the placement of individual check-ins and their queuing areas. Some check-ins generate more queues than others, and the aim is therefore to allocate counters to check-ins such that congestion is avoided. Moreover, for operational reasons (such as proximity to the ticket offices), many airlines have strong preferences for certain check-in counters, and the airport tries to respect these preferences when allocating check-in counters.

The primary planning problems of the check-in process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.10.

Problem	Optimization	Strategic	Tactical	Operational
Optimal counter opening profiles	Process	Airlines	Airlines	Airlines
Check-in counter allocation	Resource	CPH	CPH	CPH
Process management	Process	CPH	CPH	CPH
Rostering and task scheduling for check-in handling agents	Manpower	Handlers	Handlers	Handlers

Table 3.10: *Primary planning problems of the check-in process.*

3.4.2 Passenger security check

The purpose of the security check is to ensure that no items are brought airside that could potentially constitute a risk for the security of the flights arriving and departing from the airport. All local departing passengers must be security checked when crossing the border from landside to airside. In many other airports, the security check process has been outsourced to private security companies; however, at CPH the airport is responsible for the security check. At the security check, passengers are screened by metal detectors and their carry-on hand luggage is put through an X-ray machine. Furthermore, local departing and international passengers are checked at the Central Security Check (CSC) consisting of 16 lanes. As for the check-in, the goal of the airport is to minimize overall waiting times for passengers in order that they access the transit areas as quickly as possible, which is a question of providing enough lanes. However, it is clear that this may lead to excessive operational costs for the airport, and CPH therefore accepts that a small proportion of passengers will have to wait in line before they have passed the security check.

The primary planning problems of the security check process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.11.

Problem	Optimization	Strategic	Tactical	Operational
Optimal track opening profiles	Process	CPH	CPH	CPH
Process management	Process	CPH	CPH	CPH
Rostering and task scheduling for security staff	Manpower	CPH	CPH	CPH

Table 3.11: *Primary planning problems of the security check process.*

3.4.3 Emigration control

The emigration process is similar to the immigration process. The border police checks that the passengers have valid documentation allowing them to leave the country and enter the destination country. At CPH, the emigration control is handled by the national police and takes place at the same place as the immigration control. The primary planning problems of the emigration control are the same as for the immigration control, except that now, it is for departing passengers, and not arriving passengers.

3.4.4 Boarding

The boarding process consists of calling the passengers to the appropriate gate and then ensuring that they are safely brought from the terminal buildings to the aircraft. If the aircraft is parked on a contact stand allowing pier-serviced handling, passengers walk from the aircraft to the arrival gate, e.g. via a jet-bridge. At CPH, the jet-bridge is detached from the aircraft by a *gate agent* provided by the handling company taking care of departures for the airline. If the aircraft is parked on a remote or on a contact stand not allowing pier-serviced handling, passengers are instead transported by bus from the gate to the aircraft. If there are passengers with reduced mobility (PRM) they are assisted by a PRM handling agent. For aircraft parked at contact stands, many different boarding strategies may be applied like e.g. back-to-front by row, outside-in by column (window, middle, aisle), or simply "random".

The primary planning problems of the boarding process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.12.

Problem	Optimization	Strategic	Tactical	Operational
Optimal boarding policy	Process	Handlers	Handlers	Handlers
Rostering and task scheduling for gate agents	Manpower	Handlers	Handlers	Handlers
Rostering and task scheduling for bus drivers	Manpower	CPH	CPH	CPH
Rostering and task scheduling for PRM handling agents	Manpower	PRM handlers	PRM handlers	PRM handlers

Table 3.12: *Primary planning problems of the boarding process.*

3.4.5 Baggage security check

All bags must be security checked before they can be loaded onto the aircraft. The purpose of the security check is to ensure that bags do not contain any dangerous items that could potentially constitute a risk to the security of departing flights. At CPH, the airport is responsible for the security check of outbound baggage.

The primary planning problems of the baggage security check process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.13.

Problem	Optimization	Strategic	Tactical	Operational
Rostering and task scheduling for security staff	Manpower	CPH	CPH	CPH

Table 3.13: *Primary planning problems of the baggage security check process.*

3.4.6 Outbound baggage handling

The handling of outbound baggage is often seen as the most complex subprocess in baggage handling. Once a bag has been checked in, it is sent to the baggage handling facilities. On its way it passes several security checks and if the bag is finally approved as not constituting a risk for flight security, it is sent to the sorting system. The sorting system is where all bags are sorted according to their departure. It consists of a long conveyer belt with several tiltable plates and a set of ìmake-upî positions. At the make-up positions, the bags are finally grouped according to their departure and then transported to the aircraft. The bags are moved from the tiltable plates to the make-up position by a chute. Once the tiltable plate passes the corresponding make-up position, it tilts the bag down the chute, and a handling agent lifts the bag onto either a trolley or into a container (see Figure 3.6). If an aircraft requires the bags to be loaded individually, they

Figure 3.6: *A typical make-up position at CPH.*

are transported to the aircraft in trolleys; if the aircraft requires bags to be loaded in containers, they are put into containers, which are loaded directly onto the aircraft. There are three main types of containers (low, medium, and high) and the equipment available at the make-up position, determines which container type(s) it can handle. However, the vast majority of the make-up positions at CPH can be used for trolley loaded baggage.

The primary planning problems of the outbound baggage handling process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.14.

Problem	Optimization	Strategic	Tactical	Operational
Baggage make-up position allocation	Resource	CPH	Handlers	Handlers
Rostering and task scheduling for baggage handlers	Manpower	Handlers	Handlers	Handlers

Table 3.14: *Primary planning problems of the outbound baggage handling process.*

3.4.7 Baggage loading

Once the bags have been sorted at the make-up positions, they are transported to the aircraft either in containers or in trolleys depending on the load type of the aircraft.

The primary planning problems of the baggage loading process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.15.

Problem	Optimization	Strategic	Tactical	Operational
Allocation of equipment	Resource	Handlers	Handlers	Handlers
Rostering and task scheduling for baggage handlers	Manpower	Handlers	Handlers	Handlers

Table 3.15: *Primary planning problems of the baggage loading process.*

3.5 The transfer process

The transfer process encompasses all subprocesses dealing with the transfer of passengers and bags between connecting flights. For passenger handling, the primary subprocesses are the security check of non-EU passengers and immigration control on non-Schengen and non-EU passengers. For baggage handling, the primary subprocess is the security check. In some cases, a short connection time may necessitate that some bags are transferred directly between connecting flights. However, normally, transfer bags are sent to the baggage handling facilities where they are then handled in the same way as local departing bags.

3.5.1 Passenger security check

Although the vast majority of non-EU passengers have already been thoroughly checked before departing for CPH, all transferring non-EU passengers must be security checked again before they can enter the transit areas of CPH. The security check is equivalent to that performed for local departing passengers.

3.5.2 Immigration control

All transferring non-Schengen and non-EU passengers must pass an immigration control before entering the transit area. The immigration control is the same as for local arriving non-Schengen and non-EU passengers.

3.5.3 Baggge security check

As for arriving passengers, all bags arriving from non-EU destinations must be security checked before they can be transferred to their connecting flight. The security check is the same as for local departing bags.

3.6 The aircraft turn-round process

The term *aircraft turn-round* refers to the set of processes taking place while an aircraft is on ground at an airport. The length of a turn-round is defined as the time from when the aircraft parks at its arrival stand until it leaves its departure stand. These processes include the general aircraft preparation including cleaning, catering, fueling, and maintenance and in some cases towings.

3.6.1 Towing

A turn-round can be handled in many different ways. For most turn-rounds, the aircraft is parked at the same stand during its entire stay. However, for operational reasons it may sometimes be necessary to tow an aircraft from one stand to another or even park it on an intermediate remote stand while it is idle. The process of moving an aircraft from one stand to another or to/from the hangar and maintenance areas is called *towing*. Towing can be requested by the airline (e.g. if the aircraft must be moved to a hangar for maintenance) or by the airport (if moving the aircraft improves stand and gate allocation).

The primary planning problems of the towing process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.16.

Problem	Optimization	Strategic	Tactical	Operational
Stand and gate allocation	Resource	CPH	CPH	CPH
Rostering and task scheduling for tractor drivers	Manpower	Handlers	Handlers	Handlers

Table 3.16: *Primary planning problems of the towing process.*

3.6.2 Aircraft preparations

The aircraft preparations process encompasses all activities required to prepare the aircraft for the next flight. This includes among other things, cleaning, catering (food and beverages to be served to passengers), fueling, and in some cases maintenance.

The primary planning problems of the aircraft preparation process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.17.

Problem	Optimization	Strategic	Tactical	Operational
Rostering and task scheduling for cleaning staff	Manpower	Handlers	Handlers	Handlers
Dispatching of catering trucks	Resource	Handlers	Handlers	Handlers
Rostering and task scheduling for catering staff	Manpower	Handlers	Handlers	Handlers
Dispatching of fueling trucks	Resource	Handlers	Handlers	Handlers
Rostering and task scheduling for fueling staff	Manpower	Handlers	Handlers	Handlers
Rostering and task scheduling for aircraft technicians	Manpower	Handlers	Handlers	Handlers

Table 3.17: *Primary planning problems of the aircraft preparation process.*

3.7 The take-off process

The take-off process includes all subprocesses occurring between the time the aircraft is ready to leave its departure stand until it takes-off; they include push-back, taxi-out, de-icing (during winter periods), and finally, the actual take-off. Take-off processes only involve aircraft handling.

3.7.1 Push-back

Before a departing aircraft can taxi-out to the runway, it needs to start its engines. Depending on its engine type and the placement of the stand on the apron, this may happen at the stand. Alternatively, the aircraft is first moved to a designated start-up position by a push-back tractor. In the following we refer to the process that begins when an aircraft is ready to leave its departure stand and ends when it has started its engines and is ready to taxi-out to the departure runway as a *push-back option*. For each stand, multiple push-back options may exist—see Figure 3.7 for an example of the possible push-back options for stand B10 at CPH. Ground handling resources are in

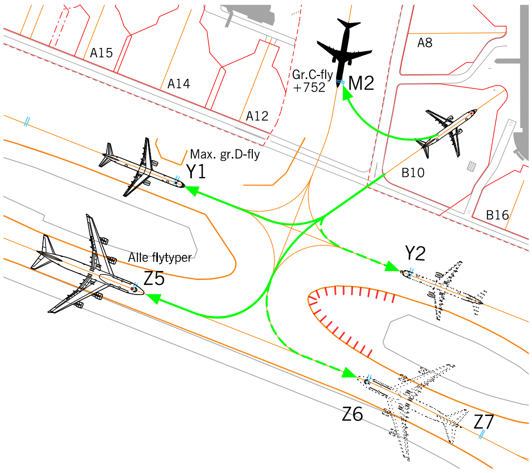

Figure 3.7: *Possible push-back options for stand B10 at Copenhagen Airport. M2, Y1, Y2, Z5, and Z6 denote the different start-up positions at taxiways M, Y, and Z respectively that should preferably be used when starting up aircraft departing from stand B10. Large aircraft must always be pushed to either start-up position Z5 or Z6, whereas small/medium sized aircraft can be pushed to any of the start-up positions. The choice of start-up position is dependent on wind direction, other traffic, and various technical conditions.*

this case the push-back tractor, taxiways and start-up positions. Which push-back option a given aircraft can use depends on the aircraft type, the stand, and various technical and operational conditions. Moreover, the various restrictions regarding simultaneous push-back described in [Haagensen, 2012] must be respected. An example of such a restriction is that two simultaneous push-backs must be separated by at least one stand for small aircraft and two stands for large aircraft, if the aircraft are being pushed back to the same taxiway. In general, for wide-body aircraft with large/multiple engines, the start-up procedure can take up to ten minutes, whereas for smaller aircraft, it only takes 2-4 minutes. The choice of push-back option is thus also dependent on other traffic within the taxiway network.

The primary planning problems of the push-back process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.18.

Problem	Optimization	Strategic	Tactical	Operational
Choice of push-back option	Resource	ATC	ATC	ATC
Dispatching of push-back tractors	Resource	Handlers	Handlers	Handlers
Rostering and task scheduling for tractor drivers	Resource	Handlers	Handlers	Handlers

Table 3.18: *Primary planning problems of the push-back process.*

3.7.2 Taxi-out

Once the aircraft has performed its start-up procedure, it is ready to taxi out to the runway for take-off. As for the taxi-in process, the aircraft may be asked to hold back en route for shorter periods in order to optimize the flow of taxiing aircraft. For a departing aircraft, another option is to hold it at the stand until a conflict-free route through the network is available. Although this minimizes the amount of time it has its engines running, it has the disadvantage that the aircraft is not immediately ready to taxi-out once the path is identified. Unpredicted delays may occur during both the push-back and the subsequent start-up process. As a consequence, many airports practice what is called *push-and-hold* in which the aircraft is first made ready for taxiing and then asked to hold back before continuing, usually at the point where it performed its start-up.

The primary planning problems of the taxi-out process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.19.

Problem	Optimization	Strategic	Tactical	Operational
Taxiway route allocation	Resource	CPH	ATC	ATC

Table 3.19: *Primary planning problems of the taxi-out process.*

3.7.3 De-icing

In winter, it may be necessary to de-ice the aircraft or apply anti-icing before it is allowed to take-off. De-icing is defined as removal of snow, ice or frost from a surface. Anti-icing is the application of chemicals that not only de-ice, but also remain on the surface and continue to delay the reformation of ice, or prevent ice adhesion (which makes mechanical removal easier). At CPH, there are two main de-icing platforms, and aircraft are handled on a first-come, first served basis.

The primary planning problems of the de-icing process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.20.

Problem	Optimization	Strategic	Tactical	Operational
De-icing platform allocation	Resource	CPH	ATC	Handlers

Table 3.20: *Primary planning problems of the de-icing process.*

3.7.4 Take-off

As for landings, sequencing aircraft for take-off must satisfy a set of fixed constraints, which relate to the sizes and routes of successive aircraft. Moreover, it must take into consideration the fact that some departing aircraft may have a "slot restriction", meaning that they must depart within a specific time interval, defined to be between -5 minutes and +10 minutes from their calculated take-off time (CTOT). Aircraft with slot restrictions are required to be at the runway and ready for departure at their CTOT. If the slot is missed (or if it is known in advance that it will be missed), the central flow management unit (CFMU) of EUROCONTROL assigns a new one.

The airport is responsible for ensuring that runways are always ready for the take-off of departing aircraft, and during winter periods, this includes snow removal and de-icing.

The primary planning problems of the take-off process and the main responsibilities for the strategic, tactical, and operational levels are shown in Table 3.21.

Problem	Optimization	Strategic	Tactical	Operational
Runway allocation	Resource	ATC	ATC	ATC
Departure sequencing	Process	ATC	ATC	ATC
Rostering and task scheduling for ground staff	Manpower	CPH	CPH	CPH

Table 3.21: *Primary planning problems of the take-off process.*

3.8 Solving airport planning problems

The overall goal when solving the planning problems described in the previous section is often to exploit existing manpower and ground handling resources to the greatest possible extent. Historically, many planning problems have been solved manually using paper-and-pen or spreadsheet-based methods. However, continuous growth in the amount of traffic and the scale of airport operations means that the planning and control of processes can no longer be adequately addressed by manual planning methods alone. As a consequence, many airports have started to introduce sophisticated decision support tools to assist both planners and airport management in general. Many of these decision support tools are based on operations research techniques or advanced simulation-based methods.

3.8.1 Simulation

Simulation refers to the imitation of an operation of a real-world process or system over time. Typically, a model representing the key characteristics or behaviors of the process is developed and the simulation mimics the operation of the process over time. The general analytical approach is to analyze the simulation of many different scenarios to indicate potential operational challenges such as flow congestion, bottlenecks, inappropriate use of resources, rather than to find an optimal solution for a single or a few scenarios.

Simulation is often used for highly-complex planning problems that contain, for example, nonlinear or stochastic elements. Examples of such planning problems are the runway allocation and departure sequencing problems of the take-off process and more generally, all processes and planning problems involving a flow of passengers, like e.g. the check-in process and the security check process for departing passengers. For examples of the application of simulation to airport operations management problems, see Table 3.22.

Process	Planning problem	Example references
Take-off	Departure sequencing	[Andreussi et al., 1981], [Fleming et al., 2002], [Khoury et al., 2006], [Martinez et al., 2001]
Check-in	Process management	[Appelt et al., 2007], [Chun and Mak, 1999], [Chung and Sodeinde, 2000], [Joustra and Van Dijk, 2001], [Takakuwa and Oyama, 2003], [van Dijk and van der Sluis, 2006]
Passenger security check	Process management	[Castaneda et al., 2007], [Chawdhry, 2009], [Malone et al., 2009], [de Barros and Tomber, 2007], [Leone and Liu, 2011], [Nie et al., 2012], [Pendergraft et al., 2004], [Wilson et al., 2006]

Table 3.22: *Examples of the application of simulation to airport operations management problems.*

3.8.2 Operations research

If an optimal or near-optimal solution to the problem is needed, the usual approach is to apply operations research techniques. Historically, the aviation industry has been a pioneer in the application of operations research in various areas. For airlines, revenue management, aircraft scheduling, and manpower planning are among the areas where the industry has gained substantially from the application of operations research for decision support. Although most airport optimization problems have been thoroughly investigated and various solutions have been implemented and tested, in many ways airports are still in the initial phases of applying operations research to their processes (according to EFOA).

When applying operations research, the optimization problem is typically formulated as either an integer programming (IP) or mixed integer programming (MIP) problem, and this problem is then solved using heuristics, exact methods, or a combination of both.

Exact methods In operations research, the term *exact method* is used for methods that find an optimal solution to the problem if one exists; however the computational time may be very long. Examples of exact methods are *branch-and-bound* and *branch-and-price*, which is a combination of branch-and-bound and *column generation*.

The branch-and-bound approach is based on the principle that the total set of feasible solutions can be partitioned in smaller subsets of solutions. These smaller subsets can then be evaluated systematically until the best solution is found. Starting from a root node, the solution space is split whenever the optimal solution of the relaxed formulation is found to be infeasible in the original formulation. Each of the subspaces is stored in a child node of the root node and the corresponding restricted problem is then solved in the child node. A complete branching scheme has the property that the relaxed solution space is split into two or more subsets, where the union of the subsets contains all feasible solution of the original formulation. The current infeasible solution is not in any of the subsets and hence each of the subsets will provide a new solution, which is either feasible or where another branching decision can be applied. If the lower bound for some child node A is greater than the upper bound for some other child node B, then A may be safely discarded from the search. This step is called pruning, and is usually implemented by maintaining a global variable shared among all nodes of the tree that records the minimum upper bound seen among all child nodes examined so far. For a detailed description of branch-and-bound, see [Wolsey, 1998].

Column generation is a methodology developed specifically for optimization problems with certain characteristics. It has its foundation in mathematical programming and has been shown to perform extremely well on rostering and routing problems, because they naturally decompose into a so-called *master problem* and *pricing problem*. Column generation considers a limited set of columns, which, depending on the context, refer to roster-lines, paths, etc. The master problem combines the available columns to cover a given demand as tightly as possible. The pricing problem is called iteratively, to generate new promising columns. The iterative procedure may be continued until an optimal solution is found. When column generation is embedded in a branch-and-bound structure, the resulting algorithm is referred to as a branch-and-price algorithm.

For examples of the application of exact methods to airport operations management problems, see Table 3.23.

Heuristics Heuristic methods can provide very efficient solutions; however, they are not guaranteed to find an optimal solution, even if one exists. Heuristics are typically used for hard problems, as a way to find high-quality solutions in a reasonable amount of time. Some heuristics mimic the corresponding manual planning process or apply simple rules to build and modify solutions. Heuristics are typically easier to implement than exact methods and they may be the method of choice for that reason. Other heuristics are built from extensive theory and are applied mainly for their computational properties. Examples of heuristics are greedy heuristics and metaheuristics, including simulated annealing, tabu search, and genetic algorithms.

Process	Planning problem	Example references
Landing	Aircraft landing sequencing	[Ernst et al., 1999], [D'Ariano et al., 2010]
Taxi-in/taxi-out	Taxiway route allocation	[Lan et al., 2006], [Marín, 2006], [Marín and Codina, 2008], [Balakrishnan and Yoon, 2007], [Montoya et al., 2010], [Roling and Visser, 2008]
Parking	Stand and gate allocation	[Babic et al., 1984], [Bihr, 1990], [Bolat, 1999], [Bolat, 2000], [Yan and Ho, 2001], [Diepen et al., 2007], [Dorndorf et al., 2007], [Li, 2008], [Diepen et al., 2009], [Jaehn, 2010], [Tang et al., 2010]
Check-in	Optimal counter opening profiles	[van Dijk and van der Sluis, 2006], [Genovese and Bruno, 2010]
	Queue management	[Parlar and Sharafali, 2008]

Table 3.23: *Examples of the application of exact methods to airport operations management problems.*

A greedy heuristic is a heuristic where a solution is found by iteratively making a locally optimal choice to solve a part of the problem. For many problems, a greedy strategy will not yield an optimal solution in the general case, but nonetheless it may be close to a global optimal solution. Greedy heuristics have the advantage that they are easy to implement and are often very efficient.

A metaheuristic is a method which combines a generic methodology with a problem-specific setup. Metaheuristics make few assumptions about the optimization problem being solved, and are therefore useful for a variety of problems. Many metaheuristics implement some form of stochastic optimization, so that the solution found is dependent on the set of random variables generated. By searching over a large set of feasible solutions, metaheuristics can often find good solutions with less computational effort than simple heuristics. Many different metaheuristics exist; one example relies on the definition of neighborhoods. From a feasible solution (or in some cases, an almost-feasible solution) a neighborhood defines a set of similar solutions. Neighborhoods are attractive, as it is often possible to find better solutions in the neighborhood of already good solutions. Neighborhood-based metaheuristics are distinguished by the way they ensure diversification, i.e. how effective they are at exploring large parts of the solution space. For a general introduction to metaheuristics, see [Glover and Kochenberger, 2003].

For examples of the application of heuristics to airport operations management problems, see Table 3.24.

One of the main goals when solving the different planning problems is often to exploit existing manpower and ground handling resources to the greatest possible extent. However, for each problem, many other objectives may exist. A set of operational *performance indicators* (PI) are typically used by the airport to monitor and control the performance of the different airport processes and activities, and obviously, these should also be included in the objective function for the individual problems. Each PI is linked to a target value, in order to evaluate whether it meets expectations or not. The measuring of the PIs and the evaluation of their values is also referred to as *airport performance management.*

3.9 Airport performance management

For airports, performance management typically appears either in the form of *airport benchmarking* and/or *operational performance management.* In airport benchmarking, the performance of multiple airports is compared either at a single point in time or over a period of time. In operational performance management, the airport monitors and controls the performance of daily operations and the focus is typically on both the airport itself and the relationship between the

Process	Planning problem	Example references
Landing	Aircraft landing sequencing	[Amrahov and Ibrahim Alsalihe, 2011]
Taxi-in/out	Taxiway route allocation	[Pesic et al., 2001],[Gotteland and Durand, 2003], [Liu and Guo, 2010], [Liu and Wang, 2011], [Baik et al., 2002], [Clare et al., 2009], [Smeltink and Soomer, 2008], [Atkin et al., 2008]
Parking	Stand and gate allocation	[Mangoubi and Mathaisel, 1985], [Bolat, 2000], [Ding et al., 2004], [Ding et al., 2005], [Drexl and Nikulin, 2008], [Genç et al., 2012], [Gu and Chung, 1999], [Hu and Di Paolo, 2007], [Hu and Di Paolo, 2009], [Lim et al., 2005], [Nikulin and Drexl, 2010], [Pintea et al., 2008], [Şeker and Noyan, 2012], [Wei and Liu, 2007], [Xu and Bailey, 2001], [Cheng, 1997], [Diepen et al., 2007], [Dorndorf et al., 2008], [Dorndorf et al., 2012], [Haghani and Chen, 1998], [Yan and Tang, 2007], [Yan et al., 2011]
Boarding	Optimal boarding policy	[Soolaki et al., 2012],
Check-in	Optimal counter opening profiles	[Yan et al., 2005], [Yan et al., 2008]

Table 3.24: *Examples of the application of heuristics to airport operations management problems.*

airport and the airlines.

3.9.1 Airport benchmarking

As the amount of available data has continued to grow, benchmarking has become an increasingly important strategic performance management tool. Airport benchmarking studies began appearing in the literature in the early 1990s. One of the first such studies was carried out by [Tolofari et al., 1990], who compared airports operated by the British Airport Authority. A series of benchmarking studies have since appeared, with a variety of geographic foci, and covering a diverse set of performance parameters.

According to [International, 2012] and [Humphreys and Francis, 2002], airport benchmarking is a powerful analytical tool, but only when used carefully. When used carelessly, the results may lead to claims that upon further analysis turn out to be far from the truth. One of the reasons for this is that airports operate under very different conditions in terms of aviation activities, commercial activities, site constraints, governance and ownership structure, etc. As a result, the most relevant and useful performance indicators differ between individual airports, making comparison very complex. For example, privatized airports are likely to focus on different financial performance indicators than non-profit government-owned airports. Likewise, larger airports are likely to focus on different performance indicators than smaller airports, and airports with large developable land areas are likely to focus on different performance indicators than tightly-constrained airports in large urban areas, etc.

Even among airports with similar characteristics, managers will have different views regarding which performance indicators are most important, and how many indicators the airport should track. A smaller set of closely-monitored performance indicators (key performance indicators) is likely to be a more effective performance management tool than a larger set of indicators that attract less focus. Furthermore, over time, the performance indicators considered to be most important to the individual airport will change, as new issues arise. A key example of this is the currently evolving area of environmental performance indicators, which until recently was not a key area for performance management for many airports.

To help airports around the world in their performance management and to standardize airport benchmarking, [International, 2012] define a set of 42 performance measures. The measures are divided into six categories, referred to as Key Performance Areas (KPAs). For each performance measure the factors that drive particular results are examined and the types of airports where the measure is applicable are identified. Moreover, the strengths and weaknesses of each measure as a benchmarking tool are discussed.

Each year, the Air Transport Research Society (ATRS) publishes the Global Airport Benchmarking Report (see [ATRS, 2014]). The purpose of this report is to measure and compare the performance of several international airports by evaluating different aspects of airport management and operation, including productivity and efficiency, unit costs and unit competitiveness, financial results, and airport charges. Furthermore, the report examines the relationships between various performance measures and airport characteristics in order to better understand observed differences in airport performance. Based on the 2003 ATRS benchmarking report, [Oum and Yu, 2004] describe how the results can used by airport managers for internal performance comparison and improvement, and they provide a review of the most important literature on airport benchmarking. [Oum and Yu, 2004] describe the benefits of systematically comparing airports in this way; however, they also highlight the challenges in obtaining data from other airports and the amount of time and effort required to perform the study.

Other examples of public airport benchmarking reports include Skytrax reviews and ratings ([Skytrax, 2014]), and the Airport Service Quality (ASQ) reports published by the Airport Council International (ACI). See [ACI, 2014] and [ASQ, 2014] for further details.

Airport benchmarking studies are a very useful tool for airport managers when making strategic decisions about future development and the positioning of the airport in relation to others. However, airport benchmarking reports are difficult to use at the tactical and operational levels, as the level of operational details is usually too low. [Francis et al., 2002] examine how benchmarking is being used by airport managers as a way to compare internal performance and make improvements. Drawing on interviews with airport managers and a survey of the world's top 200 busiest passenger airports, the paper discusses the nature, prevalence and consequences of current benchmarking practices in airports. One of the conclusions of the survey is that although a rich picture emerges of different practices, from which certain general trends can be discerned, there is a potential danger in overestimating the importance of benchmarking. Only 72% of participating airports reported that they were involved in any form of benchmarking. While the survey illustrated that many airports are engaged in Best Practice Benchmarking (46 %) it also indicated that many were not (54 %), and these airports seem to have developed other ways of satisfying their performance management requirements.

As the largest airport in Scandinavia, CPH is naturally always included in the major annual benchmarking reports prepared by the ATRS, ACI, ASQ, and Skytrax. Furthermore, CPH participates in a range of specialized benchmarking groups. As an example, the traffic department participates in benchmarking groups on on-time and winter performance (snow clearance, de-icing, etc.), while the quality department participates in the Airport Quality Club (AQC) benchmarking group which focuses on comparing the quality of the turn-round process, check-in, security, and baggage handling across 12 major European airports. The advantage of such specialized benchmarking groups is that they provide more accurate results as they only focus on a few measures at a time. Moreover, participants have access to a specialized network where experience and knowledge about best and worst practice can be exchanged. See Table 3.25 for examples of awards won by Copenhagen Airport during the last 14 years.

3.9.2 Operational performance management

Unlike airport benchmarking, operational performance management (OPM) focuses on the daily operation of the airport. To monitor and control the performance of different airport processes and activities, a set of operational *performance indicators* (PI) are used. PIs are quantitative indicators that can be used to measure the airport's current operational performance. Each PI is linked to a target value, in order to evaluate whether it meets expectations or not. Typically, an operational PI must meet S.M.A.R.T criteria (see [Doran, 1981]), i.e. it is has a **S**pecific purpose for the business of the airport, it is **M**easurable, the defined targets are **A**chievable, it is **R**elevant to the success of the organization, and finally it is **T**ime phased, which means the value or outcomes are shown for a predefined and relevant period. This requirement ensures that operational PIs

Year	Benchmarking	Award
2013	Danish Travel Awards	Best airport in Denmark
2013	ATRS	European Airport Efficiency Excellence Award
2013	Skytrax	World's best airport security
2012	ATRS	European Airport Efficiency Excellence Award
2012	Skytrax	Best airport in Northern Europe
2012	ASQ	Europe's best airport shopping center
2011	World LCC Congress	World's best low cost airport
2011	Skytrax	Best airport in Northern Europe
2011	ASQ	Europe's best airport shopping center
2011	IATA	Fast travel awards for self check-in efforts
2011	ATRS	European Airport Efficiency Excellence Award
2010	ASQ	Europe's best airport shopping center
2009	ASQ	Europe's best airport shopping center
2009	ATRS	European Airport Efficiency Excellence Award
2008	ATRS	European Airport Efficiency Excellence Award
2005	ATRS	European Airport Efficiency Excellence Award
2005	ASQ	Europe's best airport
2004	ASQ	Europe's best airport
2003	ASQ	Europe's best airport
2002	ASQ	Europe's best airport
2001	ASQ	World's best airport

Table 3.25: *Awards won by Copenhagen Airport.*

describe the current performance of the airport in a simple and easily comprehensible way; at the same time they indicate where corrective action needs to be taken and enable airport management to make decisions.

Usually, it is straightforward to define a range of valid operational PIs for a given airport process. However, defining a manageable set of PIs that can be used in daily operations is more difficult. Many processes involve several stakeholders, and although all stakeholders typically have the same overall goal, they can have conflicting interests with regards to the desired targets of PIs.

As an example, the check-in process involves the airport, airlines and handling companies. The overall goal for both the handling companies and the airport is to ensure that all passengers are checked in in time to reach their departure gate. However, handling companies are typically also concerned about staff costs when planning, and if a reduction in the number of staff required to perform a given check-in only results in a slight increase in passenger waiting time, they are likely to do so. On the other hand, the airport is primarily concerned with passenger satisfaction and flow in their terminals, and prefers to minimize overall waiting times and the demand for queuing areas - no matter what the cost for handling companies.

All airport processes can be viewed in either a strategic, tactical or operational context. Therefore, each corresponding operational PI can be used at either the strategic, tactical or operational level, depending on how it can influence decisions.

As an example, consider a PI that measures the percentage of passengers that board aircraft via a jet-way bridge. At the strategic level, this PI can be used by the airport in discussions related to capacity assessment and overall strategies for passenger handling. At the tactical and operational levels, it can be used actively to improve stand and gate allocation plans, as the percentage of passengers that board the aircraft via a jet-way bridge can be influenced by adjusting the amount of towings and busings such that the specified targets can be met. In contrast, the PI that measures overall stand utilization can only be used at the strategic level, and not the tactical and operational levels. Stand utilization describes the amount of time stands are in use. At the strategic level, this measure can be a good indication of well the airport is exploiting its available resources; it can also be used in comparisons with other airports, i.e. benchmarking. The airport can thus use this PI in capacity assessment analyses to decide if the stand configuration should be changed. However, at the tactical and operational levels, the measure does not contribute much; it is very difficult to influence stand utilization on a daily basis, as it is predetermined by the published traffic schedule and stand allocation plans.

[Enoma and Allen, 2007] present the first part of a larger project with the overall goal of developing a set of operational key performance indicators for safety and security in airports. The work is based on a series of surveys, interviews, and questionnaires. One of their conclusions is that when designing a PI framework, it is important that the proposed indicators are measurable and that data is available for individual PIs.

[Granberg and Munoz, 2013] divide airport operations into five main areas and define the primary business goals. They propose, for each business goal, a set of operational key performance indicators. PIs are ranked according to the results of a survey that was sent out to numerous airport managers in Sweden and Spain. Only 13.3% of participating airports responded to the survey; nevertheless, [Granberg and Munoz, 2013] use it to identify the most important PIs. Unfortunately, [Granberg and Munoz, 2013] do not provide any examples of an actual practical implementation of the PIs that were developed, but their proposition that the set of operational key performance indicators should be linked to both the airport's business goals and real-world operations is important.

In the past couple of years, both researchers and practitioners have recognized that measuring terminal performance through purely quantitative measures is not sufficient. Moreover, innovative techniques for studying passenger needs and their perception of service quality have also been developed. A new generation of terminal assessment models that incorporate issues such as comfort, convenience, and ambience has emerged. [Zidarova and Zografos, 2011] describe how existing models vary according to the type of decisions supported, evaluation perspective, type of measurements, and evaluation approach used. Furthermore, [Zidarova and Zografos, 2011] review the state of the art and current practice regarding methods and techniques to: assess the performance of airport passenger terminals; identify their capabilities and limitations; and propose issues that should be the subject of further research.

With respect to the monitoring and control of strategic goals and targets, CPH participates in both international airport benchmarking studies and conducts an extensive operational performance management exercise. This consists of a Lean-inspired, daily stand-up meeting entitled PULS, and an extensive Service Level Agreement (SLA) established between the airport and the airlines operating at CPH. Both at PULS meetings and in the SLA, a set of operational PIs are evaluated. The operational PIs currently measured at CPH are specified in Table 3.26, however their targets are confidential.

PULS

Every morning at 09:00, representatives of the operational departments of CPH, handling companies and the major airlines meet in the Apron Tower for PULS. The word PULS means *pulse* in Danish, and the purpose of the meeting is to give all operational stakeholders an indication of the current pulse of the airport. This is done partly by a description of current operational circumstances (heavy snow, jet-way-bridge at C10 is out of order, SAS is short of ground handling staff, etc.), and partly by an evaluation of a selected subset of operational PIs (see Table 3.26).

If the value of given PI meets the specified target, no further action needs to be taken. However, if a PI does not meet its target, possible explanations are discussed and the necessary actions are decided upon. For example, if it is known that there will be an unusually high number of people waiting for the arrival of the national football team, the passenger department can start to prepare how to control flows in the terminals, whether certain public announcements should be made, etc.

As the representatives of all operational departments, handling companies, and major airlines are present at this meeting, decision-making is often very efficient and potential misunderstandings are eliminated.

Service Level Agreement

To improve operational performance and enhance the general quality of the airport, a *Service Level Agreement* (SLA) between the airlines and CPH became effective on April 1, 2011. In BL9-15, Appendix 1, the Danish Transport Authority, Aviation (TS) fixed the framework for the SLA and defined the services which should be measured. These are: (1) check-in, (2) security, (3) baggage inbound, (4) baggage outbound, (5) transfer, (6) turnaround, and (7) passenger satisfaction. Note that handlers as such are not part of the SLA, however, many of the PIs of the SLA indirectly measure the performance of handlers.

The SLA defines a set of operational PIs for services that have a significant impact on the flow of passengers, baggage, and aircraft through the airport (hereafter SLA PIs, see Table 3.26). For some SLA PIs a critical target is specified to indicate that it is particularly important. The values of both the normal and critical targets are specified in accordance with the desired service level. If a critical target is not met, it is considered unacceptable for the operation of the airport.

As described earlier, airports operate under different conditions in terms of aviation activities, commercial activities, site constraints, governance and ownership structure, etc. However, a common feature of most airport business models is that a substantial part of their income comes from the charges paid to the airport by the airlines. These include: landing charges, passenger and cargo fees, security, parking and hangar charges, and operational charges (excluding fueling). To motivate both the airport and the airlines to meet the targets of the SLA PIs, a so-called *incentive model* was established as part of the SLA.

The idea of the incentive model is that poor performance by either the airport or the handlers has financial consequences. In practical terms, some of the charges paid by the airlines (hereafter the *SLA funds*) depend on the performance of both the airport and the airlines (handlers). In addition to the normal (and critical) targets, each SLA PI is associated with a financial weight. Based on these financial weights, the values of PIs are converted into a points system, which is then converted into a financial value and paid quarterly to the airlines and their respective ground handlers. The number of points made available each week by CPH is derived from a weighting of the passenger figures for a given week on the basis of an aggregated level over the previous five years. This means that in weeks where there are many passengers there will be more points available than in weeks with fewer passengers. For example, the weighting of week 28 will be based on the traffic share of week 28 calculated from the average over the previous five years. Points are distributed between the airlines and their handlers in accordance with a specified plan.

The incentive model works as follows. First, the performance of CPH is evaluated. If CPH does not meet the critical targets given in the SLA PIs, SLA funds are paid back to the airlines. If CPH does not meet the targets of the SLA PIs but is within the tolerance threshold for all SLA PIs, SLA funds are added to the *Airline Performance Pool*. Next, the performance of the airlines (handlers) is evaluated. If the airlines do not meet their targets, SLA funds are added to the common *SLA Investment Pool*, but if the targets are met, SLA funds are paid back to the airlines.

If CPH meets the targets set in the SLA PIs, CPH keeps the SLA funds, but adds a minimum share to the Airline Performance Pool to ensure a minimal flow of money to either the airlines or the SLA Investment Pool. This shows that the Airline Performance Pool simply acts as an intermediate account for the accumulation of SLA funds to be paid to the airlines - ultimately, SLA funds will either be kept by CPH, sent to the SLA Investment Pool, or paid as compensation to the airlines. See Figure 3.8 for an overview of the incentive model.

In general, it is in the interests of both the airlines and CPH that the funds in the SLA Investment Pool are kept as low as possible, as they demonstrate inadequate performance. However, at the same time, it is in the interests of both parties that the Investment Pool is not empty; consequently, a minimum share of SLA funds is added. The SLA Investment Pool can be used for activities or investments that contribute to improved airline or CPH performance in the services where performance has proved to be below the agreed target(s). The SLA Investment Pool may not be spent on investments in capacity or facilities that the airport would normally provide. The following is the framework for activities that the Investment Pool may be used for:

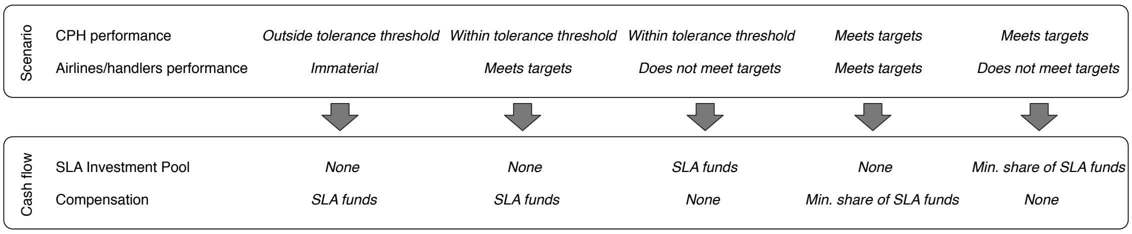

Figure 3.8: *Incentive model defined as part of the service level agreement between the airlines and Copenhagen Airport.*

- Analysis and testing of new technology to replace manual measurements
- Tests and analyses to improve performance in the selected services
- Analyses of whether PIs for additional services should be included under the SLA
- Consultant assistance with analyses
- Training of employees who carry out assignments related to PIs
- Study trips to investigate how other airports carry out PI-related functions
- Costs of auditing the overall SLA agreement

It should be noted that any costs incurred by CPH related to payments to the Airline Performance Pool or any other costs incurred as a result of the SLA cannot be included in aeronautical costs or any other costs or cost base to be paid by the airlines and handlers. The values of the SLA PIs are measured continuously and compiled into a weekly performance report, which is distributed to airline representatives, handling companies, and CPH. Any compensation is paid to the airlines on a quarterly basis.

Since the introduction of the SLA, both the airlines and the airport are more motivated to ensure an efficient, stable, and reliable operation. Weekly performance reports and the daily PULS meeting has increased managers' focus on operations and led to a better collaboration across different CPH departments. Most importantly, management has realized that in order to obtain the best possible performance of the airport, it is crucial that all processes are optimized and that optimizing operations requires sophisticated approaches. In the following chapters, various airport optimization problems are considered. All these problems share the same feature, in as much as their solutions affect a certain part of overall airport operations. Therefore, the quality of their solutions also has an impact on the operational key performance indicators described in Table 3.26. For each optimization problem we describe which operational key performance indicator(s) the problem primarily affects.

Service	Owner	Performance indicator	Unit	PULS	SLA	Frequency
Check-in	CPH	Capacity Take-away belts	%		•	Daily
	Airlines/handlers	Share of pax waiting less than 15 min	%	•	•	Daily
	Airlines/handlers	Share of pax waiting less than 35 min	%	•	•	Daily
Security	CPH	CSC passenger forecast deviation	%	•		Daily
	CPH	CSC - share of pax waiting less than 5 min	%	•	•	Daily
	CPH	CSC - share of pax waiting less than 20 min	%	•	•	Daily
	CPH	Terminal 1 - share of pax waiting less than 5 min	%	•	•	Daily
	CPH	Terminal 1 - share of pax waiting less than 20 min	%	•	•	Daily
	CPH	C23 - share of pax waiting less than 10 min	%	•	•	Daily
	CPH	C23 - share of pax waiting less than 30 min	%	•	•	Daily
	CPH	Staff West - share of pax waiting less than 3 min	%	•	•	Daily
	CPH	Staff West - share of pax waiting less than 10 min	%	•	•	Daily
	CPH	CSRA East - share of pax waiting less than 6 min	%	•	•	Daily
	CPH	CSRA East - share of pax waiting less than 10 min	%	•	•	Daily
Transfer	CPH	Share of bags with barcode read to make-up position in less than 7 min	%		•	Daily
	CPH	Share of bags with in-system time less than 7 min	%			Daily
	Airlines/Handlers	Share of bags with on-block to barcode read in less than 18 min	%		•	Daily
Turn-round	CPH	Jet-bridges availability	%	•	•	Weekly
	CPH	Jet-bridges response time - 90% quantile	Hours		•	Weekly
	CPH	Jet-bridges response time - 80% quantile	Hours		•	Weekly
	CPH	Stand availability (jet bridge, 400Hz, aircraft ventilation)	%		•	Weekly
	CPH	Snow clearing started within 30 minutes of covering snow	NA		•	Weekly
	CPH	Share of delays due to airport facilities (AF)	%		•	Weekly
	Airlines/Handlers	Share of delays due to error at check-in (PE delays)	%		•	Weekly
	Airlines/Handlers	Share of delays due to late acceptance of pax (PD delays)	%		•	Weekly
	Airlines/Handlers	Share of departures on-time ($<$ STD+15min)	%	•		Daily
Baggage inbound	CPH	Share of narrow body aircraft with last bag in less than 30 min	%	•	•	Daily
	CPH	Share of wide body aircraft with last bag in less than 50 min	%	•	•	Daily
	Airlines/Handlers	Share of narrow body aircraft with last bag in less than 30 min	%	•	•	Daily
	Airlines/Handlers	Share of wide body aircraft withe last in less than 50 min	%	•	•	Daily
Baggage outbound	CPH	Major Emergency Procedure - level 0,1,2	# occurrences		•	Weekly
	CPH	Share of bags reaching make-up position more than 20 min before STD	%		•	Daily
	Airlines/handlers	Share of BSMs without bag	%		•	Daily
	Airlines/handlers	Share of "no full notifications" lasting longer than 2 minutes	%		•	Daily
Observed pax satisfaction	CPH	Security - overall (share of pax satisfied)	%	•	•	Monthly
	CPH	Cleaning - overall (share of pax satisfied)	%	•	•	Monthly
	CPH	Terminal ambiance - way-finding (share of pax satisfied)	%	•	•	Monthly
	Airlines/Handlers	Check-in - overall satisfaction	# checks OK	•	•	Weekly
	Airlines/Handlers	Baggage reclaim - overall satisfaction	# checks OK	•	•	Weekly

Table 3.26: *Operational key performance indicators measured at Copenhagen Airport.*

Chapter 4

Allocation of Ground Handling Resources

As can be seen from Chapter 3, the predominant role of CPH is to facilitate airport processes and ensure that sufficient capacity is provided to handling companies, border police, ATC, airlines, and customs authorities. In this chapter, we introduce the problems addressed by this thesis. The problems are all resource allocation problems for which CPH is responsible at either the strategic, tactical, or operational level. For each problem we introduce the problem-specific terminology and describe how the problem is currently addressed at CPH. At the end of the chapter we introduce the basic resource allocation problem which is a more general formulation of the ground handling resource allocation problems considered.

4.1 The check-in counter allocation problem

A check-in group is a group of flights (departures), which share the same physical check-in counters for check-in or baggage drop off. Any check-in group needs a certain number of counters (called the counter demand). This can be constant or vary over time (see Figure 4.1).

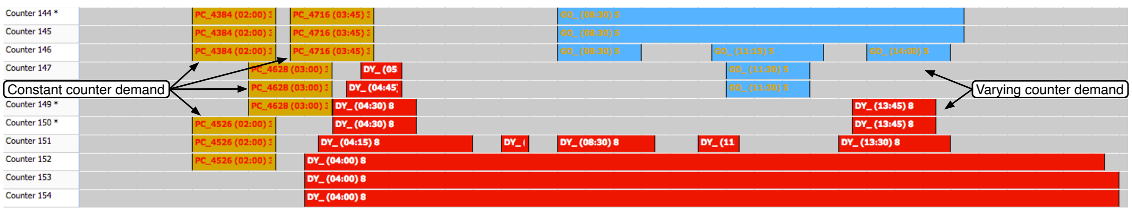

Figure 4.1: *Illustration of the difference between constant and varying counter demand.*

A check-in group can only use the counters that are a set of *allowed* check-in counters that has been specified in advance. Furthermore, some of the allowed counters may be specified in advance as *preferred* for the check-in group in question. The set of allowed and preferred check-in counters reflects the preferences of the airline, and are determined in collaboration with handling companies and the airport.

For each check-in group, the expected number of people in the queue and the expected number of bags to be checked in is calculated for each five minute time interval in the planning period. This is based on predictions made by CPH. Terminals are divided into several queuing areas and a critical level is specified for each queuing area. In the same way, the baggage take-away belts behind the check-in counters that move bags to the baggage handling facilities (security screening and sorting), are divided into several take-away blocks and a critical level is specified for each block. For both queuing areas and take-away blocks, critical levels can be exceeded, although this should be avoided if possible.

The *check-in counter allocation problem* is the problem of allocating check-in groups to check-in counters so that: counter demand is met for all check-in groups; no check-in counter is occupied by more than one check-in group at a time; each check-in group is allocated to its preferred check-in counters if possible; and critical levels in queuing areas and baggage take-away blocks are not exceeded. The check-in counter problem arises as a resource allocation problem for the passenger subprocess check-in, that is part of the overall departure process.

4.1.1 Implementation at CPH

In 2010 work began on designing a more efficient check-in allocation at CPH. Until that point handlers had been solely responsible for the bi-annual allocation of counters. However, the increasing number of passengers at peak times meant that the skills of an experienced planner were no longer sufficient to solve the counter allocation problem. CPH risked running out of capacity if planning continued to be left to the handlers. Consequently, CPH took over responsibility and started to develop an optimization tool that could not only solve the problem, but also provide solutions that could be directly applied into daily operations.

The operational environment of an airport is subject to many regulatory requirements. To ensure stable and reliable operations at all hours of the day and night, all year round, the vast majority of airport processes are standardized and conducted following process descriptions that are specified in advance. Naturally, unwritten practices have developed over time and in general, airport staff, airlines, and handling companies all prefer "things to be done as they have always been done". Consequently, changing, for example, the distribution of responsibilities for a given process or the underlying process description is a difficult task. When taking over responsibility for the check-in counter allocation problem, CPH therefore needed to prove to airlines and handlers that they were capable of solving the problem and that the changes would not lead to further difficulties in their daily operations.

Today, the check-in counter allocation problem is entirely owned by CPH and in principle it arises at strategic, tactical, and operational levels. At the strategic level, the problem is solved for forecast traffic and airport configuration scenarios and can provide detailed information about future check-in counter demand. At the tactical level, the problem is solved by the Operational and Business Analysis (OBA) department, which is responsible for the daily planning of the operational check-in counter allocation. Once the allocation plan has been created for a given day, it is made available to all handling companies and duty managers in the Customer Care department (CCA) via a web-based solution. This shows the latest version of the plan and allows selected users to make manual changes by dragging-and-dropping check-in groups. This shared view of counter allocation was very new when it was introduced in 2010, but despite the usual reluctance to change, it has helped to establish a much better relationship between CPH, airlines and handling companies. Prior to 2010, neither individual handling agents nor duty managers knew anything about each other's planned allocations. This led to a number of conflicts, and often handling companies argued over their right to certain counters. Although very busy days still occur (particularly when Terminal 2 is almost completely saturated during the morning peak) (see Figure 4.2), both airlines and handling companies are in general satisfied with the new setup.

CPH currently has 116 check-in counters distributed across three terminals; five counters in Terminal 1, 79 counters in Terminal 2, and 32 counters in Terminal 3. Terminal 1 only services domestic flights, and as these flights are infrequent the check-in counter allocation problem is only solved for Terminals 2 and 3. Solutions to the check-in counter allocation problem primarily affect/are affected by the following operational KPIs (cf. Table 3.26):

- *Capacity of take-away belts*
- *Proportion of pax waiting less than 15 min*
- *Proportion of pax waiting less than 35 min*

Figure 4.2: *Morning peak in Terminal 2 on June 30, 2012 after CPH took over responsibility for check-in counter allocation. The Gantt chart shows the planned check-in counter allocation and the picture shows passengers queuing for check-in at around 5AM.*

- *Check-in - overall satisfaction*

For further details and a review of the previous work on the problem, see Chapter 8.

4.2 The stand and gate allocation problem

An aircraft turn-round refers to the set of processes taking place while an aircraft is on the ground. The duration of a turn-round is defined as the time that elapses between the moment the aircraft is parked at its arrival stand and when it leaves its departure stand. The processes that take place include cleaning, unloading and loading of catering, fueling, maintenance, disembarkation and boarding of passengers, and unloading and loading of baggage and cargo. Once the aircraft is ready to depart, it is (typically) pushed back from its departure stand and taxis out to a pre-assigned departure runway.

Handling an aircraft turn-round is complex; from the time the aircraft lands, until the moment it departs many different ground handling resources are required at different times. A stand is required for parking. During arrival and departure, gates and gate facilities together with equipment for unloading and loading baggage and cargo are required. During push-back, both a push-back tractor and part of the taxiway network are required. From the airport's perspective, the primary resources to consider are the stands (contact and remote), gates, and taxiways used for push-back. Hangar and maintenance facilities should also be taken into account to some extent, as these provide areas that can be used for parking idle aircraft, thereby freeing regular stands.

A turn-round can be handled in many different ways. For most turn-rounds, the aircraft is parked at the same stand during its entire stay. However, for operational reasons it may sometimes be necessary to tow an aircraft from one stand to another or even park it on an intermediate remote stand while it is idle. If it is not possible to pier-service passengers, they have to be transported by bus between the aircraft and the gates of the terminal building. For each turn-round a *prioritized* set of possible *handling options* is determined by the airline, depending on the length of the turn-round, the aircraft type, etc. (see Figure 4.3).

For various commercial and operational reasons, an airline may prefer that their arrivals and departures are handled at certain stands. Likewise, it may be preferable that some aircraft types or turn-rounds arriving from/departing to specific locations are handled at certain stands. A *stand*

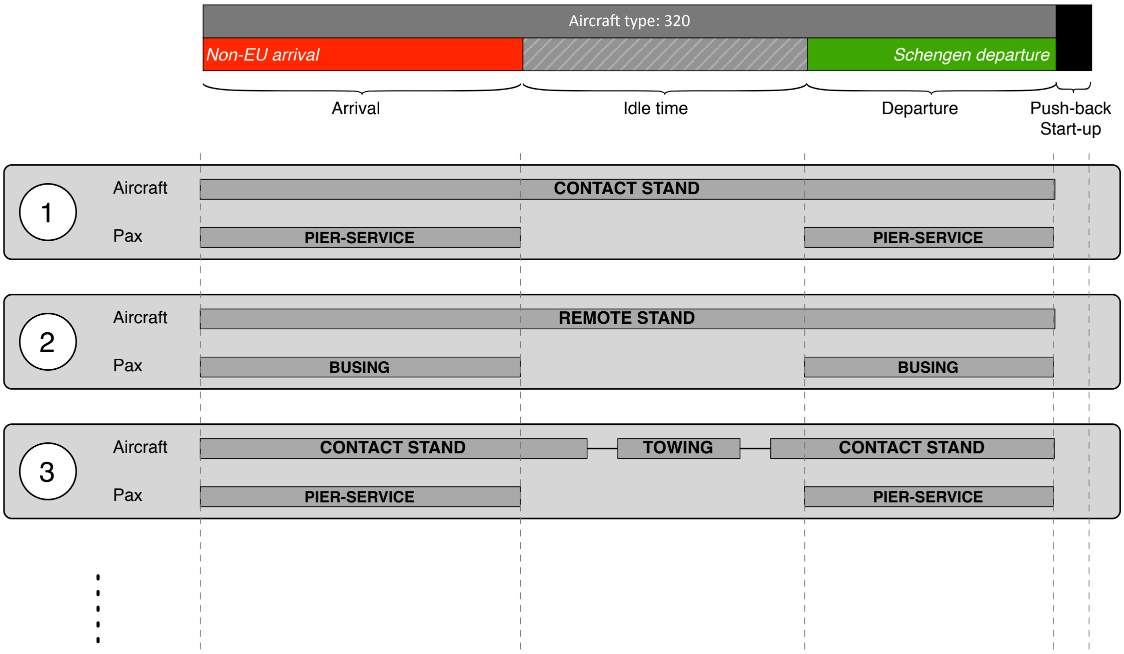

Figure 4.3: *Illustration of different handling options for a given turn-round.*

cluster is a prioritized set of stands describing such stand preferences. If a stand belongs to more than one stand cluster, a ranking of the stand clusters is given for the particular stand.

The handling of a turn-round can be described by a *handling schedule* which is the operationalization of a specific handling option, and describes what ground handling resources are needed and in what time intervals. Depending on the number of handling options, many potential handling schedules may exist for each turn-round (see Figure 4.4).

The *stand and gate allocation problem* is the problem of finding a set of handling schedules consisting of exactly one handling schedule for each turn-round so that: no two turn-rounds simultaneously attempt to claim the same ground handling resource; the best possible handling option (according to priority) is chosen for each turn-round; and stand preferences are satisfied to the greatest possible extent. The stand and gate allocation problem arises as a resource allocation problem for the aircraft subprocess parking, that is part of the overall landing process.

4.2.1 Implementation at CPH

The model and solution developed in this thesis is currently not directly operational, however, the model is solved in order to analyze future stand and gate demand at the strategic level.

CPH is currently using a stand and gate allocation system which they bought in the early 1990s. The system is used for both tactical day-to-day planning and on-the-day operations. The system functions well; it has proven to be very stable and delivers feasible solutions. However, it is a predominantly manual system in the sense that many processes still have to be performed manually, even at the planning level (for example, the allocation of stands to aircraft that must be towed twice). The system is thus more a smart way to display and handle complex data, than a piece of optimization software. CPH is currently considering replacing the current system with a more sophisticated option. The knowledge provided by this thesis will play a substantial role in this project, as it enables CPH to better evaluate and challenge potential new software providers.

CPH has 71 gates, 27 remote stands, and 58 contact stands, out of which 21 are large, wide-body stands. Solutions to the tactical stand and gate allocation problem primarily affect/are affected by the following operational KPIs (cf. Table 3.26):

Figure 4.4: *Illustration of handling schedules for a given handling option for a given turn-round.*

- *Proportion of delays due to airport facilities (AF)*
- *Proportion of delays due to check-in errors (PE delays)*
- *Proportion of delays due to late acceptance of pax (PD delays)*
- *Proportion of on-time departures (< STD+15min)*
- *Jet-bridge availability*
- *Jet-bridge response time - 90% quartile*
- *Jet-bridge response time - 80% quartile*
- *Stand availability (jet-bridge, 400Hz, aircraft ventilation)*
- *Snow clearing started within 30 minutes of covering snow*

- *Proportion of pax waiting less than 15 min*
- *Proportion of pax waiting less than 35 min*
- *Check-in - overall satisfaction*

For further details and a review of the previous work on the problem, see Chapter 9.

4.3 The turn-round re-allocation problem

The operational variant of the stand and gate allocation problem resembles the tactical variant of the problem. The difference is that when solving the operational variant, an infeasible reference solution to the problem is given. The reference solution becomes invalid as a consequence of changes to the timetable, and feasibility must be restored by re-allocating one or more turn-rounds. If a turn-round is involved in a conflict, it is said to be *disrupted*. The operational stand and gate allocation problem arises as a resource allocation problem for the aircraft subprocess parking, partly as part of the overall landing process, partly as part of the aircraft turn-round process, because the aircraft can be moved while on ground (towing).

The point in time when the *turn-round re-allocation problem* is solved is referred to as the *conflict handling time* (CHT) and at CPH, any turn-round arriving or departing within 45 minutes of the conflict handling time is considered critical. At the conflict handling time, some aircraft might already be on the ground at the airport and some turn-rounds might be in the process of being handled. Obviously, it is not desirable, if not actually impossible, to re-allocate these turn-rounds. Furthermore, for turn-rounds arriving within 15 minutes of the conflict handling time, handlers are likely to have already started preparing for their arrival. Therefore, re-allocation is also undesirable in these cases.

To minimize the impact on the reference solution, a *conflict interval* is defined. The conflict interval starts either 15 minutes after the conflict handling time, or at the earliest estimated time of arrival of a disrupted turn-round that is not on the ground. It ends at the latest departure time of a disrupted turn-round that is not on the ground. Only turn-rounds within the conflict interval can be re-allocated, i.e. turn-rounds on the ground at either the start or the end of the conflict interval (or both) cannot be re-allocated and resources that these turn-rounds claim are blocked (see Figure 4.5).

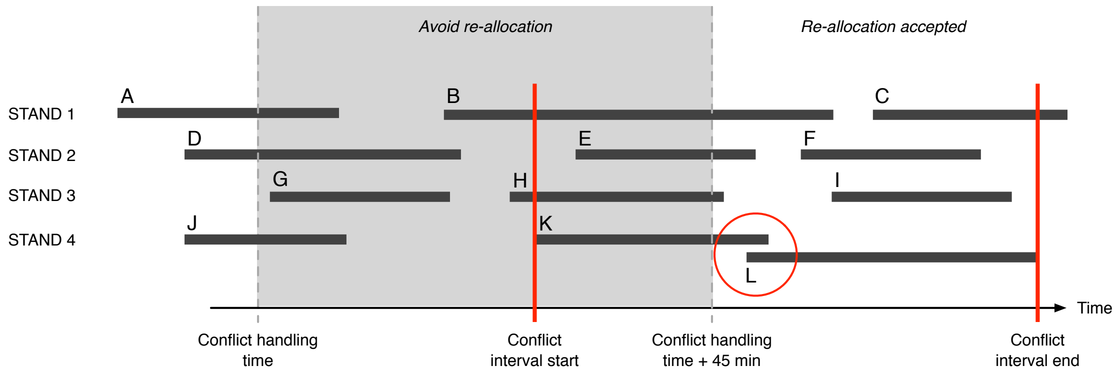

Figure 4.5: *Illustration of a conflict interval.*

4.3.1 Implementation at CPH

As mentioned before, CPH is currently running a stand and gate allocation system that they bought in the early 1990s. On the day of operation, all conflicts arising from small perturbations in timetables or other types of disruptions must be handled manually. The approach described in Chapter 4.3 motivated a new approach to the operational stand and gate allocation problem. Currently, dispatchers create a plan every evening for the next day. When this plan is complete, it is published and from that point on, the despatcher's job is to maintain this plan. The arguments in favor of this approach are multiple. First of all, information about the actual linking of arrivals and departure (the tail assignment) is not available until the very last minute before the day of operations. Airlines prefer to wait for as long as possible before publishing their tail assignments, because this enables them to optimize the use of their fleet and crew. Second, the motivation for preparing stand and gate allocations for an entire day at a time is that these plans are used by airlines and their handling companies to prepare their own staff and equipment planning.

The timetable information used when solving the stand and gate allocation problem is stored in the Airport Operational Database (AODB). Typically, the AODB holds timetable information for a season at a time and usually for at least the following 4-5 months. For the vast majority of operations estimated arrival/departure times, origin/destinations, and assumed aircraft types have already been specified when the seasonal traffic program is loaded into the AODB. As described above, the actual linking of operations is not known until the very last moment before the day of operation. However, for most airlines, the operations of a given airline follow a repetitive pattern over a season; given that the aircraft type is usually known, the linking of operations can usually be forecast based on historical data.

It still remains to investigate how link forecasting should be carried out; in the following sections we assume that a good link forecast can be given for all operations available in the AODB. The idea behind the new approach to operational stand and gate allocation is as follows: instead of only solving the tactical stand and gate allocation problem for the following day, and then maintaining this plan on the day of operation, the problem is continuously solved for all forecast and known turn-rounds in the AODB. Whenever an aircraft swap (tail assignment) is received, the correct linking is established, the stand and gate allocation is updated, and conflicts are resolved. Other updates to the timetable are handled in a similar way. The day-to-day planning of the stand and gate allocation is thus eliminated, and the job of the dispatchers is now "only" to continuously maintain and control the current stand and gate allocation. For timetable updates to non-critical turn-rounds (not arriving and/or departing within the next 45 minutes), the system should simply update the plan silently in the background, whereas for critical turn-rounds (arriving and/or departing within the next 45 minutes), the system should provide suggestions to dispatchers on how the plan can be updated.

The advantage of the approach is that a valid stand and gate allocation is always available for operations in the AODB, and dispatchers can focus on daily operations. Applying the considerations presented in Chapter 4.3, the optimizer will always be working on "small" problems which can be solved efficiently. The salient point, however, is whether is it is possible to provide a good forecast for linking. If the forecast is inadequate, too many changes will need to be handled, rendering the stand and gate allocation completely useless for handlers and airlines. Accurately forecasting linking is thus a potential area for future research that will not be discussed further in this thesis.

The main reason for considering the operational stand and gate allocation problem in this thesis is to demonstrate that the model and solution developed for the tactical stand and gate allocation problem can also be used in an operational context–regardless of the conflict handling strategy. Solutions to the operational stand and gate allocation problem primarily affect/are affected by the following operational KPIs (cf. Table 3.26):

- *Proportion of delays due to airport facilities (AF)*
- *Proportion of delays due to check-in errors (PE delays)*
- *Proportion of delays due to late acceptance of pax (PD delays)*
- *Proportion of on-time departures (< STD+15min)*
- *Jet-bridge availability*

- *Jet-bridge response time - 90% quartile*
- *Jet-bridge response time - 80% quartile*
- *Stand availability (jet-bridge, 400Hz, aircraft ventilation)*
- *Snow clearing started within 30 minutes of covering snow*
- *Proportion of pax waiting less than 15 min*
- *Proportion of pax waiting less than 35 min*
- *Check-in - overall satisfaction*

For further details and a review of the previous work on the problem, see Chapter 10.

4.4 The taxiway route allocation problem

Once an aircraft has landed, it must taxi from the runway to its assigned arrival stand. Likewise, after it has been pushed back and has started its engines, a departing aircraft must taxi out to the runway for take-off. The set of taxiways is collectively referred to as the taxiway network. In general, all arriving and departing aircraft pass through the taxiway network and the goal is to minimize overall taxi times and the number of potential conflicts (where two or more aircraft have to wait for each other).

To model the situation where an aircraft only requires a small part of a taxiway at a time, and where a taxiway can be used by many different aircraft at a time, the taxiway network is discretized into a set of non-overlapping taxiway segments (see Figure 4.6). Each taxiway segment

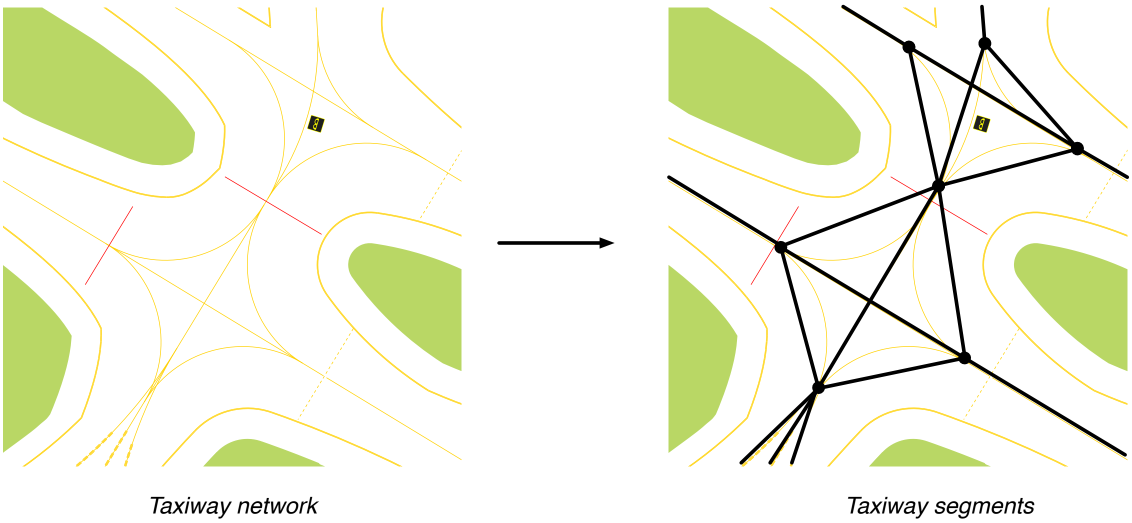

Figure 4.6: *Modeling of taxiway network. Each taxiway is discretized into a set of non-overlapping taxiway segments.*

can be considered as a ground handling resource that can be used by one aircraft at a time. A *route* is a sequence of taxiway segments leading from one point in the taxiway network to another. Depending on the layout of the taxiway network and various aircraft-type restrictions, many possible routes may exist between a given origin-destination pair in the network. A *path* refers to the traversal of a given route in time, i.e. a path consists of the sequence of taxiway segments specified by the route and an indication of the time interval during which each taxiway segment is in use. Two different aircraft can thus be assigned to the same route, as long as their corresponding paths do not use the same taxiway segment at any time.

To control the general flow of the network and to simplify communication both internally at ATC and between ATC, pilots and the airport, ATC at CPH routes aircraft according to a set of *standard routes* that have been pre-defined for each stand and runway pair for arriving and departing aircraft respectively. For a given stand and runway pair, multiple standard routes may exist, and in general, all standard routes are considered equally good, despite the fact that they may have different lengths. The idea is that by using these standard routes, all arriving or departing aircraft roughly follow their respective paths through the taxiway network. Deviations from standard routes are allowed, but should be avoided, as they may confuse pilots. Any deviation must always be communicated between ATC and pilots.

To increase the capacity of the taxiway network, it is generally accepted that arriving and departing aircraft are assigned intersecting routes and that some aircraft have to wait for others at specified *holding points* in the network. Furthermore, it is possible to delay a departing aircraft at the stand before it even enters the taxiway network. The extra time that an aircraft spends in the taxiway network due to delays en-route is referred to as *taxiway delay* and for departing aircraft being delayed at their stand, the waiting time is referred to as the *departure delay.*

Before a departing aircraft can taxi out to the runway, it needs to start up its engines. The process that begins when an aircraft is ready to leave its departure stand and ends when it has started its engines and is ready to taxi out to the departure runway is referred to as a *push-back option.* A push-back option can be described in terms of how taxiway segments are used and are included in the potential paths for departing aircraft. Consequently, the choice of push-back option from a given stand naturally becomes part of the taxiway route allocation problem.

The *taxiway route allocation problem* is the problem of assigning a set of non-conflicting paths to a set of arriving and departing aircraft, so that each aircraft is routed efficiently through the taxiway network in order to minimize taxiway and departure delay. A solution to the problem is thus a specification of how each aircraft should be routed through the taxiway network with an indication of where each aircraft should wait and for how long. The taxiway route allocation problem arises as a resource allocation problem for the aircraft subprocesses taxi-in and taxi-out, that are part of the overall landing respectively take-off processes.

4.4.1 Implementation at CPH

CPH only facilitates the process of routing arriving and departing aircraft through the taxiway network; it owns and maintains the taxiways, but routing and waiting decisions are made by the Air Traffic Control authorities. The problem is highly dynamic and often considered to be a purely operational problem. In this thesis the problem is addressed at the tactical level, and we demonstrate how it can be modeled as a ground handling resource allocation problem.

Like many other airports, CPH does not currently have a framework for analyzing the operational consequences of operations such as planned maintenance to the taxiway network, a change to standard push-back procedures at certain stands, or closure of parts of the network. The purpose of focusing on the problem at the tactical level is to develop such a framework.

For further details and a review of the previous work on the problem, see Appendix A.

4.5 The baggage make-up position allocation problem

Once a bag has been checked in, it is sent to the baggage handling facilities. On its way it passes through several security checks and if the bag is finally approved as not constituting a risk for flight security, it is sent to the sorting system. The sorting system is where all bags are sorted according to their departure. The sorting system consists of a long conveyer belt with several tiltable plates and a set of make-up positions. At the make-up positions, the bags are finally grouped according to their departure and then transported to the aircraft. The bags are led from the tiltable plates to the make-up position by a chute. Once the tiltable plate passes the corresponding make-up position, it tilts the bag down the chute, and a handling agent lifts the bag onto either a trolley or into a container. If an aircraft requires the bags to be loaded individually, the bags are transported to the aircraft in trolleys; if the aircraft requires the bags to be loaded in containers, the bags are put into containers and the containers are loaded directly onto the aircraft. There are three main types of containers (low, medium, and high) and the equipment available at the make-up position, determines which container type(s) it can handle. However, the vast majority of the make-up positions at CPH can be used for trolley loaded baggage.

As for check-in groups, a departure group is defined as a group of flights (departures), which share the same baggage handling facilities (baggage make-up positions) for sorting bags. Any departure group needs a certain number of make-up positions (the make-up position demand) and a specification of the load type of the aircraft. The make-up position demand can be either constant or vary over time. Each departure group can use make-up positions within the defined set of *allowed* positions, and there may also be defined *preferred* positions.

If bags for a given departure group arrive before the corresponding make-up positions have been opened, they are sent to the *early baggage storage system* (EBS). As with all other storage units, the EBS has a limited capacity which must be respected.

The opening and closing times of the make-up positions for each departure group are determined in Service Level Agreements (SLA), negotiated between handling companies and airlines. However, experience has led to an unwritten practice being developed that supplements the SLA. The average opening time is approximately two and a half hours before the scheduled time of departure (STD) of the flight in question, and the closing time is around 15 minutes before the STD, leaving enough time to transport the bags to the aircraft. The opening times of the make-up positions are, to some extent, flexible. To ensure that the EBS capacity is not exceeded, opening times can be adjusted according to the estimated inflow of baggage for each of the individual departure groups.

The *baggage make-up position allocation problem* concerns the determination of opening times and allocating departure groups to make-up positions so that: make-up position demand is met for all departure groups; no make-up position is occupied by more than one departure group at a time; each departure group is allocated its preferred make-up positions if possible; and the capacity of the EBS is not exceeded while still respecting the different SLAs. The baggage make-up position allocation problem arises as a resource allocation problem for the baggage subprocess outbound baggage handling, that is part of the overall departure process.

4.5.1 Implementation at CPH

As can be seen, the baggage make-up position allocation problem is almost identical to the check-in counter allocation problem, but the problem is currently only partly owned by CPH. At the strategic level, the problem is solved for forecast future traffic and airport configuration scenarios can provide information about future make-up position demand. At the tactical level, the problem is solved in the context of short-/medium-term capacity planning and used to analyze different operational scenarios. If, for instance, a given airline wants to change its current set of allowed make-up positions, or has another request, the operational consequences are analyzed using solutions of the make-up position allocation problem. For example, for the summer of 2011, the baggage make-up position problem to pinpoint a number of possible optimizations. Some were already implemented in summer 2011, the most important being earlier opening of make-up positions for selected departure groups as well as fewer make-up positions for other departure groups. CPH aims at actively applying these successful examples as a base for the implementation of a wider range of optimizations.

Daily operational planning is undertaken entirely by handling companies in collaboration with airlines and only to a limited extent with the airport. CPH only has little, if any, influence at the operational level. The baggage make-up position allocation is done by the handlers according to a number of high-level SLAs concerning the number of make-up positions per departure and opening hours of a make-up position, per departure. The SLAs have historically served to provide the necessary service level towards airlines but with the increased traffic, CPH believes that the current service level towards airlines can be sustained with a set of new and improved SLAs.

Baggage facilities at CPH are split into three so-called *baggage factories*. Domestic baggage is handled in Baggage Factory 1 (BF1), while international baggage (including transferring bags) is handled in Baggage Factory 2 (BF2) and Baggage Factory 3 (BF3). BF2 has 17 make-up positions, and BF3 has 119 make-up positions. The total capacity of the EBS is 1,206 bags. Solutions to the baggage make-up position allocation problem primarily affect/are affected by the following operational KPIs (cf. Table 3.26):

- *Proportion of bags reaching their make-up position more than 20 min before STD*
- *Major Emergency Procedure - level 0, 1, 2*
- *Share of "no full notifications" lasting longer than two minutes*

It is important to emphasize that the improved make-up position allocation is at present only partly implemented and that work is on-going on fully deploying the improved allocations into operation. The main focus of CPH is to create an incentive scheme that incentivizes the handler to do what CPH perceives as optimal. This is required as CPH is not able to man make-up positions and thus needs handlers to follow the allocation. CPH is experiencing a high degree of inertia towards the proposed change initiatives and the pace and scope of the implementation clearly reflects this. At the same time, airlines are actively putting pressure on handlers to improve service levels and lower costs. CPH aims at actively using this airline pressure as leverage for a successful implementation by showing the positive results to service and reduced costs of an improved allocation.

The current implementation of the baggge make-up position allocation model has not been fully tested, and as such, the academic work on the problem has not yet sufficiently matured to be published. In relation to this thesis, we thus only present the problem as an example of yet another resource allocation problem that share the same structure and mathematical properties as the other problems described, but details on the problem will not be further elaborated.

4.6 The basic resource allocation problem

The ground handling resource allocation problems described above applies to different airport processes and involves different stakeholders. They can all be seen as variants of the same underlying problem: a set of *activities* each consisting of a set of *tasks* and a set of *ground handling resources*. A task can claim a given resource if is approved, and the allocation of a resource to a given task is associated with a cost that reflects preferences specified by airlines, handling companies, and the airport itself. The *basic resource allocation problem* is then the problem of allocating the set of ground handling resources to the set of tasks so that each resource is dedicated to, at most, one task at a time and total allocation costs are minimized.

Chapter 5

Mathematical Models

In this chapter we first formulate a mathematical model for the basic resource allocation problem. The model is a set-packing based model where the individual ground handling resources are represented over a sequence of time intervals and where the solution is constrained by a set of generalized upper bound (GUB) set-partitioning constraints. Then, we describe how each of the ground handling resource allocation problems introduced in the previous chapter can be formulated as variants of the basic resource allocation problem.

5.1 The basic resource allocation problem

Let $\mathcal{R}$ denote the set of ground handling resources that can be allocated to the tasks of a given set of activities $\mathcal{P}$ of a given airport process.

A *schedule* describes the allocation of resources to the tasks of a given activity $p \in \mathcal{P}$. Since an activity can typically be handled in many different ways, many possible schedules may exist for each activity. Let $\mathcal{S}_p$ denote the set of possible schedules for activity $p \in \mathcal{P}$ and let $\mathcal{S} = \bigcup_{p \in \mathcal{P}} \mathcal{S}_p$ denote the set of all possible schedules for all activities. Each schedule $s \in \mathcal{S}$ is associated with a cost $c_s \in \mathbb{R}_+$ reflecting the preferences for the resources claimed in the schedule and various other operational costs. How the cost of a schedule is defined depends on the given problem.

In the general case, a given resource $r \in \mathcal{R}$ can only be claimed by one task at a time, and two schedules are said to be in *conflict* if they simultaneously claim the same resource. To model this restriction, the planning period of the problem is discretized into a sequence of time intervals, and a set-packing constraint is identified for each resource, for each time interval. In the following, let $\mathcal{I}$ denote the set of time intervals, and let $\mathcal{K}$ denote the set of resource-time interval set-packing constraints, i.e. $|\mathcal{K}| = |\mathcal{R}| \cdot |\mathcal{I}|$. The length of each time interval is dependent on the type of resource and the problem under consideration. For example, to be able to model the nature of the taxiway allocation problem, a rather fine time discretization is needed, whereas a more coarse time discretization may suffice for the stand and gate allocation problem. See Section 5.1.2 for a further discussion of the time discretization approach.

Now, define the binary parameters g_{sp}, where $g_{sp} = 1$ if schedule $s \in \mathcal{S}$ is a schedule for activity $p \in \mathcal{P}$ and $g_{sp} = 0$ otherwise and a_{sk}, where $a_{sk} = 1$ if schedule $s \in \mathcal{S}$ is claiming resource-time interval $k \in \mathcal{K}$ and $a_{sk} = 0$ otherwise. Given the resource-based constraint system defined above, any schedule can thus be represented as a column of the constraint matrix specified by the parameters g_{sp} and a_{sk}.

The basic ground handling resource allocation problem can be stated as the problem of choosing a set of non-conflicting schedules consisting of exactly one schedule for each activity, and with minimum costs. To model this, we define two set of binary decision variables: the allocation variables x_s, where $x_s = 1$ if schedule $s \in \mathcal{S}$ is included in the solution and $x_s = 0$ otherwise and the coverage variables κ_p, where $\kappa_p = 1$ if no schedule is found for activity $p \in \mathcal{P}$ and $\kappa_p = 0$ otherwise. To ensure that it is never preferable to choose a coverage variable in an optimal solution, the cost of a coverage variable, c_κ, should dominate the total cost of the most expensive schedules for each activity, i.e.

$$c_\kappa > \sum_{p\in\mathcal{P}} \max_{s\in\mathcal{S}_p} c_s$$

The basic resource allocation problem can now be formulated as the following integer program (BRAP):

$$\textbf{minimize} \qquad z = \sum_{s\in\mathcal{S}} c_s x_s + \sum_{p\in\mathcal{P}} c_\kappa \kappa_p \tag{5.1}$$

$$\textbf{subject to} \qquad \sum_{s\in\mathcal{S}} g_{st} x_s + \kappa_p = 1 \qquad \forall\, p\in\mathcal{P} \tag{5.2}$$

$$\sum_{s\in\mathcal{S}} a_{sk} x_s \le 1 \qquad \forall\, k\in\mathcal{K} \tag{5.3}$$

$$x_s \in \{0,1\} \qquad \forall\, s\in\mathcal{S} \tag{5.4}$$

$$\kappa_p \in \{0,1\} \qquad \forall\, p\in\mathcal{P} \tag{5.5}$$

The objective function (5.1) minimizes the total cost of scheduling all activities. Constraints (5.2) ensure that all activities are assigned a schedule or canceled (coverage variable equal to 1), whereas Constraints (5.3) enforce the restriction that at most one task can claim any resource at any given time interval. Finally, constraints (5.4) and (5.5) give the binary restrictions on the decision variables. Note, that the constraints 5.2 will ensure that the coverage variables are always either 0 or 1, and the coverage variables could therefore instead simply be stated as positive variables instead of binary:

$$\kappa_p \ge 0 \;\forall\, p\in\mathcal{P}$$

It is obvious that the model implicitly obtains a conflict free set of schedules for the activities. Furthermore, one can also dynamically consider additional schedules for each activity without any fundamental change to the model itself; additional schedules can thus easily be included as extra columns in the model.

5.1.1 Integer properties of BRAP

When solving integer problems, the LP relaxation of the problem is often considered, as finding LP solutions is computationally much faster than finding integer solutions, and there is currently no general way of solving an IP problem without also having the LP relaxation of the problem.

For MIP problems, the underlying integer properties and structure of the model can have a significant impact on how difficult it is to find an optimal solution. According to Definition 3.14 in [Padberg, 1974], an $m\times n$ zero-one matrix $\mathbf{A}$ with $n\le m$ is said to have the *property* $\Pi_{\beta,n}$ if

1. $\mathbf{A}$ contains an $n\times n$ non-singular submatrix $\mathbf{A}_1$ whose row and column sums are all equal to β
2. each row of $\mathbf{A}$ which is not in $\mathbf{A}_1$ is either componentwise equal to a row of $\mathbf{A}_1$ or has row sum strictly less than β

A zero-one matrix is called *perfect* if the polytope of the associated set-packing problem has integral vertices only.[Padberg, 1974] characterizes perfect matrices in terms of forbidden submatrices. Now, consider an $m\times n$ zero-one matrix $\mathbf{A}$. Theorem 3.16 in Padberg [1974], then states that the following conditions are equivalent:

1. $\mathbf{A}$ is perfect
2. For $\beta\ge 2$ and $3\le k\le n$, $\mathbf{A}$ does not contain any $m\times k$ submatrix $\mathbf{A}'$ having the property $\Pi_{\beta,k}$

A *generalized upper bound set-partitioning constraint* (GUB) is a constraint of the form $\mathbf{1}^\top \mathbf{x} = 1$ (see [Rasmussen, 2011]). Now, for a matrix with a GUB constraint, none of the $m \times k$ submatrices A_k, where $3 \leq k \leq n$, can induce a non-singular submatrix $\mathbf{A}'$ with the row sum strictly larger than the row sum corresponding to the GUB constraint. A GUB constraint will thus remove the impact of submatrices with property $\Pi_{\beta,k}$, and we therefore have that any zero-one matrix with a GUB constraint is in fact a perfect matrix (see [Rasmussen, 2011] and [Rezanova, 2009]). Consequently, if the constraint matrix of a given model is a zero-one matrix with a GUB constraint, the LP solution to the model will naturally be integer.

Now, define the zero-one matrix $\mathbf{G} = (g_{sp})$ with $|\mathcal{S}|$ columns and a row for each activity, i.e. the dimensions of $\mathbf{G}$ are $|\mathcal{P}| \times |\mathcal{S}|$, where $g_{sp} = 1$ if column $s \in \mathcal{S}$ is a schedule for activity $p \in \mathcal{P}$ and $g_{sp} = 0$ otherwise. Equivalently, define the zero-one matrix $\mathbf{A} = (a_{sk})$ with $|\mathcal{S}|$ columns and a row for each resource-time interval constraint, i.e. the dimensions of $\mathbf{A}$ are $|\mathcal{K}| \times |\mathcal{S}|$, where $a_{sk} = 1$ if resource-time interval $k \in \mathcal{K}$ is claimed by schedule $s \in \mathcal{S}$ and $a_{sk} = 0$ otherwise. Using the matrices $\mathbf{G}$ and $\mathbf{A}$, the formulation of BRAP given above is equivalent to the one given below:

$$\textbf{minimize} \qquad z = \mathbf{c}_s^\top \mathbf{x} + \mathbf{c}_\kappa^\top \kappa \tag{5.6}$$

$$\textbf{subject to} \qquad \mathbf{G}\mathbf{x} + \mathbf{D}\kappa = \mathbf{1} \tag{5.7}$$

$$\mathbf{A}\mathbf{x} \leq \mathbf{1} \tag{5.8}$$

$$\mathbf{x} \in \{0,1\}^{|\mathcal{S}|} \tag{5.9}$$

$$\kappa \in \{0,1\}^{|\mathcal{P}|} \tag{5.10}$$

where $\mathbf{D}$ is an $|\mathcal{P}| \times |\mathcal{P}|$ diagonal matrix. As can be seen, the constraint matrix of BRAP can be partitioned into $|\mathcal{T}|$ *activity blocks*, where each activity block B_p contains all possible schedules and the coverage variable for a activity $p \in \mathcal{P}$ (see Figure 5.1 for an illustration). As can be seen, each activity block is a zero-one matrix with a generalized upper bound set-partitioning constraint, i.e. each activity block constitutes a *perfect* matrix. The extreme points of the polytope associated with the set-packing problem for each activity block are thus integer, i.e. when solving the LP relaxation of BRAP, fractional solutions will never appear within a single activity block. In an optimal solution to the LP relaxation of BRAP, fractions can therefore only occur between blocks of schedules belonging to different activities. In other words, fractions will only occur, if two or more schedules for different activities are competing for the same resource-time interval.

By construction, the LP relaxation of BRAP thus has good integer properties. We exploit this structure when we develop a solution method for the different ground handling resources allocation problems.

5.1.2 Time discretization

The number of resource-time interval constraints is given by the number of resources and the chosen time discretization of the planning period. Since the resource-time interval constraints enforce that at most one activity can claim any given resource in any given time interval, the time discretization of the planning period should in principle be so fine that it corresponds to the starting times and ending times of the individual activities. Otherwise, the resource-time interval constraints might incorrectly prevent the same resources from being allocated to different activities and thereby reduce the capacity of the individual resources. See Figure 5.2 for an illustration.

Accurately modeling a given problem may thus in some cases necessitate a very fine time discretization of e.g. 1 minute or less, potentially yielding an extremely large number of resource-time interval constraints if the number of resources is very large or the planning period is very long. As an example, consider the stand and gate allocation problem. CPH has 114 stands, 96 gates, and the taxiway network is discretized into 349 taxiway segments. To correctly encapsulate the dynamic nature of a push-back, a time discretization of e.g. 15 seconds is needed. If a planning period of 24 hours is considered (at the tactical level), the model would thus have 3,219,840 resource-time interval constraints.

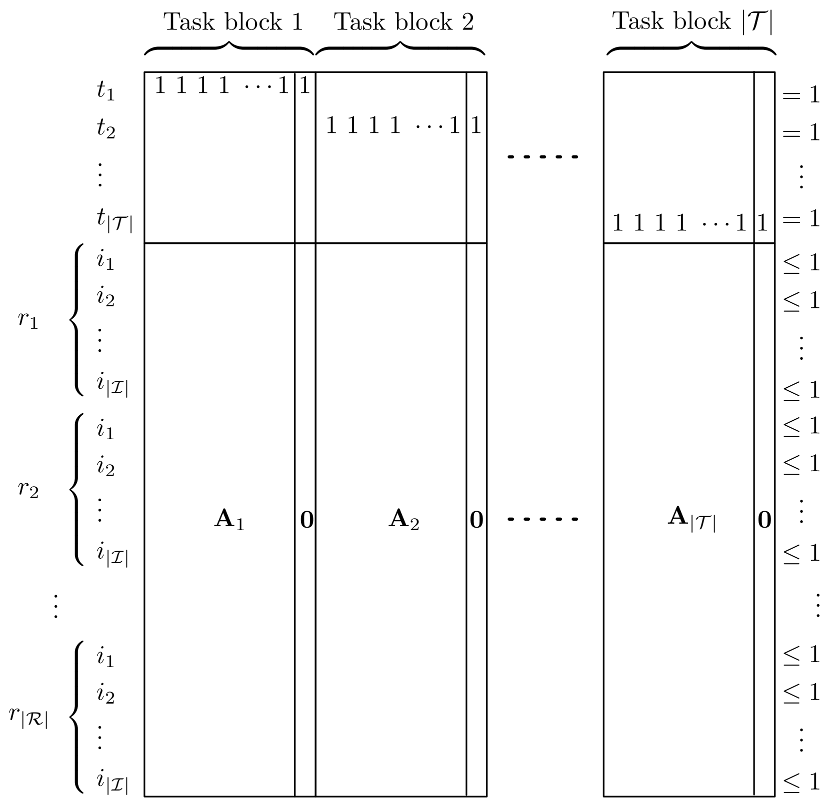

Figure 5.1: *Illustration of the* activity block partition *of the constraint matrix of BRAP.*

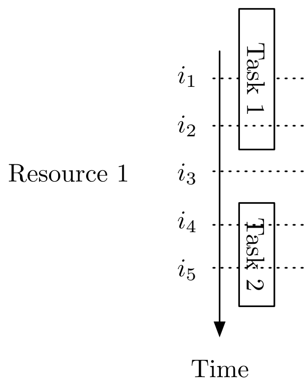

Figure 5.2: *Illustration of a time discretization. If the time intervals are set to start at the times i_2 and i_4, clearly, activity 1 and 2 will not be allocated to resource 1, as this will be prevented by the resource-time interval constraint identified for the time interval starting at time i_2. If however, a time discretization with time intervals starting at the times i_1—i_5, this will no longer be a problem.*

There are multiple ways to address this challenge. One way is to exploit that if a very fine time discretization has been chosen, many of the resource-interval constraints are likely to be redundant, or at least non-binding in an optimal solution. Redundant constraints can be removed from the model without affecting the feasibility and optimality of the solution.

Let $\mathcal{S}_k \subseteq \mathcal{S}$ denote the set of schedules that are claiming resource-time interval constraint

$k \in \mathcal{K}$:

$$\mathcal{S}_k = \{s \in \mathcal{S} \mid a_{sk} = 1\}$$

Clearly, all constraints $k \in \mathcal{K}$ for which $\mathcal{S}_k = \emptyset$ are redundant. Now, consider two constraints $k_1, k_2 \in \mathcal{K}$, $k_1 \neq k_2$. If $\mathcal{S}_{k_1} = \mathcal{S}_{k_2}$, then either k_1 or k_2 is redundant. Equivalently, if $\mathcal{S}_{k_1} \subset \mathcal{S}_{k_2}$, then k_1 is redundant, since by covering k_2, k_1 is automatically also covered. In this case, constraint k_2 is said to *dominate* k_1.

As an example of redundant constraints, consider the constraint matrix illustrated in Figure 5.3 consisting of two resources, r_1 and r_2, each represented in five time intervals, i_1—i_5, and six schedules s_1—s_6. Clearly, the resource-time interval constraints (r_1, i_5) and (r_2, i_5) are redundant as they are not claimed by any schedule. Moreover, constraint (r_1, i_4) is redundant as it is equivalent to the constraint (r_1, i_3), and equivalently, constraint (r_2, i_4) is redundant as it is equivalent to constraint (r_2, i_3). Finally, constraint (r_1, i_1) is redundant as it is dominated by constraint (r_1, i_2) and constraint (r_2, i_1) is redundant as it is dominated by constraint (r_2, i_2). Removing the redundant constraints thus reduces the constraint matrix to consist of the constraints (r_1, i_2), (r_1, i_3), (r_2, i_2), and (r_2, i_3). Assuming that the starting times and the lengths of the different activities are fixed, one could argue that resource-time-interval constraints are only needed for resource-time intervals, where a activity begins. As a consequence, it therefore suffices only to have a resource-time interval constraint for each time interval where a activity begins, i.e. in principle, at most $|\mathcal{T}| \cdot |\mathcal{R}|$ resource-time intervals are needed in order to correctly model the problem.

If the number of activities is much smaller than the number of time intervals, the number of non-redundant resource-time interval constraints will also be much smaller than the total number of resource-time interval constraints. To reduce the problem size, an obvious approach would therefore be to only add the non-redundant constraints to the model. However, as can be seen from above, this approach would require one to have knowledge about all possible schedules, and moreover, the process of analyzing all resource-time interval constraints is not trivial.

Modern solvers like Gurobi and CPLEX per default perform a *presolve* of the problem before applying the actual solution algorithm. During the presolve phase, a collection of problem reductions are applied to reduce and simplify the model, including removing redundant constraints. For LP problems the goal is to reduce problem size, whereas for MIP problems, the goal is not only to reduce problem size, but more importantly to strengthen the formulation of the problem. The problem reductions implemented are usually very sophisticated, and as a consequence, the presolve stage is often very efficient.

As can be seen, the underlying structure of BRAP in principle allows one to choose an arbitrarily fine time discretization. The model can therefore be used for strategical, tactical, and operational problems, without any particular modifications.

		s_1	s_2	s_3	s_4	s_5	s_6
	i_1			1		1	
	i_2	1		1		1	
r_1	i_3	1			1	1	
	i_4	1			1	1	
	i_5						
	i_1				1		1
	i_2		1		1		1
r_2	i_3		1	1			1
	i_4		1	1			1
	i_5						

Figure 5.3: *Constraint matrix showing examples of redundant constraints.*

5.1.3 Reducing the number of constraints

Some airport optimization problems involve many different types of resources. To further control the number of resource-time interval constraints, one option is to differentiate the time discretization of the planning period between the different resource types, reflecting how each type of resource is used in the problem. Once again, consider the stand and gate allocation problem. To correctly encapsulate the dynamic nature of a push-back, a time discretization of e.g. 30 seconds is needed, however, as arrival times and departure times are scheduled to the minutes 0,5,10,..., after the hour at CPH, a time discretization of 5 minutes will suffice for stands and gates.

The number of constraints can also be reduced by allowing resources that are less critical to be claimed by more than one activity at a time, enabling an even coarser time discretization. For instance, one could argue, that an exact planning of push-backs many hours ahead does not really make sense as it is very likely that the actual times of departure will deviate from the scheduled times. In a real-life setting, it will therefore suffice to consider the taxiway resources in time intervals of perhaps 60 seconds and then allow up to two aircraft to simultaneously claim each taxiway resource. This approach will still to some extent encapsulate the needed separation of push-backs from neighboring stands, however, now a little less accurately. If it then actually happens on the day of operation that the two aircraft both need to claim the same taxiway segment during the same 60 second time interval this conflict can be resolved simply by holding back one of the aircraft for a short time.

Let $\mathcal{M}$ denote the set of resource types and let $\mathcal{K}_m$ denote the set of resource-time interval constraints for resources of type $m \in \mathcal{M}$. Finally, let U_m denote the *capacity* of resources of type m, i.e. U_m is the upper bound for the number of activities that may simultaneously claim any resource $r \in \mathcal{R}_m$ in any given time interval $i \in \mathcal{I}_m$. A reduced formulation of the problem is then given by the following integer program (rBRAP):

$$\textbf{minimize} \qquad z = \sum_{s\in\mathcal{S}} c_s x_s + \sum_{p\in\mathcal{P}} c_\kappa \kappa_p \tag{5.11}$$

$$\textbf{subject to} \qquad \sum_{s\in\mathcal{S}} g_{st} x_s + \kappa_p = 1 \qquad \forall\, p \in \mathcal{P} \tag{5.12}$$

$$\sum_{s\in\mathcal{S}} a_{sk} x_s \le U_m \qquad \forall\, m \in \mathcal{M}\ \forall\, k \in \mathcal{K}_m \tag{5.13}$$

$$x_s \in \{0,1\} \qquad \forall\, s \in \mathcal{S} \tag{5.14}$$

$$\kappa_p \in \{0,1\} \qquad \forall\, p \in \mathcal{P} \tag{5.15}$$

Increasing the capacity of a given resource type $m \in \mathcal{M}$ is equivalent to imposing the restriction that any resource of type m can only be covered by a value of $1/U_m$ by each activity. In this case, the constraint matrix of rBRAP is no longer a zero-one matrix, however, since $1/U_m < 1$ the GUB constraints will still ensure that none of the $m \times k$ submatrices A_k, where $3 \le k \le n$, can induce a non-singular submatrix $\mathbf{A}'$ with the row sum strictly larger than the row sum corresponding to the GUB constraint. The activity blocks of rBRAP will thus still be integer, and thereby the constraint matrix of rBRAP will have the same integer properties as BRAP.

5.1.4 Blocked resources

Copenhagen Airport is in operation at all hours of the day and night and all year round. When considering an allocation problem in a given planning period, one or more resources may be *blocked* by tasks of activities that have been allocated in a preceding planning period. Let $\widetilde{\mathcal{K}}$ denote the set of blocked resource-time interval constraints. Obviously, blocked resources cannot be claimed by any task. Including blocked resources in the model can thus be obtained by setting the capacity for these resources to zero in the blocked time intervals, i.e:

$$\begin{aligned}
&\textbf{minimize} && z = \sum_{s\in\mathcal{S}} c_s x_s + \sum_{p\in\mathcal{P}} c_\kappa \kappa_p && && (5.16)\\
&\textbf{subject to} && \sum_{s\in\mathcal{S}} g_{st} x_s + \kappa_p = 1 && \forall\, p\in\mathcal{P} && (5.17)\\
& && \sum_{s\in\mathcal{S}} a_{sk} x_s \le U_m && \forall\, m\in\mathcal{M}\ \forall\, k\in\mathcal{K}_m\setminus\tilde{\mathcal{K}} && (5.18)\\
& && \sum_{s\in\mathcal{S}} a_{sk} x_s \le 0 && \forall\, k\in\tilde{\mathcal{K}} && (5.19)\\
& && x_s\in\{0,1\} && \forall\, s\in\mathcal{S} && (5.20)\\
& && \kappa_p\in\{0,1\} && \forall\, p\in\mathcal{P} && (5.21)
\end{aligned}$$

The set of blocked resources can in principle be used to reduce the number of possible schedules, as all schedules claiming one of the blocked resources $k\in\tilde{\mathcal{K}}$ will never be chosen in an optimal solution.

5.2 The check-in counter allocation problem

For the check-in counter problem, the set of resources, $\mathcal{R}$, are the check-in counters, the set of activities, $\mathcal{P}$, are the individual check-in groups, and the counter demand specify the set of tasks. A schedule is a description of which check-in counters a given check-in group is claiming and in which time intervals. The set of schedules is denoted $\mathcal{S}$. The binary parameter g_{sp} is one if schedule $s\in\mathcal{S}$ is for activity $p\in\mathcal{P}$ and zero otherwise. The binary parameter a_{sk} is one if schedule $s\in\mathcal{S}$ is covering constraint $k\in\mathcal{K}$ corresponding to a given counter $r\in\mathcal{R}$ in a given time interval $i\in\mathcal{I}$ and zero otherwise.

The parameters $\rho_{sqi}\ge 0$ denote number of passengers waiting in line in queueing area $q\in\mathcal{Q}$ in time interval $i\in\mathcal{I}$ for schedule $s\in\mathcal{S}$, and the non-negative surplus variables $y_{qi}\ge 0$ denote the excess of the critical level Q_q for queueing area q in time interval $i\in\mathcal{I}$. Equivalently, the parameters $\beta_{sbi}\ge 0$ denote the number of bags that schedule $s\in\mathcal{S}$ contributes with to baggage take-away block $b\in\mathcal{B}$ in time interval $i\in\mathcal{I}$ and the non-negative surplus variables $z_{bi}\ge 0$ denote the excess of the critical level B_b for baggage take-away block b in time interval $i\in\mathcal{I}$.

At CPH, the parameters β_{sbi} and ρ_{sqi} are assessed in five minute intervals and the opening and closing times of the check-in counters are scheduled on the minutes $0, 5, 10, \ldots, 55$ after the hour. A time discretization of five minutes is therefore suitable to accurately model the problem. Moreover, only one type of resources is considered, namely the check-in counters, and each check-in counter can be claimed by at most one check-in group in each time interval.

The set of allowed counters for check-in group $p\in\mathcal{P}$ is denoted $\mathcal{R}_p\subseteq\mathcal{R}$ and the set of preferred counters for p is $\mathcal{R}_p^+\subseteq\mathcal{R}_p$. The sets of resource-time interval constraints representing these resources are given by the sets $\mathcal{K}_p$ and $\mathcal{K}_p^+$ respectively.

The cost of a schedule $s\in\mathcal{S}$ for a given activity $p\in\mathcal{P}$ is given by the following sum, measuring the use of non-preferred counters:

$$c_s = \sum_{k\in\mathcal{K}_p\setminus\mathcal{K}_p^+} a_{sk}$$

and the parameters $c_q\ge 0$ and $c_b\ge 0$ denote the cost of exceeding the critical levels for queueing areas respectively take-away blocks.

The check-in counter allocation problem can thus be formulated as the following variant of the basic resource allocation problem (CCAP):

$$\textbf{minimize} \quad \omega_1 \sum_{s\in\mathcal{S}} c_s x_s + \omega_2 \sum_{q\in\mathcal{Q}} \sum_{i\in\mathcal{I}} c_q y_{qi} + \omega_3 \sum_{b\in\mathcal{B}} \sum_{i\in\mathcal{I}} c_b z_{bi} + \omega_4 \sum_{p\in\mathcal{P}} c_\kappa \kappa_p \tag{5.22}$$

$$\textbf{subject to} \quad \sum_{s\in\mathcal{S}} g_{st} x_s + \kappa_p = 1 \qquad \forall\, p \in \mathcal{P} \tag{5.23}$$

$$\sum_{s\in\mathcal{S}} a_{sk} x_s \leq 1 \qquad \forall\, k \in \mathcal{K} \tag{5.24}$$

$$\sum_{s\in\mathcal{S}} x_s \rho_{sqi} - y_{qi} \leq Q_q \qquad \forall\, q \in \mathcal{Q}\ \forall\, i \in \mathcal{I} \tag{5.25}$$

$$\sum_{s\in\mathcal{S}} x_s \beta_{sbi} - z_{bi} \leq B_b \qquad \forall\, b \in \mathcal{B}\ \forall\, i \in \mathcal{I} \tag{5.26}$$

$$x_s \in \{0,1\} \qquad \forall\, s \in \mathcal{S} \tag{5.27}$$

$$\kappa_p \in \{0,1\} \qquad \forall\, p \in \mathcal{P} \tag{5.28}$$

$$y_{qi} \geq 0 \qquad \forall\, q \in \mathcal{Q}\ \forall\, i \in \mathcal{I} \tag{5.29}$$

$$z_{bi} \geq 0 \qquad \forall\, b \in \mathcal{B}\ \forall\, i \in \mathcal{I} \tag{5.30}$$

where ω_1, ω_2, ω_3, and ω_4 are non-negative weights ensuring that the objective function maximizes the airline preferences compliance, distributes the queues such that the critical levels of the queueing areas are not exceeded, distributes the inflow of baggage such that the critical levels of the take-away blocks are not exceeded, and minimizes the number of unallocated check-in groups. The constraints (5.23), (5.24), (5.27), and (5.28) constitute the basic resource allocation model. The constraints (5.25), (5.26), (5.29), and (5.30) extend the basic resource allocation model. The constraints (5.25) and (5.26) controls the variables y_{qi} and z_{bi} setting these to the excess for each of the time intervals where the critical levels are exceeded and zero otherwise. Finally, the constraints (5.29) and (5.30) are positivity constraints on the slack variables. For further details, see Chapter 8.

5.3 The stand and gate allocation problem

For the stand and gate allocation problem, the resources, $\mathcal{R}$, are the stands, gates, and taxiway segments. The activities, $\mathcal{P}$, are the turn-rounds and a schedule corresponds to a handling schedule. The set of handling schedules is denoted $\mathcal{S}$. The binary parameter g_{sp} is one if schedule $s \in \mathcal{S}$ is for activity $p \in \mathcal{P}$ and zero otherwise. The binary parameter a_{sk} is one if schedule $s \in \mathcal{S}$ is covering constraint $k \in \mathcal{K}$ corresponding to a given stand, gate, or taxiway $r \in \mathcal{R}$ in a given time interval $i \in \mathcal{I}$ and zero otherwise.

At CPH, the scheduled arrival and departure times are on the minutes $0, 5, 10, \ldots, 55$ after the hour, i.e. a time discretization of five minutes is suitable for the stands and gates when considering the tactical variant of the problem, where scheduled times are used. To encapsulate the dynamics of a push-back, a fine time discretization of 15 seconds is, in principle, needed. However, as described, an exact planning of the push-backs is not required when considering the tactical planning. The taxiway segments are thus considered as less critical resources and a time discretization of 60 seconds and a capacity of two is therefore used for each taxiway segment. The hangar and maintenance areas are the least critical resources, however they should be included as a way of providing additional parking options for idle aircraft. To simplify, it is assumed that the hangar and maintenance areas have a capacity of 15 and a time discretization of 10 minutes is used.

The cost of a schedule $s \in \mathcal{S}$ is given by the following weighted sum:

$$c_s = \omega_1 \frac{\rho_s}{\rho^{max}} + \omega_2 \frac{\theta_s^a + \theta_s^d}{2\theta^{max}} + \omega_3 \frac{\phi_s^a + \phi_s^d}{2\phi^{max}} - \omega_4 \frac{n_s^a + n_s^d}{N}$$

where

- $\rho_p \in 1, \ldots, \rho^{max}$ denotes the priority of the corresponding handling option (1 is the highest and $\rho^{\max}$ the lowest priority)
- $\phi_p^a \in \{1, \ldots, \phi^{\max}\}$ denotes the priority of the arrival stand (1 is the highest priority and $\phi^{\max}$ is the lowest)
- $\phi_p^d \in \{1, \ldots, \phi^{\max}\}$ denotes the priority of the departure stand (1 is the highest priority and $\phi^{\max}$ is the lowest)
- $\theta_p^a \in \{1, \ldots, \theta^{\max}\}$ denotes the ranking of the arrival stand cluster at the arrival stand (1 is the highest priority and $\theta^{\max}$ is the lowest)
- $\theta_p^d \in \{1, \ldots, \theta^{\max}\}$ denotes the ranking of the departure stand cluster at the departure stand (1 is the highest priority and $\theta^{\max}$ is the lowest)
- n_p^a denotes the number of arriving passengers that are pier-serviced
- n_p^d denotes the number of departing passengers that are pier-serviced
- N is the total number of passengers

The first term thus measures the handling preferences compliance, the second and third term measure the stand preferences compliance, and the fourth term measures the number of pier-serviced passengers.

Now, let $\mathcal{R}_{\text{S\&G}} \subseteq \mathcal{R}$ denote the set of all stands and gates, $\mathcal{R}_{\text{TWY}} \subseteq \mathcal{R}$ the set of taxiway segments, and $\mathcal{R}_{\text{H}} \subseteq \mathcal{R}$ the set of hangar and maintenance facilities, and $\mathcal{K}_{\text{S\&G}}$, $\mathcal{K}_{\text{TWY}}$, and $\mathcal{K}_{\text{H}}$ the corresponding resource-time interval constraints. The tactical stand and gate allocation problem can then be formulated as the following variant of the basic model (T-SGAP):

$$\textbf{minimize} \quad \sum_{s \in \mathcal{S}} c_s x_s + \sum_{p \in \mathcal{P}} c_\kappa \kappa_p \tag{5.31}$$

$$\textbf{subject to} \quad \sum_{s \in \mathcal{S}} g_{st} x_s + \kappa_p = 1 \qquad \forall\, p \in \mathcal{P} \tag{5.32}$$

$$\sum_{s \in \mathcal{S}} a_{sk} x_s \leq 1 \qquad \forall\, k \in \mathcal{K}_{\text{S\&G}} \tag{5.33}$$

$$\sum_{s \in \mathcal{S}} a_{sk} x_s \leq 2 \qquad \forall\, k \in \mathcal{K}_{\text{TWY}} \tag{5.34}$$

$$\sum_{s \in \mathcal{S}} a_{sk} x_s \leq 15 \qquad \forall\, k \in \mathcal{K}_{\text{H}} \tag{5.35}$$

$$x_s \in \{0, 1\} \qquad \forall\, s \in \mathcal{S} \tag{5.36}$$

$$\kappa_p \in \{0, 1\} \qquad \forall\, p \in \mathcal{P} \tag{5.37}$$

where the constraints (5.32), (5.33), (5.34), (5.35), (5.36), and (5.37) constitute the basic resource allocation model. For further details, see Chapter 9.

5.4 The turn-round re-allocation problem

The turn-round re-allocation problem resembles the conventional stand and gate allocation problem described above, with only a few differences.

At the operational level, the estimated arrival and departure times are used, and these may be on any of the minutes $0, 1, 2, \ldots, 59$ after the hour; a time discretization of 1 minute is thus more suitable for the stands and gates, when considering the operational variant of the problem. Equivalently, as detailed plans are now needed, the time discretization for taxiway segments is set to 15 seconds and each taxiway segment can be claimed by at most one turn-round. Finally, one or more resources may be blocked by turn-rounds already on ground or arriving within a short time after the conflict handling time.

The cost of a schedule is calculated in the same way as for the stand and gate allocation problem. However, to minimize the number of re-allocations of critical turn-rounds, the binary parameter δ_s, where $\delta_s = 0$ if schedule $s \in \mathcal{P}$ is for a critical turn-round and the arrival and departure stands are equal to the current stand and gate allocation, and $\delta_s = 1$ otherwise, is introduced to the objective function.

Letting $\tilde{\mathcal{K}} \subseteq \mathcal{K}$ denote the blocked resources, the operational variant of the stand and gate allocation problem can then be formulated as the following variant of the basic resource allocation model (O-SGAP):

$$\textbf{minimize} \quad \sum_{s\in\mathcal{S}} \delta_s c_s x_s + \sum_{p\in\mathcal{P}} c_\kappa \kappa_p \tag{5.38}$$

$$\begin{aligned}
\textbf{subject to} \quad & \sum_{s\in\mathcal{S}} g_{st} x_s + \kappa_p = 1 && \forall\, p \in \mathcal{P} && (5.39)\\
& \sum_{s\in\mathcal{S}} a_{sk} x_s \le 1 && \forall\, k \in \mathcal{K}_{\text{S\&G}} \setminus \tilde{\mathcal{K}} && (5.40)\\
& \sum_{s\in\mathcal{S}} a_{sk} x_s \le 1 && \forall\, k \in \mathcal{K}_{\text{TWY}} \setminus \tilde{\mathcal{K}} && (5.41)\\
& \sum_{s\in\mathcal{S}} a_{sk} x_s \le 15 && \forall\, k \in \mathcal{K}_{\text{H}} \setminus \tilde{\mathcal{K}} && (5.42)\\
& \sum_{s\in\mathcal{S}} a_{sk} x_s \le 0 && \forall\, k \in \tilde{\mathcal{K}} && (5.43)\\
& x_s \in \{0,1\} && \forall\, s \in \mathcal{S} && (5.44)\\
& \kappa_p \in \{0,1\} && \forall\, p \in \mathcal{P} && (5.45)
\end{aligned}$$

For further details, see Chapter 10.

5.5 The taxiway route allocation problem

For the taxiway route allocation problem the resources, $\mathcal{R}$, are the taxiway segments and the activities, $\mathcal{P}$, are the arriving and departing aircraft that need to pass through the network. A schedule corresponds to a path. The set of possible paths is denoted $\mathcal{S}$ and the binary parameter a_{sk} is one if schedule $s \in \mathcal{S}$ is covering constraint $k \in \mathcal{K}$ corresponding to a given taxiway segment $r \in \mathcal{R}$ in a given time interval $i \in \mathcal{I}$ and zero otherwise. To correctly encapsulate the dynamic nature of taxiing aircraft, a time discretization of 15 seconds is used.

The binary parameter α_s is one if schedule $s \in \mathcal{S}$ represents the path of an arriving aircraft and zero otherwise. The non-negative parameters $\delta_s^T \ge 0$ and $\delta_s^S \ge 0$ denote the amounts of taxiway delay respectively stand delay for schedule $s \in \mathcal{S}$ and the cost of s is then given by the following weighted sum:

$$c_s = \omega_1 \alpha_s \delta_s^T + \omega_2 (1-\alpha_s)\delta_s^T + \omega_3 \alpha_s \delta_s^S + \omega_4 (1-\alpha_s)\delta_s^S$$

The taxiway route allocation problem can now be formulated a the following variant of the basic resource allocation model (TRAP):

$$\textbf{minimize} \quad \sum_{s\in\mathcal{S}} c_s x_s + \sum_{p\in\mathcal{P}} c_\kappa \kappa_p \tag{5.46}$$

$$\begin{aligned}
\textbf{subject to} \quad & \sum_{s\in\mathcal{S}} g_{st} x_s + \kappa_p = 1 && \forall\, p \in \mathcal{P} && (5.47)\\
& \sum_{s\in\mathcal{S}} a_{sk} x_s \le 1 && \forall\, k \in \mathcal{K} && (5.48)\\
& x_s \in \{0,1\} && \forall\, s \in \mathcal{S} && (5.49)\\
& \kappa_p \in \{0,1\} && \forall\, p \in \mathcal{P} && (5.50)
\end{aligned}$$

where the constraints (5.47), (5.48), (5.49), (5.50) constitute the basic model. For further details, see Appendix A.

5.6 The baggage make-up position allocation problem

For the baggage make-up position allocation problem, the set of resources, $\mathcal{R}$, are the make-up positions, the activities, $\mathcal{P}$, are the individual departure groups, and the resource demand is given by the make-up position demand. A schedule is a description of which make-up positions a given departure group is claiming and in which time intervals. The set of schedules is denoted $\mathcal{S}$. The binary parameter g_{sp} is one if schedule $s \in \mathcal{S}$ is for departure group $p \in \mathcal{P}$ and zero otherwise. The binary parameter a_{sk} is one if schedule $s \in \mathcal{S}$ is covering constraint $k \in \mathcal{K}$ corresponding to a given make-up position $r \in \mathcal{R}$ in a given time interval $i \in \mathcal{I}$ and zero otherwise.

In addition to the constraints of the the basic model, the model includes constraints ensuring that the capacity E of the early baggage storage (EBS) cannot be exceeded. For each schedule $s \in \mathcal{S}$ the parameter $\varepsilon_{si} \geq 0$ denotes the number of bags that s contributes with to the EBS in time interval $i \in \mathcal{I}$.

At CPH, the parameters ε_{si} are calculated in five minute intervals and the opening and closing times of the baggage make-up positions are scheduled on the minutes $0, 5, 10, \ldots, 55$ after the hour. A time discretization of five minutes is therefore suitable to accurately model the problem. Moreover, only one type of resources is considered, namely the baggage make-up positions, and each make-up position can be claim by at most one departure group in each time interval.

The set of allowed make-up positions for departure group $p \in \mathcal{P}$ is denoted $\mathcal{R}_p \subseteq \mathcal{R}$ and the set of preferred make-up positions for t is $\mathcal{R}_p^+ \subseteq \mathcal{R}_p$. The sets of resource-time interval constraints representing these resources are given by the sets $\mathcal{K}_p$ and $\mathcal{K}_p^+$ respectively.

The cost of a schedule $s \in \mathcal{S}$ for a given activity $p \in \mathcal{P}$ is given by the following sum, measuring the use of non-preferred counters:

$$c_s = \sum_{k \in \mathcal{K}_p \setminus \mathcal{K}_p^+} a_{sk}$$

The baggage make-up position allocation problem can thus be formulated as the following variant of the basic resource allocation problem:

$$\begin{aligned}
&\textbf{minimize} && \sum_{s \in \mathcal{S}} c_s x_s + \sum_{p \in \mathcal{P}} c_\kappa \kappa_p && && (5.51)\\
&\textbf{subject to} && \sum_{s \in \mathcal{S}} g_{st} x_s + \kappa_p = 1 && \forall\, p \in \mathcal{P} && (5.52)\\
& && \sum_{s \in \mathcal{S}} a_{sk} x_s \leq 1 && \forall\, k \in \mathcal{K} && (5.53)\\
& && \sum_{s \in \mathcal{S}} \varepsilon_{si} x_s \leq E && \forall\, i \in \mathcal{I} && (5.54)\\
& && x_s \in \{0,1\} && \forall\, s \in \mathcal{S} && (5.55)\\
& && \kappa_p \in \{0,1\} && \forall\, p \in \mathcal{P} && (5.56)
\end{aligned}$$

where the objective function maximizes the airline preferences compliance and minimizes the number of unallocated departure groups. The constraints (5.52), (5.53), (5.55), and (5.56) constitute the basic resource allocation model, BRAP. The constraints (5.54) ensure that the capacity of the EBS is not exceeded at any time.

Chapter 6

Solution Methods

When solving airport optimization problems, a range of real-world aspects must be taken into consideration. In general, the success of a solution approach can be measured by its **applicability**, **impact**, and **acceptance**.

The *applicability* describes how well the solution approach matches the working procedures of the operational environment in which the problem arises. For instance, it is a common misunderstanding that run times are allowed to be long for strategical and tactical problems, simply because they are dealing with future planning and consider relatively long time periods. At the strategical and tactical levels, the airport typically wants to analyze multiple different scenarios and make decisions based on these analyses; if finding near-optimal, operationally applicable solutions can be done much more efficiently than finding optimal solutions, this is therefore often preferred, as this enables a more efficient work process.

The *impact* describes whether or not the solutions generated by the solution approach improve the considered processes and decisions. Conventionally, the impact of a solution approach is measured by means of costs; if the solution approach leads to cost reductions, it is considered a success. However, the impact can also appear as simple improvements to the normal working procedures.

Finally, the *acceptance* describes how well the solution approach and the solutions are acknowledged by the different stakeholders of the problem (dispatchers, unions, employees, etc.). A criteria for acceptance is often that the solution approach encapsulates as many of the operational aspects of the problem as possible. The operational aspects of a problem are typically expressed by the airlines and handling companies (and the airport) as preferences, and the solution approach must be able to satisfy these preferences to the greatest possible extent. A way of ensuring the acceptance is thus to design the solution approach so that is allows an easy configuration of these preferences. As an example, the airlines often have strong operational preferences for certain stands of the airport. The airlines are typically only concerned about their own operation and when discussing stand preferences, each airline will only be able to describe which stands it has a preference for and how these stands are prioritized. If a stand belongs to more than one stand cluster, this is in principle not the airline's problem, and the ranking of the clusters on the particular stand should be configured solely by the airport. Equivalently, the airlines typically have strong preferences for how their turn-rounds are handled, and often these preferences are unaffiliated to the overall strategic goals of the airport concerning number of busings and towings, share of pier-serviced passengers, etc. The concept of *handling options* provides an easy way of configuring the handling preferences of the airlines, and it is then up to the airport to decide to what extent these handling preferences should be fulfilled.

The three measures of success are interdependent. Most likely, a solution approach will not be accepted if is not applicable, and equivalently, a solution approach will generally not have an impact, if it is not accepted. To summarize, a successful solution approach at CPH is characterized by being

- efficient (applicability)
- able to satisfy preferences expressed by the airlines to the greatest possible extent (impact and acceptance)
- easy to configure (acceptance)

To solve the allocation problems considered in this thesis, two different solution methods have been applied: the *total enumeration* (TE) method and the *partial enumeration* method (PE). The TE method enumerates all variables and uses a standard of-the-shelf MIP solver to find an optimal solution. The primary advantage of this method is that it is simple and relatively easy to implement. The PE is a little more complicated and is used for the problems where it is not possible to hold all variables in memory or simply computationally inefficient to consider all variables. In the following the PE method is described on a more general level. See Table 6.1 for an overview of which solution method that has been applied for the different allocation problems considered in this thesis.

6.1 The partial enumeration method

Two things can make the models described in Chapter 5 difficult to solve: the dimensions of the constraint matrix and the binary constraints on the decision variables. To deal with this, the partial enumeration method considers a restricted and relaxed version of the model, where only a restricted set of schedules, $\mathcal{S}' \subseteq \mathcal{S}$, is considered and the integrality constraints (5.4) and (5.5) are relaxed (R-BRAP):

$$\textbf{minimize} \qquad z = \sum_{s \in \mathcal{S}'} c_s x_s + \sum_{p \in \mathcal{P}} c_\kappa \kappa_p \tag{6.1}$$

$$\textbf{subject to} \qquad \sum_{s \in \mathcal{S}'} g_{sp} x_s + \kappa_p = 1 \qquad \forall\, p \in \mathcal{P} \tag{6.2}$$

$$\sum_{s \in \mathcal{S}'} a_{sk} x_s \leq 1 \qquad \forall\, k \in \mathcal{K} \tag{6.3}$$

$$x_s \geq 0 \qquad \forall\, s \in \mathcal{S}' \tag{6.4}$$

$$\kappa_p \geq 0 \qquad \forall\, p \in \mathcal{P} \tag{6.5}$$

Relaxing the integrality constraints converts the problem into an LP problem which, in the general case, is computationally much faster to solve than an IP problem. Restricting the set of considered schedules reduces the problem size to a manageble level.

Due to the presence of the coverage variables, there will always be a mathematically feasible solution to R-BRAP; however, since we have restricted the set of possible schedules, the solution might be *invalid*, in the sense that one or more activities are not assigned a schedule, although more possible schedules actually exist. The idea of the PE method is now to iteratively solve R-BRAP until a valid solution has been found or no more schedules can be added to $\mathcal{S}'$. Initially, $\mathcal{S}' = \emptyset$, i.e. R-BRAP only contains the coverage variables. After each iteration, the validity of the

Problem	Solution method
Check-in counter allocation	Total enumeration
Stand and gate allocation	Partial enumeration
Turn-round re-allocation	Partial enumeration
Taxiway route allocation	Partial enumeration (not fully implemented)

Table 6.1: *Overview of chosen solution methods.*

solution is checked, and if one or more activities are unallocated, $\mathcal{S}'$ is expanded, and R-BRAP is re-solved. To keep $\mathcal{S}'$ as small as possible, new schedules are only added for unallocated and conflicting activities. A activity is termed *conflicting* if it is claiming one of more of resource-time intervals that would otherwise have been claimed by one of the unallocated activities.

If a valid solution to the R-BRAP is fractional, we can exploit the described properties of the constraint matrix to impose constraint branching when searching for an integer solution (see [Ryan et al., 1981]).

However, as described in Chapter 1, being able to find a near-optimal integer solution efficiently is often preferred to finding an optimal integer solution, even at the strategical and tactical levels. As a consequence, the PE, resolves fractionality, simply by solving BRAP using the schedules in $\mathcal{S}'$, i.e. when a valid but fractional solution is found, R-BRAP is "unrelaxed" and solved as an integer problem using a standard IP solver.

If it is not possible to find a valid integer solution to BRAP using only the schedules in $\mathcal{S}'$, the PE method will expand $\mathcal{S}'$ by adding more schedules for all processes, and then try to re-solve the problem. In a worst case scenario, the algorithm will end up adding all schedules of $\mathcal{S}$ in this process, however, as will be shown in Chapter 9, 10, and A, this case was never observed during the computational experiments.

The main ideas of the PE method are summarized in Algorithm 1 and in the following, the steps where conflicting activities are identified, and $\mathcal{S}'$ is expanded are described in more detail. One iteration of the PE method is equivalent to one iteration of the loop in line 3.

6.1.1 Identifying conflicting activities

Let $\mathcal{S}'_p \subseteq \mathcal{S}_p$ denote the restricted set of schedules for activity $p \in \mathcal{P}$ and let $\tilde{\mathcal{P}} \subseteq \mathcal{P}$ denote the set of unallocated activities, i.e. for all $t \in \tilde{\mathcal{P}}$, $\kappa_p > 0$, after solving R-BRAP. If $\tilde{\mathcal{P}} = \emptyset$, all activities have been successfully assigned a schedule and a valid solution exists. Now, let $\mathcal{K}_s \subseteq \mathcal{K}$ denote the set of resource-time interval constraints claimed by schedule $s \in \mathcal{S}$,

$$\mathcal{K}_s = \{k \in \mathcal{K} \mid a_{sk} = 1\}$$

The set of resource-time interval constraints claimed by the schedules in $\mathcal{S}'_p$ is thus given by

$$\mathcal{K}_{\mathcal{S}'_p} = \bigcup_{s \in \mathcal{S}'_p} \mathcal{K}_s$$

Consider an unallocated activity $p \in \mathcal{P}$. Since we have set

$$c_\kappa > \sum_{p \in \mathcal{P}} \max_{s \in \mathcal{S}_p} c_s$$

we can only have that $\kappa_p \geq 0$, if it is not possible to choose any of the schedules in $\mathcal{S}'_p$. One or more of the constraints in $\mathcal{K}_p$ must hence be binding, because they have been claimed by some other activities.

For each optimal primal solution to the R-BRAP, a corresponding dual solution also exists. Let π_k denote the dual variable of constraint $k \in \mathcal{K}$. If k is binding, the value of its corresponding dual variable is positive, i.e. $\pi_k > 0$. The set of binding constraints in an optimal solution is thus given as

$$\tilde{\mathcal{K}} = \{k \in \mathcal{K} \mid \pi_k > 0\}$$

Thereby, the set of binding constraints preventing the schedules in $\mathcal{S}'_p$ from being chosen is given as

$$\tilde{\mathcal{K}}_p = \tilde{\mathcal{K}} \cap \mathcal{K}_p$$

The set of *conflicting* activities preventing t from being allocated is thus given as the set of allocated activities for which a schedule claiming one of the constraints in $\tilde{\mathcal{K}}_p$ has been chosen, i.e.

$$\hat{\mathcal{P}}_p = \left\{ q \in \mathcal{P} \mid q \neq t, \exists\ s \in \mathcal{S}'_q : x_s > 0\ \wedge\ \mathcal{K}_s \cap \tilde{\mathcal{K}}_p \neq \emptyset \right\}.$$

Algorithm 1: Pseudo-code for the PE solution method.

```
Read airport problem instance;
Set $\mathcal{S}' = \emptyset$;
repeat
    Solve R-BRAP using a standard LP solver;
    Identify the set of unallocated activities $\tilde{\mathcal{P}}$;
    Identify the set of binding constraints $\tilde{\mathcal{K}}$;
    Identify the set of conflicting activities $\hat{\mathcal{P}}$;
    for each activity $t \in \tilde{\mathcal{P}} \cup \hat{\mathcal{P}}$ do
        if $\mathcal{S}_p \setminus \mathcal{S}'_p \neq \emptyset$ then
            Generate variables by identifying the cheapest schedules from $\mathcal{S}_p \setminus \mathcal{S}'_p$ not
            claiming any of the constraints in $\tilde{\mathcal{K}}$;
            if possible to identify new schedules then
                Add new schedules to $\mathcal{S}'$;
until valid solution has been found;
if solution is fractional then
    Restrict all variables to be binary;
    repeat
        Solve BRAP using a standard IP solver;
        if solution to BRAP found then
            Save and return solution;
        else
            for each activity $r \in \mathcal{P}$ do
                Generate variables by identifying the cheapest schedules from $\mathcal{S}_p \setminus \mathcal{S}'_p$;
                if possible to identify new schedules then
                    Add new schedules to $\mathcal{S}'$;
    until valid solution has been found;
else
    Save and return solution;
```

and the set of all conflicting activities for *all* unallocated activities is then given as

$$\hat{\mathcal{P}} = \bigcup_{t \in \tilde{\mathcal{P}}} \hat{\mathcal{P}}_p$$

6.1.2 Variable generation

Expanding the set $\mathcal{S}'$ is not a trivial task. Only schedules that could potentially improve the current solution should be added to $\mathcal{S}'$, i.e. the schedules added to $\mathcal{S}'$ should preferably have negative reduced costs. In conventional column generation, columns with negative reduced costs are identified by solving a *pricing problem* exploiting the values of the dual variables. The pricing problem is essentially an optimization problem, where a solution represents a new potential column in the master problem. If the pricing problem returns a solution with negative reduced costs, it means that columns which could potentially improve the current solution exist. If, on the other hand, the pricing problem returns a solution with non-negative reduced cost, the solution to the master problem is optimal.

As described, solution methods developed for the operational environment at CPH must be able to encapsulate as many operational aspects and airline preferences as possible, mimicking the operational practice to the greatest possible extent. However, not all operational aspects can be modeled in the same way, and defining an adequate pricing problem may be a very difficult task. For instance, for the stand and gate allocation problem, many of the required resources are needed at the same time (e.g. the stand and the arrival gate), and modeling this concurrency is difficult.

As a consequence, the PE does not find new variables to add to $\mathcal{S}'$ by solving a pricing problem, but instead uses a (simple) variable generation algorithm (VGA). Essentially, the VGA pursues the idea that schedules with low costs are better than schedules with a higher cost. Initially, only "the best" (cheapest) schedules for each activity are added to $\mathcal{S}'$, and for each iteration, less attractive schedules are then added. Thereby it is assumed, that in a valid solution, only the best possible schedule is chosen for each activity, and once a valid solution has been found, it will be very close (or equal) to an optimal solution.

As described, $\mathcal{S}'$ is only expanded for unallocated and conflicting activities at each iteration. To restrict the size of $\mathcal{S}'$ even further, the PE method also restricts the the number of schedules that can be added per activity per iteration, and only schedules not claiming any of the currently binding constraints $\tilde{\mathcal{K}}$ are added.

As described in Table 6.1, the partial enumeration method is used when solving the tactical and operational stand and gate allocation problems as well as the taxiway route allocation problem. For both the tactical and the operational stand and gate allocation problems, the schedules are generated as follows: First, all handling schedules are generated for the first priority handling option and these are sorted according to their costs and added to the model when needed. When no more handling schedules exist for the first priority handling options, handling schedules are generated and added for the second priority handling option. And so forth. Although the number of handling schedules may theoretically be very large for handling options involving towings, the real-world restrictions (like e.g. aircraft type compatibility on the individual stands and status handling) severely limit the number of possible stand combinations. As will be shown in the results presented in Chapter 9, the described variable generation approach enabled the solution method to efficiently find operationally applicable, near-optimal solutions.

For the taxiway route allocation, the schedules are generated as follows: First, all paths for the standard routes are generated, without imposing any delay. Then, if more paths are needed, paths with stand delays of $15, 30, 45, \ldots, 300$ seconds are added to the existing paths, and these schedules are added to $\mathcal{S}'$. Finally, if a conflict free routing still cannot be found, paths with taxiway delays of $15, 30, 45, \ldots, 300$ seconds at the various holding points are added to the model. Although, many different delay combinations may exist, the described variable generation approach enabled the solution method to efficiently find operationally applicable, near-optimal solutions. However, the given approach is not the best, since many paths not resolving the conflicts may be added to the problem. The generation of variables for the taxiway route allocation problem thus constitute a topic for future research.

6.2 Pitfalls

As will be shown in the papers, of the advantages of the PE method is that it is easy to implement and for most problem instances, it generates near-optimal solutions very efficiently (see Chapter 9, 10, and A). Moreover, the variable generation algorithm allows one to easily incorporate any operational aspect of the problem in the variable generation phase, and it is even possible to have a non-linear cost function to calculate the costs of the schedules. However, the PE method has several (theoretical) disadvantages. First of all, there is no guarantee that an optimal solution will be found, even if one such exists. Second of all, if a non-optimal solution is found, there is no guarantee for how far from optimum the solution is. Finally, in a worst case scenario, the method might end up adding all variables to the model. However, as will be shown in Chapter 9, 10, and A, this case was never observed in the computational experiments.

Chapter 7

Conclusion

In this thesis various real-life ground handling resource allocation problems arising from the operation of an airport has been studied. Although the studies were based on the situation at Copenhagen Airport, the models and solution methods have been designed to be generally applicable.

The thesis provides a general introduction to airport operations management and introduces the primary planning and optimization problems typically arising from the different airport processes. Each planning problem is categorized as being either a manpower planning, resource allocation, or process optimization problem, and the distribution of responsibilities for the strategical, tactical, and operational management of the problem is indicated. Furthermore, the thesis describes how the performance of an airport can be monitored and controlled using airport benchmarking and a detailed set of operational key performance indicators.

7.1 Academic results

One of the findings of the project is that the considered ground handling resource allocation problems can be modeled and solved using the same underlying mathematical model and solution methods. Moreover, the developed model is proven to be applicable at both the strategical, tactical, and operational levels. These findings motivates the establishment of a common solution framework that can be used solve the different problems and potentially other problems have the same characteristics.

For the check-in counter allocation problem, the total enumeration method is used to solve the problem it is shown to perform well on a series of real-life data instances. However, it is shown that a simple greedy heuristic easily matches the current solution method in terms of both performance and solution quality.

For the stand and gate allocation problem, it is shown that the developed mathematical model can include any kind of resource demand associated with the handling of a turn-round, and conflicts are resolved implicitly. Furthermore the model relatively easily incorporates the requirements and preferences of the different stakeholders of the problem; stand preferences for different airlines are expressed via stand clusters and handling preferences for different types of turn-rounds, aircraft types, airlines, etc. are expressed via handling options. Stand dependencies are handled implicitly via push-back options, yielding a much more accurate approach than normally seen for this type of constraint. The problem is solved using the partial enumaration method and numerical results on real life test instances arising at CPH show that the proposed methodology performs well; the solution algorithm can find near-optimal valid solutions respecting a predefined prioritization of objectives within reasonable time and the solutions are as good or better than current practice.

For the operational variant of the stand and gate allocation problem, two different conflict handling strategies for handling conflicts arising when changes to the timetable are received (estimated arrival time, estimated departure time, and aircraft registration) are proposed. The instant conflict handling strategy immediately resolves every conflict that occurs, whereas the lazy conflict handling strategy only resolves conflicts every five minutes. To resolve conflicts, the stand and gate allocation problem is solved on a part of the stand and gate allocation plan defined by the conflict interval in order to reallocate one or more turn-rounds. The turn-round re-allocation problem is solved using the same model and solution method (partial enumaration) as for the general stand and gate allocation problem, and numerical results on a real life test instance arising at CPH show that the proposed methodology performs well as it can find near-optimal for all conflicts in reasonable time while maintaining a high solution quality. Both the instant and lazy conflict handling strategies perform well in that they both give rise to good values of the considred objectives, however the lazy conflict handling strategy is the most attractive, as fewer conflict handlings are needed and the KPI values in general are higher.

For the taxiway route allocation problem, the main conclusion is that the problem can be modeled and solved using the basic resource allocation model and the partial enumeration method, however, more development and more testing is needed before any final conclusions can be drawn.

The article presented in Chapter 11 shows by the example of two case studies at two different airports how project success can be reached when OR applications are applied to airport operations. Based on the presented operational aspects the most important problem properties can be determined. A thorough identification of the operational aspects will support a successful development by taking care of the relevant problem characteristics and often forgotten real-world requirements. The presented operational aspects are part of the subjective view of the authors but many discussions with other practitioners both from academia and from other airports have shown that these factors represent the common sense at airports. One of the main conclusions described in the paper, is that for all applications, the principle “keep it as simple as possible” is an important success factor. Examples are the work with mixed-integer programming and the use of standard solvers. This approach is easy understandable and can be brought into business quickly. For many problems, first applications can be generated by simplified solution models. *Acceptance* can be increased by an early involvement of the stakeholders and providing fast first exemplary results. Finally, the paper outlines that mathematical optimality is not as important to real-life airport operations management as it is often considered by the scientific community; in many cases, optimality may not even be measurable due to data uncertainty and the unknown actual behavior on the microscopical level.

7.2 Practical results

For the check-in counter allocation problem, the developed model and solution method is currently used at CPH for both short-, medium-, and long-term capacity planning and for providing the check-in counter allocations used on the day-of-operation. For the stand and gate allocation problem, the developed model and solution method is used at CPH for both short-, medium-, and long-term capacity planning, and for evaluating and challenging potential new software providers. For the taxiway route allocation problem, the developed model and solution method is used at CPH for analyzing the operational consequences of changes to both the layout of the taxiway network and the daily operational practice. For the baggage make-up position allocation problem, the developed model and solution method is currently used at CPH for both short-, medium-, and long-term capacity planning and for analyzing the operational consequences of changes to the daily operational practice. As can be seen, the developed solution tools for all models presented in this thesis, are fully implemented and used at Copenhagen Airport.

7.3 Future research

The potential areas of future research include:

- **Robustness** In a dynamic environment as that of an airport, robustness of the tactical plans is very important. In this thesis, robustness has been included as buffers separating tasks allocated to the same resource. This approach is widely used in many practical applications and in general, it has proven to be a viable approach yielding good results for the daily operation of the airport. However, a range of questions naturally arise, even for this simple approach. How long should the buffer be? Should the size of the buffer be fixed for all tasks or determined from historical knowledge about the problem? Moreover, robustness is a research field in itself, and clearly, other approaches could be considered, including stochastic programming (see e.g. [Şeker and Noyan, 2012] and [Genç et al., 2012]).

- **Variable generation** The variable generation phase described for the partial enumeration method is currently relatively simple. Although the different results show that the method provides operationally applicable and near-optimal solutions for the considered problems, other approaches to the variable generation could be investigated. An obvious alternative approach could be to use column generation to generate the variables, exploiting the values of the dual variables.

- **Optimal solutions** As argued in the thesis, being able to find near-optimal solutions efficiently is often preferred, if finding optimal solutions is a time consuming process. For the tactical stand and gate allocation problem (see 9), the developed solution method finds near-optimal solutions efficiently, however, as also shown, the more simple approach, where all variables are enumerated and an optimal solution is found using a state-of-the-art MIP solver is in fact able to solve the problem in less than 15 minutes. A topic for future research could therefore be to consider alternative exact methods and investigate if it is possible to find optimal solutions as efficiently as the near-optimal solutions are currently being found.

Bibliography

ACI, 2014. Airports council international.
URL https://www.aci-europe.org

Amrahov, S. E., Ibrahim Alsalihe, T. A., 2011. Greedy algorithm for the scheduling aircrafts landings.

Andreussi, A., Bianco, L., Ricciardelli, S., 1981. A simulation-model for aircraft sequencing in the near terminal area. European Journal of Operational Research 8 (4), 345–354.

Appelt, S., Batta, R., Lin, L., Drury, C., 2007. Simulation of passenger check-in at a medium-sized us airport. Winter Simulation Conference, 1252–1260.

ASQ, 2014. Airport service quality.
URL http://www.airportservicequality.aero/content/survey.html

Atkin, J. A. D., Burke, E. K., Greenwood, J. S., Reeson, D., 2008. A metaheuristic approach to aircraft departure scheduling at london heathrow airport. Computer-aided systems in public transport 600, 235–252.

ATRS, 2014. Air transport research society.
URL http://www.atrsworld.org

Babic, O., Teodorovic, D., Tosic, V., 1984. Aircraft stand assignment to minimize walking. Journal of Transportation Engineering 110 (1), 55–66.

Baik, H., Sherali, H. D., Trani, A. A., 2002. Time-dependent network assignment strategy for taxiway routing at airports. Transportation Research Record, 70–75.

Balakrishnan, H., Yoon, J., 2007. A framework for coordinated surface operations planning at dallas-fort worth international airport. Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2007 3, 2382–2400.

Bihr, R. A., 1990. A conceptual solution to the aircraft gate assignment problem using 0, 1 linear programming. Computers and Industrial Engineering 19 (1-4), 280–284.

Bolat, A., 1999. Assigning arriving flights at an airport to the available gates. Journal of the Operational Research Society 50 (1), 23–34.

Bolat, A., 2000. Procedures for providing robust gate assignments for arriving aircrafts. European Journal of Operational Research 120 (1), 63–80.

Castaneda, E., Gonzalez, J., Harris, S., Kim, J., 2007. Optimized airport security infrastructure system (oasis), 1–6.

Chawdhry, P. K., 2009. Risk modeling and simulation of airport passenger departures process. Proceedings - Winter Simulation Conference, 2820–2831.

Cheng, Y., 1997. A knowledge-based airport gate assignment system integrated with mathematical programming. Computers and Industrial Engineering 32 (4), 837–852.

Chun, H. W., Mak, 1999. Intelligent resource simulation for an airport check-in counter allocation system. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 29 (3), 325–335.

Chung, C. A., Sodeinde, T., 2000. Simultaneous service approach for reducing air passenger queue time. Journal of Transportation Engineering 126 (1), 85–88.

Clare, G., Richards, A., Sharma, S., 2009. Receding horizon, iterative optimization of taxiway routing and runway scheduling. AIAA Guidance, Navigation, and Control Conference and Exhibit, 2009–5646.

CPH, 2013a. Copenhagen airports a/s - annual report 2013. Web.

CPH, 2013b. Cph official website.
URL http://www.cph.dk

D'Ariano, A., D'Urgolo, P., Pacciarelli, D., Pranzo, M., 2010. Optimal sequencing of aircrafts take-off and landing at a busy airport. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, 1569–1574.

de Barros, A. G., Tomber, D. D., 2007. Quantitative analysis of passenger and baggage security screening at airports. Journal of Advanced Transportation 41 (2), 171–193.

Diepen, G., van den Akker, J. M., Hoogeveen, J. A., 2009. Integrated gate and bus assignment at amsterdam airport schiphol.

Diepen, G., van den Akker, J. M., Hoogeven, J. A., Smeltink, J. W., 2007. Using column generation for gate planning at amsterdam airport schipol. Technical report, Department of Information and Computing Sciences, Utrecht University.

Ding, H., Lim, A., Rodrigues, B., Zhu, Y., 2004. Aircraft and gate scheduling optimization at airports. Proceedings of the annual Hawaii International Conference on Systems Sciences.

Ding, H., Lim, A., Rodrigues, B., Zhu, Y., 2005. The over-constrained airport gate assignment problem. Computers and Operations Research 32 (7), 1867–1880.

Dohn, A., 2010. Rostering and task scheduling : Applications in manpower planning. Ph.D. thesis.

Doran, G. T., 1981. There's a s.m.a.r.t. way to write management's goals and objectives. Management Review 70, 35–36.

Dorndorf, U., Jaehn, F., Lin, C., Ma, H., Pesch, E., 2007. Disruption management in flight gate scheduling. Statistica Neerlandica 61 (1), 92–114.
URL http://EconPapers.repec.org/RePEc:bla:stanee:v:61:y:2007:i:1:p:92-114

Dorndorf, U., Jaehn, F., Pesch, E., 2008. Modelling robust flight-gate scheduling as a clique partitioning problem. Transportation Science 42 (3), 292–301.

Dorndorf, U., Jaehn, F., Pesch, E., 2012. Flight gate scheduling with respect to a reference schedule. Annals of Operations Research 194 (1), 177–187.

Drexl, A., Nikulin, Y., 2008. Multicriteria airport gate assignment and pareto simulated annealing. IIE Transactions 40 (4), 385–397.

Enoma, A., Allen, S., 2007. Developing key performance indicators for airport safety and security. Facilities 25 (7), 296–315.

Ernst, A. T., Krishnamoorthy, M., Storer, R. H., 1999. Heuristic and exact algorithms for scheduling aircraft landings. Networks 34 (3), 229–241.

EUROCONTROL, 2010. Long term forecast 2010-2030.

EUROCONTROL, 2012. Medium term forecast 2012-2018.

Fleming, K., Bazargan, M., Subramanian, P., 2002. A simulation study to investigate runway capacity using taam. Winter Simulation Conference Proceedings 2, 1235–1243.

Francis, G., Humphreys, I., Fry, J., 2002. The benchmarking of airport performance. Journal of Air Transport Management 8 (4), 239–247.

Genç, H., Erol, O. K., Eksin, İ., Berber, M. F., Güleryüz, B. O., 2012. A stochastic neighborhood search approach for airport gate assignment problem. Expert Systems with Applications 39 (1).

Genovese, A., Bruno, G., 2010. A mathematical model for the optimization of the airport check-in service problem. Electronic Notes in Discrete Mathematics 36 (C), 703–710.

Glover, F., Kochenberger, G. A., 2003. Handbook of metaheuristics.

Gotteland, J. B., Durand, N., 2003. Genetic algorithms applied to airport ground traffic optimization. Proceedings - CEC: 2003 Congres on evolutionary computation, 544–551.

Granberg, T. A., Munoz, A. O., 2013. Developing key performance indicators for airports.

Gu, Y., Chung, C., 1999. Genetic algorithm approach to aircraft gate reassignment problem. Journal of Transportation Engineering 125 (5), 384–389.

Haagensen, K., 2012. Local regulations/appendix 1 - taxiway routing and guidelines for push-back at copenhagen airport.

Haghani, A., Chen, M., 1998. Optimizing gate assignments at airport terminals. Transportation Research, Part A (Policy and Practice) 32A (6), 437–454.

Hu, X. B., Di Paolo, E., 2007. An efficient genetic algorithm with uniform crossover for the multi-objective airport gate assignment problem, 55–62.

Hu, X.-B., Di Paolo, E., 2009. A ripple-spreading genetic algorithm for the airport gate assignment problem, 1857–1864.

Humphreys, I., Francis, G., 2002. Performance measurement: A review of airports. International Journal of Transport Management 1 (2), 79–85.

International, A. C. (Ed.), 2012. Guide to Airport Performance Measures. Airports Council International.

Jaehn, F., 2010. Solving the flight gate assignment problem using dynamic programming. Zeitschrift für Betriebswirtschaft 80 (10), 1027–1039.

Joustra, P. E., Van Dijk, N. M., 2001. Simulation of check-in at airports. Winter Simulation Conference Proceedings 2, 1023–1028.

Khoury, Kamat, Ioannou, 2006. Simulation and visualization of air-side operations at detroit metropolitan airport.

Lan, S., Clarke, J. P., Barnhart, C., 2006. Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. Transportation Science 40 (1), 15–28.

Leone, K., Liu, R. R., 2011. Improving airport security screening checkpoint operations in the us via paced system design. Journal of Air Transport Management 17 (2), 62–67.

Li, C., 2008. Airport gate assignment: New model and implementation, –.

Lim, A., Rodrigues, B., Zhu, Y., 2005. Airport gate scheduling with time windows. Artificial Intelligence Review 24 (1), 5–31.

Liu, C., Guo, K., 2010. Airport taxi scheduling optimization based on genetic algorithm.

Liu, C., Wang, Y., 2011. Aircraft taxi path optimization based on ant colony algorithm 2, 226–229.

Malone, J. D., Kulesz, J., Delp, W. W., Lee, R., Brigantic, R. T., Muller, G. A., 2009. Simulation to assess the efficacy of us airport entry screening of passengers for pandemic influenza. International Journal of Risk Assessment and Management 12 (2-4), 290–310.

Mangoubi, R. S., Mathaisel, D. F. X., 1985. Optimizing gate assignments at airport terminals. Transportation Science 19 (2), 173–188.

Marian, O., 2012. Issues of airport's capacity. Metalurgia International 17 (8), 105–107.

Marín, A., Codina, E., 2008. Network design: taxi planning. Annals of Operations Research 157 (1), 135–151.

Marín, Á. G., 2006. Airport management: taxi planning. Annals of Operations Research 143 (1), 191–202.

Martinez, J. C., Trani, A. A., Ioannou, P. G., 2001. Modeling airside airport operations using general-purpose, activity-based, discrete-event simulation tools. Transportation Research Record (1744), 65–71.

Montoya, J., Wood, Z., Rathina, S., Malik, W., 2010. A mixed integer linear program for solving a multiple route taxi scheduling problem. American Institute of Aeronautics and Astronautics 2, 1208–1225.

Nie, X., Parab, G., Batta, R., Lin, L., 2012. Simulation-based selectee lane queueing design for passenger checkpoint screening. European Journal of Operational Research 219 (1), 146–155.

Nikulin, Y., Drexl, A., 2010. Theoretical aspects of multicriteria flight gate scheduling: deterministic and fuzzy models. Journal of Scheduling 13 (3), 261–280.

Oum, T. H., Yu, C., 2004. Measuring airports'operating efficiency: a summary of the 2003 atrs global airport benchmarking report. Transportation Research Part E: Logistics and Transportation Review 40 (6), 515–532.

Padberg, M. W., 1974. Perfect zero–one matrices. Mathematical Programming 6 (1), 180–196.

Parlar, M., Sharafali, M., 2008. Dynamic allocation of airline check-in counters: A queueing optimization approach. Management Science 54 (8), 1410–1424.

Pendergraft, D. R., Robertson, C. V., Shrader, S., 2004. Simulation of an airport passenger security system. Proceedings - Winter Simulation Conference 1, 874–878.

Pesic, B., Durand, N., Alliot, J. M., 2001. Aircraft ground traffic optimisation using a genetic algorithm. GECCO, 1397–1404.

Pintea, C.-M., Pop, P. C., Chira, C., Dumitrescu, D., 2008. A hybrid ant-based system for gate assignment problem. Hybrid Artificial Intelligence Systems 5271, 273–280.

Rasmussen, M. S., 2011. Optimisation-Based Solution Methods for Set Partitioning Models. Technical University of Denmark (DTU).

Rezanova, N. J., 2009. The Train Driver Recovery Problem - Solution Method and Decision Support System Framework.

Roling, P. C., Visser, H. G., 2008. Optimal airport surface traffic planning using mixed-integer linear programming. International Journal of Aerospace Engineering.

Ryan, Foster, Wren, 1981. An integer programming approach to scheduling. In: Computer Scheduling of Public Transport. Urban Passenger Vehicle and Crew Scheduling. Proceedings of an International Workshop — 1981.

Şeker, M., Noyan, N., 2012. Stochastic optimization models for the airport gate assignment problem. Transportation Research Part E 48 (2), 438–459.

Skytrax, 2014. Skytrax.
URL `http://www.airlinequality.com`

Smeltink, J. W., Soomer, M. J., 2008. An optimisation model for airport taxi scheduling.

Soolaki, M., Mahdavi, I., Mandavi-Amiri, N., Hassanzadeh, R., Aghajani, A., 2012. A new linear programming approach and genetic algorithm for solving airline boarding problem. Applied Mathematical Modelling 36 (9), 4060–4072.

Takakuwa, S., Oyama, T., 2003. Simulation analysis of international-departure passenger flows in an airport terminal. Winter Simulation Conference Proceedings 2, 1627–1634.

Tang, C.-H., Yan, S., Hou, Y.-Z., 2010. A gate reassignment framework for real time flight delays. 4OR 8 (3), 299–318.

Tolofari, S. R., Ashford, N. J., Caves, R. E., 1990. The Cost of Air Service Fragmentation. Department of Transport Technology, University of Technology, Loughborough.

van Dijk, N. M., van der Sluis, E., 2006. Check-in computation and optimization by simulation and ip in combination. European Journal of Operational Research 171 (3), 1152–1168.

Wei, D., Liu, C., 2007. Optimizing gate assignment at airport based on genetic-tabu algorithm. Proceedings of the IEEE International Conference on Automation and Logistics August 18 - 21, 2007, Jinan, China, 1135–1140.

Wilson, D., Roe, E. K., So, S. A., 2006. Security checkpoint optimizer (sco): An application for simulating the operations of airport security checkpoints. Proceedings - Winter Simulation Conference, 529–535.

Wolsey, L. A., 1998. Integer programming. Wiley-Interscience.

Xu, J., Bailey, G., 2001. The airport gate assignment problem: Mathematical model and a tabu search algorithm. Proceedings of the annual Hawaii International Conference on Systems Sciences.

Yan, Tang, Chen, 2008. Reassignments of common-use check-in counters following airport incidents. Journal of the Operational Research Society 59 (8), 1100–1108.

Yan, S., Chang, K. C. K.-C., Tang, C. H. C.-H., 2005. Minimizing inconsistencies in airport common-use checking counter assignments with a variable number of counters. Journal of Air Transport Management 11 (2), 107–116.

Yan, S., Ho, C., 2001. Optimization of multiple objective gate assignments. Transportation Research, Part A (Policy and Practice) 35A (5), 413–432.

Yan, S., Tang, C., 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., Hou, Y.-Z., 2011. Airport gate reassignments considering deterministic and stochastic flight departure/arrival times. Journal of Advanced Transportation 45 (4), 304–320.

Zidarova, E. D., Zografos, K. G., 2011. Measuring quality of service in airport passenger terminals. Transportation Research Record (2214), 69–76.

Part II

Scientific Papers

Chapter 8

Check-in Counter Allocation at Copenhagen Airport

Tor Fog Justesen*† Esben Kolind * Jesper Larsen †

*Airport Optimization, Copenhagen Airports A/S
Lufthavnsboulevarden 6, 2770 Kastrup

†Department of Management Engineering, Technical University of Denmark
Produktionstorvet, Building 426, DK-2800 Kgs. Lyngby, Denmark

Contact: Tor Fog Justesen (+45 23 24 62 47 / tor.justesen@cph.dk)

Abstract The present paper addresses the problem of allocating check-in counters to check-in groups (single flight or common-use areas) in airports. The problem is formulated as an integer programming problem and takes airline preferences, the distribution of queues in the terminal areas, and the distribution of baggage inflow to the take-away belts into consideration. A case study demonstrates the successful implementation of the model in an optimization tool developed at *Copenhagen Airport* (CPH). The optimization tool has helped CPH improve the distribution of queues and baggage inflow in the terminals for international departures and led to a better utilization of the existing check-in counters, enabling CPH to both reduce and postpone the investments in additional check-in counters.

Keywords: *airport operations*, *check-in counter allocation*, *optimization*

8.1 Introduction

In 2012, more than 23 million passengers departed from Copenhagen Airport (CPH). Check-in is usually the first process encountered by the passengers when traveling by air. By checking in, the passengers confirm to the airline that they actually have the intention to board the flight for which they have booked a ticket. Moreover, at check-in, the passengers have the possibility to choose, buy or change a seat, register bags, etc. Today the check-in can be performed in various ways; online, via self-service kiosks at the airport, and via traditional check-in desks, where the passengers are assisted by representatives of the airlines. In most cases, check-in is performed before the passengers reach the security check. In Europe, it is often the responsibility of the airport to provide available check-in counters, and the airlines must provide available representatives, typically by using so-called handling agents. For an illustration of the typical flow for departing passengers, see Figure 8.1.

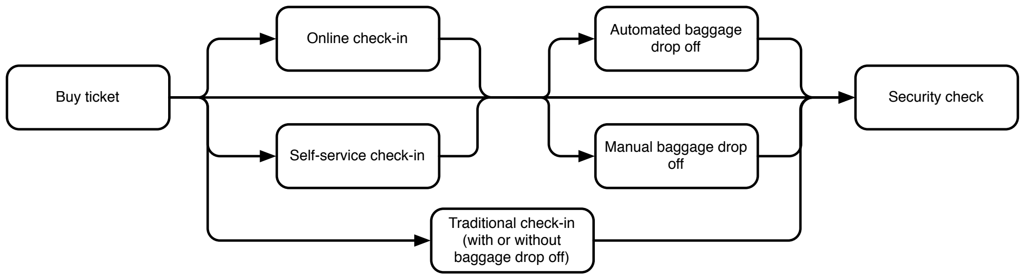

Figure 8.1: *The different possibilities for checking in and dropping off bags when traveling by air. The traditional check-in and the manual baggage drop off both require available check-in counters and staff.*

A *check-in group* is a group of flights (departures) that share the same check-in counters for check-in or baggage drop-off. A check-in group can either consist of a single flight (dedicated area) or all flights of a specific or multiple airlines (common use area). Each check-in group is associated with a counter demand over time, and the *check-in counter allocation problem* is then the problem of allocating check-in groups to available check-in counters.

Since 2001, CPH has experienced a total growth in the number of passengers of approximately 29% (cf. [CPH, 2013]) and despite the introduction of more and more self-service technologies and a continuously increasing number of common use areas (airlines sharing the same check-in facilities), the demand for check-in counters has also continued to increase. Prior to 2010, CPH had 91 check-in counters to handle international departures divided between two terminals. All departures were handled by one of two handling companies and the 91 counters provided a sufficiently large capacity to handle all check-ins in an unproblematic manner. This situation allowed for a simple approach to the check-in counter allocation problem; each handling company was assigned a fixed subset of the check-in counters, and it was then up to the handling companies themselves to plan the usage of these counters. Typically, the handlers would manually perform the allocation largely on a seasonal basis and without considering the counter allocation of the other handlers.

By early 2010, the number of passengers and the demand for check-in counters had grown so much that the simple approach to the check-in counter allocation was no longer viable. At peak hours, the handlers did not have enough counters available and the flow in the terminals was unbalanced, so one could experience situations where some parts of the terminals were congested by long queues blocking the flow while other parts of the terminals were almost empty, just to mention a few challenges.

To address these problems, CPH decided to take over the responsibility for the check-in counter allocation problem. To ensure that the correct counter demand was used for each check-in group, CPH analyzed data from all operating handlers and airlines. To have a high degree of accuracy, all necessary data was assessed in five minute intervals. Furthermore, CPH implemented an optimization tool for solving the check-in counter allocation problem as it is described in this paper. The optimization tool was deployed in May 2010, and has been running since then. The tool is used on a weekly basis to provide the airlines, the handling companies and the staff of the passenger department of CPH (PAX) with check-in counter allocation plans.

The remainder of this paper is organized as follows. In Section 8.2 we give a more detailed description of the problem and in Section 8.3 the relevant literature is presented. The modeling approach and the mathematical model is presented in Section 8.4 and in Section 8.5 we describe how the model was implemented and applied at CPH. Finally, we draw some conclusions in Section 8.6.

8.2 Problem definition

For each check-in group we have a counter demand profile over time. The correct counter demand for a given check-in group is determined by the airline(s) in dialogue with the airport and the handling companies.As an example, Table 8.1 describe the counter demand for six different check-in groups at CPH in the period 08:45—09:15 on June 1, 2011. For the allocation problem at CPH, the counter demands are determined beforehand and is thus not part of the optimization.

Check-in group	**08:45**	**08:50**	**08:55**	**09:00**	**09:05**	**09:10**
Air France and KLM	5	5	5	5	5	5
British Airways	4	4	4	4	4	4
Norwegian	6	6	6	5	5	5
EasyJet	2	2	2	2	2	2
Star Alliance - Bag drop	6	6	6	4	4	4
Star Alliance - Economy	7	7	7	6	6	6

Table 8.1: *Counter demand six different check-in groups at CPH in the period 08:45—09:15 on June 1, 2011.*

A check-in counter is said to be opened when a handling agent logs into the check-in system at the counter and is ready to check-in passengers and receive their bags. When the handling agent logs out again, the check-in counter is said to be closed. A *counter opening* refers to the opening of a given counter and its length is the time from when the counter is opened until it is closed. Two counter openings are said to be *adjacent* if they represent the opening of adjacent counters. For the sake of simplicity, we assume that all counters are in operation.

A given counter demand can be translated into different *patterns*, each consisting of a set of counter openings. Potentially many possible patterns may exist for a given counter demand, however, for practical purposes, only the patterns with the minimum number of counter openings and the maximum amount of time where the open counters are adjacent, are considered operationally valid. It is obvious, that minimizing the number of counter openings implicitly maximizes the length of the counter openings. Now, let the parameter d_t denote the counter demand in a given time interval t and assume that we have a counter demand $D = (d_{t_1}, d_{t_2}, d_{t_3}, d_{t_4}) = (4, 3, 3, 2)$ for a given check-in group. Figure 8.2 illustrates six possible patterns for this counter demand, out of which only three are considered operationally valid, as these all maximize the amount of time where the open counters are adjacent.

An *allocation option* is the allocation of a given pattern to actual counters in the airport. As an example, Figure 8.3 illustrates three possible allocation options for pattern 4 illustrated in Figure 8.2 for counter demand D. Two allocation options are said to be in conflict if they simultaneously claim the same check-in counter(s).

8.2.1 Terminal layout

Airports differ in layout and also in aim of service quality when planning. Often, the set of check-in counters is divided into multiple *counter clusters* as a consequence of the airport layout (multiple terminals) and terminal layout (stairways, shops, etc.). At CPH, a check-in group must always be allocated entirely within a single cluster such that a check-in is not handled across a stairway or in different terminals.

The floor space in front of the check-in counters is divided into multiple smaller *queueing areas*. Each queueing area is associated with a given critical level specifying the maximum number of passengers that the area can hold at any time before it becomes too crowded. If the number of passengers in a given queueing area exceeds the critical level there is a risk that flow in the terminals will be blocked and that people are so cramped that queueing becomes a very unpleasant experience. The definition of the queuing areas and the critical levels is done by the airport.

The take-away belt system behind the counters lead the bags from the terminals to the outbound baggage handling facilities of the airport. At CPH there are two baggage handling facilities

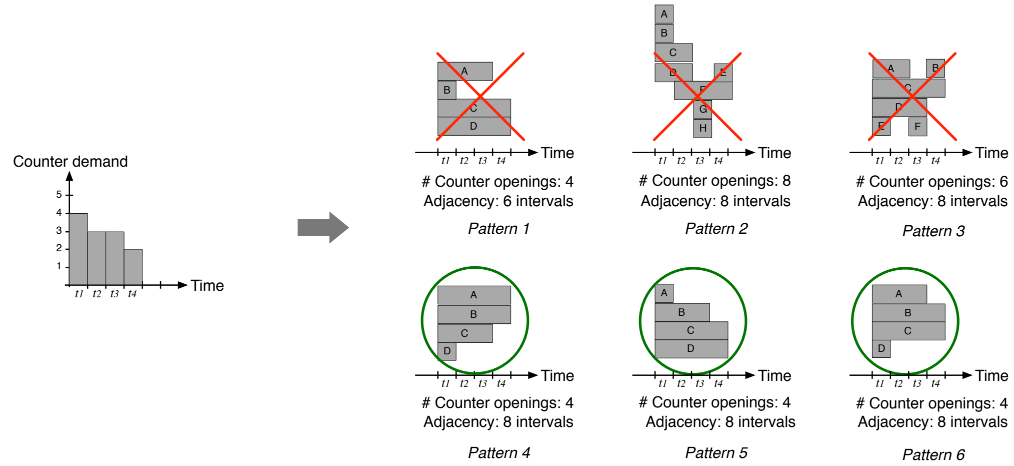

Figure 8.2: *Examples of possible patterns for the counter demand* $D = (d_{t_1}, d_{t_2}, d_{t_3}, d_{t_4}) = (4, 3, 3, 2)$*. Patterns 1—3 cover the counter demand but are considered operationally invalid, whereas the patterns 4—6 are all considered operationally valid as they maximize the amount of time where the open counters are adjacent.*

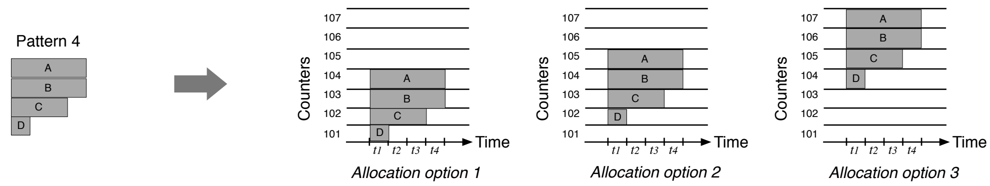

Figure 8.3: *Allocation options for pattern 4 illustrated in Figure 8.2 for counter demand* $D = (d_{t_1}, d_{t_2}, d_{t_3}, d_{t_4}) = (4, 3, 3, 2)$*.*

for the handling of bags for international flights, BF2 and BF3. The take-away belt system consists of a number of smaller take-away belt sections connected to each other. The direction of each take-away belt section can be switched, and the direction of the belt determines which baggage handling facilities that can be reached from which counters; bags sent to BF2 can also reach BF3, where bags sent to BF3 cannot reach BF2. For each check-in group, the required baggage handling facility is specified and the allocation of check-in counters must ensure that the bags of each check-in group are sent in the right direction. It is assumed that the direction of each take-away belt section is given when solving the check-in counter allocation problem.

The capacity of the take-away belt determine the maximum baggage intake allowing smooth operation. Just as the floor space, the take-away belt sections are grouped into multiple *take-away blocks*, each associated with a given critical level specifying the maximum number of bags that can be checked in at any given time before the baggage handling facilities becomes overloaded. As for queuing areas, the take-away blocks are specified by the airport and reflects the layout and the capacity of the baggage handling system.

A given check-in counter can be associated with exactly one queueing area and at exactly one take-away block, however the counter clusters, queueing areas and take-away blocks do not have to be aligned. A pair of counters can thus belong to the same cluster but different queuing areas and take-away block. As a consequence, a check-in group can be allocated entirely within a single cluster, but across multiple queuing areas and take-away blocks.

For both the queuing areas and the take-away blocks, the critical levels are indications of when the load is at maximum capacity, however they should not be seen as hard constraints. It can thus be accepted that the critical levels are exceeded. Note, that for practical purposes, there is no differentiation on how much below the critical levels, the load is. That is, whenever the number of passengers in queue or the number of bags flowing in to a given take-away block is below the critical level, this is considered to be acceptable.

To calculate the impact on the queueing areas and the take-away blocks, the expected number of passengers waiting in queue and the expected number of bags is given for each five minute interval for each check-in group. For the sake of simplicity, it is assumed that the number of passengers waiting in queue and the expected number of bags are uniformly distributed among the counters in the possible allocation options for a given check-in group.

The queueing areas and take-away blocks at CPH are specified in Table 8.2 and 8.3 and an illustration is shown in Figure 8.4.

Queueing area	**Counters**	**Critical level**
1	001—009	125
2	011—018	75
3	021—028	150
4	031—038	150
5	087—094	150
6	095—109	100
7	110—120	150
8	122—139	200
9	140—154	100

Table 8.2: *Counter clusters at CPH.*

Take-away block	**Counters**	**Critical level**
1	001—018	67
2	021—038	67
3	087—120	67
4	122—143	67
5	144—154	67

Table 8.3: *Baggage take-away blocks at CPH.*

Recall, that an *allocation option* is the allocation of a pattern to actual check-in counters for a given check-in group. With only very few restrictions, each check-in group can in principle be allocated to any set of check-in counters. However, for both commercial and historical reasons, most operators would prefer not to have their allocations differ too much from week to week or even from day to day. Therefore, limitations to the set of possible allocation options for each check-in group are introduced. For a given check-in group, a limited subset of *allowed* check-in counters is specified, and the check-in group is then only allowed to be allocated to check-in counters within this subset. To control the allocation even further, a *preferred* subset of the allowed check-in counters is specified for each check-in group. Negotiating the sets of allowed and preferred counters is in principle a dialogue between two stakeholders with contradicting interests. On one hand, the airport wants to enlarge the sets as much as possible in order to get the largest possible set of allocation options for each check-in group as this could potentially lead to better solutions with regards to load on the queueing areas and take-away belts. However, on the other hand, the airlines want to limit the sets as much as possible, as this will ensure that the corresponding check-in groups will not be moved too much around in the terminals over time.

At CPH, each check-in group is pre-assigned to a specific terminal for operational and commercial reasons. As a consequence, the largest possible set of allowed counters for each check-in group is equal to the set of counters in its pre-assigned terminal. In the following we will refer to

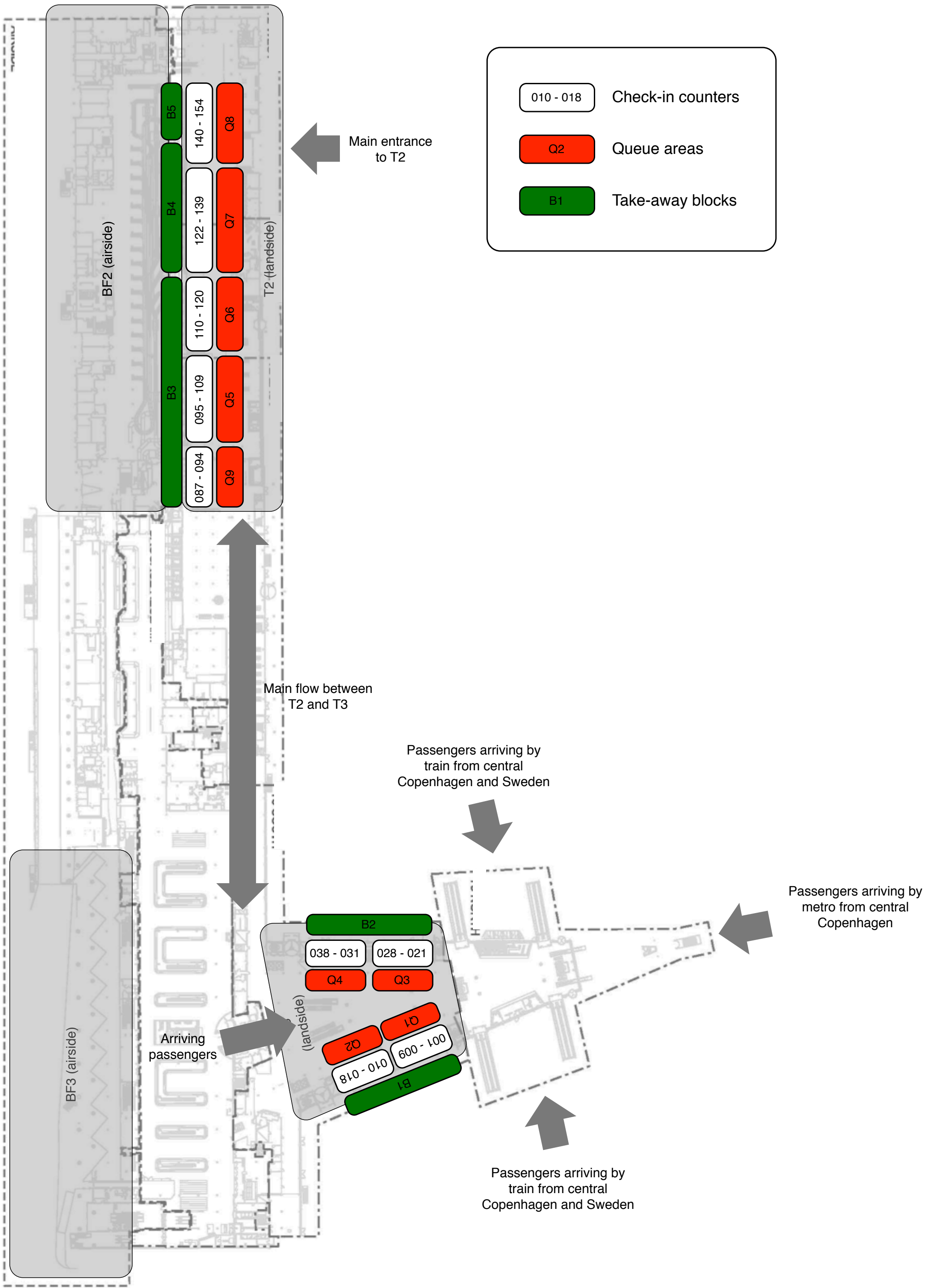

Figure 8.4: *Illustration of the counter clusters, queueing areas, and take-away blocks in Terminal 2 at CPH.*

the sets of allowed and preferred counters as the *configuration* of the check-in counter allocation problem.

By considering the counter clusters, the set of allowed check-in counters, and the requirements to take-away belt direction, the set of allocation options for each check-in group can be precalculated prior to solving the problem.

8.2.2 Objectives

When solving the check-in counter allocation problem, the following objectives have been identified at CPH (in prioritized order)

1. Minimize the number of unallocated check-in groups
2. Satisfy airline preferences to the greatest possible extent
3. Distribute queues such that the critical levels of the queueing areas are not exceeded
4. Distribute inflow of baggage such that the critical levels of the take-away blocks are not exceeded

The check-in counter allocation problem can now be stated as the problem of selecting exactly one allocation option for each check-in group such that no check-in counter is occupied by more than one check-in group at any time and that the objectives are satisfied to the greatest possible extent respecting their priorities.

8.3 Previous work

Even though operations research is central within airline and airport optimization, the relevant literature for the described check in counter allocation problem is rather sparse. The problem was originally presented in a paper by [Hon, 1996] aiming to optimize the counter allocation for *Hong Kong International Airport*. [Hon, 1996] presented a heuristic to solve a stochastic version of the problem where counter demands can vary and the impact on resources is simulated. The presented problem is almost similar to the problem of CPH, but the solution approach differs on the key point that in the case of CPH, it is assumed that all data is accurate and given beforehand.

Closely related to the problem described in this report is the *adjacent resource scheduling* problem presented by [Duin and Sluis, 2006]. Adjacent resources are resources that are placed next to each other, and the adjacent resource scheduling problem is then the problem of scheduling a set of tasks requiring this adjacency property to the available resources. To exemplify, [Duin and Sluis, 2006] describe how the problem arises when considering berth allocation of ships, scheduling of parallel processors in a computer system, and finally, when allocating check-in counters in an airport. They present mathematical formulations for the problem, and show that the decision version of the problem is strongly $\mathcal{NP}$-complete.

[Yan et al., 2004] develop an integer programming model to assist airport authorities to assign common use check-in counters at *Chiang Kai-Shek International Airport* (CKS) in *Taiwan*. In contrast to our problem [Yan et al., 2004], present a problem of monthly planning minimizing the total walking distances of passengers. The problem presented is solely two dimensional with check-in groups and blocks of counters and does not include considerations on baggage or queueing. The counter demand is assumed to be constant for each check-in group. The case of CKS is further explored by [Yan et al., 2005] who develop a model that minimizes total inconsistencies in common-use counter assignments with a variable number of counters. The model is formulated as a zero-one integer program and solved heuristically. Another heuristic approach to solving the problem is presented by [Wang Yeung and Chun, 1995] who develop an airport check-in counter allocation system using a genetic algorithm (GA) approach. The developed approach measures the quality of a possible solution by a *fitness measure*. In principle, the fitness measure can be used to incorporate the conditions from CPH (number of people waiting in line, number of bags check-in, and airline preferences), however adjacency criteria are not refined enough in the model.

A large amount of research has been put into determining the actual counter demand and most of the mentioned papers are concerned with minimizing the number of required resources (counters). [van Dijk and van der Sluis, 2006] present an approach in which the number of counters needed for each check-in group is determined as part of the optimization, and afterwards the maximum number of counters used at any time is minimized. Equivalently, [Tang, 2010] present a network model for the optimization of common use check-in groups with the goal of minimizing the number of counters required for daily operations. Other work on optimizing the number of counters needed includes the most recent paper on single flight check-in queueing estimation by [Parlar and Sharafali, 2008], the paper by [Park and Ahn, 2003] on passenger arrival, and finally the simulation paper on determining the counter usage by [Chun and Mak, 1999].

8.4 Mathematical model

Let $\mathcal{C}$ denote the set of check-in counters, $\mathcal{G}$ the set of check-in groups, and $\mathcal{I}$ the set of five minute intervals in the planning period. Furthermore, let $\mathcal{Q}$ and $\mathcal{B}$ denote the set of queueing areas respectively take-away blocks.

The set of allowed counters for check-in group $g \in \mathcal{G}$ is denoted $\mathcal{C}_g \subseteq \mathcal{C}$ and the set of preferred counters for g is $\mathcal{C}_g^+ \subseteq \mathcal{C}_g$. The set of all possible allocation options for all check-in groups is denoted $\mathcal{P}$ and the set of allocation options for check-in group $g \in \mathcal{G}$ is denoted $\mathcal{P}_g \subseteq \mathcal{P}$. The binary parameter α_{pct} is one if allocation option $p \in \mathcal{P}$ uses counter $c \in \mathcal{C}$ in time interval $t \in \mathcal{I}$ and zero otherwise.

For each allocation option $p \in \mathcal{P}$, the parameter $\beta_{pbt} \in \mathbb{N}$ denotes the number of bags that p contributes with to take-away block $b \in \mathcal{B}$ in time interval $t \in \mathcal{I}$. Equivalently, the parameter ρ_{pqt} denotes the number of passengers waiting in line in queueing area $q \in \mathcal{Q}$ in time interval $t \in \mathcal{I}$. The critical levels of the queuing areas and take-away blocks are given by the parameters Q_q and B_b respectively.

To model the problem, four sets of decision variables are defined: the binary allocation variables x_p, the coverage variables π_g, and the positive surplus variables, y_{qt} and z_{bt}. An allocation variable x_p is one if allocation option $p \in \mathcal{P}$ is chosen in the solution and zero otherwise. A coverage variable π_g is one if check-in group $g \in \mathcal{G}$ is not allocated and zero otherwise. If the critical level Q_q for queueing area $q \in \mathcal{Q}$ is exceeded in time interval $t \in \mathcal{I}$ the positive surplus variable y_{qt} is equal to the excess, otherwise it is zero. Equivalently, if the critical level B_b for take-away block $b \in \mathcal{B}$ is exceeded in time interval $t \in \mathcal{I}$, the positive surplus variable z_{bt} is equal to the excess, otherwise it is zero.

As described in Section 8.2, the objectives that have been identified at CPH are (in prioritized order)

1. Minimize the number of unallocated check-in groups

2. Satisfy airline preferences to the greatest possible extent

3. Distribute queues such that the critical levels of the queueing areas are not exceeded

4. Distribute the inflow of baggage such that the critical levels of the take-away blocks are not exceeded

To model the distribution of queues and baggage inflow, we have chosen an approach where any overrun of the critical levels is penalized. In principle, this penalization can be done in many ways, and basically, the choice of method is a question of how the airport considers operational peak periods.

Assume that we have two queue areas Q_1 and Q_2 and that in a given solution, the critical levels are exceeded as depicted in Figure 8.5. For Q_1, the excess is very high, but only for a short period, wheres as for Q_2, the excess is not that high, but for a longer period.

If an operational peak like the one seen for Q_1 is considered the most critical, the model should penalize the busiest queueing area in each time interval. This can be accomplished by introducing a decision variable $Y_t \geq 0$ for each time interval $t \in \mathcal{I}$ and by adding the constraints

$$Y_t \geq y_{qt} \;\; \forall\, q \in \mathcal{Q} \; \forall\, t \in \mathcal{I}$$

to the constraint matrix and the term

$$\sum_{t \in \mathcal{I}} Y_t$$

to the objective function.

If, however, an operational peak like the one seen for Q_2 is considered the most critical, the model should penalize the busiest queueing areas seen over the entire planning period. This can be accomplished by simply adding the term

$$\sum_{q \in \mathcal{Q}} \sum_{t \in \mathcal{I}} y_{qt}$$

to the objective function.

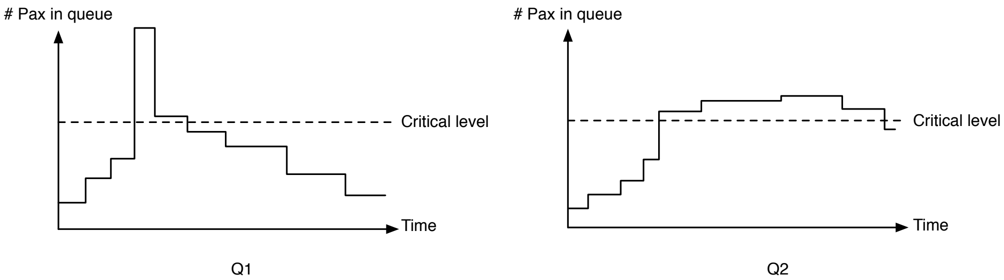

Figure 8.5: *Illustration of two different types of queue level excess.*

At CPH the general perception is that operational peaks as the one seen for Q_2 are the most critical. Handling a short period with a high excess, is a question of taking some fast decisions and in general, the amount of passengers affected by the excess is limited. In contrast to this, a long period with a constant excess might end up affecting more passengers and in general it is a more stressful situation for the handlers to deal with a long period with constant pressure. At CPH the preferred approach to modeling the distribution of queues and baggage inflow is hence the second one. An equivalent line of reasoning can be used for the inflow of baggage to the take-away blocks.

As can be seen, the check-in counter allocation problem is in essence a multi-criteria optimization problem. Multiple equally good solutions might thus exists, however, for the sake of simplicity, we have chosen to use the following weighted sum as objective function:

$$z = \omega_1 \sum_{g \in \mathcal{G}} \pi_g + \omega_2 \sum_{g \in \mathcal{G}} \sum_{p \in \mathcal{P}_g} \sum_{c \in \mathcal{C}_g \setminus \mathcal{C}_g^+} \sum_{t \in \mathcal{I}} \alpha_{pct} x_p + \omega_3 \sum_{q \in \mathcal{Q}} \sum_{t \in \mathcal{I}} y_{qt} + \omega_4 \sum_{b \in \mathcal{B}} \sum_{t \in \mathcal{I}} z_{bt}$$

Assuming that all parameters are non-negative integers, the check-in counter allocation problem can thus be formulated as the following integer programme:

$$\textbf{minimize} \quad z \tag{8.1}$$

$$\textbf{subject to} \quad \sum_{p \in \mathcal{P}_g} x_p + \pi_g = 1 \qquad \forall\, g \in \mathcal{G} \tag{8.2}$$

$$\sum_{p \in \mathcal{P}} \alpha_{pct} x_p \leq 1 \qquad \forall\, c \in \mathcal{C}\ \forall\, t \in \mathcal{I} \tag{8.3}$$

$$\sum_{p \in \mathcal{P}} x_p \rho_{pqt} - y_{qt} \leq Q_q \qquad \forall\, q \in \mathcal{Q}\ \forall\, t \in \mathcal{I} \tag{8.4}$$

$$\sum_{p \in \mathcal{P}} x_p \beta_{pbt} - z_{bt} \leq B_b \qquad \forall\, b \in \mathcal{B}\ \forall\, t \in \mathcal{I} \tag{8.5}$$

$$x_p \in \{0,1\} \qquad \forall\, p \in \mathcal{P} \tag{8.6}$$

$$\pi_g \in \{0,1\} \qquad \forall\, g \in \mathcal{G} \tag{8.7}$$

$$y_{qt} \geq 0 \qquad \forall\, q \in \mathcal{Q}\ \forall\, t \in \mathcal{I} \tag{8.8}$$

$$z_{qt} \geq 0 \qquad \forall\, z \in \mathcal{B}\ \forall\, t \in \mathcal{I} \tag{8.9}$$

where the constraints (8.2) ensure that all check-in groups are either allocated or not. Constraints (8.3) ensure that each counter is claimed by at most one check-in group in each time interval. Constraints (8.4) and (8.5) control the variables y_{qt} and z_{bt} respectively setting these to the excess for each of the time intervals where the critical levels are exceeded and zero otherwise. Finally, constraints (8.6), (8.7), (8.8), and (8.9) define the variables correctly.

8.5 Implementation at CPH

The configuration currently used at CPH is relatively restricted because the airport wants to satisfy the airline preferences as much as possible; the airlines and handling agents have accepted that CPH has taken over the planning process for the counters, however, they are still reluctant to major changes to where the individual departures are checked in. For some check-in groups at CPH, the sets are even so limited, that only one allocation option exists. As a consequence, the solution space is so restricted that it is possible to solve the problem simply by enumerating all variables and using a state-of-the-art MIP solver (Gurobi) to find an optimal solution. We refer to this method as the *exact method*.

To compare the exact method with the manual process previously applied by the handlers, we compare the exact method with a simple greedy heuristic. Obviously, a real planner uses a more complicated approach than just a simple greedy heuristic. A real planner uses experience to make the final decisions, however, as we do not have detailed knowledge about the previous planning processes, implementing these aspects is very difficult. We therefore use the simple greedy approach to give an idea of the the potential in using an efficient heuristic as an alternative to the exact method.

The greedy heuristic sorts the check-in groups according to their size in terms of total counter demand. Then, the algorithm allocates the check-in groups one by one, starting with the largest check-in groups. For each check-in group, all possible allocation options are found and their costs are calculated as described in Section 8.4. Then, the algorithm tries to find the least expensive allocation that does not violate any of the constraints (8.3). If it is not possible to find a feasible allocation option, the check-in group is considered unallocated. Once a check-in group is allocated, it cannot be moved, and the algorithm continues until it has tried to allocate all check-in groups.

8.5.1 Computational results

We consider 30 data instances from the CPH representing each day of the month of June, 2011. The summer period is typically a very busy period at CPH and especially during the morning peaks, the terminals are almost completely saturated. To evaluate the potential scaling of the developed methods, we consider the following three configurations:

A: The same sets of allowed and preferred counters as used by CPH

B: The same sets of preferred counters as used by CPH, but the sets of allowed counters are expanded to include all counters of each of the clusters of the sets of allowed counters used in A

C: The same sets of preferred counters as used by CPH, but the sets of allowed counters are expanded to include all counters in the preferred terminal of the check-in group

Configuration A gives rise to the smallest possible solution space, whereas configuration C gives rise to the largest possible solution space for the problem.

The size of a given instance is described by the number of check-in groups and the total counter demand expressed in terms of *counter minutes* (c-min) which denote the sum of all counter demands in all time intervals. Furthermore, we describe each instance in terms of the number of possible allocation options for the individual check-in groups. To do this, we introduce $G_{i-j} \geq 0$ denoting the number of check-in groups that have between i and j allocation options (both inclusive) in a given configuration and consider the sizes of G_{1-15}, G_{16-30}, G_{31-45}, and $G_{46-\infty}$. Table 8.4 presents the characteristics of the considered data instances for each of the considered configurations.

We evaluate the solution quality by analyzing the number of unallocated check-in groups, Π, the preference compliance, P, i.e. the total amount of time where check-in groups are allocated to their preferred counters, the total queue excess, Q^+, and the total baggage inflow excess, B^+. The value of P is calculated as the following sum:

$$P = \sum_{g \in \mathcal{G}} \sum_{p \in \mathcal{P}_g} \sum_{c \in \mathcal{C}_g^+} \sum_{t \in \mathcal{I}} a_{pct} x_p$$

The value of Q^+ is calculated as the sum

$$Q^+ = \sum_{q \in \mathcal{Q}} \sum_{t \in \mathcal{I}} y_{qt}$$

and equivalently, the value of B^+ is calculated as the sum

$$B^+ = \sum_{b \in \mathcal{B}} \sum_{t \in \mathcal{I}} z_{bt}$$

The performance of the solution methods is evaluated by considering the number of constraints, the number of variables, the total solve time, and the objective value.

To obtain the priorities of the objectives as described in Section 8.2 the weights used in the objective function are set to $\omega_1 = 1{,}000{,}000$, $\omega_2 = 10{,}000$, $\omega_3 = 100$, and $\omega_4 = 1$. These weights are equal to the ones currently used at CPH. The weights could be set differently, however, we will not change the values of them, as current practice show that they yield solutions which are accepted by both the handling companies and the airlines and which are applicable in the operation.

The computational results measuring the quality of the solutions are shown in Table 8.5 and the computational results measuring the performance of the solution methods are shown in Table 8.6.

As can be seen from Table 8.5, the exact method in general finds better solutions that the greedy heuristic in terms of both the number of unallocated check-in groups and preference compliance for all configurations. For configuration A, the greedy heuristic finds solutions where the preference compliance is higher than for the exact method for seven of the instances, however, for these instances, the number of unallocated check-in groups is 2.4 higher in average. Equivalently, for configuration B, the greedy heuristic finds solutions where the preference compliance is higher than

Instance	# Check-in groups	Counter demand (c-min)	A				B				C			
			G_{1-15}	G_{16-30}	G_{31-45}	$G_{46-\infty}$	G_{1-15}	G_{16-30}	G_{31-45}	$G_{46-\infty}$	G_{1-15}	G_{16-30}	G_{31-45}	$G_{46-\infty}$
2011-06-01	55	49760	26	8	7	14	6	17	17	14	3	7	3	42
2011-06-02	57	42210	35	8	4	10	11	23	13	10	2	10	5	40
2011-06-03	68	41670	43	11	5	9	9	35	15	9	1	14	7	46
2011-06-04	70	46210	44	9	7	10	11	32	17	10	2	13	6	49
2011-06-05	68	47820	44	9	5	10	10	33	15	10	1	15	6	46
2011-06-06	62	43415	33	11	4	14	8	25	15	14	2	10	3	47
2011-06-07	62	42490	36	9	5	12	10	26	14	12	2	8	7	45
2011-06-08	61	44230	33	8	7	13	7	25	16	13	2	7	4	48
2011-06-09	61	44840	36	8	4	13	12	23	13	13	2	10	6	43
2011-06-10	75	48150	46	10	5	14	12	34	15	14	3	15	5	52
2011-06-11	69	47170	44	9	7	9	11	32	17	9	2	13	6	48
2011-06-12	68	44540	45	9	5	9	10	34	15	9	1	15	6	46
2011-06-13	57	42825	33	11	4	9	8	25	15	9	2	9	4	42
2011-06-14	65	45210	38	9	5	13	10	28	14	13	2	10	5	48
2011-06-15	64	47230	36	8	7	13	8	27	16	13	2	9	3	50
2011-06-16	62	45750	37	8	4	13	12	24	13	13	2	11	6	43
2011-06-17	73	47735	45	10	6	12	12	33	16	12	3	15	5	50
2011-06-18	72	48870	47	9	7	9	11	35	17	9	2	15	6	49
2011-06-19	75	48620	47	9	7	12	10	36	17	12	1	18	5	51
2011-06-20	54	64580	34	10	5	5	8	27	14	5	1	10	4	39
2011-06-21	67	45900	40	9	5	13	10	30	14	13	2	11	5	49
2011-06-22	71	48355	41	9	7	14	11	30	16	14	2	13	5	51
2011-06-23	68	47640	39	9	6	14	12	27	15	14	2	12	6	48
2011-06-24	79	49045	51	10	5	13	12	39	15	13	3	17	5	54
2011-06-25	77	51460	51	9	7	10	10	40	17	10	3	13	7	54
2011-06-26	83	53080	55	9	7	12	11	43	17	12	2	19	5	57
2011-06-27	75	50720	44	11	6	14	12	32	17	14	4	13	4	54
2011-06-28	74	49610	45	9	6	14	11	34	15	14	3	12	5	54
2011-06-29	80	52660	49	10	7	14	14	36	16	14	3	17	4	56
2011-06-30	71	48690	43	8	6	14	13	29	15	14	3	14	6	48
Minimum	54	41670	26	8	4	5	6	17	13	5	1	7	3	39
Maximum	83	64580	55	11	7	14	14	43	17	14	4	19	7	57
Average	68	47682	41.3	9.2	5.7	11.8	10.4	30.5	15.4	11.8	2.2	12.5	5.1	48.3

Table 8.4: *Considered data instances*

	Exact method												Greedy heuristic											
	Π			P (c-min)			Q^+			B^+			Π			P (c-min)			Q^+			B^+		
Instance	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
2011-06-01	0	1	0	11665	10960	12405	247.4	247.4	247.4	2.2	0.0	2.2	3	4	0	11070	9230	10325	247.4	247.4	247.4	2.2	0.0	12.3
2011-06-02	3	1	1	12660	13050	13050	127.8	127.8	127.8	0.0	0.0	0.0	3	2	1	11805	10730	11105	127.8	127.8	127.8	0.0	0.0	0.0
2011-06-03	3	0	0	13180	13695	13750	0.0	0.0	0.0	0.0	0.0	0.0	4	3	0	13280	11940	12145	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-04	7	3	2	11250	10695	11790	34.0	34.0	34.0	0.0	0.0	0.0	8	8	4	11130	10260	10600	34.0	42.3	34.0	0.0	0.0	0.0
2011-06-05	5	1	0	11590	11985	12550	56.3	56.3	56.3	0.0	0.0	0.0	6	5	1	11390	10565	10970	56.3	56.3	56.3	0.0	0.0	0.0
2011-06-06	1	1	0	13130	13460	13490	0.0	0.0	0.0	0.0	0.0	0.0	2	3	0	11545	10350	11250	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-07	1	0	0	12790	13165	13135	0.0	0.0	0.0	0.0	0.0	0.0	2	2	0	12035	10955	11225	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-08	2	0	0	12510	12895	13275	0.0	0.0	0.0	0.0	0.0	0.0	4	3	0	12695	11215	11680	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-09	3	0	0	12615	12855	13285	29.9	29.9	29.9	0.0	0.0	0.0	5	3	1	11920	10875	11400	29.9	29.9	29.9	0.0	0.0	0.0
2011-06-10	7	1	1	12970	11910	13645	32.1	32.1	32.1	0.0	0.0	0.0	11	6	4	11705	10755	11550	32.1	32.1	32.1	0.0	0.0	0.0
2011-06-11	9	6	3	10795	11165	11685	41.5	41.5	41.5	0.0	0.0	0.0	11	11	6	10615	9620	10080	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-12	5	0	0	11780	10955	12735	0.0	0.0	0.0	0.0	0.0	0.0	7	6	0	12010	11245	11350	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-13	1	1	0	13130	13460	13490	38.2	38.2	38.2	0.0	0.0	0.0	2	3	0	13235	11950	12265	38.2	38.2	38.2	0.0	0.0	0.0
2011-06-14	2	1	0	13115	13340	13460	3.3	3.3	3.3	0.0	0.0	0.0	3	3	1	12360	11130	11340	3.3	3.3	3.3	0.0	0.0	0.0
2011-06-15	3	0	0	12565	13100	13540	20.8	20.8	20.8	0.0	0.0	0.0	6	3	1	12270	11210	11315	20.8	20.8	20.8	0.0	0.0	0.0
2011-06-16	3	0	0	12680	12920	13350	61.6	61.6	61.6	0.0	0.0	0.0	5	3	2	11985	10940	11615	61.6	61.6	61.6	0.0	0.0	0.0
2011-06-17	5	0	0	13200	13725	13980	0.0	0.0	0.0	0.0	0.0	0.0	8	5	1	11920	10860	11775	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-18	10	6	4	11615	10640	12065	0.0	0.0	0.0	0.0	0.0	0.0	12	13	6	11225	10265	10665	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-19	8	2	2	11970	11265	13045	0.0	0.0	0.0	0.0	0.0	0.0	11	8	3	12200	11555	11660	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-20	7	5	2	16015	16930	19475	549.4	549.4	549.4	0.0	0.0	0.0	14	11	5	17610	18225	19045	549.4	549.4	549.4	0.0	0.0	0.0
2011-06-21	4	0	0	12880	13225	13345	15.6	15.6	15.6	0.0	0.0	0.0	5	2	2	12045	10910	11315	15.6	15.6	15.6	0.0	0.0	0.0
2011-06-22	5	2	1	12640	13280	13660	0.0	0.0	0.0	0.0	0.0	0.0	8	5	3	12360	11300	11435	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-23	2	1	0	13170	13140	13570	10.4	10.4	10.4	0.0	0.0	0.0	6	5	1	12085	11075	11450	10.4	10.4	10.4	0.0	0.0	0.0
2011-06-24	12	4	3	12995	11905	13820	0.7	0.7	0.7	0.0	0.0	0.0	15	10	4	12010	10830	11295	0.7	0.7	0.7	0.0	0.0	0.0
2011-06-25	14	8	4	11360	10595	12310	148.9	148.9	148.9	0.0	0.0	0.0	15	13	7	11080	10085	9995	0.0	0.0	0.0	0.0	0.0	0.0
2011-06-26	13	7	4	12345	12240	13240	417.5	417.5	417.5	7.5	0.0	7.5	14	14	7	12555	11740	12085	417.5	417.5	417.5	0.0	0.0	0.0
2011-06-27	6	4	2	13840	14290	14630	653.1	653.1	717.8	0.6	0.6	0.6	10	7	4	13075	11595	12465	650.6	650.6	626.6	0.6	0.6	0.6
2011-06-28	7	2	1	13630	12235	14155	257.1	257.1	257.1	0.0	0.0	0.0	13	8	4	12905	11465	11795	257.1	255.0	255.0	0.0	0.0	0.0
2011-06-29	13	8	6	12890	13365	14010	109.1	70.2	70.2	0.0	0.0	0.0	17	13	8	12780	11210	11500	70.2	70.2	70.2	0.0	0.0	0.0
2011-06-30	8	2	2	13205	11955	13625	187.7	187.7	187.7	0.0	0.0	0.0	10	7	5	12770	11170	11720	187.7	176.7	176.7	0.0	0.0	0.0
Minimum	0	0	0	10795	10595	11685	0.0	0.0	0.0	0.0	0.0	0.0	2	2	0	10615	9230	9995	0.0	0.0	0.0	0.0	0.0	0.0
Maximum	14	8	6	16015	16930	19475	653.1	653.1	717.8	7.5	0.6	7.5	17	14	8	17610	18225	19045	650.6	650.6	626.6	2.2	0.6	12.3
Average	5	2	1	12672	12613	13452	101.4	100.1	102.3	0.3	0.0	0.3	8	6	2	12255	11175	11613	93.7	93.5	92.4	0.1	0.0	0.4

Table 8.5: *Computational results measuring the quality of solutions.*

	Exact method												Greedy heuristic					
	# cons			# vars			Solve time (s)			Obj. val.			Solve time (s)			Obj. val.		
Instance	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C	A	B	C
2011-06-01	19267	18065	26513	5643	6114	8489	4.12	5.53	15.96	73382262	53192209	71902262	0.05	0.08	0.19	72080000	54160000	76030004
2011-06-02	18752	19736	25004	5395	5899	8493	3.46	4.94	15.89	130295400	73195400	73195400	0.04	0.06	0.18	57720000	60860000	61070000
2011-06-03	19052	20060	25394	5456	6098	9162	3.57	5.33	18.60	94460000	53910000	53800000	0.05	0.07	0.18	53840000	55080000	57010000
2011-06-04	18650	19432	23941	5602	6218	9380	3.91	5.80	22.98	352965400	242895400	99285400	0.05	0.06	0.19	62720000	63940000	69300000
2011-06-05	20214	21163	27113	5508	6136	9166	3.73	5.73	20.96	163268400	83738400	67028400	0.04	0.07	0.19	66700000	68890000	69820000
2011-06-06	19259	19877	24768	5718	6264	9080	4.26	5.98	18.58	78290000	77630000	58050000	0.05	0.07	0.19	60500000	61930000	62530000
2011-06-07	19536	20333	25429	5640	6162	8990	4.06	5.58	17.62	67080000	56570000	56630000	0.05	0.07	0.18	57630000	59070000	60450000
2011-06-08	19260	19993	25941	5675	6223	9149	4.10	5.74	20.14	112410000	59150000	58390000	0.04	0.08	0.19	57540000	60050000	61580000
2011-06-09	19667	20286	24611	5611	6119	8827	4.22	5.52	18.71	138855700	62535700	61675700	0.05	0.07	0.19	60650000	64210000	64240000
2011-06-10	19450	20308	26040	5841	6503	9751	4.56	6.63	25.44	194315709	84585709	81115709	0.04	0.08	0.20	64020000	67200000	66600000
2011-06-11	18404	19242	23761	5543	6159	9255	3.77	5.68	22.09	390895700	286675700	116995699	0.05	0.08	0.19	62770000	64200000	69020000
2011-06-12	19817	20790	25574	5462	6104	9172	3.55	5.41	20.57	156080000	64370002	60810000	0.04	0.07	0.19	58380000	59570000	63580000
2011-06-13	19024	19915	24998	5349	5895	8549	3.36	5.04	17.43	78555400	77895400	58315400	0.04	0.07	0.18	57380000	58990000	60760000
2011-06-14	19723	20385	25947	5719	6271	9239	4.23	5.89	21.32	149490600	92340600	61060600	0.05	0.07	0.19	60040000	62840000	64340000
2011-06-15	19313	20031	26307	5698	6276	9336	4.23	5.95	20.50	167285002	64745002	63865002	0.06	0.07	0.20	62410000	66060000	67350000
2011-06-16	19684	20284	24918	5618	6132	8838	4.02	5.58	19.05	140547900	64227900	63367900	0.05	0.07	0.19	62340000	65900000	64430000
2011-06-17	19416	20235	25994	5721	6363	9533	4.25	6.17	21.66	141900000	65260000	64750000	0.05	0.08	0.20	64820000	66880000	68200000
2011-06-18	19047	19758	25883	5572	6220	9398	3.89	5.83	22.88	407630300	291060300	135210300	0.05	0.08	0.20	65970000	66830000	72330000
2011-06-19	20466	21267	27069	5779	6447	9731	4.35	6.40	25.49	209520403	101990416	94490418	0.04	0.07	0.20	62800000	65130000	68240000
2011-06-20	22499	22980	27199	5074	5606	8246	3.24	9.02	20.53	255459504	213619504	157969504	0.08	0.11	0.27	75830000	76150000	82130000
2011-06-21	19670	20502	25957	5739	6311	9373	4.32	6.02	22.34	163261806	61871806	61631806	0.05	0.08	0.20	60840000	64580000	63290000
2011-06-22	19723	20486	26789	5837	6429	9557	4.60	6.41	22.48	172660000	97490000	77150000	0.05	0.08	0.21	64000000	66990000	68220000
2011-06-23	19979	20589	25415	5820	6368	9376	4.55	6.23	22.28	154502612	98242612	66342612	0.05	0.07	0.20	64480000	66530000	69140000
2011-06-24	19578	20381	26518	5825	6535	9991	4.54	6.80	26.56	260931000	134041000	115591000	0.05	0.08	0.20	62870000	66400000	69180000
2011-06-25	19385	19957	25898	5725	6437	9985	4.26	7.11	26.29	478647100	337797100	138917100	0.05	0.08	0.21	67440000	69310000	76350000
2011-06-26	20643	21407	27445	5853	6611	10355	4.63	7.29	37.58	392658615	233648519	134228618	0.05	0.07	0.22	67270000	68100000	72170000
2011-06-27	20247	20872	25893	5879	6541	9883	4.71	6.77	23.41	166486140	125576140	93743240	0.05	0.07	0.20	68425500	72615500	72682800
2011-06-28	20015	20572	25832	5886	6526	9886	4.66	6.61	23.71	230272356	113982356	79102356	0.06	0.07	0.20	62650000	67050000	69030000
2011-06-29	19959	20749	26655	5930	6600	10106	4.86	6.87	26.55	345515506	193180706	149090706	0.06	0.08	0.23	66130000	71920000	74600000
2011-06-30	19947	20664	25615	5811	6399	9397	4.46	6.23	21.14	219856131	94976131	91636131	0.05	0.07	0.19	63650000	69280000	69100000
Minimum	18404	18065	23761	5074	5606	8246	3.24	4.94	15.89	67080000	53192209	53800000	0.04	0.06	0.18	53840000	54160000	57010000
Maximum	22499	22980	27445	5930	6611	10355	4.86	9.02	37.58	478647100	337797100	157969504	0.08	0.11	0.27	75830000	76150000	82130000
Average	19654	20343	25814	5664	6265	9323	4.15	6.14	21.96	202915964	122013133	85511375	0.05	0.07	0.20	63129850	65023850	67759093

Table 8.6: *Computational results measuring the performance of the developed solution methods.*

for the exact method for three of the instances, and for these instances, the number of unallocated check-in groups is 6 higher in average. This indicates that two or more check-in groups have overlapping sets of preferred counters, and choosing not to allocate one or more of these check-in groups leads to a better overall preference compliance. In today's practice at CPH, unallocated check-in groups are allocated manually. In this manual step, the counter demand and sets of allowed counters are typically adjusted so that all check-in groups are allocated before the final solution is published to the handling agents, airlines and duty managers.

For 18 of the instances, expanding the sets of allowed check-in counters (configuration B and C) yielded a better preference compliance than if using the sets of allowed counters currently used at CPH. In average, the preference compliance was improved by 6.1% for configuration C and 3.3% for configuration B. This sort of contradicts the general perception of the airlines and handling agents, that allowing too much flexibility to the optimization tool will lead to a worse preference compliance. In fact, the opposite is shown, and for all instances, configuration C even yields a better preference compliance. CPH should therefore consider using this configuration instead of configuration A. Moreover, it is shown that the current weight setting so strongly favors preference compliance, that the impact on queue excess and baggage inflow excess is almost equal to zero. In fact, for the exact method the excess is the same for all configurations for all but two instances (2011-06-27 and 2011-06-29) and for the greedy heuristic, the excess is the same for all configurations for all but four instances (2011-06-01, 2011-06-04, 2011-06-27, and 2011-06-30).

Table 8.5 shows that the greedy heuristic in general performs almost as well as the exact method. This proves the potential for a heuristic approach to solving the problem.

This potential is also shown when considering the performance of the algorithms. As can be seen from Table 8.6, both algorithms performs best for configuration A and worst for configuration C. For the exact algorithm, the longest solve time is 37.58s for instance 2011-06-26 in configuration C as opposed to only 3.24s for instance 2011-06-20 in configuration A. Equivalently, for the greedy heuristic, the longest solve time is 0.27s for instance 2011-06-20 in configuration C as opposed to only 0.04s for e.g. instance 2011-06-02 in configuration A. Both algorithms thus scale badly, however, given that the problem is considered at the tactical level, the solution times for both algorithms in all configurations are currently acceptable.

As can be seen from the results, a heuristic approach to solving the problem indeed has potential. First of all, the algorithm solves the problem very efficiently for all instances in all configurations. Second of all, the quality of the solutions is rather good. The total queue excess and baggage excess is more or less the same as for the exact method. The preference compliance is a little less, and the number of unallocated check-in groups a little higher, however, the developed algorithm is the simplest form of greedy heuristic and one could easily imagine that a more sophisticated heuristic could overcome these issues.

The computational results demonstrates that using the exact method, is viable for CPH, given the current layout of the airport, the current configuration of the problem (A) and the current amount of traffic. The advantage of the chosen modeling approach is that it provides a general, yet simple setup, capable of modeling all of the operational needs at CPH, and the chosen solution approach is guaranteed to find an optimal solution to the problem, if one such exists.

8.6 Conclusions

In this paper, the problem of allocating check-in counters to check-in groups (single flight of common use areas) has been addressed. A formulation of the problem taking airline preferences, available queuing area in the terminals and the capacity of the baggage handling facilities has been presented.

The model has been implemented as part of an optimization tool at CPH. The responsibility had previously been distributed among several different stakeholders in an uneven flow. After CPH took over, the process was simplified to having only one responsible with a well defined and transparent set of processes. The optimization tool has helped CPH improve the distribution of queues and baggage inflow in the terminals for international departures and led to a better utilization of the existing check-in counters, enabling CPH to both reduce and postpone the investments in additional check-in counters. Moreover, CPH has obtained a valuable insight into the check-in process. CPH actively uses this insight in an effort to make the airlines and handling agents change non-optimal opening patterns agreed in their contracts, thereby changing the counter demand. The optimization tool is still running (January 2014) and CPH is continuously improving parameters used in the model and the optimization tool itself.

The solution method currently implemented at CPH solves the problem to optimality by enumerating all variables and using a state-of-the-art MIP solver. This method is fully viable to the current setup at CPH and it is shown to perform well on a series of real-life data instances. However, it is shown that a simple greedy heuristic easily matches the current solution method in terms of both performance and solution quality.

Bibliography

Chun, H. W., Mak, 1999. Intelligent resource simulation for an airport check-in counter allocation system. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 29 (3), 325–335.

CPH, 2013. Cph official website.
URL `http://www.cph.dk`

Duin, C. W., Sluis, E. V. D., 2006. On the complexity of adjacent resource scheduling. Journal of Scheduling 9 (1), 49–62.

Hon, W. C., 1996. Scheduling as a multi-dimensional placement problem. Engineering Applications of Artificial Intelligence 9 (3), 261–273.

Park, Y., Ahn, S. B., 2003. Optimal assignment for check-in counters based on passenger arrival behaviour at an airport. Transportation Planning and Technology 26 (5), 397–416.

Parlar, M., Sharafali, M., 2008. Dynamic allocation of airline check-in counters: A queueing optimization approach. Management Science 54 (8), 1410–1424.

Tang, C.-H., 2010. A network model for airport common use check-in counter assignments. Journal of the Operational Research Society 61 (11), 1607–1618.

van Dijk, N. M., van der Sluis, E., 2006. Check-in computation and optimization by simulation and ip in combination. European Journal of Operational Research 171 (3), 1152–1168.

Wang Yeung, B. K., Chun, H. W., 1995. Check-in counter allocation using genetic algorithm. EXPERSYS, 197–202.

Yan, S., Chang, K. C. K.-C., Tang, C. H. C.-H., 2005. Minimizing inconsistencies in airport common-use checking counter assignments with a variable number of counters. Journal of Air Transport Management 11 (2), 107–116.

Yan, S., Tang, C.-H., Chen, M., 2004. A model and a solution algorithm for airport common use check-in counter assignments. Transportation Research, Part A (Policy and Practice) 38A (2), 101–125.

Chapter 9

Aircraft Stand Allocation with Associated Resource Scheduling

Tor Fog Justesen*† Jesper Larsen † David Ryan ‡ Richard Lusby † Anders Høeg Dohn§

*Airport Optimization, Copenhagen Airports A/S
Lufthavnsboulevarden 6, 2770 Kastrup

†Department of Management Engineering, Technical University of Denmark
Produktionstorvet, Building 426, DK-2800 Kgs. Lyngby, Denmark

‡Department of Engineering Science, The University of Auckland
70 Symonds Street, Auckland, New Zealand

§Security Resource Management, Copenhagen Airports A/S
Lufthavnsboulevarden 6, 2770 Kastrup

Contact: Tor Fog Justesen (+45 23 24 62 47 / tor.justesen@cph.dk)

Abstract An aircraft turn-round refers to the set of processes taking place from when an aircraft parks at its arrival stand until the time it departs from its departure stand. When handling a turn-round, the different processes involved (arrival, disembarkation of passengers, cleaning, etc.) require different ground handling resources (taxiways, aircraft stands, gates, etc) at different times. Each resource can be claimed by at most one turn-round at a time. The *aircraft stand allocation problem with associated resource scheduling* is the problem of allocating the required ground handling resources to handle a given set of aircraft turn-rounds. We develop a set packing-based model formulation of the problem which is both flexible in the sense that it can encapsulate any type of resource required during the handling of a turn-round and strong in the sense that conflicts that occur when two or more turn-rounds simultaneously claim the same resource are handled implicitly. To solve the model, an LP-based heuristic is developed. The heuristic iteratively solves a relaxed, restricted version of the problem, adding extra variables at each iteration if needed. The additional variables are identified by a cost-based partial enumeration of the possible variables for each turn-round and the heuristic stops when the first feasible solution is encountered. The heuristic has been tested on real data from Copenhagen Airport, Denmark, with a special focus on tactical day-to-day planning. The results show that the method generates high-quality feasible solutions within reasonable time for tactical planning.

Keywords: *aircraft turn-round*, *stand allocation*, *set packing*, *optimization*, *column generation*, *KPI*

Acknowledgements: The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7-PEOPLE-2009-IRSES) under grant agreement no 246647 (project OptALI).

9.1 Introduction

The operation of an airport gives rise to many different resource allocation and scheduling problems that can be modeled and solved using operations research techniques. In this paper we address the handling of aircraft turn-rounds. When an aircraft has landed at an airport, it typically taxis directly from the runway to a pre-allocated aircraft stand (a parking area for an aircraft). After passenger disembarkation the aircraft might be idle for a while before the departure. While idle, the aircraft may be towed from one stand to another.

Handling an aircraft turn-round is complex; from the time the aircraft lands at an airport until it departs again, many different ground handling resources are required at different times. For example, during taxi-in and taxi-out, the aircraft requires a free taxiway route between the apron and the runways. While parked, an available aircraft stand is required, and available gates and gate facilities as well as equipment for unloading and loading baggage and cargo are required.

In most airports, ground handling resources such as runways, taxiways, de-icing platforms, stands, and gates are scarce and it is typically either very expensive (or sometimes even impossible) to add more of them. It is therefore crucial for an airport that the usage of the available ground handling resources is optimized.

When determining the best way to handle a turn-round, the most critical ground handling resource to allocate is often the stand(s) at which the aircraft is parked, and in most of the earlier previous work on this topic (see e.g. Babic et al. [1984], Cheng [1997], Haghani and Chen [1998], Bolat [1999], Yan and Ho [2001], Ding et al. [2003], and Ding et al. [2005]), stands are the only ground handling resources considered. The problem of handling turn-rounds at airports is therefore often referred to as the *stand allocation problem* and a solution to the problem typically only specifies the usage of the stands. In this paper we focus on the problem in a broader perspective and formulate a model that can also take other ground handling resources into account. The handling of a turn-round can be described by a *handling schedule* which is an exact description of which ground handling resources are needed and at what times. Since a turn-round can be handled in many different ways, many possible handling schedules may exist for each turn-round. For instance, there may be more than one stand at which it is possible to handle the turn-round, or it may be decided that the aircraft should be moved to an intermediate parking position while being idle in order to free up the most requested stands. A solution to the problem is a set of handling schedules consisting of exactly one handling schedule for each turn-round, and a feasible solution is a set of *non-conflicting* handling schedules. A set of handling schedules is termed non-conflicting if no two turn-rounds attempt to claim the same resource simultaneously.

Major European airports are daily visited by several hundred aircraft and while optimizing the usage of the ground handling resources, one should also take into account the requirements and preferences expressed by the different stakeholders of the problem, namely the airlines, the passengers and the airport itself.

Though the problem of handling turn-rounds in an airport is most obviously perceived as an operational problem, a different variant of it also arises at the strategical and tactical levels.

At the strategic level, one typically considers a long-term planning horizon (e.g. from six months to twenty years) and the primary focus is on capacity assessment in relation to a forecasted resource demand. The tactical-level variant resembles the strategic-level variant; however, at the tactical level one typically considers a short-/medium-term planning horizon (e.g. from one day to six months) and the available data is very accurate and only few assumptions are needed. The emphasis is on finding an optimal or near-optimal feasible solution where all aircraft turn-rounds are handled and where airline, airport, and passenger preferences are satisfied to the greatest possible extent.

Operational problems are defined as those that occur on a real-time day-to-day basis when predetermined resource allocation plans need to be adjusted due to unforeseen disruptions. Problems occurring at the operational level often need almost immediate resolution; however, typically the considered planning horizon is very short, and all necessary data is given and only very few assumptions are needed (if any). The aim is to handle any type of disruption without changing the original plan, however, in some cases it might be necessary to re-allocate one or more aircraft in order to re-establish a feasible plan.

In this paper we address the tactical-level variant of the problem and present a set packing-based model with a resource-based constraint system which easily allows the possibility to dynamically modify the problem and which accurately models any type of ground handling resources involved in the handling of a turn-round. To solve the problem, we consider two different methods: total enumeration (TE) and partial enumeration (PE). The TE method finds an optimal solution by enumerating all variables and solving the problem using a standard MIP solver. Inspired by column generation, the PE method iteratively solves a relaxed, restricted version of the problem using a standard LP solver, and introduces columns as they are needed to ensure that all turn-rounds are allocated and that no conflict occurs. The additional columns are identified by a simple heuristic embodying expert knowledge about the problem and a cost-based partial enumeration of the possible variables for each turn-round.

Previous work on this problem tends to pursue approaches based on heuristics only, as will be discussed in Section 9.3. Seen from a theoretical point of view, such approaches have the advantage that they can deal with large problems, however they suffer from two disadvantages; they are not guaranteed to find a feasible solution even if one such exists and they provide no guarantee of optimality or even how far from optimality the heuristic solution is. Nevertheless, many of the software systems implemented at the different airports are based on heuristics. The approach developed in this paper has not previously been used for this problem, and the goal is to show that solution algorithms combining optimization techniques and heuristics can be a viable alternative to solution methods based on heuristics only.

The remainder of this paper is organized as follows. Section 9.2 provides a more detailed description of the problem and introduces some airport terminology and Section 9.3 provides a review of previous work. A description of our set packing-based model is given in Section 9.4 and an introduction to the developed solution algorithm is presented in Section 9.5. Computational experiments are given in Section 9.6, and finally we draw some conclusions in Section 9.7.

9.2 Problem Definition

The work presented in this paper was done in collaboration with the company Copenhagen Airports A/S that owns and operates the airports at Copenhagen (CPH) and Roskilde (RKE) in Denmark. In the following, all terminology will be described in accordance with the terminology used by Copenhagen Airports A/S as well as the European Organization for the Safety of Air Navigation, EUROCONTROL.

An aircraft stand (or just stand) is a designated area on the apron intended to be used for parking an aircraft. A stand can be approved for many different aircraft types, but can only hold one aircraft at a time. The approval of a stand for a given aircraft type depends on the size of the stand, the equipment available at the stand (Docking Guidance System (DGS), power supply, air supply, fuel pit, etc.), and the accessibility of the stand.

If a stand allows the passengers to disembark or board the aircraft via a jet bridge or if the passengers can walk directly between the gate and the aircraft, the stand is also referred to as a contact stand, otherwise it is referred to as a remote stand. If an aircraft is parked at a remote stand, the passengers must be transported by bus to and from the gate. The combination of contact / remote stands and availability of walkways varies from airport to airport.

The origin and destination airport determine the *inbound* respectively *outbound status* of an aircraft. The status of the passengers is identical to the status of the aircraft and determines which set of rules for immigration and security that must be applied when handling transferring and local arriving respectively local departing passengers.

A *gate* is the passageway through which passengers proceed in order to get from the terminal buildings to the aircraft (and vice versa). If the passengers can walk between the gate and the aircraft (e.g. via a jet-way bridge), the passengers are said to be pier-serviced. If not, the passengers are typically transported with a bus, in which case the passengers are said to be bussed.

Depending on its facilities (lounge, security check point, etc.) and placement within the airport (before or after passport control, possibility to separate passengers from other passengers, etc.), a gate can allow the handling of one or more statuses. Passengers must always either board or disembark the aircraft through a gate that can handle their status.

Due to safety restrictions and operational considerations, a given gate usually only allows pier-service of the passengers for a very limited set of contact stands (one—three). Pier-service of the passengers is per definition not possible for remote stands. When busing, any gate capable of handling the status of the passengers can in principle be allocated to the given arrival/departure. However, to avoid using expensive stands for buses, many airports have a set of designated bus gates, where space for the bus and loading/unloading of the passengers is allocated in front of the gate.

An *aircraft turn-round* (or just turn-round) refers to the set of processes taking place from the time an aircraft at its *arrival stand* until it departs from its *departure stand*. The arrival stand refers to the first stand where the aircraft is parked upon arrival, and the last stand refers to the last stand where the aircraft is parked before departing. The moment the aircraft parks on the arrival stand is referred to as the *in-block time* and the moment the aircraft releases it breaks at the departure stand is referred to as the *off-block time*. A turn-round can be seen as the linking of an *arrival operation*, a *departure operation*, and an aircraft. Typically, the following information is thus available (as a minimum): aircraft type, origin airport, scheduled and estimated time of arrival, inbound flight type (position flight, cargo flight, passenger flight, charter flight, etc.), number of arriving passengers (transfer/local), number of departing passengers (transfer/local), outbound flight type (position flight, cargo flight, passenger flight, charter flight, etc.), scheduled and estimated time of departure, and destination airport.

A turn-round is said to have a *status change* if the inbound status is different from the outbound status. When allocating a turn-round to a contact stand, the stand should preferably allow both the arriving and departing passengers to be pier-serviced. See Figure 9.1 for an illustration of a turn-round with a status change.

9.2.1 Stand clusters

For various commercial and operational reasons, an airline may prefer that their arrivals and departures are handled at certain stands. Equivalently, it may be preferred that some aircraft types or turn-rounds arriving from/departing to specific locations are handled at certain stands. A *stand cluster* is a prioritized set of stands describing such preferences. The airline, type of aircraft, origin/destination, time of day, etc. determine which stand cluster a given operation is qualified for. For airlines only visiting the airport infrequently the choice of stand is less important, however, as a general rule, these airlines prefer to be parked as close to the terminal buildings as possible. To accommodate this request, a default stand cluster is defined. The default cluster contains all stands of the airport, and the stands are prioritized according to their distance to the terminal buildings. Airlines that have not expressed specific stand preferences are automatically qualified for the default cluster.

An operation can only be qualified for one stand cluster and in the following we refer to the stand clusters of the arrival and departure operations of a turn-round as the *arrival stand cluster* respectively *departure stand cluster* of the turn-round. In the US, stands and gates are to a large extent dedicated to the individual airlines, i.e. the different stand clusters are in principle

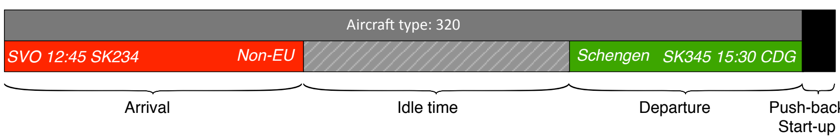

Figure 9.1: *Illustration of a turn-round with status change; the inbound status is Non-EU (arrival from Moscow) and the outbound status is Schengen (departure for Paris).*

enclosed areas, where **only** operations of the particular airline are allowed to be handled. In most European airports, however, the stand clusters mainly express *preferences*, i.e. each operation can in principle be allocated to any stand of the airport, regardless of which stand cluster the particular stand belongs to, yet the airports always strive to fulfill the stand preferences to the greatest possible extent. If a stand belongs to more than one cluster, the clusters involving the stand are ranked in accordance with the internal commercial and strategic prioritization of the airlines operating in the airport. All other clusters and the default stand cluster are given the lowest possible ranking.

The idea of stand clusters is to provide an easy way of configuring the stand preferences expressed by the airlines. Focusing on their own operation, each airline can only describe which stands they prefer and with what priority. Correspondingly, the airport should not decide upon the preferences of the airlines, but only act when possible conflicts are identified; if two or more airlines prefer the same stand, the airport should then rank these airlines, such that conflicts where the airlines request the stand at the same time can be resolved. Table 9.1 describe the most "most officially expressed" stand clusters at CPH.

9.2.2 Handling schedules

A turn-round may be handled in many different ways. The general perception is often that the preferred way of handling a turn-round is to park the aircraft at a contact stand that allows pier-service of both the arriving and departing passengers and then leave the aircraft at this stand during its entire stay in the airport. However, in many airports, contact stands constitute a scarce resource, and the airport must pay close attention to how these stands are used. As an example, the airport may decide to move the aircraft of very long turn-rounds to e.g. a remote stand while the aircraft is idle.

The process of moving an aircraft between two points (typically stands) is called a *towing*. The use of towings and busings generally increases the operational flexibility of the airport, however, it also increases the total operational costs for both the airlines and the airport. A general minimization of the number of towings and busings is therefore often included as one of the objectives when solving the stand and gate allocation problem (see e.g. Bolat [1999], Bolat [2000], Dorndorf et al. [2007], Dorndorf et al. [2008], Nikulin and Drexl [2010], and Dorndorf et al. [2012]). The problem with this approach, however, is that it suffer from the lack of capability to encapsulate the specific handling preferences that are often given by both the airlines and the airport. As an example, for large aircraft carrying more than 350 passengers, towing is preferred to busing of the passengers, whereas for small/medium sized aircraft, busing of the passengers is preferred to towing. Moreover, some airlines may prefer to have their turn-rounds handled on remote stands,

Cluster	**Airline**	**Stands**	**Criteria**
SK-HST-MORNING	SK	D1—D4	All ARN and OSL departures in the interval 06:00—10:00
SK-BIG6	SK	B2—B7	All ARN and OSL operations not in the interval 06:00–10:00 and all GOT, BGO and HEL operations
SK-DSA	SK	Pier B/D1—D4	All operations not qualified for SK-HST-MORNING and SK-BIG6
DY-SCHENGEN	DY	A4/A6/A8/A12—A17	All Schengen operations
FI	FI	B3/B5/B7/B9	All operations
AFKL	AF and KL	A18—A25	All operations
WF	WF	F95/F97	All operations
TP	TP	Pier B/A12—A17	All operations
LX	LX	A14/A15/A18—A25	All operations
LH	LH	Pier B/A14/A15	All operations
Default	All	All stands	All operations

Table 9.1: *Stand clusters used in CPH.*

even though the aircraft type is in fact approved for one or more contact stand, i.e. for these airlines, busing is obviously preferred to pier-service of the passengers.

To accomodate the different operational preferences expressed by both the airlines and the airport, we introduce the notion of a *handling option*. A handling option is a generic description of how a turn-round can be handled encapsulating both how the arriving and departing passengers should be handled (pier-service/bus), how the aircraft should be handled (towings) and where the aircraft should be parked (contact/remote). The idea is then to associate each turn-round with a prioritized set of possible handling options determined by e.g. the airline, the aircraft type, the inbound and outbound status of the aircraft, the inbound and outbound flight type, and the length of the turn-round, and the time of the day.

By using handling options, the airport can easily present to the airlines, the different handling possibilities for their turn-rounds in the airport, and equivalently, the airlines can more easily describe their handling preferences. The most commonly applied handling options at CPH are described in Table 9.2. A *handling schedule* is the operationalization of a handling option specifying

Id	Description
1	No towing, contact stand, pier-serviced arrival and departure
2	No towing, contact stand, busing of arrival, pier-serviced departure, pier-serviced departure
3	No towing, contact stand, pier-services arrival, busing of departure
4	No towing, remote stand, busing of arrival and departure
5	One towing, pier-serviced arrival at contact stand, pier-serviced departure at contact stand
6	One towing, arrival at remote stand, pier-serviced departure at contact stand
7	One towing, pier-serviced arrival at contact stand, departure at remote stand
8	Two towings, pier-serviced arrival at contact stand, pier-serviced departure at contact stand, idle on remote stand
9	Two towings, pier-serviced arrival at contact stand, pier-serviced departure at contact stand, idle in hangar
10	Two towings, pier-serviced arrival at contact stand, pier-serviced departure at contact stand, idle on cargo stand
11	No towing, GO stand, pier-serviced arrival and departure
12	No towing, cargo stand, no passenger handling
13	No towing, hangar, no passenger handling
14	One towing, pier-serviced arrival at contact stand, departure from hangar (no passenger handling)
15	No towing, remote stand, passenger handling
16	No towing, contact stand, no arriving passengers, pier-serviced departure
17	No towing, contact stand, pier-serviced arrival, no departing passengers

Table 9.2: *The most commonly used handling options at CPH.*

exactly which resources are needed and at what time. Two handling schedules are said to be in conflict if they simultaneously claim the same ground handling resource.

9.2.3 Push-back option

Before a departing aircraft can taxi out to the runway, it needs to start up its engines. Dependent on the engine type of the aircraft and the placement of the stand on the apron, the aircraft may start up its engines at the stand. Alternatively, the aircraft must first be moved to a designated start-up position by a push-back tractor. In the following we refer to the entire process from when an aircraft is ready to leave its departure stand until it has started its engines and is ready to taxi out to the departure runway as a *push-back option*. For each stand, multiple push-back options may exist—see Figure 9.2 for an example of the possible push-back options for stand B10 at CPH.

To formally describe a push-back option one can identify which ground handling resources the push-back option requires at what times. The ground handling resources are in this case the push-back tractor and the taxiways. To model that an aircraft only claims a small part of a taxiway at a time, and that a taxiway can be used by many different aircraft at a time, the taxiway network is discretized into a set of non-overlapping *taxiway segments*. Each taxiway segment can

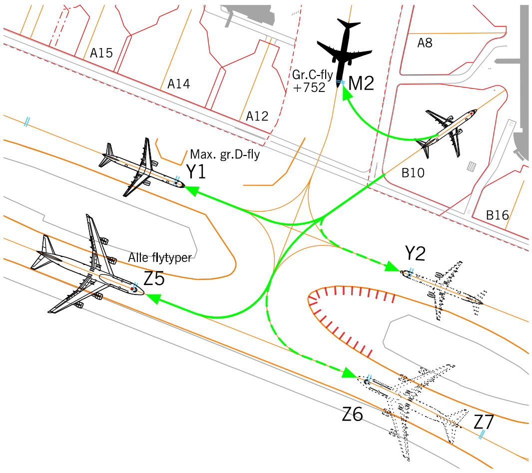

Figure 9.2: *Possible push-back options for stand B10 at Copenhagen Airport.*

be perceived as a ground handling resource that can be claimed by the turn-rounds. Which push-back option a given aircraft can use depends on the aircraft type, the stand, and various technical and operational conditions. Moreover, the various restrictions regarding simultaneous push-back described in [Haagensen, 2012] must be respected. An example of such a restriction is that two simultaneous push-backs must be separated by at least one stand for small aircraft and two stands for large aircraft if the aircraft are being pushed back to the same taxiway.

The start-up positions are typically placed on the taxiways, i.e. while starting its engines and running through the start-up procedure, the aircraft will be occupying a part of the taxiway network. A *push-back schedule* is the operationalization of a given push-back option, accurately describing which taxiway segments are need and at what time, in order to complete the push-back. The time an aircraft needs for its start-up procedure is dependent on the aircraft type; in general, for large wide-body aircraft with large/multiple engines, the start-up procedure can take up to 10 minutes, whereas for smaller aircraft, the start-up procedure only takes 2-4 minutes. The choice of push-back option is thus also dependent on other traffic within the taxiway network.

9.2.4 Stand dependencies

Two stands are said to be dependent if the handling of a turn-round on one of them influences the handling of turn-rounds on the other. Stands that overlap are naturally dependent, however, two stands might also be dependent simply because they are situated close to each other (neighboring stands). One approach to modeling stand dependencies is to have different types of constraints added to the model (see e.g. cf. [Dorndorf et al., 2007], [Dorndorf et al., 2008], [Jaehn, 2010], and Diepen et al. [2012]FIXME!); however, this approach often require one to pre-identify all potential conflicts between all pairs of turn-rounds in the timetable. Furthermore, simply stating a dependency between a pair of stands does not necessarily encapsulate the real nature of the dependency and the constraints might be too restrictive.

A pair of neighboring stands is typically declared dependent in order to prevent either simultaneous movements to or from the stands (push-back, taxi-in, or taxi-out). Recall, that when arriving at stand or leaving a stand, the aircraft moves along the taxiways and basically the stand dependencies are imposed simply to prevent that two or more aircraft simultaneously claim the same piece of taxiway segment. By imposing the restriction that no taxiway segment can be claimed by more than one turn-round at a time, and by including the push-back schedules in the handling schedules, stand dependencies are thus implicitly given by the resource-time interval constraints.

9.2.5 Buffers

To be able to cope with the many small perturbations that occur in a real world timetable, the turn-rounds must be allocated such that they are separated by a small buffer. This buffer can be assigned to both the arrival and the departure of the turn-round, and the size of the buffer can be either fixed, solely based on e.g. the size of the aircraft, or variable, based on statistical knowledge about a given arrival or departure. The buffers can be included, simply by letting each turn-round claim the relevant ground handling resources for the duration of the buffer, even though the turn-round is in principle no longer occupying them.

9.2.6 Objectives

A turn-round is termed *unallocated* if it is not possible to find a conflict free handling schedule for it. An obvious objective when solving the stand and gate allocation problem is to ensure that the number of unallocated turn-rounds is minimized. To ensure that the handling preference specified by the airlines and the airport are satisfied to the greatest possible extent, a second objective is to ensure that the best possible handling option is chosen for each turn-round. Equivalently, CPH wants to ensure that the stand preferences specified by the airlines are satisfied to the greatest possible extent. The stand preferences are expressed via the specified stand clusters and a third objective is therefore to ensure that each operation is allocated to the best possible stand (highest priority) of the cluster it is qualified for, respecting the ranking of the clusters on the individual stands. Finally, CPH wants to maximize the number of passengers that are pier-serviced. Pier-serviced arrivals and departures with many passengers should thus have a higher priority than pier-serviced arrivals and departures with few passengers. The described objectives should be prioritized as follows:

1. Minimize the number of unallocated turn-rounds
2. Maximize handling preference compliance
3. Maximize stand preference compliance
4. Maximize the number of pier-serviced passengers

As described in 9.1, short-/medium-term planning horizons are typically considered at the tactical level. Today, the stand and gate dispatchers at CPH create a plan every evening for the following day. Once the plan is made, it is released and on the day of operation, the job of the dispatchers is to maintain this plan. Given such a setting, one could argue that the efficiency of the solution mehtod is less important and that only optimal solutions should be accepted in order to provide the best starting point for each day of operation. However, in the near future, CPH will replace its current stand and gate allocation system, with a new and modern system capable of planning up to a couple of weeks ahead. The advantage of this, is that the plans can be published to the handlers and airlines more in advance, enabling them to make better planning of their staff and equipment. In such a setting, the dispatchers would have to solve the problem for multiple days at a time, perhaps sequentially. Moreover, the dispatchers would have to maintain these plans, as changes to the timetables are received. To make this process as efficient as possible, the efficiency of the solution method is thus of crucial importance; therfore, if finding near-optimal, operationally applicable solutions can be done much more efficently than finding optimal solutions, this is preferred. Moreover, since the timetables are likely to change anyway, near-optimal solutions are often just as good as optimal solutions, seen from a practical point of view.

9.3 Previous Work

In this section we give an overview of the previous work on the problem of handling turn-rounds in airports. Our aim is to give a concise yet comprehensive view of the literature in this field, focusing only on the tactical variant of the problem.

When looking at the different papers written on this topic, one can observe that most of the papers claim to be dealing with gate allocation (assignment/scheduling). However, in most cases, what they are really considering is the allocation of turn-rounds to aircraft stands. As previously described in Section 9.2, a gate is the part of the terminal building used for passenger handling, whereas an aircraft stand is simply the parking position for the aircraft. A stand can be associated with multiple gates, and the status of the passengers, determine which gate to be use. In the following we will use the term stand allocation where this is in fact the correct term to use.

An extensive survey within different research areas of aircraft stand allocation is given by [Dorndorf et al., 2007]. They describe the problem in a general manner and state the different constraints and objectives typically dealt with when solving this problem. Furthermore, they propose a classification of the primary research directions, some of which are still very open.

Among the first to apply operations research techniques and integer programming modeling to the stand allocation problem were [Babic et al., 1984] and [Mangoubi and Mathaisel, 1985]. [Babic et al., 1984] formulate the gate assignment problem as a linear 0-1 integer programming model. They use a branch-and-bound algorithm to find the optimal solution, however they do not consider transfer passengers. [Mangoubi and Mathaisel, 1985] solve the problem of [Babic et al., 1984] using an LP relaxation and greedy heuristics. They consider the transfer passenger walking distance based on a uniformly distributed gate-to-gate transfer pattern. Furthermore, they formulate the problem as a linearized *quadratic assignment problem*, assuming that all turn-rounds can be allocated to all gates, however, they do not work any further on the proposed model because it is too difficult to solve efficiently. [Haghani and Chen, 1998] also formulate the problem as a quadratic assignment problem introducing some time-indexed binary variables that indicate the assignment of a particular flight to some stand in a given time slot. They develop a branch-and-bound based solution algorithm as well as a heuristic solution algorithm to solve the problem.

Also focusing on minimizing the total walking distance for passengers, [Yan and Chang, 1998] formulate the problem as a multi-commodity network flow problem and develop both an algorithm based on Lagrangian relaxation with sub-gradient methods, accompanied by a shortest path algorithm and a Lagrangian heuristic to solve it. They compare their solution algorithm with the heuristic developed in [Mangoubi and Mathaisel, 1985] and conclude that their solution algorithm is faster.

[Lim et al., 2005] consider an approach to the stand allocation problem where flight arrivals and departures are associated with a time window instead of having a fixed schedule to capture the more realistic situation where flight arrival and departure times can change. An equivalent approach is taken by [Yan and Ho, 2001] who formulate a model that includes several delay choices to each flight.

[Diepen et al., 2007] solve the problem on the tactical level for Amsterdam Airport Schiphol. The main objective is to obtain solutions that can cope with small perturbations in the traffic without the need for re-planning big parts of the schedule. They introduce the concept of a *stand plan* representing a possible series of flights assigned to the same stand and each stand plan is associated with a cost describing its robustness as an arc-tangent based function of the total amount of idle time in the stand plan. The problem is solved by a two-phase procedure; in the first phase, a near-optimal stand plan is found for each stand by the use of column generation, and then in the second phase, the problem of ensuring that the stand plans found in phase one also respect the various stand dependencies (shadow restrictions) is solved manually by the planners at Amsterdam Airport.

[Dorndorf, 2002] model the problem as a resource-constrained project scheduling problem; the handling of a turn-round consists of a set of activities and these must be scheduling, respecting both time windows and certain precedence constraints. [Dorndorf et al., 2007] pursues the idea of considering the handling of a turn-round as a problem of scheduling a set of activities and formulate a model, where each turn-round is split into three activities: an arrival activity, an optional intermediate parking activity, the length of which depends on the length of the turn-round, and a departure activity. In a related approach, [Dorndorf et al., 2008] formulate the stand allocation problem as a clique partitioning problem. Once again, turn-rounds are modeled as *activities*, and long turn-rounds are split into an arrival activity, a parking activity, and a departure activity. The clique partitioning formulation of corresponding multi-project scheduling problem.the stand allocation problem is extended by [Dorndorf et al., 2012] to minimize the deviations from a reference schedule.

[Bergeron and Vanderstraten, 1988], [Gosling, 1990], and [Muthukrishnan, 1991] worked on solving the stand allocation problem by using expert systems. The aircraft-to-stand allocation procedure simulates the logic-based procedure of stand allocation by experienced experts at the airport. [Cheng, 1997] propose a knowledge-based airport stand allocation system that is integrated with mathematical programming. Other work based on simulation can be found in, for example, Cheng [1998] and [Yan et al., 2002]. In the work of [Yan et al., 2002], a simulation framework was proposed that not only can analyze the effects of stochastic flight delays on static gate assignments, but also can evaluate flexible buffer times and real-time gate assignment rules that the airport authority uses. A simulation based on airport operations at Chiang Kai-Shek airport is also used to evaluate the gate assignment performance.

In general, many researchers have tried various heuristic approaches. [Haghani and Chen, 1998] propose a heuristic that assigns flights with relatively more passengers to gates having smaller walking distance coefficients when there is no overlapping of the ground time of the flights. [Xu and Bailey, 2001] provide a Tabu Search (TS) for the gate assignment problem. Their algorithm exploits the special properties of different types of neighborhood moves, and adopts an effective candidate list strategy. Equivalently, [Ding et al., 2004] design a greedy algorithm and use a TS meta heuristic to solve the problem. The greedy algorithm minimizes ungated flights while providing initial feasible solutions and a new neighborhood search technique, the Interval Exchange Move, which is more flexible and more general than previously employed exchange moves used for this problem, is then used to find good quality solutions. [Ding et al., 2005] compare the Interval Exchange Tabu Search (ITS) approach to a Simulated Annealing (SA) approach, and the results show that the ITS approach produces better results than the SA approach, but that the SA approach is faster than the ITS approach. This leads to the development of a hybrid method which inserts TS steps into SA, and this method proves to give very good results. Other works pursuing the idea of heuristics have been done by [Wei and Liu, 2007] who combine genetic algorithms with TS and meta-heuristics and [Drexl and Nikulin, 2008] who model the problem as an integer quadratic assignment problem with multiple objectives and solve it by examining the Pareto front found by Pareto SA. Most recently, [Cheng et al., 2012] solve the problem using different meta-heuristics, namely, genetic algorithm (GA), TS, SA and a hybrid approach based on SA and TS.

9.4 Mathematical Model

Let $\mathcal{R}$ denote the set of turn-rounds, $\mathcal{P}$ the set of all handling schedules, and $\mathcal{P}_r$ the set of all handling schedules for turn-round $r \in \mathcal{R}$ ($\mathcal{P} = \bigcup_{r\in\mathcal{R}} \mathcal{P}_r$). The set of ground handling resources is denoted $\mathcal{R}$ and the set of time intervals obtained through a discretization of the planning period is denoted $\mathcal{I}$.

To model the restriction that no two turn-rounds may claim the same resource simultaneously, one can identify a constraint for each ground handling resource $g \in \mathcal{R}$ in each time interval $i \in \mathcal{I}$. In the following let $\mathcal{K}$ denote the set of *resource-time interval constraints*, i.e. $|\mathcal{K}| = |\mathcal{R}| \times |\mathcal{I}|$, and define the binary parameter a_{kp}, with $a_{kp} = 1$ if handling schedule $p \in \mathcal{P}$ is claiming resource-time interval constraint $k \in \mathcal{K}$ and $a_{kp} = 0$ otherwise.

To model the problem, two sets of decision variables are defined: the binary allocation variables x_p where $x_p = 1$ if handling schedule $p \in \mathcal{P}$ is included in the final solution and $x_p = 0$ otherwise and the binary coverage variables π_r, where $\pi_r = 1$ if turn-round $r \in \mathcal{R}$ is unallocated and $\pi_r = 0$ otherwise.

As described in Section 9.2, the objectives that have been identified at CPH are (in prioritized order):

1. Minimize the number of unallocated turn-rounds
2. Maximize handling preference compliance
3. Maximize stand preference compliance
4. Maximize the number of pier-serviced passengers

Minimizing the number of unallocated turn-rounds is equivalent to minimizing the sum

$$\sum_{r \in \mathcal{R}} \pi_r \tag{9.1}$$

As described in Section 9.2 each turn-round is associated with a prioritized set of possible handling options and for each handling option a set of handling schedules can be created. Each handling schedule $p \in \mathcal{P}$ is thus uniquely associated with both a handling option and a turn-round and the parameter $\rho_p \in 1, \ldots, \rho^{max}$ denotes the corresponding handling option priority, where 1 is the highest and $\rho^{\max}$ the lowest priority. Maximizing the handling preference compliance is thus equivalent to minimizing the sum

$$\sum_{p \in \mathcal{P}} \rho_p x_p \tag{9.2}$$

Recall, that due to the default stand cluster, all operations will be associated with a stand cluster. Now, let $\phi_p^a \in \{1, \ldots, \phi^{\max}\}$ and $\phi_p^d \in \{1, \ldots, \phi^{\max}\}$ denote the priority of the arrival stand respectively departure stand of handling schedule $p \in \mathcal{P}$, where 1 is the highest priority and $\phi^{\max}$ is the lowest. If the arrival stand is not part of the cluster,

Equivalently, let $\theta_p^a \in \{1, \ldots, \theta^{\max}\}$ and $\theta_p^d \in \{1, \ldots, \theta^{\max}\}$ denote the ranking of the arrival stand cluster at the arrival stand respectively the departure stand cluster at the departure stand of handling schedule $p \in \mathcal{P}$, where 1 is the highest priority and $\theta^{\max}$ is the lowest. If

Maximizing the stand preference compliance, is thus equivalent to minimizing the sums

$$\sum_{p \in \mathcal{P}} (\phi_p^a + \phi_p^d) x_p \tag{9.3}$$

and

$$\sum_{p \in \mathcal{P}} (\theta_p^a + \theta_p^d) x_p \tag{9.4}$$

Finally, let n_p^a and n_p^d denote the number of arriving respectively departing passengers that are pier-serviced for a given handling schedule $p \in \mathcal{P}$, i.e. $n_p^a = 0$ if the arrival is handled on a remote stand or the passengers are bussed and $n_p^d = 0$ if the departure is handled on a remote stand or the passengers are bussed. Maximizing the number of pier-serviced passengers is thus equivalent to minimizing the sum

$$-1 \cdot \sum_{p \in \mathcal{P}} (n_p^a + n_p^d) x_p \tag{9.5}$$

The objectives are aggregated into a single objective function using the weighted sum approach. This results in a standard single objective optimization problem. Assuming the weights are all positive, the objective function of thus becomes

$$\begin{aligned} z = \quad & \omega_1 \sum_{r \in \mathcal{R}} \pi_r + \\ & \omega_2 \sum_{p \in \mathcal{P}} \rho_p x_p + \\ & \omega_3 \sum_{p \in \mathcal{P}} (\theta_p^a + \theta_p^d) x_p + \omega_4 \sum_{p \in \mathcal{P}} (\phi_p^a + \phi_p^d) x_p - \\ & \omega_5 \sum_{p \in \mathcal{P}} (n_p^a + n_p^d) x_p \end{aligned} \tag{9.6}$$

Setting the correct values of the weights can be a challenging task. Especially given the fact, that the magnitudes of the parameters may be very different. To deal with this, we therefore consider the normalized version of the objective function:

$$\begin{aligned}\hat{z} = \quad & \omega_1 \sum_{r\in\mathcal{R}} \pi_r + \\ & \omega_2 \sum_{p\in\mathcal{P}} \frac{\rho_p}{\rho^{max}} x_p + \\ & \omega_3 \sum_{p\in\mathcal{P}} \left(\frac{\theta_p^a + \theta_p^d}{2\theta^{max}}\right) x_p + \omega_4 \sum_{p\in\mathcal{P}} \left(\frac{\phi_p^a + \phi_p^d}{2\phi^{max}}\right) x_p - \\ & \omega_5 \sum_{p\in\mathcal{P}} \left(\frac{n_p^a + n_p^d}{N}\right) x_p \end{aligned} \tag{9.7}$$

where N is the total number of arriving and departing passengers.

The stand and gate allocation problem can thus be formulated as the following integer programme (SGAP):

$$\begin{aligned} \textbf{minimize} \quad & \hat{z} & & (9.8) \\ \textbf{subject to} \quad & \sum_{p\in\mathcal{P}} x_p + \pi_r = 1 & \forall\, r \in \mathcal{R} \quad & (9.9) \\ & \sum_{p\in\mathcal{P}} a_{kp} x_p \le 1 & \forall\, k \in \mathcal{K} \quad & (9.10) \\ & x_p \in \{0,1\} & \forall\, p \in \mathcal{P} \quad & (9.11) \\ & \pi_r \in \{0,1\} & \forall\, r \in \mathcal{R} \quad & (9.12) \end{aligned}$$

where the objective function (9.8) minimizes the total cost of handling all aircraft turn-rounds. Constraints (9.9) ensure that all aircraft turn-rounds are assigned a handling schedule, whereas Constraints (9.10) enforce the restriction that at most one turn-round can claim any ground handling resource at any time interval. Finally, Constraints (9.11) and (9.12) define the binary restrictions on the decision variables. Note, that the variables π_r could also just have been defined as positive variables ($\pi_r \ge 0\ \forall\ r \in \mathcal{R}$), as 9.9 will ensure, that they always get a value of either 1 or 0 in an optimal solution.

Given the nature of the constraint system, it is obvious that the model implicitly obtains a conflict free set of handling schedules for the turn-rounds. Furthermore, one can also dynamically consider additional handling schedules for turn-rounds without any fundamental change to the model itself. Additional handling schedules can easily be included as extra columns in the model. We exploit this property when we develop an efficient solution algorithm in Section 9.5.

9.4.1 Integer solutions

Define the 0-1 matrix $T = (T_{rp})$ with $|\mathcal{P}|$ columns and a row for each turn-round, i.e. the dimension of T is $|\mathcal{R}| \times |\mathcal{P}|$, where $T_{rp} = 1$ if column $p \in \mathcal{P}$ is a handling schedule for turn-round $r \in \mathcal{R}$ (each column of T contains just one nonzero element) and $T_{rp} = 0$ otherwise. Equivalently, define the 0-1 matrix $A = (A_{kp})$ with $|\mathcal{P}|$ columns and a row for each resource-time interval constraint, i.e. the dimension of A is $\left(\sum_{m\in\mathcal{M}} |\mathcal{I}_m|\right) \times |\mathcal{P}|$, where $A_{kp} = 1$ if resource-time interval $k \in \mathcal{K}$ is claimed by handling schedule $p \in \mathcal{P}$ and $A_{kp} = 0$ otherwise. Using the matrices T and A, the formulation of SGAP given above is equivalent to the one given below:

$$\textbf{minimize} \qquad z = \mathbf{c}^\top \mathbf{x} \tag{9.13}$$

$$\textbf{subject to} \qquad T\mathbf{x} = 1 \tag{9.14}$$

$$A\mathbf{x} \leq \mathbf{U} \tag{9.15}$$

$$\mathbf{x} \in \{0,1\}^{|\mathcal{P}|} \tag{9.16}$$

As can be seen, the constraint matrix of SGAP can be partitioned into $|\mathcal{R}|$ *turn-round blocks*, where each turn-round block B_r, $r \in \mathcal{R}$

$$B_r = \begin{bmatrix} T_r \\ A_r \end{bmatrix} \tag{9.17}$$

contains all possible handling schedules for a particular turn-round. Each turn-round block is a zero-one matrix with a generalized upper bound set partitioning constraint. Exactly as described in Lusby et al. [2011], we thus have that each turn-round block is a *perfect* matrix (cf. Padberg [1972]), and we therefore have a guarantee that the extreme points of the polytope $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{B_r x} \leq \mathbf{1}, \mathbf{x} \geq \mathbf{0}\}$ are integer, see, e.g. [Rezanova and Ryan, 2010] for further details and references. This ensures that if the solution to the LP relaxation of SGAP is fractional, then fractions can only be due to the presence of odd-order two-cycles across multiple turn-round blocks, i.e. fractions will only occur, if two or more turn-rounds are competing for the same resource-time interval.

9.5 Solution Algorithm

Two things can make SGAP difficult to solve: the dimensions of the constraint matrix and the binary constraints on the decision variables. To deal with this, the developed solution method considers a relaxed and restricted version of the problem, R-SGAP, in which the set of handling schedules is restricted to $\mathcal{P}' \subseteq \mathcal{P}$ and the integrality constraints (9.11) and (9.12) are relaxed.

Relaxing the integrality constraints converts the problem into an LP problem which in the general case is computationally much faster to solve than an IP problem. Restricting the set of considered handling schedules leverages the presumption that only a subset of the handling schedules in $\mathcal{P}$ needs to be considered when solving the problem, since most of the possible handling schedules will be non-basic and assume a value of zero in an optimal solution anyway.

Due to the presence of the coverage variables, there will always be a feasible solution to R-SGAP; however, since we have restricted the set possible handling schedules, the solution might be *invalid*, in the sense that one or more turn-rounds are left unallocated, while more possible handling schedules actually exist.

The idea of the solution method is as follows. Initially, $\mathcal{P}' = \emptyset$, i.e. R-SGAP only contains the coverage variables. Then R-SGAP is solved, and the set of unallocated turn-rounds

$$\tilde{\mathcal{R}} = \{r \in \mathcal{R} \mid \pi_r > 0\}$$

is identified. If $\tilde{\mathcal{R}} = \emptyset$, all turn-rounds have been successfully assigned a handling schedule and a valid solution has been found. If, however, $\tilde{\mathcal{R}} \neq \emptyset$, more handling schedules are added to $\mathcal{P}'$, and R-SGAP is re-solved. The solution methods continues this iterative loop until a valid solution has been found or no more handling schedules can be added to $\mathcal{P}'$.

Adding handling schedules to $\mathcal{P}'$ is a two-stage process. First the set of turn-rounds for which more handling schedules must be added to $\mathcal{P}'$ is identified. Then, new handling schedules are identified for each of these turn-rounds, and these are then added to $\mathcal{P}'$.

The main ideas of the solution approach are summarized in Algorithm 2. In the following, we refer to one iteration of the loop in line 3 as one iteration of the solution algorithm.

Algorithm 2: Pseudo-code for the described solution method.

Read airport data, timetable and parameters;
Set $\mathcal{P}' = \emptyset$;
repeat
    Solve R-SGAP using a standard LP solver;
    Identify the set of unallocated turn-rounds $\tilde{\mathcal{R}}$;
    Identify the set of binding constraints $\tilde{\mathcal{K}}$;
    Identify the set of conflicting turn-rounds $\hat{\mathcal{R}}$;
    for *each turn-round* $r \in \tilde{\mathcal{R}} \cup \hat{\mathcal{R}}$ **do**
        if $\mathcal{P}_r \setminus \mathcal{P}'_r \neq \emptyset$ **then**
            Identify the cheapest handling schedules from $\mathcal{P}_r \setminus \mathcal{P}'_r$ compatible with $\tilde{\mathcal{K}}$;
            if *possible to identify new handling schedules* **then**
                Add new handling schedules to $\mathcal{P}'$;
until *valid solution has been found*;
if *solution is fractional* **then**
    Restrict all variables to be binary;
    repeat
        Solve SGAP using a standard IP solver;
        if *solution to SGAP found* **then**
            Save and return solution;
        else
            for *each turn-round* $r \in \mathcal{R}$ **do**
                Identify the cheapest handling schedules from $\mathcal{P}_r \setminus \mathcal{P}'_r$;
                if *possible to identify new handling schedules* **then**
                    Add new handling schedules to $\mathcal{P}'$;
    until *valid solution has been found*;
else
    Save and return solution;

9.5.1 Identifying conflicting turn-rounds

A reasonable explanation to why a given turn-round $r \in \tilde{\mathcal{R}}$ is left unallocated, is that $\mathcal{P}'_r$ is too restricted. However, another explanation could also be that the sets of handling schedules for the turn-rounds preventing r from being allocated are too restricted.

Now, let $\mathcal{K}_p$ denote the set of resource-time interval constraints claimed by handling schedule $p \in \mathcal{P}$:

$$\mathcal{K}_p = \{k \in \mathcal{K} \mid a_{kp} = 1\}$$

The set of resource-time interval constraints claimed by all the handling schedules in $\mathcal{P}'_r$ for a given turn-round $r \in \mathcal{R}$ is thus given by

$$\mathcal{K}_r = \bigcup_{p \in \mathcal{P}'_r} \mathcal{K}_p$$

Now, consider an unallocated turn-round $r \in \tilde{\mathcal{R}}$. Since r is unallocated, none of the handling schedules in the set $\mathcal{P}'_r$ have been chosen in the solution and consequently, a set of binding constraints preventing r from being allocated must exist.

For each optimal primal solution to the R-SGAP, a corresponding dual solution also exists. Let σ_k denote the dual variable of constraint $k \in \mathcal{K}$. If k is binding, the value of its corresponding dual variable is positive, i.e. $\sigma_k > 0$. The set of binding constraints in an optimal solution is thus given as

$$\tilde{\mathcal{K}} = \{k \in \mathcal{K} \mid \sigma_k > 0\}$$

Thereby, the set of binding constraints preventing r from being allocated is given as

$$\tilde{\mathcal{K}}_r = \tilde{\mathcal{K}} \cap \mathcal{K}_r$$

and correspondingly, the set of *conflicting* turn-rounds preventing r from being allocated is given as

$$\hat{\mathcal{R}}_r = \left\{ q \in \mathcal{R} \mid q \neq r, \exists\, p \in \mathcal{P}'_q : x_p > 0 \ \wedge \ \mathcal{K}_p \cap \tilde{\mathcal{K}}_r \neq \emptyset \right\}.$$

and the set of all conflicting turn-rounds for *all* unallocated turn-rounds is then given as

$$\hat{\mathcal{R}} = \bigcup_{r \in \tilde{\mathcal{R}}} \hat{\mathcal{R}}_r$$

To resolve the problem of an invalid solution, the solution method therefore tries to expand $\mathcal{P}'$ by adding more handling schedules for each turn-round $r \in \tilde{\mathcal{R}} \bigcup \hat{\mathcal{R}}$.

9.5.2 Identifying new handling schedules

Consider a conflicting turn-round $r \in \mathcal{R}$ and let $\mathcal{P}'_r \subseteq \mathcal{P}_r$ denote the set of handling schedules already added to $\mathcal{P}'_r$ for r at a given iteration of the solution method. The set of new handling schedules for r must hence be found in the set $\mathcal{P}_r \setminus \mathcal{P}'_r$.

Given the objective function described in Section 9.4 it is clear that the quality of each handling schedule is reflected in its cost; the cheapest handling schedules are the most attractive and the most expensive handling schedules are the least attractive.

Now the idea of the solution method is to perform an enumeration of the handling schedules in $\mathcal{P}_r \setminus \mathcal{P}'_r$ and then select the cheapest of these to add to $\mathcal{P}'$. To keep the set $\mathcal{P}'$ as small as possible, only a subset of the handling schedules in $\mathcal{P}_r \setminus \mathcal{P}'_r$ are added in each iteration. The size of this subset is determined empirically from practical testing of the algorithm (see Section 9.6).

To avoid adding handling schedules, that will not resolve the conflicts of the invalid solution, the solution method will first try to identify handling schedules to the master problem that do not claim any of the binding resource-time interval constraints from the set $\tilde{\mathcal{K}}$. If it is not possible to find any such handling schedules, but more unadded handling schedules exist, these are added.

First, all handling schedules are generated for the first priority handling option and these are sorted according to their costs and added to the model when needed. When no more handling schedules exist for the first priority handling options, handling schedules are generated and added for the second priority handling option. And so forth. Although the number of handling schedules may theoretically be very large for handling options involving towings, the real-world restrictions (like e.g. aircraft type compatibility on the individual stands and status handling) severely limit the number of possible stand combinations. As will be shown in the results presented in Section 9.6, the described variable generation approach enables the solution method to efficiently find operationally applicable, near-optimal solutions.

As can be seen, the R-SGAP initially only contains the best (cheapest) handling schedules for each turn-round, and for each iteration, the handling schedules added gets less and less attractive. In a worst case scenario, the solution method will have to add new handling plans $|\mathcal{H}_r|$ times for a given turn-round $r \in \mathcal{R}$ and eventually all possible handling schedules for r might be added to the problem, i.e. $\mathcal{P}'_r = \mathcal{P}_r$. However, as will be presented in Section 9.6 this case has not yet been observed during the computational experiments.

9.5.3 Integer solutions

Since we are solving the LP relaxation of the restricted master problem, we might end up with a valid solution of the LP that is fractional. However, as described in Section 9.4, the problem will by construction have good integer properties, and it is therefore likely that the solutions to R-SGAP are either integer or not that far from being integer. If the solution to R-SGAP is fractional, we can exploit the described properties of the constraint matrix to impose Ryan-Foster constraint branching when searching for an optimal integer solution (see [Ryan et al., 1981]). However, as the described in Section 9.2 being able to find a near-optimal integer solution efficiently is often preferred to finding an optimal integer solution, even at the strategical and tactical levels. As a consequence, the developed solution method handles fractional solutions, simply by solving SGAP using the handling schedules in $\mathcal{P}'$, i.e. when a valid but fractional solution is found, R-SGAP is "unrelaxed" and solved as an integer problem using a standard IP solver.

If it is not possible to find a valid integer solution to SGAP using only the handling schedules in $\mathcal{P}'$, the solution method will iteratively add handling schedules to $\mathcal{P}'$ for **all** turn-rounds until a valid solution to SGAP is found. In a worst case scenario, the algorithm will end up adding all handling schedules from $\mathcal{P}$ in this process, however, as will be presented in Section 9.6 this case has not yet been observed during the computational experiments.

9.5.4 Optimality

By terminating the solution method the first time a valid solution is encountered, we have no guarantee that the found solution is optimal. However, since we are only adding the best handling schedules for each turn-round in each iteration, it is likely that the objective value of the found solution is not that far from the objective value of the optimal solution (see Section 9.6).

9.5.5 Reducing the number of constraints

The number of constraints, $|\mathcal{K}|$, is determined by the number of resources, $|\mathcal{R}|$, and the number of time intervals, $|\mathcal{I}|$. Moreover, the number of time intervals is given by the granularity of the time discretization and the length of the planning period. In a tactical setting, one typically considers a planning period of at least 12 hours. Finding the right time discretization is essentially a tradeoff between accuracy and complexity. A fine time discretization accurately models the problem, but the number of constraints increase the complexity of the model. On the other hand, a more coarse discretization, reduces the number of constraints, but it also correspondingly decreases the capacity of the resources.

One simple way of reducing the number of constraints is to use different time discretizations for the different *types* of resources considered. For example, to be able to model the nature of a push-back, a rather fine time discretization is needed, whereas a more coarse time discretization may suffice for stands.

The number of constraints can also be reduced by allowing that resources that are less critical can be claimed by more than one turn-round at a time, enabling an even coarser time discretization. At the tactical level, one example of such less critical resources are the taxiways. As earlier described, one would need a very fine time discretization (perhaps 15 seconds) to fully capture the nature of a push-back. However, an exact planning of push-backs many hours ahead does not really make sense as it is very likely that the actual time of for each departure will deviate from the scheduled times. In a real life setting, it will therefore suffice to consider the taxiway resources in time intervals of perhaps 60 seconds and then allow up to two aircraft to simultaneously claim each taxiway resource. This approach will still encapsulate the needed separation of departures on neighboring stands in order to avoid simultaneously push-backs. Potential conflicts arising if two turn-rounds have been planned to claim the same taxiway segment within the same time interval at the tactical level can be resolved on the actual day of operation, simply by holding back one of the turn-rounds for a very short amount of time.

Let $\mathcal{M}$ denote the set of resource types and let $\mathcal{R}_m \subseteq \mathcal{R}$ denote the set of all resources of type $m \in \mathcal{M}$. Furthermore, let $\mathcal{I}_m$ denote the set of time intervals obtained from the discretization of the planning period for resource type $m\mathcal{M}$ and $\mathcal{K}_m$ the set of resource-time interval constraints for resources of type $m\mathcal{M}$. Finally, let U_m denote the upper bound for the number of turn-rounds that may claim any resource $r \in \mathcal{R}_m$ in any given time interval $i \in \mathcal{I}_m$. A more compact formulation of the stand and gate allocation problem is thus given by the following integer program (SGAP), which can also be solved using the described solution method:

$$\textbf{minimize} \quad \hat{z} \tag{9.18}$$

$$\textbf{subject to} \quad \sum_{p\in\mathcal{P}} x_p + \pi_r = 1 \qquad \forall\ r \in \mathcal{R} \tag{9.19}$$

$$\sum_{p\in\mathcal{P}} a_{kp} x_p \leq U_m \qquad \forall\ m \in \mathcal{M}\ \forall\ k \in \mathcal{K}_m \tag{9.20}$$

$$x_p \in \{0,1\} \qquad \forall\ p \in \mathcal{P} \tag{9.21}$$

$$\kappa_r \in \{0,1\} \qquad \forall\ r \in \mathcal{R} \tag{9.22}$$

9.6 Results

The proposed methodology has been tested by allocating 15 real-life timetables representing typical busy days during the summer and winter seasons at CPH. For each timetable, the same stands, gates, and taxiways are included in the model. The characteristics of the timetables considered are given in Table 9.3. To reflect the current process at CPH and to include the turn-rounds

Timetable	# Turn-rounds	# Turn-rounds - small/medium aircraft	# Turn-rounds - large aircraft	# Turn-rounds - no status change	# Turn-rounds - status change	# Turn-rounds - peak hour	Peak hour (UTC)	# Operations - Domestic	# Operations - Schengen	# Operations - Non-Schengen	# Operations - Non-EU	# Arriving pax - total	# Departing pax - total
2012-07-02	484	454	30	382	102	56	04:50—05:50	79	666	98	125	52,194	54,258
2012-07-03	442	414	28	358	84	55	04:50—05:50	78	618	90	98	45,519	46,162
2012-07-04	451	420	31	360	91	58	04:50—05:50	66	634	82	120	46,531	46,632
2012-07-05	448	416	32	365	83	57	04:50—05:50	63	635	85	113	48,050	47,329
2012-07-06	442	411	31	341	101	61	22:05—23:05	59	615	91	119	50,301	50,001
2012-07-07	372	337	35	295	77	65	23:00—00:00	26	530	62	126	45,897	47,016
2012-07-08	401	372	29	324	77	56	02:55—03:55	46	573	69	114	47,628	48,146
2012-07-18	312	290	22	244	68	58	22:05—23:05	43	432	67	82	32,664	32,693
2012-10-08	483	461	22	390	93	47	05:00—06:00	110	685	89	82	50,963	52,157
2012-10-09	447	429	18	381	66	47	04:35—05:35	105	648	82	59	44,366	44,446
2012-10-10	460	438	22	390	70	48	04:50—05:50	104	658	76	82	46,500	46,318
2012-10-11	473	451	22	387	86	49	05:00—06:00	106	678	83	79	50,705	50,054
2012-10-12	470	446	24	379	91	47	01:40—02:40	96	652	84	108	52,759	52,074
2012-10-13	320	298	22	258	62	47	22:45—23:45	40	429	55	116	36,794	37,280
2012-10-14	350	333	17	281	69	45	04:45—05:45	64	489	59	88	39,116	40,009

Table 9.3: *Timetables used for testing.*

that arrive on the day in consideration and stays on ground until the day after, each timetable contains all turn-rounds arriving from 18h on the previous day until 06h on the next day. As an example, the timetable 2012-07-02 contains all turn-rounds arriving between 2012-07-01 18:00:00 and 2012-07-03 06:00:00. The timetables considered thus have overlapping turn-rounds, however, for the sake of simplicity, we consider each timetable as a stand-alone optimization problem.

At CPH, scheduled arrival and departure times are always on the minutes 0, 5, 10,. . . ,55 after the hour. We therefore consider a time discretization of 5 minutes and a maximum capacity of 1 for all stands and gates. The hangar and maintenance areas at CPH are not divided into sets of stands. We therefore consider the hangar and maintenance areas as less critical resources. To incorporate this, the hangar and maintenance areas are modeled as a single resource with a maximum capacity of 15 aircraft in each time interval.

Since we are considering planning periods of at least 36 hours, one can argue that an exact planning of the push-back of each of the departures is not important; the exact time of the push-back is going to change anyway. The taxiways used for push-back are thus less critical compared to stands and gates and we will therefore use a time discretization of one minute and a maximum capacity of two for all taxiway resources.

As described in Section 9.5 the objective function is

$$\begin{aligned}\hat{z} = \quad & \omega_1 \sum_{r \in \mathcal{R}} \kappa_r + \\ & \omega_2 \sum_{p \in \mathcal{P}} \frac{\rho_p}{\rho^{max}} x_p + \\ & \omega_3 \sum_{p \in \mathcal{P}} \left(\frac{\theta_p^a + \theta_p^d}{2\theta^{max}} \right) x_p + \omega_4 \sum_{p \in \mathcal{P}} \left(\frac{\phi_p^a + \phi_p^d}{2\phi^{max}} \right) x_p - \\ & \omega_5 \sum_{p \in \mathcal{P}} \left(\frac{n_p^a + n_p^d}{N} \right) x_p \end{aligned} \tag{9.23}$$

where N is the total number of arriving and departing passengers. In the testing we have set $\omega_1 = 10,000$, $\omega_2 = 1,000$, $\omega_3 = 10$, $\omega_4 = 5$, and $\omega_5 = -1$.

9.6.1 Computational results

The tests have been run on a Linux server running SuSE Linux Enterprise 11 64 bit with 16 GB RAM and a single Intel Merom single core processor. We have used the Gurobi Optimizer version 5.5 as both the LP and IP solver. For each timetable, the turn-rounds have been allocated using the developed LP based heuristics described in Section 9.5. Moreover, the optimal solution of each instance has been found by enumerating all variables and solving SGAP.

As described the solution algorithm iteratively solves R-SGAP adding extra handling schedules in each iteration until a valid solution has been found or no more handling schedules can be added to $\mathcal{P}'$. The algorithm identifies the new handling schedules by a simple partial enumeration of the handling schedules in $\mathcal{P}_r \backslash \mathcal{P}'_r$ for each unallocated or conflicting turn-round $r \in \mathcal{R}$ and adds a subset of these to $\mathcal{P}'$. Determining the size of these subsets is essentially a trade-off between efficiency of the algorithm and the quality of the found solutions. To find an appropriate value, we have run the solution algorithm for different sizes of the subsets. For the sake of simplicity, however, we only report on tests that gave rise to the best performance. The results of the performance tests are shown in Table 9.4 and Table 9.5 and the quality of the solutions is described by Table 9.6 and 9.7.

As can be seen from Table 9.4 and Table 9.5 the developed solution algorithm performs very well for a tactical problem. For all timetables, the LP based heuristic was able to solve the problem in less than 30 seconds whereas the optimal solution was found in 424 seconds in average. For 13 of the 15 timetables, the final solution of the LP relaxed problem is fractional, however for each of these timetables, solving the MIP takes less than 2 seconds. Furthermore, Table 9.4 shows that the solution algorithm finds a valid solution in relatively few iterations (for 10 of the 15 timetables only four iterations are needed) and less than 8.7% of the generated handling schedules and less

Timetable	# handling plans - total	# handling plans - generated	# handling plans - added to R-SGAP	# iterations	LP integer	Obj. val. - LP solution (R-SGAP)	Obj. val. - MIP solution (SGAP)	Obj. val. - optimal solution	GAP (SGAP vs. optimal)
2012-07-02	750,389	183,064	12,686	5	No	51,094	51,095	50,852	0.5%
2012-07-03	754,561	146,584	10,449	4	Yes	46,842	46,842	46,402	0.9%
2012-07-04	751,333	130,710	10,916	4	No	47,568	47,568	47,147	0.9%
2012-07-05	739,391	130,133	11,015	4	No	48,151	48,154	47,118	2.2%
2012-07-06	745,485	153,510	11,001	4	No	48,031	48,100	46,525	3.4%
2012-07-07	665,614	113,187	8,337	4	No	39,272	39,273	38,813	1.2%
2012-07-08	714,037	143,931	9,550	4	No	42,398	42,399	42,166	0.6%
2012-07-18	460,543	84,918	7,369	6	No	33,585	33,586	33,109	1.4%
2012-10-08	789,840	174,791	12,219	4	No	51,444	51,445	50,938	1.0%
2012-10-09	758,090	162,535	11,119	4	Yes	47,113	47,113	46,848	0.6%
2012-10-10	738,857	142,021	11,281	4	No	48,862	48,863	48,618	0.5%
2012-10-11	780,355	163,773	12,675	5	No	50,188	50,236	49,776	0.9%
2012-10-12	759,780	167,080	12,407	5	No	50,342	50,343	49,769	1.2%
2012-10-13	547,847	125,747	7,037	4	No	33,646	33,648	33,570	0.2%
2012-10-14	602,394	156,032	9,555	5	No	36,850	36,850	36,681	0.5%

Table 9.4: *Solution algorithm log.*

Timetable	**Build master problem (s)**	**Inspect solution (s)**	**Create handling plans (s)**	**Solve LP (s)**	**Solve MIP (s)**	**Total (s)**	**Total - optimal solution (s)**
2012-07-02	2.1	0.3	10.3	3.0	1.8	25.6	444.2
2012-07-03	2.1	0.1	8.1	2.2	0.0	17.1	444.6
2012-07-04	2.1	0.1	7.2	2.4	1.5	18.4	432.3
2012-07-05	1.9	0.1	7.0	2.6	1.7	19.2	464.7
2012-07-06	2.1	0.1	9.2	2.6	1.8	23.8	831.3
2012-07-07	1.8	0.1	7.2	1.9	1.2	16.7	485.4
2012-07-08	2.1	0.1	8.9	2.5	1.5	23.9	465.4
2012-07-18	1.9	0.1	4.9	1.9	1.0	13.2	255.3
2012-10-08	2.0	0.3	9.2	1.9	1.6	21.5	404.8
2012-10-09	2.0	0.2	8.8	2.2	0.0	18.5	396.9
2012-10-10	2.1	0.3	7.3	1.9	1.5	18.4	418.8
2012-10-11	2.0	0.3	8.6	2.4	1.7	22.7	368.1
2012-10-12	2.0	0.3	9.4	2.5	1.6	23.1	404.6
2012-10-13	2.1	0.1	7.9	1.7	1.0	16.1	301.3
2012-10-14	2.2	0.3	8.9	1.9	1.2	23.7	262.9

Table 9.5: *Run times of the solution algorithm.*

than 1.6% of the total number of handling schedules need to be added to R-SGAP in order for the algorithm to be able to find a valid solution.

For all timetables, the process influencing the most on the final run time of the solution algorithm is the process of generating handling schedules. For most timetables more than 37% of the final run time of the solution algorithm is spend on generating handling schedules. An obvious improvement to the algorithm is thus to implement a more efficient way of generating, storing and iterating through the set of handling schedules.

With regards to the solution quality, the results indicate that the solution algorithm is capable of finding good near-optimal feasible solutions. In all cases, the objective value of the returned valid integer solution is in average 1% away from the optimal solution. Furthermore, for all timetables, the handling preference compliance is 97% or more, the stand preference compliance is 75% or more, the stand priority compliance is 74% or more, and the share of passengers at contact stand is 97% or more. The average real-life share of passengers at contact stand for the timetables is 96%.

Table 9.7 gives more detailed view on the stand cluster compliance for each of the stand clusters described in Table 9.1. As can be seen the stand cluster compliance is best for all the stand clusters involving relatively few operations. However, for the stand clusters, SK-HST-MORNING, SK-BIG6, SK-HST and DY-SCHENGEN, the stand cluster compliance is not as good. The explanation to this is most likely that the airlines SK and DY both have a home base in CPH. The focus of these airlines in CPH is thus on servicing their different arrivals and departures while optimizing the usage of their aircraft fleet and crew (flight deck and cabin). As a consequence, the turn-rounds of these airlines might contain status changes. There is therefore a risk that some of the operations qualified for a given stand cluster are combined with other operations in such a way that it is not possible to allocate the turn-round to the given cluster. As an example the timetables show that SK has made a turn-round combining an arrival from OSL (Oslo) with a departure to LHR (London Heathrow). In principle, the arrival from OSL is

Timetable	Handling preference compliance	Stand preference compliance	Share of pax at contact stand	# Towings	# Bussings	# Bussed pax
2012-07-02	98%	76%	98%	51	14	1469
2012-07-03	98%	78%	98%	35	12	1278
2012-07-04	99%	79%	99%	43	5	458
2012-07-05	97%	77%	98%	45	17	1868
2012-07-06	96%	76%	97%	59	28	3127
2012-07-07	98%	81%	99%	68	8	1045
2012-07-08	98%	83%	99%	61	10	1196
2012-07-18	97%	82%	98%	46	11	1078
2012-10-08	98%	75%	98%	22	18	2038
2012-10-09	99%	77%	99%	30	8	866
2012-10-10	98%	78%	99%	28	10	1054
2012-10-11	98%	76%	98%	49	13	1512
2012-10-12	97%	78%	98%	52	18	2036
2012-10-13	99%	82%	100%	51	4	288
2012-10-14	99%	77%	99%	37	4	400

Table 9.6: *Value of the KPIs described in Section 9.2.6.*

Timetable	Overall	SK-HST-MORNING	SK-BIG6	SK-DSA	DY-SCHENGEN	WF	TP	AFKL	FI	LX	LH
2012-07-02	76%	2/10	49/67	153/214	54/61	14/20	6/6	28/28	12/12	6/6	22/22
2012-07-03	78%	1/10	58/68	150/203	57/64	12/18	4/4	26/26	10/10	6/6	22/22
2012-07-04	79%	0/10	60/69	159/206	58/65	14/18	6/6	26/26	8/8	6/6	22/22
2012-07-05	77%	3/10	52/66	156/212	53/62	12/18	6/6	28/28	10/10	4/6	20/20
2012-07-06	76%	1/7	50/61	150/205	42/51	12/18	6/6	26/26	12/12	6/6	20/20
2012-07-07	81%	1/3	34/49	122/146	58/66	12/12	6/6	28/28	8/8	6/6	20/22
2012-07-08	83%	4/5	45/56	136/171	68/71	14/16	6/6	26/26	8/10	8/8	20/22
2012-07-18	82%	2/4	36/42	104/129	37/43	12/14	4/4	22/22	6/6	6/6	18/18
2012-10-08	75%	1/12	76/90	156/221	44/55	16/24	6/6	28/28	6/6	6/6	20/22
2012-10-09	77%	5/12	74/87	158/212	46/51	12/20	4/4	26/26	4/4	6/6	18/22
2012-10-10	78%	5/12	83/94	166/216	45/51	12/20	4/4	26/26	4/4	6/6	18/22
2012-10-11	76%	3/11	77/90	158/222	47/50	14/20	4/4	28/28	6/6	6/6	20/22
2012-10-12	78%	2/7	77/89	171/218	34/39	14/22	6/6	26/26	6/6	6/6	20/22
2012-10-13	82%	1/3	42/52	130/154	5/7	8/10	6/6	28/28	4/4	6/6	20/22
2012-10-14	77%	5/6	49/61	137/176	12/16	12/20	6/6	26/26	6/6	6/6	22/22

Table 9.7: *Stand cluster compliance.*

qualified for the SK-BIG6 stand cluster, however as none of the SK-BIG6 stands can handle the NonSchengen departure to LHR, this turn-round will most likely be allocated to another stand that allows the Schengen-NonSchengen status changes. The OSL arrival will thus not be allocated to the SK-BIG6 cluster. For all timetables such status-changing turn-rounds for SK alone amounts to 31% of SKs turn-rounds.

9.7 Conclusion

In this paper we have considered an important tactical problem in the airport industry, namely the stand allocation problem with associated resource scheduling. We have proposed a set packing-based formulation with a resource-based constraint system that implicitly resolves conflicts and developed a solution approach that exploits the flexibility of the model to be dynamically updated. The solution approach attempts to expand an invalid, restricted problem by identifying conflicting turn-rounds and intelligently include extra flexibility without creating an intractable problem. Extra flexibility is provided in the form of additional handling schedules. A key benefit of the model proposed is that it is relatively simple to include any kind of resource demand associated with the handling of a turn-round, and conflicts are resolved implicitly. Furthermore the model relatively easily incorporates the requirements and preferences of the different stakeholders of the problem; stand preferences for different airlines are expressed via stand clusters and handling preferences for different types of turn-rounds, aircraft types, airlines, etc. are expressed via handling options. Stand dependencies are handled implicitly via push-back options, yielding a much more accurate approach than normally seen for this type of constraint.

Numerical results on real life test instances arising in Copenhagen Airport show that the proposed methodology performs well; the solution algorithm can find near-optimal valid solutions respecting a predefined prioritization of objectives within reasonable time and the solutions are as good or better than current practice.

Bibliography

Babic, O., Teodorovic, D., Tosic, V., 1984. Aircraft stand assignment to minimize walking. Journal of Transportation Engineering 110 (1), 55–66.

Bergeron, M., Vanderstraten, G., 1988. Automatic assignment of aircraft to gates at a terminal. Computers and Industrial Engineering 14 (1), 15–25.

Bolat, A., 1999. Assigning arriving flights at an airport to the available gates. Journal of the Operational Research Society 50 (1), 23–34.

Bolat, A., 2000. Procedures for providing robust gate assignments for arriving aircrafts. European Journal of Operational Research 120 (1), 63–80.

Cheng, C., Ho, S., Kwan, C., 2012. The use of meta-heuristics for airport gate assignment. Expert Systems With Applications 39 (16), 12430–12437.

Cheng, Y., 1997. A knowledge-based airport gate assignment system integrated with mathematical programming. Computers and Industrial Engineering 32 (4), 837–852.

Cheng, Y., 1998. Solving push-out conflicts in apron taxiways of airports by a network-based simulation. Computers and Industrial Engineering 34 (2), 351–369.

Diepen, G., van den Akker, J. M., Hoogeveen, J. A., Smeltink, J. W., 2012. Finding a robust assignment of flights to gates at amsterdam airport schiphol. Journal of Scheduling 15 (6), 703–715.

Diepen, G., van den Akker, J. M., Hoogeven, J. A., Smeltink, J. W., 2007. Using column generation for gate planning at amsterdam airport schipol. Technical report, Department of Information and Computing Sciences, Utrecht University.

Ding, H., Lim, A., Rodrigues, B., Zhu, Y., 2004. Aircraft and gate scheduling optimization at airports. Proceedings of the annual Hawaii International Conference on Systems Sciences.

Ding, H., Lim, A., Rodrigues, B., Zhu, Y., 2005. The over-constrained airport gate assignment problem. Computers and Operations Research 32 (7), 1867–1880.

Ding, H., Lim, A., Zhu, Y., Rodrigues, B., 2003. The airport gate assignment problem. Proceedings - Annual Meeting of the Decision Sciences Institute, 1711–1716.

Dorndorf, U., 2002. Project Scheduling with Time Windows - From theory to applications. Springer.

Dorndorf, U., Drexl, A., Nikulin, Y., Pesch, E., 2007. Flight gate scheduling: State-of-the-art and recent developments. Omega 35 (3), 326–334.

Dorndorf, U., Jaehn, F., Pesch, E., 2008. Modelling robust flight-gate scheduling as a clique partitioning problem. Transportation Science 42 (3), 292–301.

Dorndorf, U., Jaehn, F., Pesch, E., 2012. Flight gate scheduling with respect to a reference schedule. Annals of Operations Research 194 (1), 177–187.

Drexl, A., Nikulin, Y., 2008. Multicriteria airport gate assignment and pareto simulated annealing. IIE Transactions 40 (4), 385–397.

Gosling, G., 1990. Design of an expert system for aircraft gate assignment. Transportation Research, Part A (General) 24A (1), 59–69.

Haagensen, K., 2012. Local regulations/appendix 1 - taxiway routing and guidelines for push-back at copenhagen airport.

Haghani, A., Chen, M., 1998. Optimizing gate assignments at airport terminals. Transportation Research, Part A (Policy and Practice) 32A (6), 437–454.

Jaehn, F., 2010. Solving the flight gate assignment problem using dynamic programming. Zeitschrift für Betriebswirtschaft 80 (10), 1027–1039.

Lim, A., Rodrigues, B., Zhu, Y., 2005. Airport gate scheduling with time windows. Artificial Intelligence Review 24 (1), 5–31.

Lusby, R., Larsen, J., Ryan, D., Ehrgott, M., 2011. Routing trains through railway junctions: A new set-packing approach. Transportation Science 45 (2), 228–245.

Mangoubi, R. S., Mathaisel, D. F. X., 1985. Optimizing gate assignments at airport terminals. Transportation Science 19 (2), 173–188.

Muthukrishnan, R., 1991. An expert system methodology for aircraft-gate assignment. Computers and Industrial Engineering 21 (1-4), 101–105.

Nikulin, Y., Drexl, A., 2010. Theoretical aspects of multicriteria flight gate scheduling: deterministic and fuzzy models. Journal of Scheduling 13 (3), 261–280.

Padberg, M. W., 1972. On the facial structure of set packing polyhedra. Mathematical Programming 5 (1), 199–215.

Rezanova, N. J., Ryan, D., 2010. The train driver recovery problem—a set partitioning based model and solution method. Computers and Operations Research 37 (5), 845–856.

Ryan, Foster, Wren, 1981. An integer programming approach to scheduling. In: Computer Scheduling of Public Transport. Urban Passenger Vehicle and Crew Scheduling. Proceedings of an International Workshop — 1981.

Wei, D., Liu, C., 2007. Optimizing gate assignment at airport based on genetic-tabu algorithm. Proceedings of the IEEE International Conference on Automation and Logistics August 18 - 21, 2007, Jinan, China, 1135–1140.

Xu, J., Bailey, G., 2001. The airport gate assignment problem: Mathematical model and a tabu search algorithm. Proceedings of the annual Hawaii International Conference on Systems Sciences.

Yan, S., Chang, C., 1998. A network model for gate assignment. Journal of Advanced Transportation 32 (2), 176–189.

Yan, S., Ho, C., 2001. Optimization of multiple objective gate assignments. Transportation Research, Part A (Policy and Practice) 35A (5), 413–432.

Yan, S., Shieh, C., Chen, M., 2002. A simulation framework for evaluating airport gate assignments. Transportation Research Part A: Policy and Practice 36 (10), 885–898.

Chapter 10

Operational Stand and Gate Allocation at Copenhagen Airport

Tor Fog Justesen* Simon Thoustrup † Jesper Larsen ‡

*Airport Optimization, Copenhagen Airports A/S
Lufthavnsboulevarden 6, 2770 Kastrup

†Traffic & Airside Operations, Copenhagen Airports A/S
Lufthavnsboulevarden 6, 2770 Kastrup

‡Department of Management Engineering, Technical University of Denmark
Produktionstorvet, Building 426, DK-2800 Kgs. Lyngby, Denmark

Contact: Tor Fog Justesen (+45 23 24 62 47 / tor.justesen@cph.dk)

Abstract In this paper we consider the stand and gate allocation problem at the operational level. At the tactical level, the stand and gate allocation for a given day is planned and published to the involved stakeholders (airlines, handling companies, airport staff) either the day or a couple of days before. Throughout the day of operation, multiple changes to the timetable are received in the form of flight delays, aircraft swaps, cancellations, etc. Each change to the timetable is processed by updating the timetable and the current stand and gate allocation. If the timetable change does not impose any conflicts where two or more turn-rounds are claiming the same resource(s), the timetable change is said to be *disturbing*. If, on the other hand, the timetable change induces conflicts to the stand and gate allocation, the timetable change is said to be *disrupting*. Each operation (arrival or departure) may receive multiple timetable changes, both disturbing and disrupting. In the case of a disrupting timetable change, one or more turn-rounds must be re-allocated to restore a feasible stand and gate allocation. We present two different strategies for this conflict handling and show how the turn-round re-allocation problem can be solved using a model and solution method developed for the tactical stand and gate allocation problem. To test the conflict handling strategies and the developed solution method, we simulate a typical busy day of operation at Copenhagen Airport (CPH) considering flight delays and aircraft swaps as these are predominant types of timetable changes experienced at CPH. The numerical results show that the proposed methodology performs well as it can find near-optimal solutions for all conflicts in reasonable time while maintaining a high solution quality for both of the considered conflict handling strategies.

Keywords: *aircraft turn-round*, *operational stand and gate allocation*, *set packing*, *optimization*

10.1 Introduction

Operating an airport gives rise to many different resource allocation and scheduling problems that can be modeled and solved using operations research techniques. In this paper we address the handling of aircraft turn-rounds at the operational level. An aircraft turn-round refers to the set of processes taking place while an aircraft is on ground at an airport and the length of a turn-round is defined as the time from when the aircraft parks at its arrival stand until it departs from its departure stand. When handling a turn-round, the processes involved (arrival, disembarkation of passengers, cleaning, etc.) require different ground handling resources at various times, including taxiways, aircraft stands, gates, etc. Each resource can be claimed by at most one turn-round at a time. The *stand and gate allocation problem* is the problem of allocating turn-rounds to available ground handling resources without imposing any conflicts. A conflict occurs if two or more turn-rounds simultaneously claim the same resource.

For many airports, the conventional setup is as follows: For a given day of operation, a tactical stand and gate allocation is made either the day or a couple of days before and published to the relevant stakeholders (airlines, handling companies, airport staff), and then on the day of operation, the job of the stand and gate dispatchers is to maintain this plan. In the tactical stand and gate allocations, the turn-rounds are typically separated by a small buffer. The size of the buffer can be either fixed, solely based on e.g. the size of the aircraft, or variable, based on historical knowledge about a the actual arrival or departure times. The purpose of the buffers is to make the stand and gate allocation tolerant of the many small perturbations that occur in a real-world timetable. The buffers primarily address changes to the arrival and departure times of the turn-rounds, however, in a real-world setting, many other types of changes are received, including aircraft swaps, cancellations, new flights, equipment failures, etc.

A *timetable alteration* (TA) is processed by updating the timetable and the current stand and gate allocation. Each operation may receive multiple TAs during the day of operation. If a TA causes the current stand and gate allocation to become infeasible, i.e. two or more turn-rounds are claiming the same resource(s), one or more turn-rounds needs to be re-allocated. The process of re-allocating one or more turn-rounds is referred to as *conflict handling*. There may be many different strategies for resolving conflicts. In this paper we consider two strategies: the instant conflict handling strategy, where the conflict handling is triggered each time a conflict is detected, and the lazy conflict handling strategy, where the conflict handling is only triggered when a specified set of conditions are fulfilled.

At the core of every conflict handling, the *turn-round re-allocation problem* is solved. The turn-round re-allocation problem is essentially equivalent to the stand and gate allocation problem solved at the tactical level, when the tactical stand and gate allocations are generated. The difference is, that when solving the problem, some resources are blocked by on-going processes or aircraft already on ground, and for some turn-rounds deviations from the current allocation should be avoided. [Justesen et al., 2014] provide a thorough introduction to the tactical stand and gate allocation problem and describe how the problem can be formulated as a set packing-based model with a resource-based constraint system. The model easily includes any type of ground handling resource involved in the handling of a turn-round and conflicts are re-solved implicitly. The model is solved with an LP based heuristic using a partial enumeration of the variables. Given the promising results presented by [Justesen et al., 2014], the aim of this paper is to demonstrate, that the suggested approach can also be used to solve the turn-round re-allocation problems occurring during the day of operation. By using the the solution approach on both the tactical and operational level, CPH will be able to obtain a consensus between decisions made at the tactical level and decisions made on the day of operation.

Obviously, the two strategies yield turn-round re-allocation problems with different characteristics and of various sizes. To investigate the potential of the suggested solution approach we simulate a typical busy day of operation at Copenhagen Airport (CPH) applying both the instant conflict handling strategy and the lazy conflict handling strategy. Copenhagen Airport is Scandinavia's main airport hub and in 2012 CPH handled 242,992 operations (arrivals and departures) and serviced 23,336,187 million passengers (see [CPH, 2012a]). Close to 90% of all turn-rounds departed on time, making CPH the most punctual airport in Europe in 2012 (see [CPH, 2012b]). For the sake of simplicity, we only consider flight delays and aircraft swaps as these are the predominant TAs experienced at CPH.

The remainder of this paper is organized as follows. Section 10.2 provides a review of previous work and Section 10.3 describes how updates to the timetables are processed and presents the two conflict handling strategies in more detail. Section 10.5 provides a detailed description of the turn-round re-allocation problem and shows how it can be modeled and solved using a set packing-based model. The computational results of the experiments are given in Section 10.6 and finally, we draw some conclusions in Section 10.7.

10.2 Previous work

Most of the previous work on the operational stand and gate allocation problem is focused on making the tactical stand and gate allocation plans less sensitive to minor changes to the arrival and departures times of the flights in the timetable, i.e. delays. [Dorndorf et al., 2007a] provide a survey about robustness in stand and gate allocation and describe how fluctuations in arrival and departure times can be address either via robust optimization (see e.g. [Yan and Chang, 1998], [Bolat, 1999], [Bolat, 2000], [Yan and Ho, 2001], [Wei and Liu, 2007], [Wei and Liu, 2009], and [Diepen et al., 2012]) or stochastic programming (see e.g. [Lim et al., 2005], [Yan and Tang, 2007], [Şeker and Noyan, 2012], and [Genç et al., 2012]).

Obviously, having robust stand and gate allocations is a prerequisite for having a stable and reliable operation of the airport. However, no matter how robust a stand and gate allocation is made, some updates to the timetable may still impose conflicts, necessitating a re-allocation of one or more turn-rounds. Having an efficient turn-round re-allocation strategy is thus just as important to an airport as being able to provide robust stand and gate allocation plans, but nevertheless, only little attention has been given to this problem in the literature.

[Tang et al., 2010] develop a turn-round re-allocation framework and a systematic computerized tool for repeatedly handling stand re-allocations given varied flight delay information. They consider the problem as consisting of two phases; in phase one, the preprocessing phase, the flight departure and arrival times are updated and the number of available stands, the available time window for each stand, the number of disrupted flights, and the flight schedules are preprocessed. In phase two, the re-allocation construction phase, the re-allocation of flights to stands is performed. They test their algorithm on real data instances provided by Taiwan Taoyan International Airport and compare the results of the developed algorithm with the results of handling all disruptions manually. The results show that the method is not time-consuming and can be applied in real-time operations. The objective of their algorithm is to minimize the amount that the re-allocation of the flights to stands deviates from the initial plan. To quantify this deviation in a simple manner, they introduce two types of inconsistencies; space inconsistency, which occurs when a flight is re-allocated to a gate different than the one it was allocated to in its initial plan, and time inconsistency, which occurs when a flight's starting time is altered.

[Yan et al., 2011] divide a given set of flights into deterministic flights and stochastic flights; for the deterministic flights, departure and arrival times are considered certain, whereas for stochastic flights, the departure and arrival times are considered stochastic. They then develop a 0-1 integer programming stand re-allocation model designed to consider both deterministic and stochastic flights. Like [Tang et al., 2010] they define space and time inconsistencies as a measure of the quality of stand re-allocations, and the objective is to minimize these inconsistencies, thereby minimizing the deviation of re-allocations from the original planned allocation. To meet the demands of a practical real-time tool, the developed stand re-allocation model is applied to a dynamic framework for repeated handling of stand re-allocations. Numerical tests also utilizing data from Taiwan Taoyan International Airport operations were performed. All findings demonstrate the good theoretical and practical performance of the developed framework.

One of the first attempts to describe a more general framework with focus on both robustness and turn-round re-allocation strategies is given by [Dorndorf et al., 2007b]. They describe how changes to the timetable managed in airline scheduling (aircraft and crew), indirectly impact stand and gate allocations at airports. They define disruption management as the efforts both to avoid and recover from disruptions that require changes to the initial stand and gate allocation plan. Disruption management thus comprises both robustness strategies and recovery strategies. As for the objectives of disruption management they consider both the maximization of real profits and the desire to return to the original plan as soon as possible; two objectives that in some cases can be contradictory. The survey for recovery strategies is focused on the aircraft recovery problem since the ways to solve this problem (delaying, swapping or canceling flights) can be directly applied to the operational stand and gate allocation problem as well as the analysis for decision costs.

10.3 Timetable alterations

Timetable alterations are typically received for a single operation at a time and consists of a timestamp, an operation id, the field updated and the new value of the updated field. There are many reasons to why changes to the timetable occur. In some cases, they are imposed deliberately by the airlines (holding back a departure in order to wait for some crew, cancellations due to technical problems, etc.), but in most cases they are caused by one or more of the processes involved in the turn-round (prolonged boarding process, late fueling, etc.). Changes to the timetable can occur either at the airport or at one of the airports the aircraft is visiting. In any case, the airport has little influence, and must simply accede to whatever change that occurs.

Alterations to the ETA/ETD are often referred to as *flight delays*. In principle, there are four types of delay: early arrival, late arrival, early departure, and late departure. In all cases, a delay will cause the turn-round of the delayed operation to claim its allocated resources at times different from the times in the current stand and gate allocation plan. If the size of the delay is small enough it may be absorbed by either the idle time of the turn-round or the idle time between consecutive flights on the stand. However, if the delay cannot be absorbed and leads to a conflict, re-allocations of the affected turn-rounds are needed.

An *aircraft swap* refers to the event that an airline decides to change the *tail assignments* of a given set of flights. The tail assignment specifies exactly which aircraft is servicing a particular flight operation. A turn-round is the linking of an arrival operation, a departure operation, and an aircraft. Changing the tail assignments for a set of flights will therefore most likely lead to the creation of a set of new turn-rounds that may be different from the turn-rounds initially serviced by the swapped aircraft in both length, inbound/outbound status, and aircraft type. Aircraft swaps occur if, for instance, an aircraft gets a technical problem and needs to be grounded for a period. To avoid cancellations and delays, the flights originally assigned to the aircraft thus need to be re-assigned to other aircraft. Other reasons for aircraft swaps could be lack of crew, cancellations of flights due to weather, late incoming aircraft, etc. As described in [Dorndorf et al., 2007b] aircraft swaps and cancellations are imposed by the airlines when they are solving the *aircraft recovery problem*. Since a TA only specifies a tail assignment change for one operation at a time, aircraft swaps usually result in multiple TAs so that all turn-rounds once again links an arrival to a departure. See Figure 10.1 for an illustration of an aircraft swap.

In general, there are two strategies for how to handle conflicts during the day of operation.

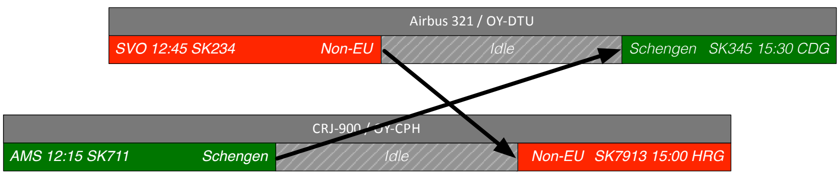

Figure 10.1: *Illustration of a pair of aircraft swaps.*

The idea of the *instant conflict handling* strategy (ICH) is to remove every conflict as soon as it is detected. The advantage of this strategy is that only one conflict is handled at a time and the stand and gate allocation plan is constantly up-to-date and conflict free. However, sometimes, multiple TAs are received for a single operation, and some conflicts may resolve themselves over time without the need to re-allocate any turn-rounds. A disadvantage of the ICH approach is therefore that some turn-rounds may be unnecessarily re-allocated, perhaps multiple times. As an example, multiple TAs are typically received for the arrival time of a turn-round; while the aircraft is airborne, the ETA is based on the calculated flight path, however, this flight path does not include e.g. congestion and holdings in the airspace around the airport, and the ETA is often adjusted multiple times within the last minutes before the aircraft lands. Equivalently, if an aircraft is suddenly taken out of of operation due to e.g. technical problems, many tail assignment changes may be received within a very short period, while the airline is re-assigning the flights of the defective aircraft to other aircraft.

An alternative to the ICH strategy is the *lazy conflict handling* strategy (LCH), where conflict handling is only performed when certain conditions are fulfilled, except, off course, if the conflicts necessitates immediate resolution. The conditions could e.g. be that a fixed time interval has passed since the last conflict handling, or that a conflict occurring within a specified time interval, e.g. the next 60 minutes, has been detected. In this way, one is more certain that the conflicts handled are indeed true conflicts. In principle, any type of conditions could be chosen, and the conditions may vary during the day of operation. For instance, conflict handling could be done every minute during the peak periods, whereas during the off-peak periods, conflicts are only handled when they become critical to the operation. See Figure 10.2 for an illustration of the two conflict handling strategies. The current practice at CPH is somewhat of a mix of the two strategies and is highly depending on the experience of the dispatcher at work. The current operational stand and gate allocation system is predominantly manual in the sense that all conflicts are handled manually, and the dispatchers can apply whatever conflict handling strategy he or she prefers. Some dispatchers prefer the plan to be constantly updated and apply a strategy practically resembling the instant conflict handling strategy, whereas other dispatchers apply a lazy conflict handling strategy and always resolve conflicts at the very last minute. As will be shown in Section 10.6, the different strategies yield different results, and the current approach therefore lacks the control of the consistency in how well the overall business objectives of the stand and gate allocation are fulfilled.

10.4 Conflict interval

The time where a conflict handling is performed is referred to as the *conflict handling time* (CHT). At the conflict handling time, some aircraft might already be on ground and in the process of being handled. Obviously, these turn-rounds cannot be re-allocated, and we refer to these turn-rounds as being *locked*. Turn-rounds on ground that are idle at the time of the conflict handling might also be re-allocated if the length of the turn-round allows it, however, for the sake of simplicity, we have omitted this option, and therefore also consider these turn-rounds to be locked.

Turn-rounds arriving or departing within the next 45 minutes after the conflict handling time, can, in principle, be re-allocated, however, it should be avoided, since gate announcements or other handling-related preparations have typically already been initiated. In the following we will say that a turn-round is *critical* if it arrives within the next 45 minutes after the conflict handling time, i.e. in the interval [CHT,CHT+45min]. Turn-rounds arriving later than 45 minutes after the conflict handling time are said to be *open*. The choice of 45 minutes reflects the current operational practice at Copenhagen Airport, however, one could easily imagine a longer critical turn-round interval of up to e.g. 90 minutes after the the conflict handling time. Essentially, the longer interval, the less impact on the current stand and gate allocation, but also, the less flexibility in the re-allocation as more turn-rounds will be critical.

The current practice at CPH is to resolve conflicts manually by re-allocating one turn-round at a time by simply looking for alternative allocation options in the current stand and gate allocation.

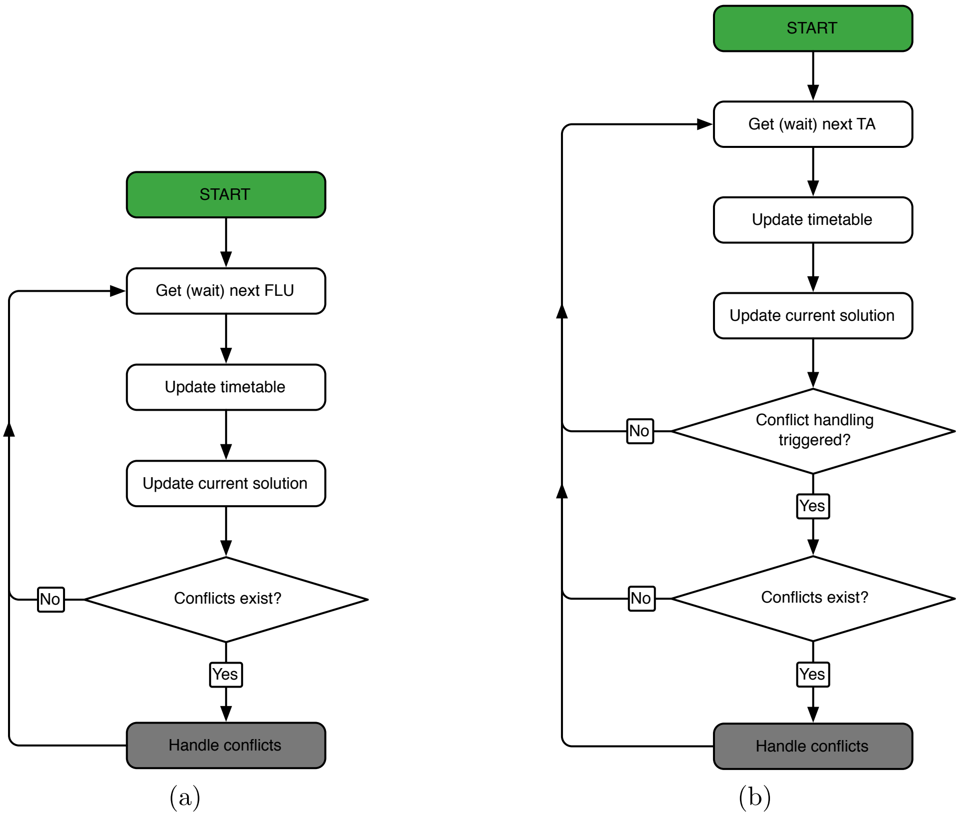

Figure 10.2: *Illustration of (a) instant conflict handling and (b) lazy conflict handling.*

The advantage of this approach is that it is very simple and intuitive to understand, however, in some cases it may lead to a cascade of re-allocations, and finding a good solution to the conflict may be very difficult. Obviously, a better approach to the turn-round re-allocation problem would be to include all open and critical turn-rounds at every conflict handling, however, this approach is also unattractive, as the number of open turn-rounds may be very large. Moreover, since the current stand and gate allocation has already been published to the airlines and handling companies, as few turn-rounds as possible should be re-allocated.

An alternative approach is to solve the turn-round re-allocation problem for a limited *conflict interval* of the planning period. To simplify, we define the conflict interval to start at the earliest ETA of the disrupted turn-rounds, unless this time is before the conflict handling time, in which case, the interval starts at the conflict handling time. The interval ends at the latest ETD of the disrupted turn-rounds. Only turn-rounds arriving within the conflict interval can be re-allocated, i.e. turn-rounds "on ground" at the start of the conflict interval cannot be re-allocated and are considered locked. We refer to the time between the conflict handling time and the beginning of the conflict interval as the *lead time* of the conflict handling. If the lead time is more than 45 minutes, all turn-rounds in the conflict interval can be re-allocated without a penalty. If the lead time is less than 45 minutes, some of the turn-rounds may be critical, in which case, any deviation from the current solution should be penalized. From an practical perspective, the allocation of a turn-round is said to deviate from the current solution, if either the arrival stand or the departure stand of the turn-round has been changed in the new solution. See Figure 10.3 for an illustration of a conflict interval and corresponding locked, critical, and open turn-rounds. Resources claimed by locked turn-rounds are said to be *blocked* and cannot be claimed by other turn-rounds.

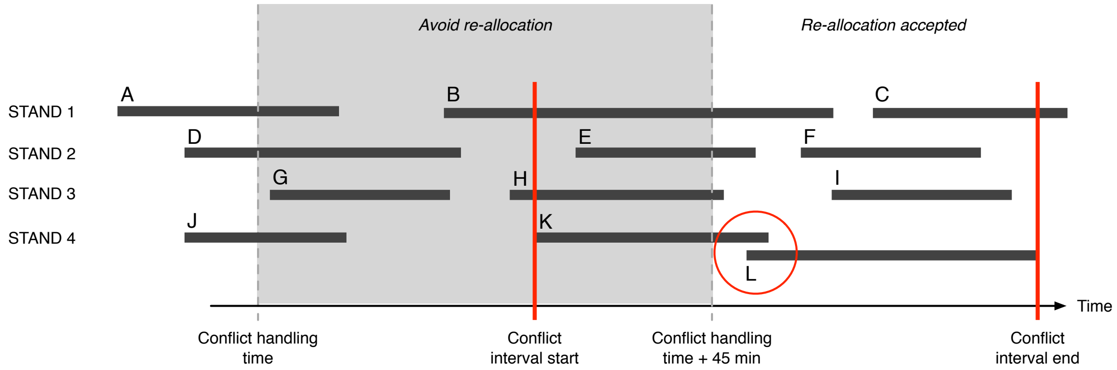

Figure 10.3: *Illustration of a conflict interval. The conflict is between turn-round J and K on stand 4. Turn-rounds A, B, F, G, and I are all locked, i.e. these turn-rounds cannot be re-allocated. Turn-rounds C, D, E, H, J, K can all be re-allocated, however, the turn-rounds D and J are critical turn-rounds, i.e. any deviations from the current solution (Stand 2 respectively Stand 4) must be penalized for these turn-rounds.*

10.5 Re-allocating turn-rounds

In the following we summarize the main principles of the approach for tactical stand and gate allocation described in [Justesen et al., 2014] and describe how the model and solution method should be adopted to be applicable to the turn-round re-allocation problem. The main idea of the approach is to describe the possible *handling options* for each turn-round by a corresponding set of *handling schedules*.

A handling option describes if the aircraft should be handled on a contact stand or a remote stand, if the passengers should be pier-serviced or bussed, if the aircraft should be towed, etc. The set of possible handling options and their priorities is given according to some business logic specified by the airlines and/or airport. A *handling schedule* is the operationalization of a given handling option and describes exactly which ground handling resources are needed and at what times. Since a turn-round can be handled in many different ways and at different stands, many possible handling schedules may exist for each turn-round. The *stand and gate allocation problem* is the problem of finding a set of non-conflicting handling schedules consisting of exactly one handling schedule for each turn-round. A set of handling schedules is said to be conflicting, if two or more turn-rounds simultaneously claim the same resource(s).

To model that each resource can be claimed by at most one turn-round at a time, a constraint is identified for each resource in a sequence of time intervals. The time interval is obtained through a discretization of the considered planning period. Any handling schedule can then be represented as a column of the constraint matrix with binary components; a one in a particular row indicates that the turn-round is claiming the corresponding resource-time interval, whereas a zero indicates otherwise. Given the nature of the constraint system, it is obvious that the model implicitly obtains a conflict free set of handling schedules for the turn-rounds. Furthermore, one can also dynamically consider additional handling schedules for turn-rounds without any fundamental change to the model itself. Additional handling schedules can easily be included as extra columns in the model.

For each stand, multiple push-back options may exist—see Figure 10.4 for an example of the possible push-back options for stand B10 at CPH. To model that an aircraft only claims a small part of a taxiway at a time, and that a taxiway can be used by many different aircraft at a time, the taxiway network is discretized into a set of non-overlapping *taxiway segments*. Each taxiway segment can be perceived as a ground handling resource that can be claimed by the turn-rounds. A *push-back schedule* is the operationalization of a given push-back option, accurately describing which taxiway segments are needed and at what time, in order to complete the push-back.

Two resources are said to be *dependent* if a turn-round claiming one of them, also automatically

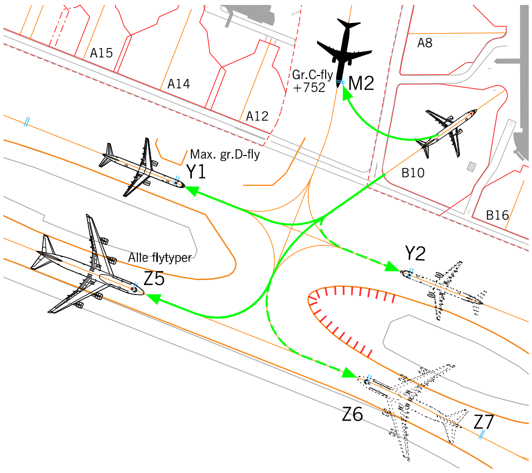

Figure 10.4: *Possible push-back options for stand B10 at Copenhagen Airport. M2, Y1, Y2, Z5, and Z6 denote the different start-up positions at the taxiways M, Y, and Z respectively that should preferably be used when starting up aircraft departing from B10. Large aircraft must always be pushed to either start-up position Z5 or Z6, whereas small/medium sized aircraft can be pushed any of the start-up positions. The choice of start-up position is dependent on wind direction, other traffic, and various technical conditions.*

claims the other. For taxiway segments, dependencies occur if two segment are located so close that aircraft cannot pass each due to the wingspan. For stands, the dependencies are either given explicitly in the form overlapping stands (Multiple Aircraft Ramp System stands), or implicitly in the form of restrictions on the simultaneous handling of turn-rounds on the stands. By imposing the restriction that no taxiway segment can be claimed by more than one turn-round at a time, and by including the push-back schedules in the handling schedules, stand dependencies are implicitly given by the resource-time interval constraints. This approach is a much more accurate approach than normally seen for stand dependencies (cf. [Dorndorf et al., 2007a], [Dorndorf et al., 2008], [Jaehn, 2010], and [Diepen et al., 2012]) as the very dynamic nature of a push-back is encapsulated indirectly.

For various commercial and operational reasons, an airline may prefer that their arrivals and departures are handled at certain stands. Equivalently, it may be preferred that some aircraft types or turn-rounds arriving from/departing to specific locations are handled at certain stands. A *stand cluster* is a prioritized set of stands describing such preferences. The airline, type of aircraft, origin/destination, time of day, etc. determine which stand cluster a given operation is qualified for. For airlines only visiting the airport infrequently the choice of stand is less important, however, as a general rule, these airlines prefer to be parked as close to the terminal buildings as possible. To accommodate this request, a default stand cluster is defined. The default cluster contains all stands of the airport, and the stands are prioritized according to their distance to the terminal buildings. Airlines that have not expressed specific stand preferences are automatically qualified for the default cluster.

An operation can only be qualified for one stand cluster and in the following we refer to the stand clusters of the arrival and departure operations of a turn-round as the *arrival stand cluster* respectively *departure stand cluster* of the turn-round. In the US, stands and gates are to a large extent dedicated to the individual airlines, i.e. the different stand clusters are in principle enclosed areas, where **only** operations of the particular operation are allowed to be handled. In most European airports, however, the stand clusters mainly express *preferences*, i.e. each operation can in principle be allocated to any stand of the airport, regardless of which stand cluster the particular stand belongs to, yet the airports always strive to fulfill the stand preferences to the greatest possible extent. If a stand belongs to more than one cluster, the clusters involving the stand are ranked in accordance with the internal commercial and strategic prioritization of the airlines operating in the airport. All other clusters and the default stand cluster are given the lowest possible ranking.

The origin and destination airport of the turn-round determine the *inbound* respectively *outbound status* of the aircraft. The status of the passengers is identical to the status of the aircraft and determines which set of rules for immigration and security that must be applied when handling transferring and local arriving respectively local departing passengers. In Europe, the status of an aircraft can be either *domestic*, *Schengen*, *Non-Schengen* (EU, but not part of the Schengen Agreement), or *Non-EU* (outside EU and not part of the Schengen Agreement). With a few exceptions, the status is determined solely by the country of the airport. A turn-round is said to have a *status change* if the inbound status is different from the outbound status.

A *gate* is the passageway through which passengers proceed in order to get from the terminal buildings to the aircraft (and vice versa). Passengers can either walk (e.g. via a jet-bridge) or be transported by a bus. Depending on its facilities (lounge, security check point, etc.) and placement within the airport terminal buildings (before or after passport control, possibility to separate passengers from other passengers, etc.), a gate can allow the handling of one or more statuses. An operation is said to be *pier serviced* if the aircraft is parked at a contact stand and it is possible for the passengers to walk from the aircraft to a gate that allows the handling of the status of the operation. If the passengers need to be transported by bus in order to get to a suitable gate, the operation is said to be *bussed*.

10.5.1 Mathematical model

The model described in [Justesen et al., 2014] does not distinguish between critical and non-critical turn-rounds, however, as described, this is required when considering the turn-round re-allocation problem. Now, let $\mathcal{R}$ denote the set of turn-rounds, $\mathcal{P}$ the set of handling schedules, and $\mathcal{P}_r \subseteq \mathcal{P}$ the set of handling schedules for each turn-round $r \in \mathcal{R}$. The set of ground handling resources is denoted $\mathcal{R}$ and the set of resource-time interval constraints is denoted $\mathcal{K}$. The binary parameter a_{kp} is one if handling schedule $p \in \mathcal{P}$ is claiming resource-time interval constraint $k \in \mathcal{K}$ and zero otherwise. Finally, the set of resource-time intervals that are blocked as a consequence of locked turn-rounds is denoted $\tilde{\mathcal{K}} \subseteq \mathcal{K}$.

Two sets of decision variables are defined: the binary allocation variables x_p where $x_p = 1$ if handling schedule $p \in \mathcal{P}$ is included in the final solution and $x_p = 0$ otherwise and the binary coverage variables π_r, where $\pi_r = 1$ if turn-round $r \in \mathcal{R}$ is unallocated and $\pi_r = 0$ otherwise.

As described in [Justesen et al., 2014], the normalized objective function for the tactical stand and gate allocation is given as follows:

$$\begin{aligned}
\hat{z} = \quad & \omega_1 \sum_{r\in\mathcal{R}} \pi_r + \\
& \omega_2 \sum_{p\in\mathcal{P}} \frac{\rho_p}{\rho^{max}} x_p + \\
& \omega_3 \sum_{p\in\mathcal{P}} \left(\frac{\theta_p^a + \theta_p^d}{2\theta^{max}} \right) x_p + \omega_4 \sum_{p\in\mathcal{P}} \left(\frac{\phi_p^a + \phi_p^d}{2\phi^{max}} \right) x_p - \\
& \omega_5 \sum_{p\in\mathcal{P}} \left(\frac{n_p^a + n_p^d}{N} \right) x_p
\end{aligned} \tag{10.1}$$

where for a given handling schedule $p \in \mathcal{P}$

- $\rho_p \in 1, \ldots, \rho^{max}$ denotes the priority of the corresponding handling option (1 is the highest and $\rho^{\max}$ the lowest priority)
- $\phi_p^a \in \{1, \ldots, \phi^{\max}\}$ denotes the priority of the arrival stand (1 is the highest priority and $\phi^{\max}$ is the lowest)

- $\phi_p^d \in \{1, \ldots, \phi^{\max}\}$ denotes the priority of the departure stand (1 is the highest priority and $\phi^{\max}$ is the lowest)
- $\theta_p^a \in \{1, \ldots, \theta^{\max}\}$ denotes the ranking of the arrival stand cluster at the arrival stand (1 is the highest priority and $\theta^{\max}$ is the lowest)
- $\theta_p^d \in \{1, \ldots, \theta^{\max}\}$ denotes the ranking of the departure stand cluster at the departure stand (1 is the highest priority and $\theta^{\max}$ is the lowest)
- n_p^a denotes the number of arriving passengers that are pier-serviced
- n_p^d denotes the number of departing passengers that are pier-serviced
- N is the total number of passengers

The allocation of a turn-round is said to deviate from the current solution if either the arrival stand or the departure stand of the turn-round has been changed in the new solution. In other words, for a critical turn-round, handling schedules for which the arrival and departure stands are the same as in the current stand and gate allocation of the turn-round, are strongly preferred. Now, define the binary parameter δ_p, where $\delta_p = 0$ if handling schedule $p \in \mathcal{P}$ is for a critical turn-round and the arrival and departure stands are equal to the current stand and gate allocation, and $\delta_p = 1$ otherwise. We now modify the objective function for the tactical stand and gate allocation as follows

$$\begin{aligned}\hat{z} = \quad & \omega_1 \sum_{r \in \mathcal{R}} \pi_r + \\ & \omega_2 \sum_{p \in \mathcal{P}} \delta_p \frac{\rho_p}{\rho^{max}} x_p + \\ & \omega_3 \sum_{p \in \mathcal{P}} \delta_p \left(\frac{\theta_p^a + \theta_p^d}{2\theta^{max}} \right) x_p + \omega_4 \sum_{p \in \mathcal{P}} \delta_p \left(\frac{\phi_p^a + \phi_p^d}{2\phi^{max}} \right) x_p - \\ & \omega_5 \sum_{p \in \mathcal{P}} \delta_p \left(\frac{n_p^a + n_p^d}{N} \right) x_p \end{aligned} \tag{10.2}$$

Thereby, all non-deviating handling schedules for critical turn-rounds are given the cost of zero, strongly favoring these handling schedules. The turn-round re-allocation problem can then be formulated as the following integer programme (OSGAP):

$$\textbf{minimize} \quad \hat{z} \tag{10.3}$$

$$\textbf{subject to} \quad \sum_{p \in \mathcal{P}} x_p + \pi_r = 1 \qquad \forall\, r \in \mathcal{R} \tag{10.4}$$

$$\sum_{p \in \mathcal{P}} a_{kp} x_p \leq 1 \qquad \forall\, k \in \mathcal{K} \setminus \tilde{\mathcal{K}} \tag{10.5}$$

$$\sum_{p \in \mathcal{P}} a_{kp} x_p \leq 0 \qquad \forall\, k \in \tilde{\mathcal{K}} \tag{10.6}$$

$$x_p \in \{0, 1\} \qquad \forall\, p \in \mathcal{P} \tag{10.7}$$

$$\pi_r \in \{0, 1\} \qquad \forall\, r \in \mathcal{R} \tag{10.8}$$

where the objective function (10.3) minimizes the total cost of handling all aircraft turn-rounds. Constraints (10.4) (i.e. the GUB constraints) ensure that all aircraft turn-rounds are assigned a handling schedule, whereas Constraints (10.5) and Constraints (10.6) enforce the restriction that at most one turn-round can claim any non-blocked ground handling resource at any time interval and no turn-round can claim a blocked resource-time interval. Finally, Constraints (10.7) and (10.8) define the binary restrictions on the decision variables.

10.5.2 Solution algorithm

Figure 10.5 shows the frequencies with which TAs are received for arrivals and departures scheduled to take place on July 2, 2012, which represents a typical busy summer day at CPH. Not all TAs lead to conflicts, however, given the very high frequencies of the TAs, it is clear that when solving OSGAP, efficiency and feasibility is more important than being able to find optimal solutions, as the problem needs to be solved quite often, and a solution potentially only stays valid for a short period anyway.

To solve the tactical stand and gate allocation problem, [Justesen et al., 2014] develop an LP-based heuristic. The results presented by [Justesen et al., 2014] show that the heuristic generates **high-quality** feasible solutions within reasonable time for tactical planning. The idea is now to apply the same heuristic when solving the OSGAP.

The heuristic iteratively solves a relaxed, restricted version of the problem, R-OSGAP, adding extra variables at each iteration if needed. The additional variables are identified by a cost-based partial enumeration of the possible variables for each turn-round and the heuristic stops when the first feasible solution is encountered. Initially only the "best" handling schedules for each turn-round are considered, and for each iteration, the handling schedules added gets less and less attractive. In a worst case scenario, the solution method will have to add new handling plans $|\mathcal{H}_r|$ times for a given turn-round $r \in \mathcal{R}$ and eventually all possible handling schedules for r might be added to the problem, i.e. $\mathcal{P}'_r = \mathcal{P}_r$. However, as will be presented in Section 10.6 this case has not yet been observed during the computational experiments.

To limit the number of handling schedules added, only unallocated and *conflicting* turn-rounds are considered at each iteration. A turn-round $r \in \mathcal{R}$ is termed unallocated if $\pi_r > 0$ in the current solution. A turn-round is termed conflicting if it claims one or more resource-time intervals that could otherwise have been claimed by one of the currently unallocated turn-rounds. The set of conflicting turn-rounds is identified by analyzing the set of binding resource-time interval constraints, i.e. resource-time interval constraints for which the corresponding dual variable has a positive value.

First, all handling schedules are generated for the first priority handling option and these are sorted according to their costs and added to the model when needed. When no more handling schedules exist for the first priority handling options, handling schedules are generated and added for the second priority handling option. And so forth. Although the number of handling schedules may theoretically be very large for handling options involving towings, the real-world restrictions (like e.g. aircraft type compatibility on the individual stands and status handling) severely limit the number of possible stand combinations. As will be shown in the results presented in Section 10.6, the described variable generation approach enables the solution method to efficiently find operationally applicable, near-optimal solutions.

In an operational setting, where things are likely change anyway, being able to find a near-optimal integer solution efficiently is often preferred to finding an optimal integer solution. As a consequence, the developed solution method handles fractional solutions, simply by solving OSGAP using the handling schedules in $\mathcal{P}'$, i.e. when a valid but fractional solution is found, R-OSGAP is "unrelaxed" and solved as an integer problem using a standard IP solver.

The method terminates the first time a valid integer solution is encountered. We thus have no guarantee that the found solution is optimal. However, since we are only adding the "best" handling schedules for each turn-round in each iteration, it is likely that the objective value of the found solution is not that far from the objective value of the optimal solution (see Section 10.6).

The main ideas of the solution approach are summarized in Algorithm 3. In the following, we refer to one iteration of the loop in line three as one iteration of the solution algorithm. For further detail about the solution method, please see [Justesen et al., 2014].

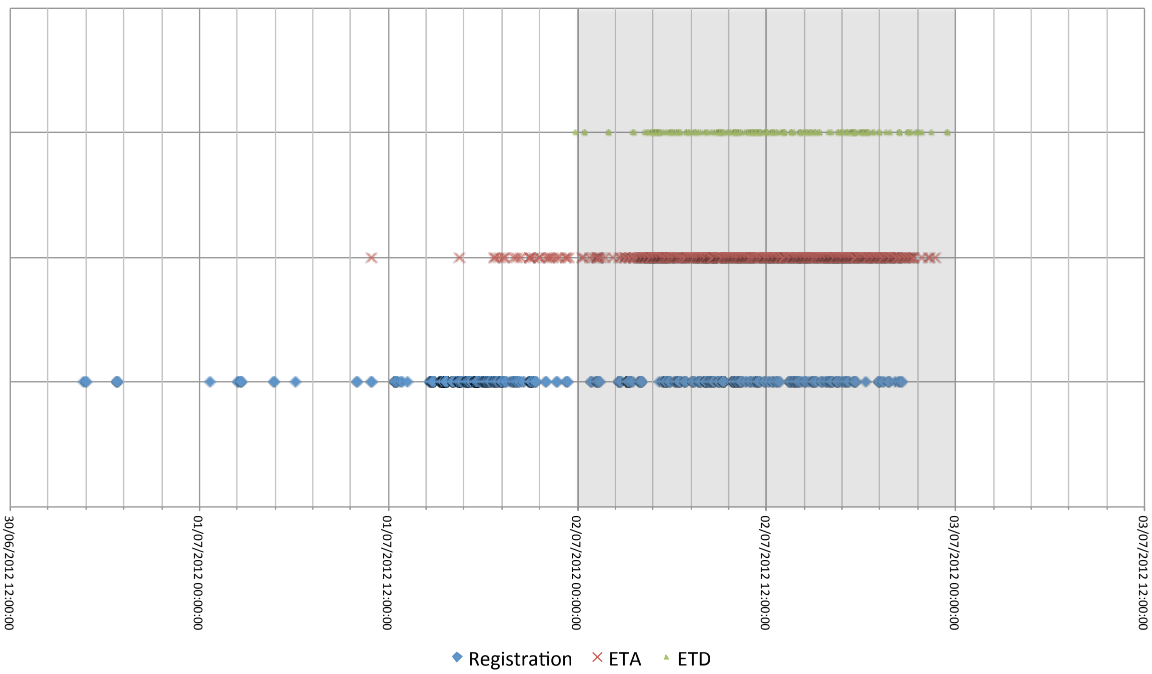

Figure 10.5: *TA frequencies for arrivals and departures scheduled to take place on July 2, 2012 at CPH. As can be seen, many tail assignment changes are received even before the day of operation, where the majority of the ETD and ETA TAs are received during the actual day of operation.*

10.6 Results

To test the proposed methodology, we consider a typical busy summer day at Copenhagen Airport (CPH) and mimic the handling of the corresponding set of timetable changes. Details of the timetable composition can be found in Table 10.1 and details of the set of corresponding TAs are shown in figures 10.6 and 10.7. The total number of timetable changes received for July 2, 2012 is 2309 and as can be seen from Figure 10.6 more than 50% of the timetable changes are changes to the aircraft registration. Figure 10.7 shows the sizes of the delays (relative to scheduled time) specified in the changes to the ETAs and reveals that the majority of the flights considered arrive earlier than scheduled.

To mimic the handling of the timetable changes received for the considered day of operation, we iterate over the set of timetable changes and process them consecutively. Conflicts are handled in accordance with the chosen conflict handling strategy (ICH or LCH). In principle any frequency for the LCH could be chosen, and the frequency does not have to be regular, but can vary during the day of operation depending on the TAs received. However, for the sake of simplicity, and to demonstrate the concept of the LCH strategy, we apply an approach where conflicts are handled every five minutes.

For both strategies, the turn-round re-allocation problem is solved using the solution method described in Section 10.5.2. As described, the solution method potentially only considers a fraction of the possible handling schedules, and it stops the first time a feasible solution to OSGAP has been found, i.e. there is thus no guarantee that an optimal solution has been found. To evaluate the quality of the solutions, we also apply an approach where an optimal solution is found at each conflict handling. The optimal solution is found by enumerating all variables and solving OSGAP using a standard MIP solver. In the following we refer to the solution method adapted from [Justesen et al., 2014] as the *partial enumeration* method (PE) and the exact method as the

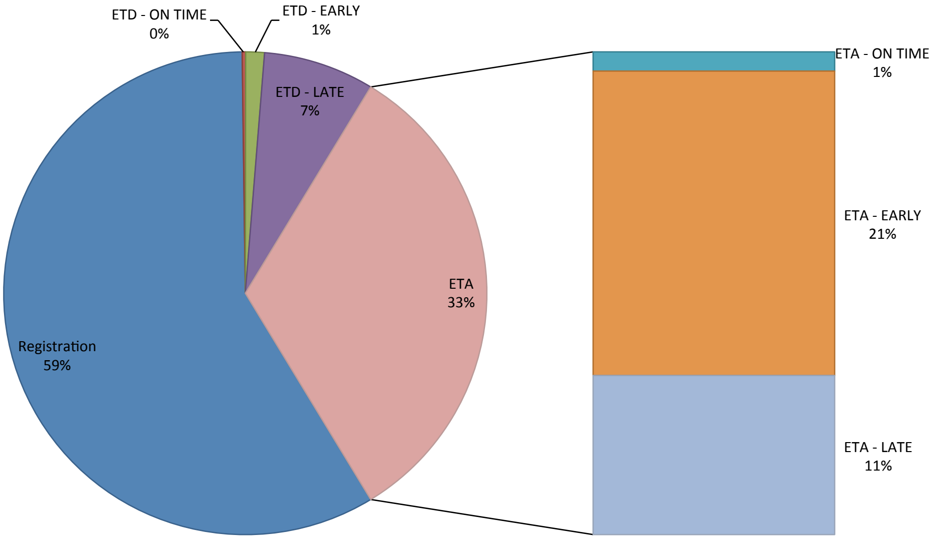

Figure 10.6: *Distribution of timetable changes for July 2, 2012 at CPH.*

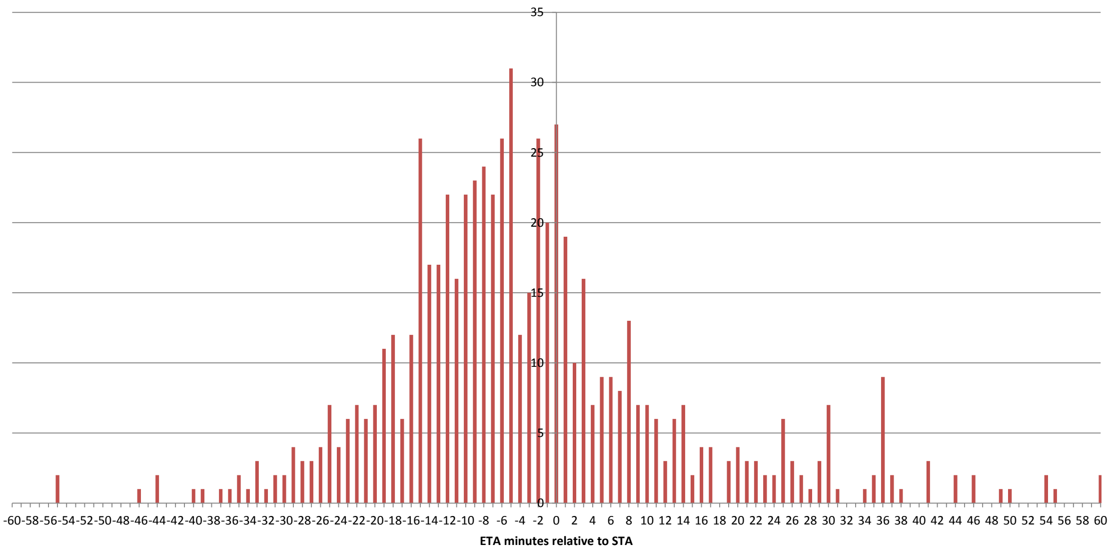

Figure 10.7: *Delay sizes for changes to the ETA for July 2, 2012 at CPH.*

Algorithm 3: Pseudocode for the described solution method.

```
Read airport data, timetable and parameters;
Set $\mathcal{P}' = \emptyset$;
repeat
    Solve R-OSGAP using a standard LP solver;
    Identify the set of unallocated turn-rounds $\tilde{\mathcal{R}}$;
    Identify the set of binding constraints $\tilde{\mathcal{K}}$;
    Identify the set of conflicting turn-rounds $\hat{\mathcal{R}}$;
    for each turn-round $r \in \tilde{\mathcal{R}} \cup \hat{\mathcal{R}}$ do
        if $\mathcal{P}_r \setminus \mathcal{P}'_r \neq \emptyset$ then
            Identify the cheapest handling schedules from $\mathcal{P}_r \setminus \mathcal{P}'_r$ compatible with $\tilde{\mathcal{K}}$;
            if possible to identify new handling schedules then
                Add new handling schedules to $\mathcal{P}'$;
until valid solution has been found;
if solution is fractional then
    Restrict all variables to be binary;
    repeat
        Solve OSGAP using a standard IP solver;
        if solution to OSGAP found then
            Save and return solution;
        else
            for each turn-round $r \in \mathcal{R}$ do
                Identify the cheapest handling schedules from $\mathcal{P}_r \setminus \mathcal{P}'_r$;
                if possible to identify new handling schedules then
                    Add new handling schedules to $\mathcal{P}'$;
    until valid solution has been found;
else
    Save and return solution;
```

# Operations	960
# Operations - Domestic	79
# Operations - Schengen	661
# Operations - Non-Schengen	98
# Operations - Non-EU	122
# Arriving pax - total	51,568
# Departing pax - total	53,620
# Bussed pax - Actual	3,483
# Towings - Actual	10

Table 10.1: *Characteristics of the timetable used for testing.*

total enumeration method (TE).

For each conflict handling we report how the algorithm performs and evaluate the values of the different objectives in the objective function. To simplify, we assume that all parameters except for the timetable, i.e. stands, stand clusters, push-back options, etc. remain unchanged during the entire day of operation.

For both PE and TE the objective function is given as

$$\begin{aligned}\hat{z} = \quad & \omega_1 \sum_{r\in\mathcal{R}} \pi_r + \\ & \omega_2 \sum_{p\in\mathcal{P}} \delta_p \frac{\rho_p}{\rho^{max}} x_p + \\ & \omega_3 \sum_{p\in\mathcal{P}} \delta_p \left(\frac{\theta_p^a + \theta_p^d}{2\theta^{max}}\right) x_p + \omega_4 \sum_{p\in\mathcal{P}} \delta_p \left(\frac{\phi_p^a + \phi_p^d}{2\phi^{max}}\right) x_p - \\ & \omega_5 \sum_{p\in\mathcal{P}} \delta_p \left(\frac{n_p^a + n_p^d}{N}\right) x_p \end{aligned} \tag{10.9}$$

where we set $\omega_1 = 10000$, $\omega_2 = 1000$, $\omega_3 = 10$, $\omega_3 = 5$, and $\omega_4 = -1$, as these weights reflect the prioritization of the objectives currently applied at CPH. Moreover, we have that $\rho^{max} = 10$, $\theta^{max} = 10$, $\phi^{max} = 10$, but N is dependent on the turn-rounds in $\mathcal{R}$ and may therefore be different for each conflict handling.

The tests have been run on a Linux server running SuSE Linux Enterprise 11 64 bit with 16 GB RAM and a single Intel Merom single core processor. We have used the Gurobi Optimizer version 5.5.0 as both the LP and MIP solver and algorithms have been coded in the C++ programming language.

The performance of the solution method is summarized in Table 10.2 and 10.3 and the quality of the solution is illustrated in the Figure 10.8, 10.9, and 10.10.

	PE				**TE**			
	Min	**Max**	**Avg.**	**Std.dev.**	**Min**	**Max**	**Avg.**	**Std.dev.**
Num. solutions generated	—	—	739	—	—	—	745	—
Elapsed time - total	0.2	13.4	2.9	1.9	0.3	194.4	35.5	34.8
Num. iterations	2	26	5.8	3.7	—	—	—	—
LP integer	—	—	96%	—	—	—	—	—
Num. turn-rounds in conflict interval	1	146	56	42	1	157	57	41
Num. handling schedules generated	14	35,932	13,875	10,364	—	—	—	—
Num. handling schedules added to $\mathcal{P}'$	14	6,577	1,593	1,081	—	—	—	—
Num. re-allocated critical turn-rounds	0	5	0.4	0.8	0	8	0.5	1.0
Handling preference compliance	74%	78%	76%	1%	76%	80%	78%	1%
Number of bussed pax	1,186	2,552	1,916	510	1,259	2,189	1,727	158
Number of towings	9	27	14	5	49	125	83	25
Stand preference compliance	74%	78%	76%	1%	76%	80%	78%	1%

Table 10.2: *Performance of the solution method - instant conflict handling (ICH).*

	PE				**TE**			
	Min	**Max**	**Avg.**	**Std.dev.**	**Min**	**Max**	**Avg.**	**Std.dev.**
Num. solutions generated	—	—	257	—	—	—	262	—
Elapsed time - total	0.3	13.6	3.8	2.5	0.4	201.3	48.7	43.7
Num. iterations	2	30	6.4	3.9	—	—	—	—
LP integer	—	—	90%	—	—	—	—	—
Num. turn-rounds in conflict interval	1	262	72	55	1	262	71	55
Num. handling schedules generated	14	66,382	18,238	12,617	—	—	—	—
Num. handling schedules added to $\mathcal{P}'$	14	6,896	2,058	1,352	—	—	—	—
Num. re-allocated critical turn-rounds	0	3	0.5	0.8	0	6	0.7	1.0
Handling preference compliance	98%	98%	98%	0%	81%	83%	82%	0%
Number of bussed pax	1,186	1,982	1,584	337	1,186	2,054	1,574	372
Number of towings	7	26	12	6	47	110	72	22
Stand preference compliance	78%	80%	79%	0%	81%	83%	82%	0%

Table 10.3: *Performance of the solution method - lazy conflict handling (LCH).*

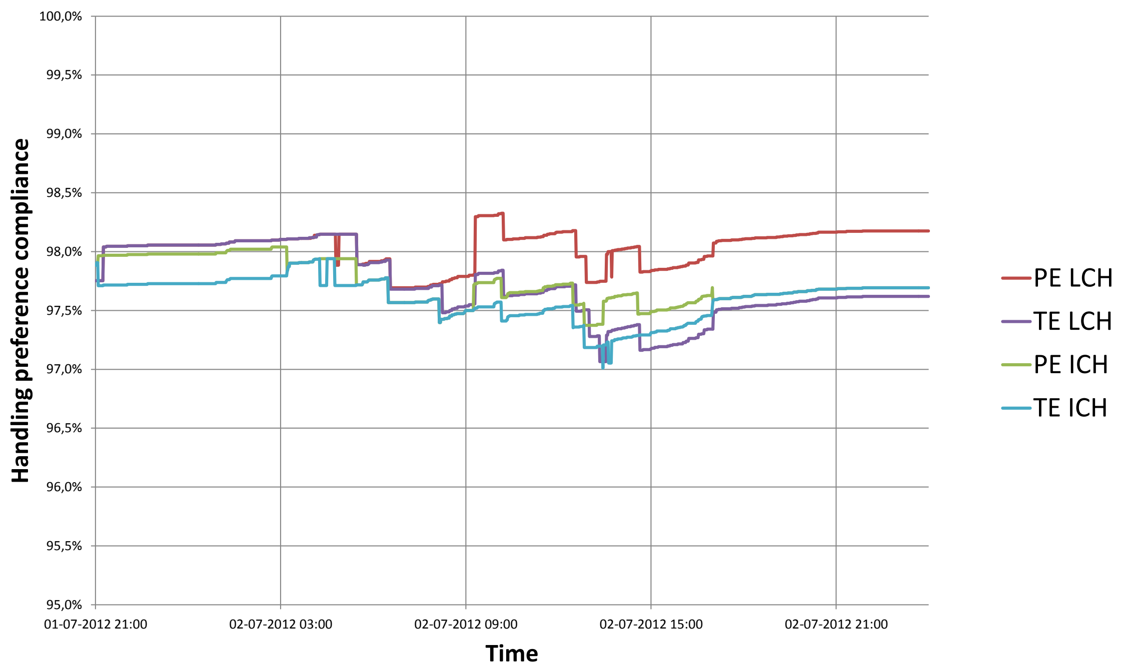

Figure 10.8: *Handling preference compliance.*

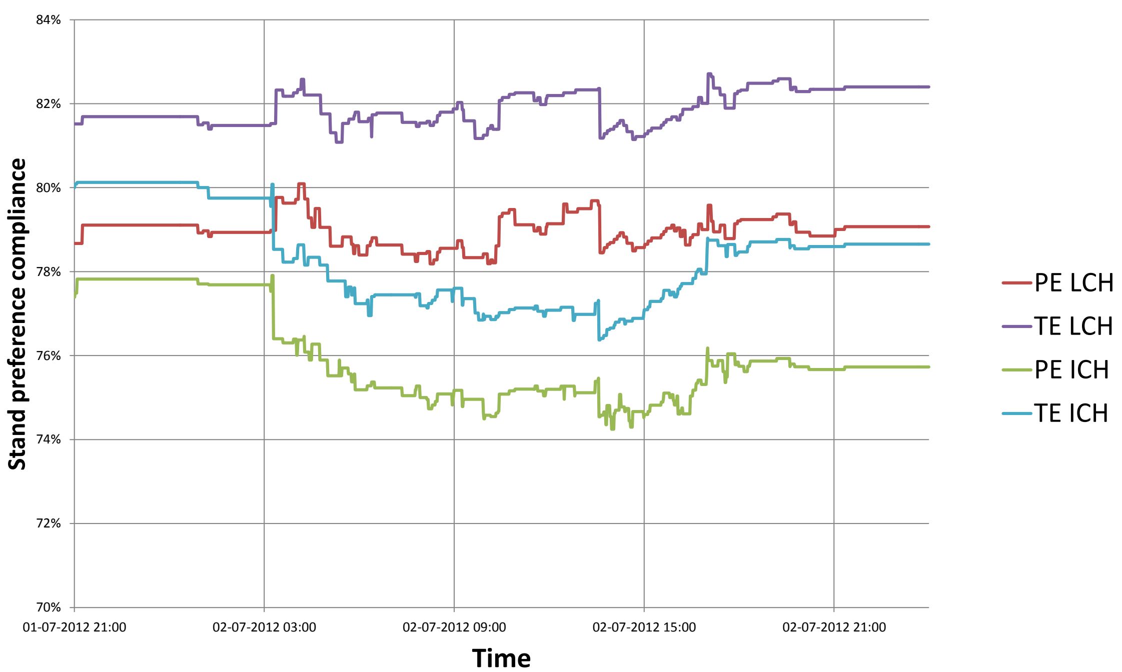

Figure 10.9: *Stand preference compliance.*

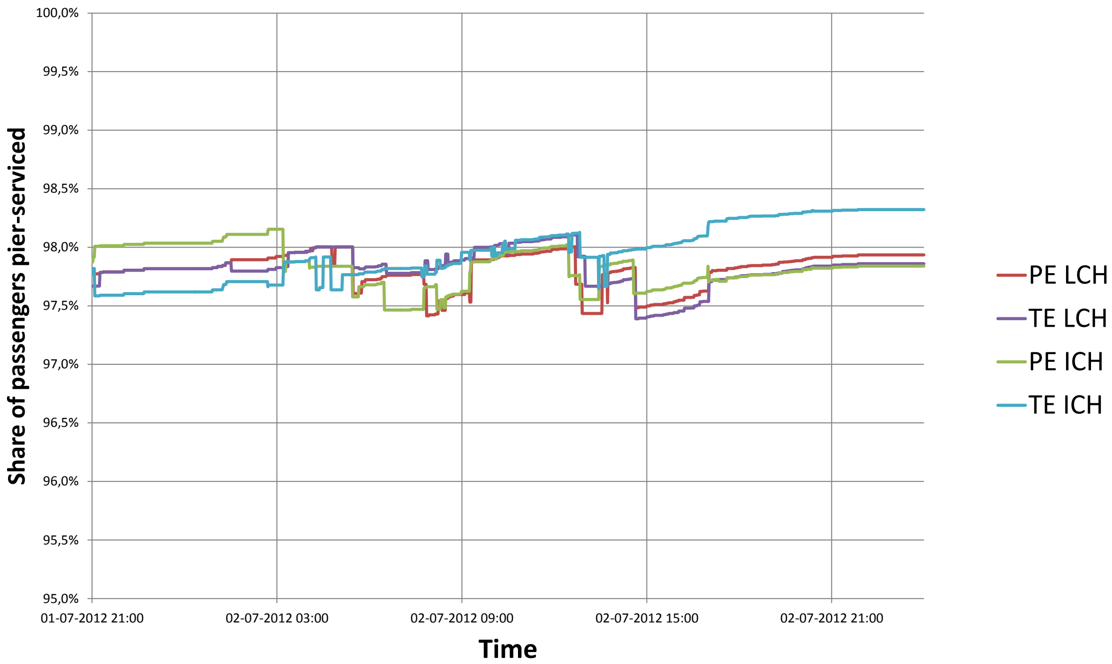

Figure 10.10: *Share of passengers that are pier-serviced.*

As can be seen from Table 10.2 and 10.3 conflict handling is triggered only around 250 times for the LCH strategy as opposed to around 740 times for the ICH strategy. This result was what we expected, as the LCH only resolves conflicts every five minutes, while timetable changes are received every 90 seconds in average for all TAs and every 45 seconds for TAs received in the period 06:00-09:00 on July 2, 2012. Moreover, the number of turn-rounds in the conflict intervals, $|\mathcal{R}|$, was 56 in average with a standard deviation of 42 for ICH strategy, but 72 in average with a standard deviation of 55 for the LCH strategy. However, this was also what we expected, since for the LCH strategy, more conflicts may by handled per conflict handling.

In average, the PE methods solves the problem very efficiently (in average 1.9 seconds for the ICH and 2.5 seconds for the LCH strategies) making the method applicable to operational stand and gate allocation. One of reasons for this is that the number of handling schedules added to $\mathcal{P}'$ is kept relatively small compared to the number of handling schedules generated (in average, only 11% of the generated schedules are added for the ICH strategy, and only 11% are added in average for the LCH strategy). Furthermore, Table 10.2 and 10.3 show that only few iterations of the solution algorithm (see Algorithm 3) are needed to find a solution. This indicates that the relatively simple approach of partially enumerating the handling schedules and just adding them sequentially with constantly increasing costs, is a viable approach in this context.

The peak in the solution time for the lazy conflict handling strategy is due to the fact that the majority of aircraft registration changes is received late in the afternoon on the day before the day of operation in consideration. Once the tail assignment has been finished by the airlines it is sent out as one message describing all tail assignment, however this message is handled in the stand and gate allocation system as a large number of tail assignment changes received more or less at the same time. Late in the afternoon on the day before the day of operation in consideration the number of conflicts to be handled is thus very large.

With regards to the quality of the solution, the PE method generally performs well. In Table 10.3 it can be seen that for the PE LCH approach, only a few more passengers are bussed with both max and average values that lies within 70 pax of the TE LCH method. Stand preference compliance, which is important for the customers of the airport, namely the airlines, are equally good in the PE LCH approach yielding an average compliance only 3%-points lower than the TE LCH methods result of 82%.

Similar results are obtained for the PE ICH and TE ICH approaches; however it is apparent that the LCH approach yields better results in general than the ICH approach. For all the parameters that are important for the operations of the airport and the airlines (handling preference compliance, number of bussed pax, number of towings and stand preference compliance) the LCH approach outperforms the ICH approach. Even when comparing the PE LCH method to the TE ICH method, the results of the LCH approach are superior on all the KPIs.

It is interesting that the TE method results in a lot more towings than the PE approach. This is a result of the cost setting and the prioritization of the handling plans. As stand preference compliance (maximize) and number of bussed pax (minimize) are most important, the optimizer will for the TE method reach the solution that yields the best results on these parameters. In Table 10.3 it can be seen that while figures for number of towings and handling preference compliance are somewhat degraded for the TE method, the figures for number of bussed pax and stand preference compliance KPIs are improved in the TE method compared to the PE method. These results are generated using the same cost settings and handling plan priorities for both the PE and TE method. Different settings for the PE and the TE methods may be required to yield operationally optimal results for both methods.

When comparing the results on bussed pax and towings to the real-life results for the timetable used for testing, we find that the real-life number of towings (10) falls within the min and max values obtained for the PE method. However the number of real-life bussed pax (3,483) is 37% higher than the max value obtained for either approach.

From an operational point of view, the PE method yields results that fulfills the need and are considered optimum for the operational environment.

10.7 Conclusion

In this paper we have considered the operational variant of the stand and gate allocation problem. We have proposed two different strategies for handling conflicts in the stand and gate allocation plan that arise when changes to the timetable are received (estimated arrival time, estimated departure time, and aircraft registration). The instant conflict handling strategy immediately resolves every conflict that occurs, whereas the lazy conflict handling strategy only resolves conflicts every five minutes. To resolve conflicts, the stand and gate allocation problem is solved on a part of the stand and gate allocation plan defined by the conflict interval in order to reallocate one or more turn-rounds. To solve the stand and gate allocation problem we have proposed a set packing inspired formulation with a resource based constraint system that implicitly resolves conflicts and developed a solution approach that exploits the flexibility of the model to be dynamically updated. The solution approach attempts to expand an invalid, restricted problem by identifying conflicting turn-rounds and intelligently include extra flexibility without creating an intractable problem. Extra flexibility is provided in the form of additional handling schedules. A key benefit of the model proposed is that it is relatively simple to include any kind of resource demand associated with the handling of a turn-round, and conflicts are resolved implicitly. Furthermore the model relatively easily incorporates the requirements and preferences of the different stakeholders of the problem; stand preferences for different airlines are expressed via stand clusters and handling preferences for different types of turn-rounds, aircraft types, airlines, etc. are expressed via handling options. Stand dependencies are handled implicitly via push-back options, yielding a much more accurate approach than normally seen for this type of constraint.

Numerical results on a real life test instance arising in Copenhagen Airport show that the proposed methodology performs well as it can find near-optimal for all conflicts in reasonable time while maintaining a high solution quality. Both the instant and lazy conflict handling strategies perform well in that they both give rise to good values of the KPIs, however the lazy conflict handling strategy is the most attractive, as fewer conflict handlings are needed and the KPI values in general are higher.

Bibliography

Bolat, A., 1999. Assigning arriving flights at an airport to the available gates. Journal of the Operational Research Society 50 (1), 23–34.

Bolat, A., 2000. Procedures for providing robust gate assignments for arriving aircrafts. European Journal of Operational Research 120 (1), 63–80.

CPH, 2012a. Copenhagen airports a/s - annual report 2012. Web.

CPH, 2012b. Official website. Web.
URL www.cph.dk

Diepen, G., van den Akker, J. M., Hoogeveen, J. A., Smeltink, J. W., 2012. Finding a robust assignment of flights to gates at amsterdam airport schiphol. Journal of Scheduling 15 (6), 703–715.

Dorndorf, U., Drexl, A., Nikulin, Y., 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., Pesch, E., 2007b. Disruption management in flight gate scheduling. Statistica Neerlandica 61 (1), 92–114.
URL http://EconPapers.repec.org/RePEc:bla:stanee:v:61:y:2007:i:1:p:92-114

Dorndorf, U., Jaehn, F., Pesch, E., 2008. Modelling robust flight-gate scheduling as a clique partitioning problem. Transportation Science 42 (3), 292–301.

Genç, H., Erol, O. K., Eksin, İ., Berber, M. F., Güleryüz, B. O., 2012. A stochastic neighborhood search approach for airport gate assignment problem. Expert Systems with Applications 39 (1).

Jaehn, F., 2010. Solving the flight gate assignment problem using dynamic programming. Zeitschrift für Betriebswirtschaft 80 (10), 1027–1039.

Justesen, T. F., Larsen, J., Ryan, D., Lusby, R., Dohn, A., 2014. Aircraft stand allocation with associated resource scheduling. Special issue on Airport Operations Management, Computers and Operations Research.

Lim, A., Rodrigues, B., Zhu, Y., 2005. Airport gate scheduling with time windows. Artificial Intelligence Review 24 (1), 5–31.

Şeker, M., Noyan, N., 2012. Stochastic optimization models for the airport gate assignment problem. Transportation Research Part E 48 (2), 438–459.

Tang, C.-H., Yan, S., Hou, Y.-Z., 2010. A gate reassignment framework for real time flight delays. 4OR 8 (3), 299–318.

Wei, D., Liu, C., 2007. Optimizing gate assignment at airport based on genetic-tabu algorithm. Proceedings of the IEEE International Conference on Automation and Logistics August 18 - 21, 2007, Jinan, China, 1135–1140.

Wei, D., Liu, C., 2009. Fuzzy model and optimization for airport gate assignment problem. 2009 IEEE International conference on intelligent computing and intelligent systems, Proceedings, Vol 2, 828–832.

Yan, S., Chang, C., 1998. A network model for gate assignment. Journal of Advanced Transportation 32 (2), 176–189.

Yan, S., Ho, C., 2001. Optimization of multiple objective gate assignments. Transportation Research, Part A (Policy and Practice) 35A (5), 413–432.

Yan, S., Tang, C., 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., Hou, Y.-Z., 2011. Airport gate reassignments considering deterministic and stochastic flight departure/arrival times. Journal of Advanced Transportation 45 (4), 304–320.

Chapter 11

Making OR a Success: Application of Optimization to Airport Operations

Torben C. Barth [*†] Tor Fog Justesen[*‡] Frauke Böckmann [†]

[*]Department of Management Engineering, Technical University of Denmark
Produktionstorvet, Building 426, DK-2800 Kgs. Lyngby, Denmark

[†]Information and Telecommunications, Fraport AG
D-60547 Frankfurt am Main, Germany

t.barth@fraport.de [‡]Planning, Analysis & Projects, Copenhagen Airports A/S
Lufthavnsboulevarden 6, 2770 Kastrup

Contact: Tor Fog Justesen (+45 23 24 62 47 / tor.justesen@cph.dk)

Abstract Operations Research is considered to be an applied research field, but why do most OR projects end up as academic exercises with no practical impact? Based on a dozen of successful projects implemented in major airports, the authors give an overview of enablers and challenges that need to be addressed to get an overall acceptance of a developed application or system. The discussion takes its origin in two major projects implementing decision support tools for check-in counter allocation and baggage handling. To have OR projects accepted by the organization, employees, and decision makers, a range of real-life operational requirements must be taken into consideration. Unfortunately, many researchers do not have direct access to the operational environment of an airport. Consequently, correctly incorporating and modeling the operational requirements can be difficult. The paper presents a general discussion of the most important operational aspects to consider when developing models and solution methods for airport optimization problems, and proposes a new step in the development process where these aspects are considered. The idea is that this new step mimics the existence of an operational environment and thus increases the probability that the developed application can have an impact on the operation and be accepted by the stakeholders of the problem.

11.1 Introduction

Operating an airport gives rise to a broad variety of optimization problems. Many of these can be modeled and solved using operations research techniques. The field of airport operations management is an emerging field of research within operations research and a continuously growing number of airport optimization problems are being dealt with in the literature.

Seen from an airport operator's perspective, an important success criterion when developing applications for solving the different problems is that the solutions are applicable in the real-life operation. This may seem obvious, however, during the work on several problems arising at two larger European airports, we discovered that the task of finding applicable solutions was more difficult than anticipated.

The success of an application can be measured by the *applicability*, the *impact*, and the *acceptance*. The applicability describes how well the application performs on data instances originating from the operational environment of the airport. The impact describes whether or not the application improves the considered processes and decisions. Finally, the acceptance describes how well the application is acknowledged by the different stakeholders of the problem (dispatchers, unions, blue-collar workers, etc.).

11.1.1 Development process

Developing and deploying an application to a real-world optimization problem involves two processes: a *development process* where the model and the solution method are developed and refined and an *operational process* where solutions to the problem are deployed in the real-world setting in which the problem arises.

As described in Rönnqvist [2012], the initial step of the development process is typically to *describe and define* the problem. This is normally done in an abstract language. It is necessary to identify, simplify and aggregate the real-world problem into a tractable optimization problem. In the *modeling* step, the parameters, constraints, decision variables, and objective function are identified and the problem is formulated as a mathematical model. To solve the model, a *solution method* needs to be defined. The solution method can be either a heuristic, an exact method or a combination of both. Finally, the solution method is *implemented* and the results are ready to be deployed in the organization. These three steps are typically performed sequentially, however they are continuously adjusted using *feedback loops*. The development process is thus an iterative process.

The operational process consists of the following three steps: *planning*, *execution and monitoring*, and *follow-up*. In the planning phase, the generated solutions are used to plan the different activities in the considered process. The result of the planning phase can be, e.g. a work plan for a group of staff, an investment plan, or a resource allocation plan. The time horizon of the plan depends on the specific problem and the airport. It is important to note that the planning decisions can even be in real-time followed directly by the execution phase like, for instance, the assignment of flights to baggage carousels. In the execution and monitoring phase, the planned activities are performed. To ensure that everything goes as planned and to be able to handle perturbations to the initial plan due to e.g. delayed flights or technical issues with the equipment, the activities are usually monitored while they are being executed. This monitoring may either be via systems and data or it can be manually. In the follow-up phase, learnings from already executed processes are evaluated and the knowledge obtained is used to either take precautions in the next planning of the process or perhaps even change elements in the planning step on a more permanent basis.

The development process and operational processes can in principle be seen as decoupled. However, our experience is, that the two processes should be seen as interdependent. The developed solution approach forms the basis of the planning step and conclusions from the follow-up step should be fed back into the development process to improve both the modeling and the solution method. See Figure 11.1 for an illustration of the two processes and how they are connected.

11.1.2 Motivation and outline

Our experience is that the developed application is more likely to become successful because the interdependencies between the two processes can be exploited when having access to both

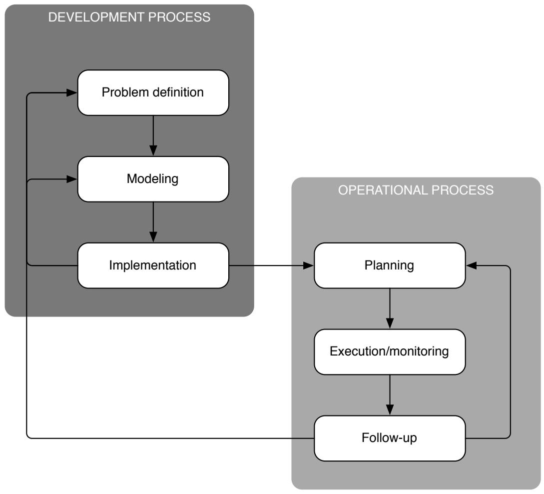

Figure 11.1: *Illustration of the development process and the operational process and their interdependencies.*

the development process and the operational process. One way of doing this is by initially only considering a simple variant of the problem, not necessarily incorporating all operational details. Solutions to this problem are then deployed in the operational process, and the feedback from the follow-up step is then used to adjust and refine the model and the solution method. This interaction is then repeated until a formulation of the problem leading to acceptable solutions has been found. The advantages of this approach are multiple. First of all, the operation is indirectly part of the development process. This gives the people from operations a sense of ownership, which increases the probability that the final model and solution method will be accepted. Furthermore, it is ensured that the primary requirements are correctly modeled, which increases the probability that the applications will also have an impact. And finally, the applicability is continuously tested, and contingencies arising due to e.g. incorrect or insufficient data or requirements only existing as tacit knowledge are quickly discovered and the necessary precautions can be made.

Unfortunately, most researchers do not have access to the operational practice, and if they do, they are rarely in a position where they can influence it. As a consequence, the vast majority of the articles published about airport optimization problems is focused on the mathematical formulation and theoretical aspects of the different problems. The developed models are based on extensive literature reviews and the developed solution methods take mathematical optimality, bounds, and performance of the algorithm into consideration. Typically, the solutions are proven to be either optimal or near-optimal and in most cases better than some defined "current practice". This is by no means negative and it is not the intention of this article to dispute neither the scientists having already published their work nor the quality of the different scientific journals publishing papers on airport optimization problems. The published research articles all contribute to the ongoing research within the application of operations research to airport operations management.

The aim of this article is to provide researchers and practitioners with a better understanding of the different aspects of the operational process and the environment in which the different optimization problems arise. We do this by introducing a new step in the development process. The step is called *operational aspects* and is placed after the problem definition, but before the modeling step. The purpose of this step is to ensure that the following operational aspects are taken into consideration: the role of the airport, the considered planning horizon, the problem variability, the qualifications of the decision makers, the stakeholders, the interdependency to other processes, the objectives of the problem, the goals of the project, the definition of optimality, the data quality, the degree of automation, and the deployment options. In the following we describe each of these aspects in more detail and discuss how they might have an impact on the development of the model and solution method. The idea is, that this step mimics the existence of an operational process, thereby enabling researchers and practitioners to develop successful applications. See Figure 11.2 for an illustration of the modified development process.

To illustrate the advantages of taking the described operational aspects into consideration, we present two different case studies. The first case study deals with the allocation of check-in counters to departures at Copenhagen Airport (CPH) and the second case study presents three different projects related to the handling of baggage at Frankfurt Airport (FRA).

All authors have several years of experience working within airport operations management at CPH respectively FRA. The article summarizes the experiences obtained while working on the projects described in the case studies. Common for all projects is that the applications have all been deployed at the respective airport and have been successfully running since then. All authors are part of internal departments of the companies operating the airports and responsible for supporting the operational departments with analyses and solution development. The two airports are operated by two independent companies.

The remainder of the article is structured as follows. In Section 11.2 we give a brief overview of the related literature. In Section 11.3 the suggested operational aspects are described. Section 11.4 presents the case about check-in counter allocation at CPH and in Section 11.5 the case about baggage handling at FRA is presented. Finally, we draw some conclusions in Section 11.6.

11.2 Background

Figure 11.3 illustrates the different processes that are part of the operation of an airport. Each process gives rise to operational decision problems that may be supported by optimization algorithms. The amount of research in the area of airport operations is in no terms comparable to the connected field of airline management. Hounsgaard and Justesen [2012] and Böckmann et al. [2012] presented lessons learned during the introduction of optimization based applications in the area of airport operations. These presentations clearly show the need for having the operational context of the airport in mind when developing models and solution methods.

Fortuin and Zijlstra [1989] present the experiences of an OR group in industry. They show that OR can succeed as part of a company. Haneyah et al. [2012] study generic control architecture for automated material handling systems and compare the practical requirements versus the existing theory. They use the control of the baggage handling system in an airport as one of their examples. The conclusion of the paper is that practical requirements need to be incorporated in the systems. Furthermore, the generic architectural design should be “flexible, modular, robust, and scalable”. Their field of research is not concerned with OR applications but their conclusions can be translated

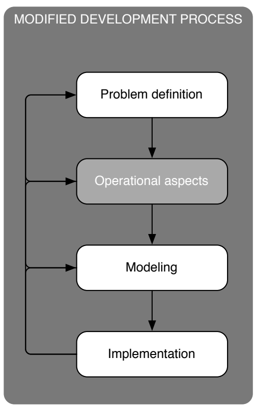

Figure 11.2: *Illustration of the modified development process including the step considering operational aspects.*

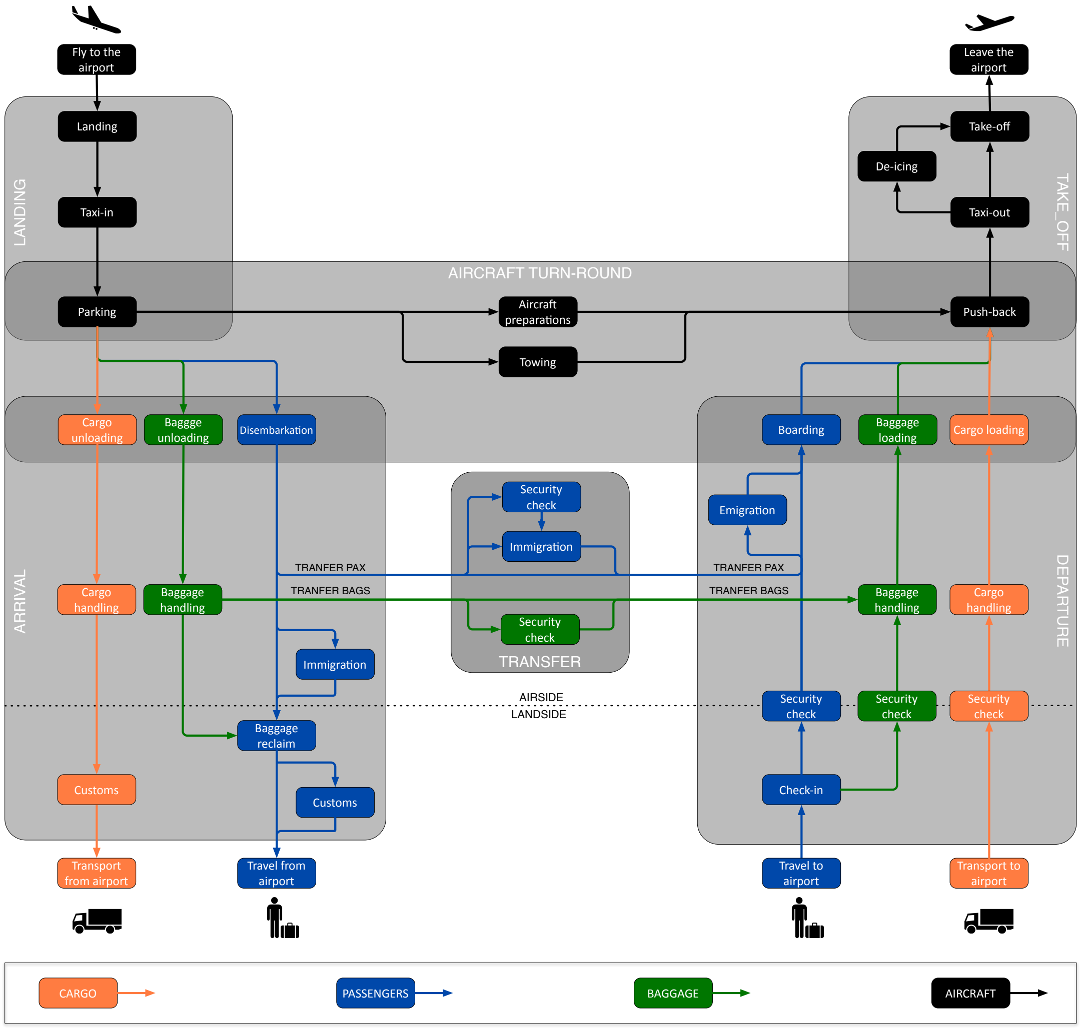

Figure 11.3: *Overview of the different processes in airport operations.*

into the OR context.

Hildebrandt [1980] focuses on the different roles during the implementation of OR. He identifies important problems in the implementation process and concludes that implementation is one of the major obstacles for the use of OR. Hildebrandt [1981] studies important implementation strategies and their limitations. Tilanus [1985] discusses success and failure reasons on different cases collected in the Dutch industry. Schneeweiss [1987] introduces a formalization of quantitative model building that also respects soft and vague factors.

To celebrate the 20th anniversary of EURO, a special issue of EJOR was published in 1995. The special issue was partly concerned with the practical issues when doing OR. For instance the article from Ranyard [1995] presented a review of OR practice in the UK and the article by Corbett et al. [1995] showed that to draw conclusions about successful approaches it is necessary to study not just a single project, but a whole range of projects.

Murphy [2001] studies the role of practitioners in the field of OR. In the "The Art and Science of Practice" series of Interfaces several articles concerning the practical implementation of OR applications are discussed (see for further details [Murphy, 2005a], [Murphy, 2005b], [Murphy, 2005c], [Hew, 2007], [Liberatore and Luo, 2013]).

Semini [2011] studied in his PhD Thesis applicability of OR in manufacturing logistics. Rönnqvist [2012] discusses in his article OR challenges and experiences from solving industrial applications. Several cases are described and the learnings from the projects are presented.

11.3 Operational aspects

In the following we describe the most important operational aspects to consider when developing applications for airport optimization problems.

11.3.1 Role of the airport

The role of the airport differs from process to process. The airport can be either *process owner* (planning, execution/monitoring, and follow-up) or *process facilitator* (monitoring and follow-up). It is hence not possible to give a description of the division of responsibilities that is common for all processes. Moreover, many processes are either handled by the airlines themselves or outsourced to external handling companies (e.g. catering and fueling). For these processes, the primary job of the airport is to ensure that the optimal working conditions are present and that all required ground handling assets are in operation. Nevertheless, the airport often monitors all processes and has a constant dialogue with all of the handling companies to ensure that all processes are continuously optimized and that all customers (passengers, airlines, etc.) are satisfied.

If the airport is process owner entirely, the solutions of the applications can be deployed as operational requirements. If the airport is process facilitator, the solutions can only be deployed as some kind of decision support, however the solutions may still be valuable for the airport as input to e.g. capacity analyses.

11.3.2 Planning horizon

A problem typically appears in three different variants, namely a *strategical*, a *tactical*, and an *operational* variant. The specification depends on the individual problem and the strategical variant may be the same as the tactical variant, and equivalently the tactical variant may be the same as the operational variant.

At the strategic level, one typically considers a long-term planning horizon (e.g. from six months to several years) and the primary focus is on capacity assessment in relation to a forecasted resource demand. Obviously, the airport wants to exploit its existing resources to the greatest possible extent. The question is how long this will be possible before resource acquisition is needed. The strategic level is furthermore characterized by a high degree of uncertainty on the available data as it is often prognosis data and based on general assumptions. It can thus be accepted that the level of detail is not that high for the model formulations of the strategic variant of a given problem.

The tactical level typically considers a short-/medium-term planning horizon (e.g. from one day to six months). Data availability increases and is extended by prognosis data. The emphasis is on finding an optimal or near-optimal feasible solution.

Operational problems occur on a day-to-day basis and often need almost immediate resolutions, e.g. when predetermined solutions need to be adjusted due to unforeseen disruptions. Typically, the considered planning horizon is very short, and one can assume that all necessary data is available. Nevertheless, the data can still be subject to uncertainty. The aim is to handle any type of disruption that causes the current solution to become invalid (late aircraft arrival or departure, unavailability of resources due to maintenance or technical issues, canceled departures, etc.). As the plans are often already disclosed or at least known by the dispatcher, the new solution must impose a minimum amount of changes to the original solution.

In short terms, the strategic level is where the preconditions for optimality are analyzed and decided upon (e.g. increase/decrease resource supply), the tactical level is where optimality is created (e.g. actual usage of the resources) and the operational level is where optimality is maintained (e.g. contingency plans). The focus shifts from cost and efficiency oriented goals at the strategic level to more cost and quality driven goals at the operational level. At the strategic level, the airport considers how to obtain a given level of service with the least costs. At the operational level the cost is to some extent fixed and the best possible quality is sought under the given settings.

A common misunderstanding is that run times are allowed to be long for strategical problems as they are dealing with future planning and consider relatively long time periods. However, an airport typically wants to analyze multiple different scenarios rather than providing detailed solutions. The efficiency of the solution method is thus also very important in this case. Equivalently, the tactical variant of a given problem should also be solved efficiently as even tactical plans are subject to many changes the more accurate data becomes.

11.3.3 Problem variability

A fast changing environment is typical for the air traffic business. For example, new regulations or the introduction of new aircraft types can cause changes in the demand structure and alterations in the existing processes. When solving an operational optimization problem it is important to correctly determine the variability of the problem. The variability of a given problem are given by the dynamics of the problem domain and by the dynamics of the problem itself.

The *domain dynamics* describe how often, e.g. the constraints and the objective function of the problem are changed. Every time any such change is encountered, the underlying model and solution method must be changed accordingly.

The *problem dynamics* describe how often the problem must be solved, the variation in the sizes of the instances, etc. Exemplary high problem dynamics can be found in cases where many changes in the input data occur closely before the final event.

The dynamics of the problem have an impact on the design of the data processing and the solution algorithm. If one is considering a problem with low domain dynamics but high problem dynamics, the primary focus should be on developing an efficient solution algorithm and it can be accepted that the model is formulated in a more complicated way to favor the solution algorithm. On the other hand if one is considering a problem with high domain dynamics but low problem dynamics, the model should be formulated in such a way that e.g. constraints and parameters can be easily added, modified, or removed, and focus can be less on the efficiency of the solution algorithm.

Equivalently, if the goal of solving the problem is to provide plans to be followed and the input data changes every thirty seconds, having a solution algorithm that takes a couple of minutes to run, does not make sense.

11.3.4 Qualifications of the decision makers

The qualifications of the end users are an important factor influencing the way decisions are taken. In an airport, there is typically a broad range of qualifications present among the different decision makers, ranging from blue-collars with low education to managers with a higher education. Though the qualifications of the decision maker are unlikely to have an impact on the chosen modeling approach and solution method, one needs to consider this aspect when considering how the applications should be deployed.

Many operational decision makers are used to take decisions primarily based on their gut feeling. Experiences and solutions provided by a system are per definition not as good their own solutions. However, given the continuously increasing complexity of airport processes, decision support applications are becoming more and more common. The developed model and solution method should thus support the dispatchers in monitoring the process. Furthermore, the suggested solutions have to be explainable in a simple way.

11.3.5 Stakeholders

In most cases, many stakeholders are involved in the different airport operations processes. The stakeholders can be either directly or indirectly connected to the process. Examples of directly connected stakeholders are the decision makers and the stakeholders affected by solutions to the problem (airlines, passengers, handling companies, etc.), and these are often considered as the most important. Examples of indirectly connected stakeholders include other departments within the airport organization, airport authorities, police, customs, etc.

If the solution to a given problem involves many different stakeholders, one should analyze the dependencies between these stakeholders. If the stakeholders are highly interdependent, there is the risk that if just one of the stakeholders does not follow the specified plan, it will have a negative impact on the plans for all of the other stakeholders as well. In this case, one should consider an approach to the problem in which the problem is solved for each of the stakeholders individually, taking the interdependencies to the other stakeholders into consideration by e.g. including different robustness measures. If, on the other hand, the stakeholders are not that interdependent, one can consider formulating a single model providing a solution to all stakeholders involved.

Focusing on the relevant decision is important for applicability. In the modeling step of the solution development, the relevant decision variables need to be identified. A common pitfall is that decisions are included in the model which will in reality be part of the operational doing and cannot be influenced by the planning decision, like e.g. the trip buildup for baggage transportation is not decided by the dispatcher who decides the transport destinations for each single unit.

11.3.6 Process dependencies

In addition to the stakeholders, the dependencies to other processes also have to be considered. If two or more processes are interdependent, solutions found to one process may have an influence on the solutions found to the other processes. The objectives of interdependent processes should be aligned and master data shared to the greatest possible extent.

Data enrichment and reuse can increase the impact of an application. Examples of enriched data can be any calculations during the determination of final solutions, e.g. the data processing for transfer baggage handling can be used to derive minimum and maximum connection times between apron position areas.

Often it is possible to reuse generated data in other contexts to improve neighboring processes.

11.3.7 Objectives

Having identified the stakeholders and the process dependencies of the problem, the next step is to identify both the *objectives of the problem* and the *project goal*. The objectives typically involve costs expressed either directly or indirectly and different quality measurements. With regards to the goal of solving the problem one should consider whether or not solutions to the problem are just seen as some kind of decision support or if the solutions actually specifies plans to be followed (depends on the role of the airport).

Many stakeholders with different needs and understandings of the processes and objectives can lead to contradicting objectives and perceptions of the solutions. This situation can complicate both the applicability and the acceptance of the application. For example, the management focus is on fulfilling airline and passenger preferences whereas the operators may also have, e.g. the workload for each team in mind. Problems with multiple objectives need specialized solution methods and restrict the choice of solution methods. The balancing of the different criteria is often a difficult task and should involve the stakeholders.

The introduction of the application can trigger a change of roles of the decision makers and is a typical project goal. The changed understanding of the roles can increase process quality and the impact of the application. Another possible project goal is the introduction of high quality suggestions to decrease workload of the dispatcher. The focus of the decision maker can be to ensure quality by checking consistency and reacting proactively in critical situations. In the best case, the bounds or constraints can be adjusted by manual interference when bottlenecks are determined. Examples can be the use of a temporary support team in peak periods. Models could provide information about bottlenecks or even suggest the change of constraints.

For many optimization problems the overall goal is to save costs. However, aspects of fairness and robustness are also likely to occur when considering real-life problems. In a multi-stakeholder environment as that of an airport a possible project goal is to make the decision support less dependent of individual planners and dispatchers. Knowledge about a given process should be available to all stakeholders, and the systems providing the decision support should be as general as possible.

For psychological reasons the objective often becomes very complex, since every stakeholder should be considered with his criteria. Some of the criteria will end with small weights and have almost no impact on the solution but including the criteria in the objective function is important for the acceptance of the application. Including all criteria also opens up for a different weighting in the future caused, for example, by a change in responsibility or business processes.

11.3.8 Optimality

Having defined the objectives and goals, it is necessary to specify what defines a good solution to the problem. In some cases, it is necessary to find the optimal solution in order to improve existing solutions, however, in other cases, finding near-optimal solutions might suffice. In tune with the three criteria of success described in Section 11.1 it is more important to find solutions that fit into the operational process, improve current practice and are accepted by the end users. If finding near-optimal solutions is much more efficient than finding the optimal solution, this is in many cases acceptable, especially if one is dealing with a very variable problem.

11.3.9 Data quality

An important aspect when dealing with real-world problems is the availability of data and the reliability of the available data. If the considered problem is highly dynamic but the data is only available in some kind of aggregated form with a level of detail lower than the dynamics of the problem, it is unlikely that solutions to the problem can be applied. Changing and missing data are a common problem in airport operations. The developed solutions and methods need to cope with the resulting high level of uncertainty to ensure applicability.

Furthermore, one should ensure that the level of detail in the provided solutions also match the uncertainty and the nature of the problem. If e.g. one is considering a highly uncertain problem but the solutions only consider time intervals which are longer than required by the dynamics of the problem, then the solutions provided will most likely never be applicable and continuously need to be changed.

The right degree of data resolution is often underestimated as an influence factor on the success of the application. A danger is that the applications model reality in too much detail. After feeding the models with data it can be seen that the uncertainty leads to unpredictable results in the too detailed view. For example, it is often not possible to predict the order of the arrival of the bags in the inbound baggage process since it is not known where in the container the bags are loaded and when they will be unloaded of the aircraft. Furthermore, all process times like transportation or unloading are only estimated and can change for every trip, e.g. due to weather conditions.

11.3.10 Automation

The *degree of automation* of the solutions describes the trade-off between manual decision making and system driven decisions. The degree of automation can be *manual* supported by changeable or confirmable suggestions, *semi-automated* supported by suggestions which will be transformed to decisions if the end user does not interfere or *automated* where the solutions are transformed in decisions with no or only limited interference possibilities by a decision maker.

Our experience is that for operational problems with a high degree of uncertainty and failure prone data sources, a human dispatcher is necessary for final responsibility. To have the overview the dispatcher needs data on the entire process and on the specific suggestions/decisions leading to a higher transparency. The optimal degree of automation depends on the specific setting. We observed major problems with operational solutions where changes of suggestions are not possible or time-consuming. The optimal degree for the automation is for most dynamic decisions in daily operations a continuous support by suggestions to react to changes in the environment. Depending on the properties of the application the solution methods need to respect that the solutions can be changed by the end user.

11.3.11 Deployment

Transparency is an important point for acceptance and is especially important with increasing participation of external partners. Transparency can be increased by visualization and traceability. The solutions can, e.g., be put into graphics so that the results can be easily interpreted by the end user. Furthermore, it is necessary that the impact of changes of the suggestions is measurable and can be studied, e.g. by comparing the objective function value and its components of the unchanged with the changed suggestions. Another important point is that the results and logfiles should be saved to provide reproducibility. Thus they can be used for further analysis. The results and intermediate steps need to be accessible. This is especially relevant for heuristics where it can be hard to include these requirements. The applications should be easy to control. The recognition of changes in the problem structure should be supported by the application or by a constant monitoring. An example is an increased solution time caused by higher problem complexity due to a growing traffic volume.

Furthermore, *maintenance* is an often neglected factor for acceptance in the development of models in research and practice. For companies an application should run successfully over years. We observed that models are often hard to maintain or experts are necessary for the adaptation. Without a constant support, applicability is lost during the lifetime of an application in many cases. The design of the applications should offer an easy check-up of the results. Furthermore, the master data needs to stay up-to-date and should not be part of the application and the model itself. Additionally, it is necessary to keep track of changing processes and the impact on the model.

11.3.12 Operational aspects in real-life

The above described operational aspects were considered when developing several successful projects at CPH and FRA. In the following sections two cases are presented in detail. The incorporation of the operational aspects in the development process is illustrated and shown at specific examples. The description of the cases aim to increase the understanding of the presented operational aspects. The two cases stand exemplarily for many other projects at CPH and FRA. They were chosen because of their significance on the scientific and industrial level. The operational aspects haven shown to be major success factors for reaching *applicability*, *acceptance* and *impact*. The described applications represent different problem types and are categorized differently in terms of the operational aspects.

11.4 Case one: Check-in counter allocation at Copenhagen Airport

In 2012, more than 23 million passengers departed from Copenhagen Airport (CPH). Check-in is usually the first process encountered by the passengers when travelling by air. By checking in, the passenger confirms to the airline that he or she actually has the intention to board the flight for which he or she has booked a ticket. Moreover, at check-in, the passenger has the possibility to choose, buy or change a seat, register bags, etc. Today the check-in can be performed in various ways; online, via self-service kiosks at the airport, or via traditional check-in desks where the passengers are assisted by representatives of the airlines. In most cases, check-in is performed before the passengers reach the security check. In Europe, it is usually the responsibility of the airport to provide available check-in counters, and the airlines must provide available representatives, typically by using so-called handling agents. For an illustration of the typical flow for departing passengers see Figure 11.4.

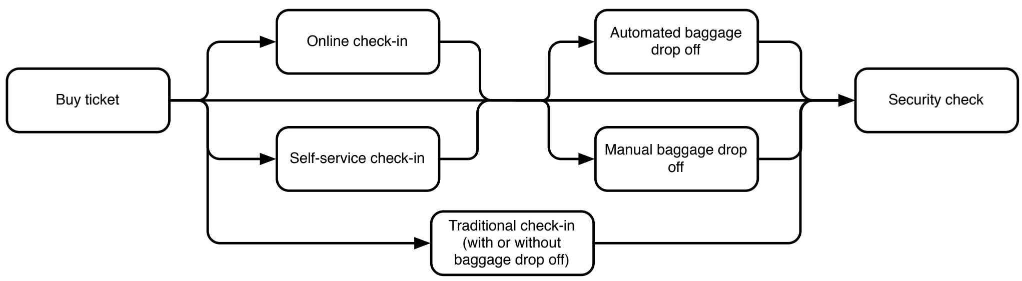

Figure 11.4: *The different possibilities for checking in and dropping off bags when travelling by air. The traditional check-in and the manual baggage drop off both require available check-in counters and staff.*

From 2001 to 2012 the number of passengers using CPH has increased by 29% and correspondingly the demand for check-in counters has also increased. Prior to 2010, CPH had 91 check-in counters divided between two terminals to handle international departures. All departures were handled by one of two handling companies and the 91 counters provided a sufficiently large capacity to handle all check-ins in an unproblematic manner. This situation allowed for a simple approach to the allocation of check-ins to check-in counters; each handling company was assigned a fixed subset of the check-in counters, and it was then up to the handling companies themselves to plan the usage of these counters. Typically, the handlers would manually perform this allocation largely on a seasonal basis and without considering the counter allocation of the other handlers.

By early 2010, the number of passengers and the demand for check-in counters had grown so much that the simple approach to the check-in counter allocation was no longer viable. At peak hours, the handlers did not have enough counters available and the flow in the terminals was unbalanced. One could experience situations where some parts of the terminals were congested by long queues blocking the flow while other parts of the terminals were almost empty. Moreover, the baggage inflow to the take-away belts behind the check-in counters was unbalanced leading to congestion and overload of the baggage handling facilities, ultimately causing major breakdowns of the entire operation. Each handler was primarily concerned about his own area, i.e. there was not a common view of the situation in the terminals, and the handlers did not see the need to communicate, unless if they wanted to request for permission to use some of the counters of another handler's area.

In order to address these problems and to ensure a robust and stable operation, CPH decided to take over the responsibility for the check-in counter allocation problem. CPH initiated the Check-in Optimization Project. The project was carried out in two phases. In the first phase, all the necessary data and parameters were identified and made available. Among other things, CPH conducted a comprehensive, systematic registration of the show-up profiles of the passengers, the processing times and the waiting times for each check-in group. To have a high degree of accuracy, all data were assessed in five minute intervals. Furthermore, CPH collected data from all operating handlers and airlines to ensure that the correct counter demand was used for each check-in group. Subsequently, data was analyzed and reconciled with the handlers and airlines. In the second phase, CPH implemented an optimization tool for solving the *check-in counter allocation problem.*

11.4.1 Problem definition

A check-in group is a group of flights (departures) that share the same check-in counters for check-in or baggage drop-off. A check-in group can either consist of a single flight (dedicated area) or all flights of a specific or multiple airlines (common use area). Each check-in group is associated with a given counter demand, and the check-in counter allocation problem is then the problem of allocating check-in groups to available check-in counters while ensuring that congestions in the flow of the terminals are avoided and that the load of the facilities for handling outbound baggage is as uniformly distributed as possible. Finally, in order to maintain a good relationship to its primary customer, the airlines, CPH also has as goal that preferences, to where the different check-in groups are allocated, are respected to the greatest possible extent.

11.4.2 Operational aspects

The check-in counter allocation process is currently owned by CPH and in principle the problem appears on the strategical, tactical and operational levels. Though the focus at the strategical and tactical levels is different, CPH has chosen an approach in which the same variant of the problem is used to perform both capacity analyses and operational planning. CPH also is responsible for the operational variant of the problem, but currently only acts as process facilitator on the day of operation.

The domain dynamics are very low, as the constraints of the problem are rarely changed, and the problem dynamics are medium, in the sense that operational plans have to be made every two weeks.

The stakeholders of the problem are the airport, the handling agents and the airlines. The airport is stakeholder partly because the check-in counters and terminal areas are the responsibility of the airport, partly because the duty managers of the passenger department in the airport have the responsibility to overlook the terminal and transit areas and ensure that everything is working as planned. The handling agents are the ones performing the check-in and they need to know in which counters to handle which check-in groups. The airlines are only an implicit stakeholder, as they are paying the handling agents to perform the check-in process, but in the solutions found to the problem, preferences for both counters and opening patters expressed by the airlines, must be taken into consideration.

When solving the check-in counter allocation problem, the primary objectives are to satisfy the counter preferences expressed by the airlines to the greatest possible extent and to spread out the passenger flow and load on the baggage handling facilities as much as possible in order to avoid congestion in the terminals and overload of the baggage handling facilities.

The data needed to solve the problem consists of layout of the terminals, airline preferences, (expected) opening patterns for each of the check-in groups, forecasted queue lengths, and forecasted inflow of baggage per check-in group. For the opening patterns (counter demand), queue lengths and baggage inflow, data is relatively accurate and given per five minute interval. Solutions to the check-in counter allocation problem are published to the airlines, handlers, and airport duty managers as soon as they are made. To constantly improve the generated solutions and the utilization of the check-in counters, a follow-up is made for all published counter allocation plans. The follow-up compares the actual opening of counters with the planned opening of counters and logs all changes made to the published plan.

11.4.3 Modeling

Even though OR is central within airline and airport optimization, the relevant literature for the described check-in counter allocation problem is rather sparse. The problem was originally presented in a paper by Hon [1996]. The paper presented a heuristic to solve a stochastic version of the problem where counter demands can vary and the impact on resources is simulated. The presented problem is closely connected to the problem of CPH, but the solution approach differs on the key point that in the CPH case, it is assumed that all data is accurate and given beforehand.

The problem of CPH has also large similarities to the *adjacent resource scheduling* problem presented by Duin and Sluis [2006]. This problem works with deterministic data. The motivating example is counter allocation in airports. Mathematical formulations for the problem are presented, and the decision version is shown to be strongly $\mathcal{NP}$-complete.

Yan et al. [2004] develop an integer programming model to assist airport authorities to assign common use check-in counters. The goal is to minimize the total walking distances of passengers in monthly planning. The problem presented does not include considerations on baggage or queueing, and the counter demand is assumed to be constant for each check-in group. The case is further explored by Yan et al. [2005] who develop a model that minimizes total inconsistencies in common-use counter assignments with a variable number of counters. The model is formulated as a zero-one integer program and solved heuristically. Another heuristic approach is presented by Wang Yeung and Chun [1995] who use a genetic algorithm (GA) approach. Quality of a possible solution is given by a *fitness measure*. In principle, the fitness measure can incorporate the conditions from CPH (number of people waiting in line, number of bags check-in, and airline preferences), however adjacency criteria are not refined enough in the model.

A large amount of research has been put into determining the actual counter demand and most of the mentioned papers are concerned with minimizing the number of resources (counters) used. van Dijk and van der Sluis [2006] present an approach in which the number of counters needed for each check-in group is determined as part of the optimization, and afterwards the maximum number of counters used at any time is minimized. Equivalently, Tang [2010] present a network model for the optimization of common use check-in groups with the goal of minimizing the number of counters required for daily operations. Other work on optimizing the number of counters needed includes the most recent paper on single flight check-in queuing estimation by Parlar and Sharafali [2008], the paper by Park and Ahn [2003] on passenger arrival, and finally the simulation paper on determining the counter usage by Chun and Mak [1999].

The check-in counter allocation problem can also be formulated with a set packing based integer model, where each check-in counter is represented in a sequence of time intervals and the solution is constrained by a set of generalized upper bound constraints governing the allocation of each check-in group. The problem is a multi-criteria decision problem and the developed objective function is a weighted sum of the different criteria.

The number of possible allocation options for a given check-in group may potentially be very large, however, for practical purposes, several restrictions must be taken into consideration. First of all, the number of counters required to handle the given check-in group is minimized. Secondly, only allocation options in which the check-in group is allocated to adjacent counters are considered. Finally, the considered allocation options should be as compact as possible, and not have "holes" in them. An allocation option fulfilling these operational requirements is said to be operationally valid.

Airlines and handling companies prefer the allocation of the same counters every day. However, including such a restriction limits the flexibility and the possibilities for the airport to control the usage of the check-in counters and the distribution of queues and inflow and baggage in the terminals. Since customer satisfaction is an important area of focus, the airport wants to satisfy the preferences of the airlines to the greatest possible extent.

As a compromise *allowed* and *preferred* check-in counters are introduced. For a given check-in group a limited subset of *allowed* check-in counters is specified. The *preferred* check-in counters are a subset of the allowed check-in counters. The set of allowed and preferred check-in counters is defined in a negotiation between the airport and airlines and is fixed prior to solving the problem. The objective is to maximize the allocation on preferred check-in counters.

11.4.4 Implementation

The mathematical model was implemented and solved using Gurobi in a C++ environment. The traffic schedule is fetched from CPH Airport Operation Database and all input parameters are stored and maintained in a Microsoft Access database.

Because of the many practical restrictions to the model, the final solution space is very limited. As a consequence, the model can be solved to optimality in less than two seconds using a standard MIP solver by enumerating all variables.

The tool does not yet have real-time and automated capabilities and it requires a planner to actively start the generation of allocations. Any changes to the allocation after publication (typically due to flight cancellations, new flights, delays or by airline request) are updated manually.

The check-in counter allocation tool has been in operation at CPH since May 2010 and recently CPH has engaged with an external software developing company in order to develop a real piece of software for the check-in counter allocation tool.

11.4.5 Feedback loops

The optimization tool was deployed in May 2010, and has been running since then. The tool is used on a weekly basis to provide the airlines, handling companies and staff of the passenger department with check-in counter allocation plans.

To ensure the transparency of the new check-in counter allocation process, all check-in counter allocation plans are made easily accessible to the duty managers of CPH, the handling companies and the airlines. Employees can view the plans online as PDF files with both Gantt charts and timetables for each flight. Furthermore, a web-based application has been introduced, showing the latest version of the allocation and allowing selected users to make manual changes by drag-and-drop of check-in groups. For the staff at the check-in counters, it is important that the check-in counter allocation is always available and therefore a small room with a monitor and a printer was established in Terminal 2 of CPH.

Since the counter allocation is created up to a week in advance and based on forecasted passenger numbers and show-up profiles, the actual counter opening patterns on the day of operation may differ somewhat from the plan as the handling agents adapt the plan on the fly to account for e.g. large groups arriving early for check-in. CPH collects electronic data on the counter usage, including counter opening/closing times and time stamps for check-in bags, and in combination with waiting time measurements and passenger surveys, the data is used to analyze and dynamically improve the counter utilization by updating relevant parameters in the allocation tool (e.g. process times and show-up profiles).

Furthermore, CPH actively uses the insight in an effort to make the airlines and handling agents change non-optimal opening patterns agreed in their contracts. Often, airlines request a check-in service for a fixed time for a certain aircraft type, e.g. 3 counters from 1.5 hours until 30 minutes before departure for an aircraft with 180 seats. This gives an average process time of one minute per passenger if the flight is full. However, such an opening pattern may lead to long waiting times if in fact the passengers show up early for check-in. Continuing the example above, if the show-up profile indicates that passengers start showing up two hours before departure, it may be beneficial to instead open 2 counters from 2 hours until 30 minutes before departure. This pattern contains the same number of man hours, but the handling company only needs two check-in agents instead of three, the passenger waiting time decreases and CPH gains a check-in counter to allocate to a different check-in group, which can be essential in peak periods where capacity is scarce.

Since 2010, CPH has successfully suggested changes to opening patterns similar (and more complex) to the example above to a number of airlines, resulting in a better utilization of the check-in counters and ultimately a better passenger experience.

11.4.6 Measures of success

The development of the check-in counter allocation tool has to a large extent exploited the interdependencies between the development process and the operational process. Initially, the set of allowed counters was deliberately limited to only a very few counters centered around the preferred counters for each check-in group. This enabled CPH to deploy the tool without imposing too many changes to the way the check-in groups had been allocated prior to 2010, which led to a relatively trouble-free *acceptance* of the new solutions by both the handling agents and the airlines. Throughout the development of the tool, it was important for CPH to maintain a positive dialogue with both the handling companies and airlines. Alongside the development of the check-in counter allocation tool, CPH organized several orientation meetings for the handling companies and airlines, and in general CPH paid a lot of attention to the feedback and wishes given by the handlers and airlines.

The developed check-in counter allocation tool has had a huge *impact* on the daily operation of the check-in areas. It has led to a transparent, consistent, and fair distribution of the available check-in counter capacity at CPH. Furthermore, a much better relationship between CPH, the airlines, and the handling companies have been established, and in general both the airlines and the handlers are satisfied with the new setup. Furthermore, the average process time per passenger has been reduced and as result, CPH has seen a decrease in the overall waiting time for the passengers, a reduction in the size of the queues and a better flow in the terminals in general.

Since 2010 the sets of allowed and preferred counters have then gradually been adjusted. Concurrently, with the handling companies and airlines getting used to that CPH is responsible for the counter allocation, CPH has gradually expanded the set of allowed counters for each check-in group, thereby increasing the number of possible solutions. In all cases, the developed application has been able to solve the problem and it is thus considered *applicable*.

11.5 Case two: Baggage handling at Frankfurt Airport

The case described in this section covers the development of optimization based decision support applications for the control of baggage handling at the infrastructure of Frankfurt Airport. FRA is with more than 56 million passengers per year one of the largest airports in the world and has a high degree of transfer passengers which makes an efficient baggage handling even more important in terms of quality and competitiveness.

The SITA baggage report shows impressively the need for improving baggage handling quality [SITA, 2013]. In total over all airports 8.83 bags per 1000 passengers were mishandled in 2012. The tasks performed at the airports have major impact on the baggage handling quality e.g. mishandled bags due to the transfer baggage process. In the last years, cost pressure and raised competition between airports has forced airports and ground handling companies to implement more efficient processes. This has caused a strong need for better decision making and support. In the past, the decisions were often only supported by simple rule based suggestions and in most cases a human dispatcher took the final decision manually.

The described project consists of three different applications which were implemented independently by a team of internal OR analysts/consultants. Each of the applications is concerned with decisions on the operational level. The different parts of the applications were introduced over a period of six years beginning with transfer baggage handling in 2008 and ending with the outbound baggage handling in 2013. The development and the implementation of each of these applications lasted about one year.

11.5.1 Problem definition

Figure 11.5 gives an overview of the different baggage handling processes at an airport. Baggage handling consists of handling the bags of arriving passengers (*inbound baggage*), transferring passengers (*transfer baggage*) and departing passengers (*outbound baggage*). Outbound baggage combines the baggage streams of local check-in and transfer baggage. The core of baggage handling is the baggage handling systems (BHS) for automatically handling the bags on an infrastructure level in the terminal buildings. The BHS often consists of a connected conveyor belt network and storage facilities.

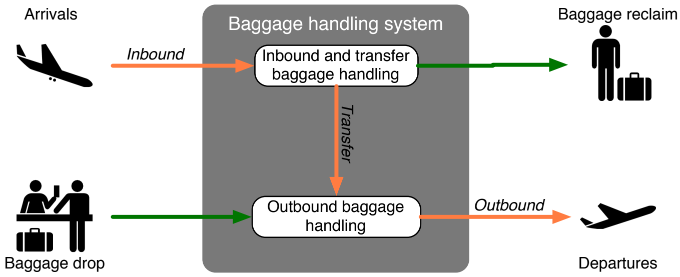

Figure 11.5: *Connections of the different parts of baggage handling.*

The considered project developed decision support applications for assigning flight events to baggage handling resources on the day of operation. A description of the baggage handling processes at Frankfurt Airport can be found in [Kipper, 2012]. The developed applications are used for assigning baggage carousels and infeed stations for the transfer baggage to arriving aircraft, and handling facilities for the outbound baggage to departing aircraft. These decisions have major impact on handling quality and efficiency. The final decision is in the responsibility of a dispatcher who is supported by suggestions of the optimization applications. For uncritical events, decisions can be activated without confirmation of the dispatcher. The applications help decreasing the workload for the dispatcher and support them to focus on critical events. Especially, in situations with major disruptions, the applications help adjusting schedules quickly to the new situation. The objective of the applications can be summarized as a combination of quality (e.g. less missed connections, robustness and passenger satisfaction) and efficiency goals (e.g. short transportation times).

Inbound baggage handling

For the inbound baggage for each arriving aircraft a *baggage carousel* has to be chosen. The most important objectives are:

- airline preferences, e.g. proximity to lost and found from airline
- avoidance of simultaneous handling of several flights at one baggage carousel
- avoidance of simultaneous handling of large flights at neighboring baggage carousels
- robustness of schedules by avoiding simultaneous handling caused by delays or early arrival
- balanced use of baggage claim halls and baggage carousels
- operational suitability of baggage carousels
- stability of solutions

Transfer baggage handling

For each trip with containers loaded with transfer bags from an arriving flight, the infeed station into the BHS has to be selected. The most important objectives for the transfer baggage process are:

- minimize missed connections
- robustness of decisions (buffer in capacities and connecting time)
- minimize transportation time (at the apron and in the BHS)
- balance resource use between infeed areas

Outbound baggage handling

For outbound baggage handling the handling start time and the handling facility have to be determined. The objectives of the problem are:

- minimize distance between handling facility and aircraft parking position
- robustness in handling (all bags should be handled before departure)
- airline preferences
- stability of solutions

Furthermore, important soft constraints are the capacity of working stations (limiting the number of flights) and container parking positions (depending on the sorting criteria of a flight).

11.5.2 Operational aspects

The airport is process owner of the baggage handling processes. The decisions have to be carried out by the staff at the handling facilities and infeed stations. Furthermore, the other stakeholders in the process such as baggage transportation have to respect the decisions. The applications are used in 24-hour planning for preparing operations and more important for the final assignment decision during operations. The degree of domain dynamics can be seen in the changes of the business processes. During the project, several changes occurred which demanded changes in the underlying objectives and constraints of the model. Examples were new objectives or security regulations restricting the choice. The operational decisions should be made as close as possible to the final arrival or departure of the aircraft since the problem dynamics are very high. Furthermore, the decisions depend on an enormous number of very dynamic data e.g. arrival times, parking positions or connection information. The decision of one flight can impact the handling of other flights since handling resources are limited.

The qualifications of the decision makers were one of the challenges in the project. The dispatchers are part of the physical baggage handling and often have no higher education. Therefore they do not understand the details of a sophisticated optimization application. It was important to convince them with a high degree of transparency and a constant involvement in the development process. Besides staff from the airport operator, the process involves many different stakeholders such as airlines, passengers, baggage transportation and baggage loading. Furthermore, other departments within the airport and public authorities were part of the processes. The needs of the different stakeholders are respected in the objective and the impact on their processes is considered in the generated solutions. The applications have large impact on other processes and the arising interdependencies were considered on an IT architectural level. Furthermore, parts of the calculations are provided to other stakeholders to increase their understanding of the processes and harmonizing the business processes.

The objectives of the problem consist, as described above, of several efficiency and quality goals. The project goal was to increase quality and to support the dispatchers. The aspired quality improvement was respected when defining the objective. The models are solved to optimality with a MIP solver but one needs to consider that the decisions are taken on an aggregated level and problem complexity is reduced thereby. The aggregation takes the data quality into account. The decisions from the dispatchers set the frame for the operational handling and leave flexibility in the operational handling. The flexibility is used to balance the workload and to bring the events in the right order during the handling processes.

The applications are used as decision support. They are designed in such a way that they adapt to changes by the dispatchers. Additionally, the applications enable automatic decision making. At Frankfurt Airport the solutions of the applications will be automatically transferred into decisions if the dispatcher does not interfere. Furthermore, a certain amount of events is decided automatically. The degree of automation was slowly increased to keep the credit of dispatchers in the application. An important property of the application is that the dispatcher can still change all suggestions easily.

The core of the project team for each project consisted of a project management of the baggage infrastructure department and OR experts from the internal IT department of Fraport. In the whole process, the dispatcher and their supervisors and as well the management were included. The objective and the frame conditions were regularly updated and agreed. It was important to align the objective criteria with the management. Therefore, the results and the impact of specific settings of the objective criteria were visualized. Furthermore, the solutions of the optimization based applications and the decisions of the dispatchers were compared visually and in numbers. Simulations generating aggregated output and tools with a GUI similar to the real dispatching systems were helpful in this process. During the discussions, important new process insights were generated and the applications were adapted. The objective criteria and the constraints were designed flexible so that the feedback of dispatcher and management can be included quickly.

The description of the projects was continuously improved. This process was supported by a quick prototyping approach which helped visualizing the results. A key part is to identify the core of the decisions and only include decision relevant factors.

11.5.3 Modeling

The model in the core of the problems is a mixed-integer assignment problem with additional soft constraints and different weighted objective criteria. The objectives consist of direct cost, e.g. transportation time, missed connections and soft constraints, e.g. capacity handling and balancing of workload. The soft constraints depend on the combination of the handling of several flight events. All possible assignment decisions of flights to handling resources are generated in a preparation step. When solving the model, the optimal feasible combination of the assignments is determined.

Many of the operational constraints can be expressed as soft constraints and the violation is penalized in the objective. This helps dealing with incomplete data and to be sure to get a feasible solution, e.g. capacity and usage cannot be calculated accurately and therefore the violation of the capacity constraint is allowed. The concept of soft constraints as part of the objective showed to represent the real world very well.

The general model is flexible and has been adjusted several times in the past years due to changes in the business processes, e.g. by deactivating constraints or introducing new constraints. Additionally, the data processing has been adjusted to new needs.

Although, the applications for the processes of inbound, transfer and outbound baggage handling are based on the same general model, the models are adjusted to the specific needs of the processes. The models differ in the objectives and the modeling of the handling capacities.

Inbound baggage handling

The general model used in FRA is presented in [Barth and Böckmann, 2012] and [Barth, 2013]. The model considers the above identified direct costs such as airline preferences and soft constraints such as avoiding simultaneous handling or balancing the workload which are weighted in a linear objective. The implementation at FRA comprises 5 direct cost terms and more than 10 soft constraints. Based on the model the optimal time horizon and point time of the decision is determined in [Barth and Böckmann, 2012]. The sensitivity of the model to changes of the weights of the different criteria and of the problem characteristics such as airport layout is studied in [Barth, 2013].

A related model was presented by Delonge [2012]. The approach is based on constraint programming and focuses on load balancing between the baggage carousels. Frey et al. [2012] present a greedy heuristic for balancing the workload and minimizing passenger walking distances. The model includes the underlying stochasticity of the data.

Transfer baggage handling

The model for transfer baggage handling is described in [Barth et al., 2013]. The model combines the above identified objectives. The capacity of the infeed areas is defined in a step-wise soft constraint. The concept of soft constraints was chosen since the duration of the baggage handling and the arrival order of the bags cannot be exactly determined. The step-wise approach reserves capacity for unforeseen events and avoids technical breakdowns by a constant use at the capacity limit. Furthermore, the process time of bags with short connecting times is minimized by choosing the optimal infeed area. The value of the saved process time decreases with increasing buffer between the available connecting time and the needed process time.

A closely related problem was presented by Kiermaier and Kolisch [2012]. The problem is as well modeled as a MIP. The objective is to minimize the handling delay. The model contains a constraint which defines a limit on the number of overload periods. As solution approach a genetic algorithm is proposed. Clausen and Pisinger [2010] consider the related routing problem for short transfer baggage. They propose a weighted greedy algorithm for dispatching the vehicles for the transport of the bags.

Outbound baggage handling

The topic of outbound baggage handling at FRA is covered by Barth and Pisinger [2012]. In this work two different solution approaches are compared. On the one hand a heuristic based on a greedy randomized search procedure and on the other hand decomposition approach based on the above mentioned assignment model are presented. The decomposition approach splits the decision into the aspect of assigning a handling facility and of assigning the handling start time to a flight.

The outbound baggage handling was originally studied by Abdelghany et al. [2006]. They consider a case with exclusive assignment of handling resources to flights. The objective of the problem is defined based on preferred resources and a balancing of the workload. The problem is solved with a activity selection problem algorithm which considers ordered flight events sequentially. Frey and Kolisch [2012] propose a column generation approach to reduce the maximal workload at the handling facilities while respecting several operational constraints. Asco et al. [2011] describe a similar problem for scheduling outbound baggage on exclusive handling resources. They propose several constructive algorithms which consider single flight events consecutively. The choice of the algorithm depends on different factors such as the airport layout and flight density. Based on this work an evolutionary solution algorithm was proposed in [Asco et al., 2012].

11.5.4 Implementation

It turned out that all models can be solved with a standard MIP solver. In a first step, the data is prepared and collected. Afterwards all feasible assignments of flight events to resources are generated. The MIP models then select the optimal solution according to a linear combination of different objectives. The generation of the assignments is implemented in .net C# and the model is built with IBM Ilog OPL and solved using IBM ILOG Cplex. Clearly, the solution approaches are not exact but heuristic since many issues are generalized or aggregated e.g. the calculation of capacity use. The solutions were included in the GUI of the existing dispatch systems. The process of decision making was only slightly changed by a more reliable and advanced decision support.

11.5.5 Feedback loop

The model and the objective criteria were discussed regularly with all stakeholders during workshops and regular process and solution reviews. Components of the review and workshops of the feedback loop can be reused for maintenance. The models were adapted and detailed several times until the final operational use. The results were visualized and presented in an easy understandable way.

11.5.6 Measures of success

FRA reported growing passenger numbers and baggage volume since the project start. With the help of the applications it was possible to keep a high handling quality and to avoid the need of additional dispatchers (constant personnel cost). The introduction of additional dispatchers arise the need to split the decisions which is likely to result in a loss of quality since artificial borders and constraints are introduced to the problems. Furthermore, the dispatch strategy over all shifts was standardized and the applications helped to discuss strategies by measuring the solutions in a mathematical model. This has led to a high process transparency. Additionally, all decisions can be referred and compared to the defined model and the optimal solution. The results and calculations of the model are used at other places of the baggage handling processes for improving baggage handling quality e.g. visualizing critical transfer containers.

The developed applications help improving decision making and are an important basis for additional analytic applications and projects. It showed that especially the deployment process including the calibration of the models was time consuming and took considerably longer than in comparable software development projects without decision support. Including the soft factors and operational aspects during modeling were key factors for project success. The use of soft constraints allowed to generate applicable solutions which are accepted by all stakeholders. The high degree of uncertainty inhibits the use of hard operational constraints e.g. for capacities. Furthermore, the real execution can only be estimated and cannot be modeled and considered in detail. Hard constraints assume a certainty which does not exist. Additionally, the solutions of the applications can be used in different phases of decision making. Furthermore, they help understanding the way of decision making and can lead to a standardization of decision making.

In general, we observed an increasing acceptance of optimization based applications due to the project success. For each of the sub problems it can be shown that the objective of the model can be improved by 10%. The improvement can mean either a quality or efficiency increase. The reason is a better combination of the different objectives and consideration of the dependency between the different events.

For the process of decision making, important insights can be gained. It can be theoretically proved that fixing the decisions for single flights before the landing results in a major loss in quality. Furthermore, it is necessary to look at all flights which will arrive in the next 45 minutes (or larger intervals). This result shows that manual decision making which considers only a limited number of events at one time and tends to decide early is not optimal.

In terms of the described success measurements it can be stated that *applicability* was proved by solving the problems at Frankfurt Airport. The models produce in all situations including disrupted days e.g. major delays satisfying results. The results can easily be transferred in real-world decisions or be applied in an automated way to the business process. The *impact* of the application was described above and can be summarized by an increased handling quality while bag volume increased with constant personnel. *Acceptance* from the dispatchers is continuously measured. The confirmation rate is about 80% for all considered flights but the overall acceptance is very good on all hierarchical levels. The confirmation rate is mainly driven by the management strategy since the dispatchers should stay involved in the process and have the final responsibility for operations. The management itself is satisfied with the applications and supports the roll-out in related areas. Furthermore, the project has established an environment for an internal development of decision support applications based on OR.

11.6 Concluding remarks

The presented article shows by the example of two case studies at two different airports how project success can be reached when OR applications are applied to airport operations. Based on the presented operational aspects the most important problem properties can be determined. A thorough identification of the operational aspects will support a successful development by taking care of the relevant problem characteristics and often forgotten real-world requirements.

During the development process, the consideration of the operational aspects needs to be included in practice and research projects. For practical projects all involved parties need to be aware of this step. This is especially important if external partners are involved.

The presented operational aspects are part of the subjective view of the authors but many discussions with other practitioners both from academia and from other airports have shown that these factors represent the common sense at airports. Discussions with academics have often showed the absence of awareness for some of the mentioned problems. We want to encourage the scientific community to include these aspects as a critical part of their work. There still exists the danger that the gap between research and practice is even growing and the field of OR looses the foundations in the field of airport operations.

It will be an ongoing challenge for the OR community to solve the different problems and support the airports with appropriate approaches. Most of the described problems are unique in their structure and hard to solve. Furthermore, it will be a critical point for the airport industry that the different decision problems are combined and that the dependency between these decision problems are considered. For further research projects in the area of airport operations it will be essential to consider the issue of *applicability.*

During our work we have learned that for all applications, the principle "keep it as simple as possible" is an important success factor. Examples are the work with mixed-integer programming and the use of standard solvers. This approach is easy understandable and can be brought into business quickly. For many problems first applications can be generated by simplified solution models. *Acceptance* can be increased by an early involvement of the stakeholders and providing fast first exemplary results. Including the different criteria of the stakeholder can lead to complex objectives and contradicts the principle to keep things simple. Therefore, it is important to make a trade-off between the competing goals. Furthermore, it has been shown that mathematical optimality is not as important as in the scientific community and improvements of the existing situation are accepted as goal of a project. Optimality is often even not measurable in reality due to data uncertainty and the unknown actual behavior on the microscopical level.

Bibliography

Abdelghany, A., Abdelghany, K., Narasimhan, R., 2006. Scheduling baggage-handling facilities in congested airports. Journal of Air Transport Management 12 (2), 76–81.

Asco, A., Atkin, J. A. D., Burke, E. K., 2011. The airport baggage sorting station allocation problem. In: Proceedings of the 5th Multidisciplinary International Scheduling Conference. pp. 419–444.

Asco, A., Atkin, J. A. D., Burke, E. K., 2012. An evolutionary algorithm for the over-constrained airport baggage sorting station assignment problem. In: Simulated Evolution and Learning. Springer, pp. 32–41.

Barth, T., 2013. Optimal assignment of incoming flights to baggage carousels at airports. Tech. Rep. Report 5, DTU Management Engineering.

Barth, T., Böckmann, F., 2012. Baggage carousel assignment at airports: Model and case study. In: International Conference for Airport Operations Management in Munich. pp. 27 – 30.

Barth, T., Holm, J. T., Larsen, J. L., 2013. A model for the transfer baggage handling at airports. Tech. rep., DTU Management Engineering.

Barth, T., Pisinger, D., 2012. Scheduling of outbound luggage handling at airports. In: Klatte, D., L, thi, H.-J., Schmedders, K. (Eds.), Operations Research Proceedings 2011. Springer, pp. 251–256.

Böckmann, F., Schäfer, A., Barth, T., 2012. Optimization solutions at frankfurt airport. In: International Annual Conference of the German Operations Research Society in Hannover. p. 196.

Chun, H. W., Mak, 1999. Intelligent resource simulation for an airport check-in counter allocation system. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 29 (3), 325–335.

Clausen, T., Pisinger, D., 2010. Dynamic routing of short transfer baggage. Tech. rep., DTU Management Engineering.

Corbett, C. J., Overmeer, W. J. A. M., Van Wassenhove, L. N., 1995. Strands of practice in or (the practitioner's dilemma). European journal of operational research 87 (3), 484–499.

Delonge, F., 2012. Balancing load distribution on baggage belts at airports. In: International Annual Conference of the German Operations Research Society in Hannover. p. 196.

Duin, C. W., Sluis, E. V. D., 2006. On the complexity of adjacent resource scheduling. Journal of Scheduling 9 (1), 49–62.

Fortuin, L., Zijlstra, M., 1989. Operational research in practice: Experiences of an or group in industry. European journal of operational research 41 (1), 108–121.

Frey, M., Kiermaier, F., Kolisch, R., 2012. Assignment of local baggage streams to claiming carousels at airports. In: International Conference for Airport Operations Management in Munich. pp. 28 – 34.

Frey, M., Kolisch, R., 2012. Heuristic procedures for planning the outbound baggage handling process. In: International Conference for Airport Operations Management in Munich. pp. 19 – 22.

Haneyah, S. W. A., Schutten, J. M. J., Schuur, P. C., Zijm, W. H. M., 2012. Generic planning and control of automated material handling systems: Practical requirements versus existing theory. Computers in industry.

Hew, P., 2007. The art and science of practice: Direct advisors vs. tool developers and implications for practice. Interfaces 37 (3), 279–284.

Hildebrandt, S., 1980. The changing role of analysts in effective implementation of operations research and management science. European Journal of Operational Research 5 (6), 359–365.

Hildebrandt, S., 1981. Implementation of the bottleneck of operations research: the state of the art. European Journal of Operational Research 6 (1), 4–12.

Hon, W. C., 1996. Scheduling as a multi-dimensional placement problem. Engineering Applications of Artificial Intelligence 9 (3), 261–273.

Hounsgaard, K. D., Justesen, T. F., 2012. Implementing an improved chute allocation at copenhagen airports. In: International Conference for Airport Operations Management in Munich. pp. 51 – 55.

Kiermaier, F., Kolisch, R., 2012. Planning the transfer baggage handling at airports. In: International Conference for Airport Operations Management in Munich. pp. 23 – 26.

Kipper, U., 2012. Baggage management and infrastructure.

Liberatore, M., Luo, W., 2013. The art and science of practice: Comparison of technical and soft skill requirements for analytics and or professionals. Interfaces 43 (2), 194–197.

Murphy, F. H., 2001. The practice of operations research and the role of practice and practitioners in informs. Interfaces 31 (6), 98–111.

Murphy, F. H., 2005a. The art and science of practice: Elements of a theory of the practice of operations research: A framework. Interfaces 35 (2), 154–163.

Murphy, F. H., 2005b. The art and science of practice: Elements of a theory of the practice of operations research: Expertise in practice. Interfaces 35 (4), 313–322.

Murphy, F. H., 2005c. The art and science of practice: Elements of a theory of the practice of operations research: Practice as a business. Interfaces 35 (6), 524–530.

Park, Y., Ahn, S. B., 2003. Optimal assignment for check-in counters based on passenger arrival behaviour at an airport. Transportation Planning and Technology 26 (5), 397–416.

Parlar, M., Sharafali, M., 2008. Dynamic allocation of airline check-in counters: A queueing optimization approach. Management Science 54 (8), 1410–1424.

Ranyard, J. C., 1995. Supporting real decisions: A review of or practice in the uk. European journal of operational research 87 (3), 474–482.

Rönnqvist, M., 2012. Or challenges and experiences from solving industrial applications. International Transactions in Operational Research 19 (1-2), 227–251.

Schneeweiss, C., 1987. On a formalisation of the process of quantitative model building. European journal of operational research 29 (1), 24–41.

Semini, M., 2011. Applicability of operations research in manufacturing logistics. Ph.D. thesis, Norwegian University of Science and Technology.

SITA, 2013. Baggage report 2013. Web.
URL http://www.sita.aero/surveys-reports/industry-surveys-reports/baggage-report-2013

Tang, C.-H., 2010. A network model for airport common use check-in counter assignments. Journal of the Operational Research Society 61 (11), 1607–1618.

Tilanus, C. B., 1985. Failures and successes of quantitative methods in management. European Journal of Operational Research 19 (2), 170–175.

van Dijk, N. M., van der Sluis, E., 2006. Check-in computation and optimization by simulation and ip in combination. European Journal of Operational Research 171 (3), 1152–1168.

Wang Yeung, B. K., Chun, H. W., 1995. Check-in counter allocation using genetic algorithm. EXPERSYS, 197–202.

Yan, S., Chang, K. C. K.-C., Tang, C. H. C.-H., 2005. Minimizing inconsistencies in airport common-use checking counter assignments with a variable number of counters. Journal of Air Transport Management 11 (2), 107–116.

Yan, S., Tang, C.-H., Chen, M., 2004. A model and a solution algorithm for airport common use check-in counter assignments. Transportation Research, Part A (Policy and Practice) 38A (2), 101–125.

Part III

Appendix

Appendix A

Tactical Taxiway Route Allocation at Copenhagen Airport

Tor Fog Justesen[*†] Claudia Munk Billing[*] Jesper Larsen [†]

[*]Airport Optimization, Copenhagen Airports A/S
Lufthavnsboulevarden 6, 2770 Kastrup

[†]Department of Management Engineering, Technical University of Denmark
Produktionstorvet, Building 426, DK-2800 Kgs. Lyngby, Denmark

Contact: Tor Fog Justesen (+45 23 24 62 47 / tor.justesen@cph.dk)

Abstract In this paper we address the problem of routing arriving and departing aircraft through the taxiway network of an airport. We formulate the problem as a set packing model with a resource-time interval based constraint system and develop an LP based heuristic to solve it. To illustrate the usage of the model and the solution method, we consider a case study from Copenhagen Airport. In the case study the operational consequences of moving the start-up positions for selected stands are analyzed. The analysis considers solutions to the taxiway route allocation problem for two different scenarios: Scenario A, where the start-up positions are as today and Scenario B, where the start-up positions have been changed. The scenarios are compared by evaluating a set of key performance indicators including the number of unallocated aircraft, the amount of delay needed to provide a conflict free routing of the aircraft, and the overall utilization of the individual taxiways.

Keywords: *airport operations*, *taxiway route allocation*, *set packing*

A.1 Introduction

According to [EUROCONTROL, 2010] the number of flights in Europe will almost double between 2009 and 2030 and the primary limitation to the future air traffic will be the capacity at the airports. Efficiency in critical airport operations will hence be more and more important in the near future. One of the operations which affect the capacity of an airport is the routing of aircraft through the taxiway network.

A taxiway network consists of a set of taxiways which are the roads connecting runways with aircraft stands, hangars, terminals and other facilities. The taxiways are primarily used for taxiing aircraft; arriving aircraft use the taxiways to taxi in from the runway to their arrival stands, and departing aircraft use the taxiways during push-back and to taxi out to the runway for take-off.

Already today, many airports are experiencing congestions in the taxiway network and in the surrounding terminal areas. Besides creating a bottleneck in the airport operation, taxiway network congestion increases the time each aircraft spends in the network, leading to an increase in the fuel costs for the airlines and the carbon footprint of the airport (see e.g. [Ravizza et al., 2012]).

The *taxiway route allocation problem* is the problem of routing all arriving and departing aircraft through the taxiway network while minimizing the overall taxitimes and the number of potential conflicts where two or more aircraft have to hold back for each other. The taxiway route allocation problem is often perceived as a strictly operational problem due to the high degree of uncertainty associated with the parameters of the problem (arrival and departure times, runway entry and exit points, aircraft speed, etc.). However, as airports are being challenged on their capacity, the strategical and to some degree also tactical variants of the problem are becoming an important part of airport capacity planning.

At the strategic level, one typically considers a long-term planning horizon (e.g. from several months to several years) and the primary focus is on assessing capacity in relation to forecasted resource demand. In planning for the future, airports must evaluate the infrastructural capacity of not only the existing taxiway network, but also any modifications to the network they are intending to make. Assessing the capacity of a taxiway network thus involves determining the maximum number of aircraft that can be routed through the network over a given time horizon.

The tactical-level variant resembles the strategic-level variant; however, at the tactical level one typically considers a short-/medium-term planning horizon (e.g. from a couple of hours to several months) and the emphasis is more on analyzing the potential operational consequences of different scenarios. These could include planned maintenance resulting in the closure of parts of the network or changes to the start-up procedures for certain areas of the airport.

Operational problems are defined as those that occur on a day-to-day basis when predetermined operating policies need to be adjusted due to unforeseen disruptions or changes to the prerequisites of the problem (weather conditions, congestions in the airspace, etc.). The highly dynamic environment in which these problems occur necessitates almost immediate resolution. The routing of aircraft through taxiway networks is very susceptible to disruption and the routes assigned to the aircraft are highly interdependent.

In this paper we address the tactical variant of the problem as it is defined at Copenhagen Airport (CPH). At CPH the operational routing of arriving and departing aircraft is handled by the Air Traffic Control authorities (ATC), however, the taxiways are owned and maintained by the airport. It is thus the responsibility of CPH to ensure that the taxiways are in a good conditions and that the taxiway network provides sufficient capacity to accommodate the demand. We formulate the taxiway route allocation problem as a set packing model with a resource based constraint system and develop an LP based heuristic mimicking the operational practice at ATC to solve the problem. To exemplify the usage of the model we present an analysis conducted at CPH in 2013, analyzing the operational consequences of changing the start-up position for a selected group of aircraft stands.

The remainder of this report is organized as follows. Section A.2 provides a review of previous work and briefly introduces the developed model and solution approach. Section A.3 provides a more detailed description of the problem and introduces some airport terminology. A description of our set packing-based model is given in Section A.4 and the developed solution method is presented in Section A.5. The case study from CPH is presented in Section A.6 and finally, we draw some conclusions in Section A.7.

A.2 Previous work

In the following section we give an overview of the previous work on the problem of routing aircraft through taxiway networks. Our aim is to give a brief yet encompassing view of the literature in this field, and hence we include work on all variants of the problem.

[Atkin et al., 2010] consider the more general *ground movement problem* and provide a thorough description of the different ground movement problems associated with the handling of arriving and departing aircraft in an airport, including the taxiway route allocation problem. As described by [Atkin et al., 2010], a typical approach to modeling the taxiway network is to represent it by a directed graph $G = (V, E)$ where each vertex represents an intersection of taxiways and an edge represents a directed taxiway between two intersections. Furthermore, [Atkin et al., 2010] describe how the previous work done on the taxiway route allocation problem tend to pursue two different approaches.

The first approach involves the development of a mixed integer linear programming (MIP) formulation and typically, the problem is solved using commercial solvers or heuristics. The MIP formulations can furthermore be classified into *exact position* approaches and *ordering* approaches. In the exact position approaches, the path of each aircraft is specified by a set of time and location pairs, indicating exactly where the aircraft is at what time. Typically, the considered planning horizon is discretized into a set of short time intervals and a copy of the taxiway network is then created for each of these time intervals. For examples of MIP formulations using the exact position approach, see e.g. [Marín, 2006], [Balakrishnan and Yoon, 2007], [Marín and Codina, 2008], [Marín and Salmerón, 2008], and [Roling and Visser, 2008].

In the ordering approach, rather than dealing directly with timings of each aircraft, the solution algorithm first aims to decide upon the sequencing and then uses this information to schedule times for each aircraft at each vertex or edge in the network. The ordering approach only require the spatial network and the sequencing constraints are modeled using binary variables, where the variables for a pair (i, j) of aircraft at a node/edge are equal to one if and only if aircraft i passes the node/edge before aircraft j. The advantage of the ordering approach is that the times for each aircraft can be modeled as continuous variables, avoiding the disadvantages of time discretization. For examples of MILP formulations using the ordering approach, see e.g. [Smeltink and Soomer, 2008], [Rathinam et al., 2008], [Clare et al., 2009], [Clare and Richards, 2011], and [Anderson and Milutinović, 2013]

The second approach to modeling and solving the taxiway route allocation problem presented by [Atkin et al., 2010] is to use genetic algorithms (GAs). GAs are search methods that incorporate the ideas of natural selection, mutation, and crossover inspired by evolutionary biology (see e.g. [Mitchell, 1998]). For examples of GA approaches to the taxiway route allocation problem see e.g. [Pesic et al., 2001], [Baik et al., 2002], [Gotteland and Durand, 2003], [García et al., 2005], [Li et al., 2011], [Liu and Guo, 2010], and [Liu and Wang, 2011].

The taxiway route allocation problem is a highly dynamic problem, and exact information about e.g. when an aircraft is ready to be pushed back from its stand, when an aircraft will land, what runway exit an arriving aircraft will take, what runway entry a departing aircraft will take, or even the exact location of each aircraft within the taxiway network, can be very hard to predict. The further the information is in the future, the less accurate it is. Moreover, the underlying conditions to the taxiway route allocation problem may be changed several times during the day of operations due to e.g. the weather conditions.

As a consequence, solutions to the taxiway route allocation problem quickly become invalid and must be continuously updated. This problem can be addressed either by providing a framework in which the problem is decomposed into smaller sub-problems which are then solved using a shifted windows approach where the problem is solved for a fixed time interval every Δ minutes (see e.g. [Pesic et al., 2001]). Alternatively, a rolling horizon approach can be applied. Here, the planning horizon is split into disjoint time intervals determined either by a fixed length or by the number of aircraft handled, and each of these are then handled consecutively (see e.g. [Smeltink and Soomer, 2008] and [Marín and Salmerón, 2008]).

Considering the operational variant of the problem [Cheng et al., 2001], [Koeners et al., 2011], [Koeners and Rademaker, 2012], [Mehta et al., 2011], [Khadilkar and Balakrishnan, 2012], and [D'Ariano et al., 2012] each develop methods and decision support systems for controlling and maintaining already planned ground movements with focus on handling conflicts as they occur.

The problem of routing aircraft through taxiway networks has many similarities to the problem of routing trains through railway junctions. A railway junction can be viewed as a graph of nodes and edges, where the edges represent railway track segments and the nodes indicate the connections between track segments. Given the detailed track layout of a junction as well as the respective arrival and departure times and entry and exit points for a set of trains, the railway track allocation problem is the problem of routing all trains through the junction in a conflict free manner. [Lusby et al., 2011a] provide an extensive survey of the different models and methods to solve the railway track allocation problem.

[Lusby et al., 2011b] suggest an alternative modeling and solution approach, where the problem is formulated as a set packing model with a resource-based constraint system. The model easily allows the possibility to dynamically modify the problem and conflicts are handled implicitly by the constraints. The primal variables of the problem represent possible paths for each train and these are constructed a priori; for each train, all potential routes through the junction are identified, and for each route, the set of possible paths (the temporal traversal of a route) are identified, given that a train can either halt, speed up or continue in same speed when entering each track section. Although this new model potentially has many constraints, [Lusby et al., 2011b] demonstrate how a tailored branch-and-price based algorithm that exploits structural properties of the model can solve the problem efficiently.

In the following we present a formulation of the tactical variant of the taxiway route allocation problem similar to the formulation of the railway track allocation problem suggested by [Lusby et al., 2011b]. Instead of solving the problem with a branch-and-price based algorithm, we develop an LP based heuristic in which an LP relaxed version of the problem is solved for a restricted set of paths for each aircraft. In the case that one or more aircraft are not assigned a path, additional paths are added to the model. The algorithm continues until all aircraft have been assigned a path or no more paths can be added to the model. We assume that landing times and runway exit points points are known a priori for all arriving aircraft and that departure stands and runway entry points are known for all departing aircraft.

A.3 Problem definition

The *maneuvering area* is the part of an airport used by aircraft for landing, take-off, towing and taxiing. The *apron* is the area designated for aircraft during disembarkation/boarding of passengers, handling of cargo and baggage, fueling, parking and maintenance. An *aircraft stand* (or just stand) is a designated area on the apron intended to be used for parking an aircraft and a *taxiway* is a road within the maneuvering area connecting runways with stands, hangars, terminals and other facilities. In most airports there are multiple taxiways, and these are collectively referred to as the *taxiway network* of the airport. The taxiway network of CPH is shown in Figure A.1.

To model that an aircraft only claims a small part of a taxiway at a time, and that a taxiway can be used by many different aircraft at a time, we model the taxiway network as a graph, where the edges are obtained by discretizing each taxiway is into a set of non-overlapping *taxiway segments* and the vertices describe how the taxiway segments are connected—see Figure A.2 for an illustration. Each taxiway segment can be claimed by at most one aircraft at a time.

A *route* is a sequence of taxiway segments, leading from one point in the network to another. Dependent on the layout of the taxiway network and various aircraft type restrictions (see [Haagensen, 2012]), many possible routes may exist between a given origin-destination pair in the network. We distinguish a route from a *path*, which refers to the temporal traversal of a given route. A path thus consists of the sequence of taxiway segment time interval pairs (also referred to as resource-time intervals) specifying when the aircraft occupies each taxiway segment of the route. Two different aircraft can thus be assigned to the same route, as long as their corresponding paths are not in *conflict*, i.e. simultaneously claim the same taxiway segment.

Before a departing aircraft can taxi out to the runway, it needs to start up its engines. Dependent on the engine type of the aircraft and the placement of the stand on the apron, the aircraft may start up its engines at the stand. Alternatively, the aircraft must first be moved to a designated start-up position by a push-back tractor. In the following we refer to the entire process from when an aircraft is ready to leave its departure stand until it has started its engines and is ready

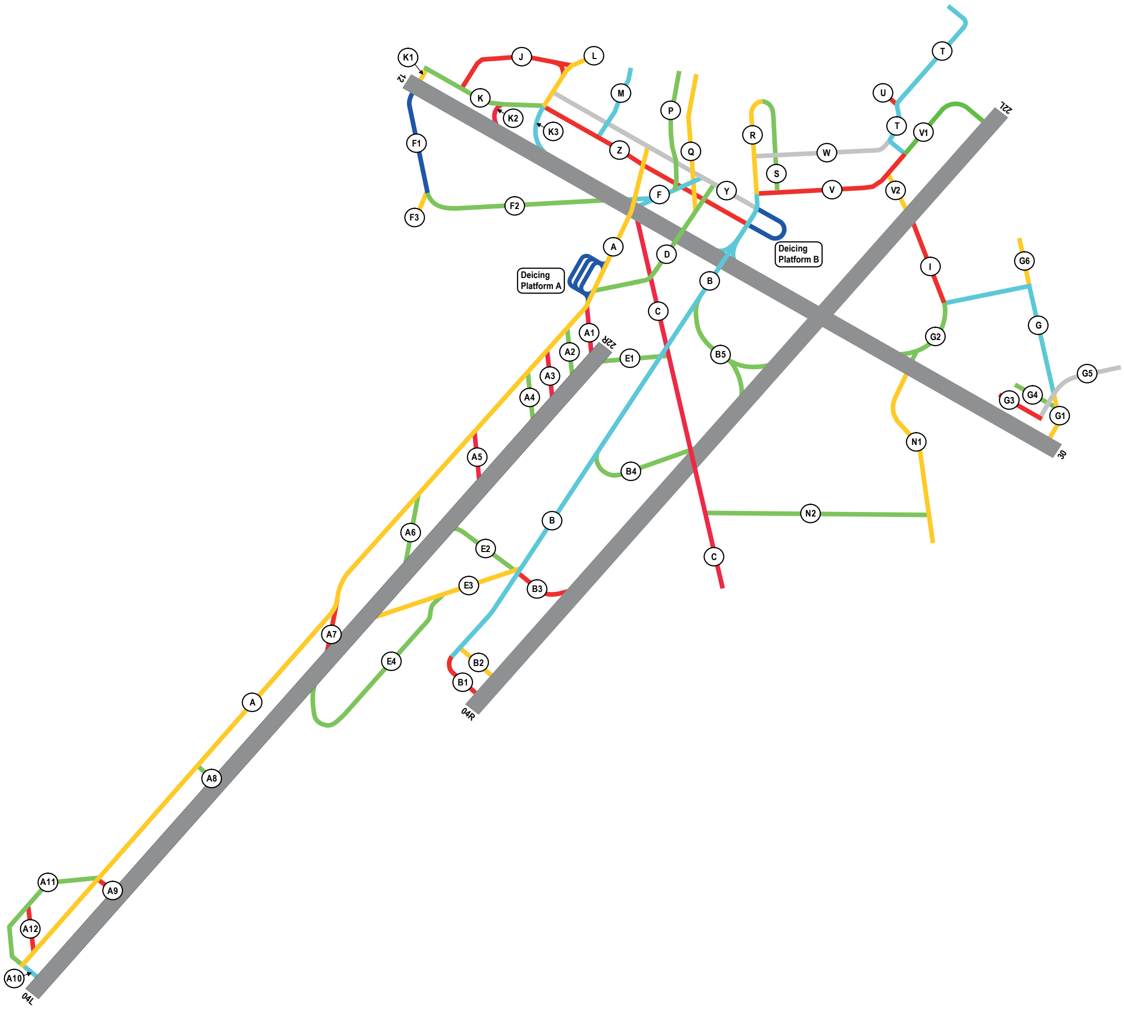

Figure A.1: *The taxiway network of Copenhagen Airport. The gray lines are the runways and the colored lines are the individual taxiways forming the taxiway network.*

to taxi out to the departure runway as a *push-back option*. For each stand, multiple push-back options may exist—see Figure A.3 for an example of the possible push-back options for B10 at CPH.

Which push-back option a given aircraft can use depends on the aircraft type, the stand, and various technical conditions. The start-up positions are typically placed on the taxiways, i.e. while starting its engines and running through the start-up procedure, the aircraft will be occupying a part of the taxiway network. The time an aircraft needs for its start-up procedure is dependent on the aircraft type, but can take up to ten minutes. The choice of push-back option is thus also dependent on other traffic within the taxiway network. A *push-back schedule* is the operationalization of a given push-back option, accurately describing which taxiway segments are need and at what time, in order to complete the push-back. We have therefore chosen to model push-back options as being part of the possible routes for departing aircraft. Thereby the choice of push-back option from a given stand naturally becomes part of the route allocation problem. Moreover, by imposing the restriction that no taxiway segment can be claimed by more than one turn-round at a time, and by including the push-back schedules in the paths for departing aircraft,

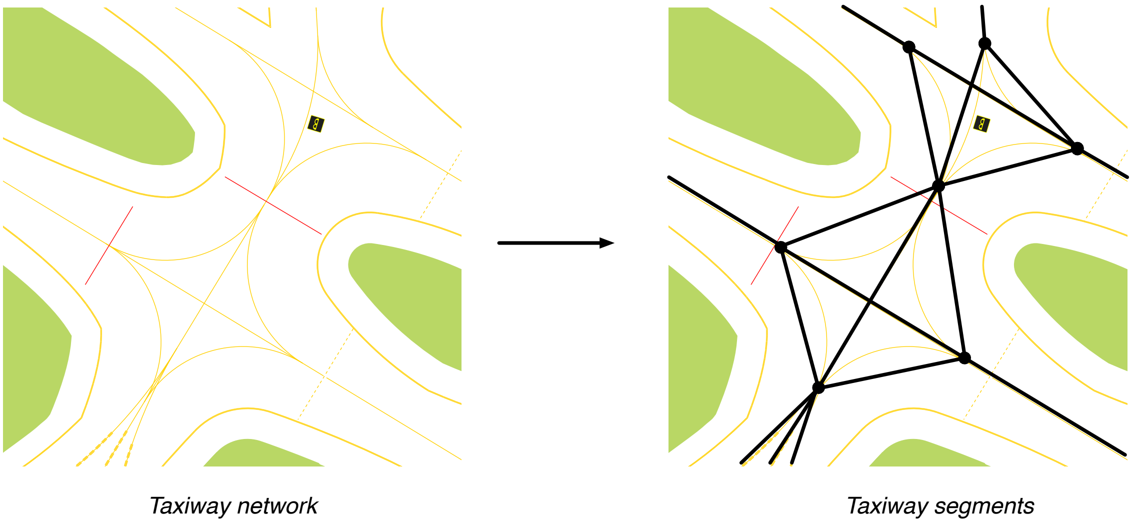

Figure A.2: *Modeling the taxiway network as a graph. The edges are obtained by discretizing each taxiway into a set of non-overlapping taxiway segments and the vertices describe how the taxiway segments are connected.*

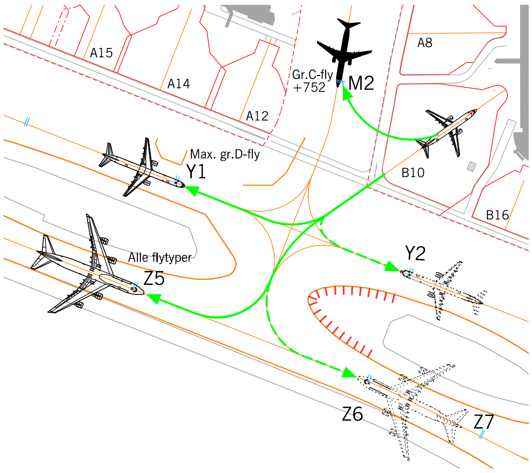

Figure A.3: *Possible push-back options for stand B10 at Copenhagen Airport. M2, Y1, Y2, Z5, and Z6 denote the different start-up positions at the taxiways M, Y, and Z respectively that should preferably be used when starting up aircraft departing from B10. Large aircraft must always be pushed to either start-up position Z5 or Z6, whereas small/medium sized aircraft can be pushed any of the start-up positions. The choice of start-up position is dependent on wind direction, other traffic, and various technical conditions.*

any type of push-back restrictions for neighboring stand thus also implicitly handled. The time it takes to complete a push-back option is referred to as the *push-back time*.

In the ideal situation, all aircraft are routed through the taxiway network in such a way that no aircraft has to hold back for other aircraft at any time. However, for many airports, the taxiway network is almost completely saturated during the peak hours, and requiring that each aircraft should be able to taxi through the network without any hold-backs would severely limit the capacity of the network. It is therefore accepted, that routes intersect and that some aircraft have to hold back for each other at specified *holding points* in the network. A holding point can in principle be anywhere in the network, however, for the sake of simplicity we assume that the holding points are always at the end points of the taxiway segments.

For departing aircraft another option is to hold back at the stand until a conflict free route through the network is available. Thereby, the amount of time that the aircraft has its engines running is minimized. A disadvantage of this approach, however, is that the aircraft is not immediately ready to taxi out once a conflict free path through the taxiway network is identified. Unpredicted delays may occur during both the push-back and the subsequent start-up process. As a consequence, many airports practice what is called *push-and-hold* in which the aircraft is first made ready for taxiing and then asked to hold back before continuing, usually at the point where it performed its start-up.

As can be seen, many alternative paths representing different delay strategies may potentially exist for each aircraft, for each route through the network. In the following we refer to the extra taxi time that an aircraft spends in the taxiway network due to hold backs en-route or push-and-hold as *taxiway delay* and for departing aircraft being held back at the stand, the hold back time is referred to as *stand delay*. In general, both types of delay should be avoided, as they prolong the total taxi time of the aircraft, however, stand delay is typically preferred (by the airlines) as this reduces the overall fuel consumption. In the following, we will let *taxitime* denote the total time an aircraft spends in the taxiway network including push-back time, taxiway delay, and stand delay.

For safety reasons, aircraft are not allowed to be too close to each other while taxiing. To ensure this separation, we will let each aircraft claim each taxiway segment along a given route from the time it enters the taxiway segment with its nose until the tail of the aircraft has left the segment. Assuming that the differences in speed among the taxiing aircraft are very small due to the general speed restrictions of the taxiway network (see [Haagensen, 2012]), this approach will ensure that taxiing aircraft are separated at least by the length of the shortest taxiway segment. This separation is assumed sufficient.

At some parts of the network, the taxiways may be placed so close to each other that the wings of an aircraft traversing one of them will also cover the others. In the following we will say that two taxiway segments are *dependent* if an aircraft traversing one of them, prevents simultaneous usage of the other. Intersecting taxiway segments are always dependent. When an aircraft traverses a route it will thus not only have to claim the taxiway segments of its route but also the corresponding dependent taxiway segments.

Taxiway networks can have rather complex layouts, and handling the taxiway route allocation problem in real airports involves extensive communication between the ATC, the pilots, and the airport. Once an arriving aircraft has landed at CPH, ATC allocates it a route to the arrival stand. Equivalently, once a departing aircraft is ready for push-back, ATC decides exactly when and how it should perform the push-back and start-up procedure (the push-back option) and allocates it a route to the departure runway. Finally, ATC is responsible for deciding when and where taxiing aircraft should hold back for each other while taxiing though the taxiway network.

In general, ATC strives to impose some degree of fairness when routing the aircraft through the taxiway network by ensuring that a single aircraft is not being held back too much, that aircraft which are known to be moving faster through the network are prioritized, and that departing aircraft with restrictions on their take-off time are given priority. Modeling all of these fairness measures is very difficult, especially given, that most of them only exist as tacit knowledge, i.e. they are not written down and well defined, and they are difficult to quantify. For the sake of simplicity we thus model these aspects of fairness by imposing restrictions on the maximum number of times and the maximum amount of time each aircraft is allowed to be held back and by prioritizing departing aircraft over arriving aircraft when imposing hold backs.

Today, ATC manually allocates routes to arriving and departing aircraft. To control the general flow in the network, ATC has established a set of *standard routes* for each stand and runway pair for both arriving and departing aircraft. The number of standard routes for the different stand and runway pairs vary from 1 to 4, and if multiple routes may exist, these are in principle all considered equally good, despite the fact that they may have different lengths. Deviations from standard routes are of course allowed, but should be avoided, as this may unnecessarily confuse the pilots. In this paper, we do not model deviations. Instead, we assume that the set of possible routes between any pair of stand and runway only consists of the standard routes, and for each aircraft we then allocated the path imposing the least amount of delay when following one of the standard routes.

The *taxiway route allocation problem* can now be stated as the problem of assigning a set of non-conflicting paths to a set of arriving and departing aircraft, such that the number of unallocated aircraft is minimized, and the amount of taxiway and stand delay are minimized. In general, a higher priority should be given to departing aircraft. A solution to the problem specifies how each aircraft should be routed through the taxiway network with an indication of where each aircraft should hold back and for how long. An aircraft is said to be *unallocated* if is not possible to route it through the network given the restrictions on the maximum amount of time the aircraft is allowed to be delayed.

A.4 Mathematical model

Let $\mathcal{M}$ denote the set of aircraft, $\mathcal{S}$ the set of taxiway segments, and $\mathcal{I}$ a set of time intervals obtained through a discretization of the planning period. Furthermore, let $\mathcal{M}^a \subseteq \mathcal{M}$ and $\mathcal{M}^d \subseteq \mathcal{M}$ denote the set of arriving respectively departing aircraft, i.e. we have that $\mathcal{M}^a \cap \mathcal{M}^d = \emptyset$ and $\mathcal{M}^a \cup \mathcal{M}^d = \mathcal{M}$. The set of all possible paths for all aircraft is denoted $\mathcal{P}$ and the set of paths for aircraft $m \in \mathcal{M}$ is denoted $\mathcal{P}_m \subseteq \mathcal{P}$. The binary parameter a_{pst} is one if path $p \in \mathcal{P}$ uses taxiway segment $s \in \mathcal{S}$ in time interval $t \in \mathcal{I}$ and zero otherwise.

To model the problem, two sets of decision variables are defined: the binary allocation variables x_p and the coverage variables π_m. An allocation variable x_p is one if path $p \in \mathcal{P}$ is chosen in the final solution and zero otherwise. A coverage variable π_m is one if aircraft $m \in \mathcal{M}$ is not allocated and zero otherwise.

Now, let the positive parameters $\delta_p^T \geq 0$ and $\delta_p^S \geq 0$ denote the amount of taxiway delay respectively stand delay for path $p \in \mathcal{P}$. The objective function is then given by the following weighted sum:

$$\begin{aligned} z = \quad & \omega_1 \sum_{m \in \mathcal{M}} \pi_m \\ & \omega_2 \sum_{m \in \mathcal{M}^d} \sum_{p \in \mathcal{P}_m} \delta_p^T x_p + \omega_3 \sum_{m \in \mathcal{M}^a} \sum_{p \in \mathcal{P}_m} \delta_p^T x_p + \\ & \omega_4 \sum_{m \in \mathcal{M}^d} \sum_{p \in \mathcal{P}_m} \delta_p^S x_p + \omega_5 \sum_{m \in \mathcal{M}^a} \sum_{p \in \mathcal{P}_m} \delta_p^S x_p \end{aligned} \tag{A.1}$$

Assuming that all parameters are non-negative integers, the taxiway route allocation problem can thus be formulated as the following integer programme (TRAP):

$$\begin{aligned} \textbf{minimize} \quad & z & & \text{(A.2)} \\ \textbf{subject to} \quad & \sum_{p \in \mathcal{P}_m} x_p + \pi_m = 1 & \forall\, m \in \mathcal{M} \quad & \text{(A.3)} \\ & \sum_{p \in \mathcal{P}} a_{pst} x_p \leq 1 & \forall\, s \in \mathcal{S}\ \forall\, t \in \mathcal{I} \quad & \text{(A.4)} \\ & x_p \in \{0,1\} & \forall\, p \in \mathcal{P} \quad & \text{(A.5)} \\ & \pi_m \in \{0,1\} & \forall\, m \in \mathcal{M} \quad & \text{(A.6)} \end{aligned}$$

where the objective function (A.2) minimizes the total cost of routing all aircraft through the taxiway network. Constraints (A.3) ensure that all aircraft are either allocated a route or not and constraints (A.4) ensure that each taxiway segment is claimed by at most one aircraft in each time interval. Finally, constraints (A.5) and (A.6) are the binary restrictions on the allocation variables and the coverage variables.

Given the nature of the constraint system, it is obvious that the model implicitly obtains a conflict free set of paths for the aircraft. Furthermore, one can also dynamically consider additional paths for aircraft without any fundamental change to the model itself. Additional paths can easily be included as extra columns in the model.

A.5 Solution method

One of the main challenges when solving TRAP is the dimensions of the constraint matrix. Although the set of possible routes is restricted to the set of standard routes, and that limitations to the number of times and the amount of time each aircraft is allowed to be held back are imposed, the number of delay combinations, and thus the number of paths, may still be very large for each aircraft. Consider an arriving aircraft and assume that there are two possible standard routes the aircraft can follow to get to its arrival stand. Moreover, assume that the aircraft can be held back at two different positions in the network for each standard route and at each hold-back point, we can impose a delay of 15, 30, 45, and 60 seconds, but the aircraft cannot be held back no more than one minute in total. The number of possible paths for this aircraft is thus 18. This is not much, but obviously the number quickly becomes much larger as the number of hold back positions and possible delay options are increased. Also number of resource-time interval constraints may be very large. Given that a taxiing aircraft move with a speed of approximately 15km/h, a time discretization of at most 15 seconds should be used in order to accurately model the problem. The taxiway network graph at CPH consists of 349 taxiway segments, i.e. for a two hour period the model will have 167,520 resource-time interval constraints. Another challenge is the binary constraints on the decision variables; MIP problems are non-convex and hence in the general case computationally difficult to solve.

To deal with these two challenges, the developed solution method considers a relaxed and restricted version of the problem, R-TRAP, in which the set of paths is restricted to $\mathcal{P}' \subseteq \mathcal{P}$ and the integrality constraints (A.5) and (A.6) are relaxed.

Relaxing the integrality constraints converts the problem into an LP problem which in the general case is computationally much faster to solve than an IP problem. Restricting the set of considered paths leverages the presumption that only a subset of the paths in $\mathcal{P}$ needs to be considered when solving the problem, since most of the possible paths will be non-basic and assume a value of zero in an optimal solution anyway.

Due to the presence of the coverage variables, there will always be a feasible solution to R-TRAP; however, since we have restricted the set of possible paths, the solution might be *invalid*, in the sense that one or more aircrafts are left unallocated, while more possible paths for these aircraft actually exist.

The idea of the solution method is as follows. Initially, $\mathcal{P}' = \emptyset$, i.e. R-TRAP only contains the coverage variables. Then R-TRAP is solved, and the set of unallocated aircraft

$$\tilde{\mathcal{M}} = \{m \in \mathcal{M} \mid \pi_m > 0\}$$

is identified. If $\tilde{\mathcal{M}} = \emptyset$, all aircraft have been successfully assigned a path and a valid solution has been found. If, however, $\tilde{\mathcal{M}} \neq \emptyset$, more paths are added to $\mathcal{P}'$, and R-TRAP is re-solved. The solution methods continues this iterative loop until a valid solution has been found or no more paths can be added to $\mathcal{P}'$.

Adding paths to $\mathcal{P}'$ is a two-stage process. First the set of aircraft for which more paths must be added to $\mathcal{P}'$ is identified. Then, new paths are identified for each of these aircraft, and these are added to $\mathcal{P}'$.

The main ideas of the solution approach are summarized in Algorithm 4. In the following, we refer to one iteration of the loop in line 3 as one iteration of the solution algorithm.

Algorithm 4: Pseudo-code for the described solution method.

```
Read airport data, timetable and parameters;
Set $\mathcal{P}' = \emptyset$;
repeat
    Solve R-TRAP using a standard LP solver;
    Identify the set of unallocated aircraft $\tilde{\mathcal{M}}$;
    Identify the set of binding constraints $\tilde{\mathcal{K}}$;
    Identify the set of conflicting aircraft $\hat{\mathcal{M}}$;
    for each aircraft $m \in \tilde{\mathcal{M}} \cup \hat{\mathcal{M}}$ do
        if $\mathcal{P}_r \setminus \mathcal{P}'_r \neq \emptyset$ then
            Identify the cheapest paths from $\mathcal{P}_r \setminus \mathcal{P}'_r$ compatible with $\tilde{\mathcal{K}}$;
            if possible to identify new paths then
                Add new paths to $\mathcal{P}'$;
until valid solution has been found;
if solution is fractional then
    Restrict all variables to be binary;
    repeat
        Solve SGAP using a standard IP solver;
        if solution to SGAP found then
            Save and return solution;
        else
            for each aircraft $m \in \mathcal{M}$ do
                Identify the cheapest paths from $\mathcal{P}_r \setminus \mathcal{P}'_r$;
                if possible to identify new paths then
                    Add new paths to $\mathcal{P}'$;
    until valid solution has been found;
else
    Save and return solution;
```

A.5.1 Identifying conflicting aircraft

A reasonable explanation to why a given aircraft $m \in \tilde{\mathcal{M}}$ is left unallocated, is that $\mathcal{P}'_m$ is too restricted. However, another explanation could be that the sets of paths for the aircraft preventing m from being allocated are too restricted.

Now, let $\mathcal{K}_p$ denote the set of resource-time interval constraints claimed by path $p \in \mathcal{P}$:

$$\mathcal{K}_p = \left\{ k \in \mathcal{K} \;\middle|\; a_{kp} = 1 \right\}$$

The set of resource-time interval constraints claimed by all the paths in $\mathcal{P}'_m$ for a given aircraft $m \in \mathcal{M}$ is thus given by

$$\mathcal{K}_m = \bigcup_{p \in \mathcal{P}'_m} \mathcal{K}_p$$

Now, consider an unallocated aircraft $m \in \tilde{\mathcal{M}}$. Since m is unallocated, none of the paths in the set $\mathcal{P}'_m$ have been chosen in the solution and consequently, a set of binding constraints preventing m from being allocated must exist.

For each optimal primal solution to the R-TRAP, a corresponding dual solution also exists. Let σ_k denote the dual variable of constraint $k \in \mathcal{K}$. If k is binding, the value of its corresponding dual variable is positive, i.e. $\sigma_k > 0$. The set of binding constraints in an optimal solution is thus given as

$$\tilde{\mathcal{K}} = \left\{ k \in \mathcal{K} \;\middle|\; \sigma_k > 0 \right\}$$

Thereby, the set of binding constraints preventing m from being allocated is given as

$$\tilde{\mathcal{K}}_m = \tilde{\mathcal{K}} \cap \mathcal{K}_m$$

and correspondingly, the set of *conflicting* aircraft preventing m from being allocated is given as

$$\hat{\mathcal{M}}_m = \left\{ q \in \mathcal{M} \;\middle|\; q \neq m, \exists\, p \in \mathcal{P}'_q : x_p > 0 \;\wedge\; \mathcal{K}_p \cap \tilde{\mathcal{K}}_m \neq \emptyset \right\}.$$

and the set of all conflicting aircraft for *all* unallocated aircraft is then given as

$$\hat{\mathcal{M}} = \bigcup_{m \in \tilde{\mathcal{M}}} \hat{\mathcal{M}}_m$$

To resolve the problem of an invalid solution, the solution method therefore tries to expand $\mathcal{P}'$ by adding more paths for each aircraft $m \in \tilde{\mathcal{M}} \bigcup \hat{\mathcal{M}}$.

A.5.2 Identifying new paths

Consider a conflicting aircraft $m \in \mathcal{M}$ and let $\mathcal{P}'_m \subseteq \mathcal{P}_m$ denote the set of paths already added to $\mathcal{P}'_m$ for m at a given iteration of the solution method. The set of new paths for m must hence be found in the set $\mathcal{P}_m \setminus \mathcal{P}'_m$.

Given the objective function described in Section A.4 it is clear that the quality of each path is reflected in its cost; the cheapest paths are the most attractive and the most expensive paths are the least attractive.

Now the idea of the solution method is simply to perform an enumeration of the paths in $\mathcal{P}_m \setminus \mathcal{P}'_m$ and then select the cheapest of these to add to $\mathcal{P}'$. To keep the set $\mathcal{P}'$ as small as possible, a maximum number of paths that can be added to R-TRAP at each iteration is specified.

First, all paths for the standard routes are generated, without imposing any delay. Then, if more paths are needed, paths with stand delays of $15, 30, 45, \ldots, 300$ seconds are added to $\mathcal{P}'$ and finally, if a conflict free routing still cannot be found, paths with taxiway delays of $15, 30, 45, \ldots, 300$ seconds at the various holding points. In the real world operation the delays imposed can be of any says, however, for the sake of simplicity and to align the paths with the chosen time discretization, we have chosen to let the delay sizes be multiples of 15 seconds. Morover, in a real world setting, there is in principle no upper limit of the amount of time a given aircraft can be asked to wait at a given holding point, however, in the general case, it should be avoided to hold back an aircraft for more than five minutes at a single holding point.

To avoid adding paths that will not resolve the conflicts of the invalid solution, the solution method will first try to identify paths that do not claim any of the binding resource-time interval constraints from the set $\tilde{\mathcal{K}}$. If it is not possible to find any such paths, but more unadded paths exist, these are added.

As can be seen, the R-TRAP initially only contains the "best" paths for each aircraft, and for each iteration, the paths added gets less and less attractive. In a worst case scenario, the solution method will have to add all possible paths for a given aircraft $m \in \mathcal{M}$, i.e. $\mathcal{P}'_m = \mathcal{P}_m$. However, as will be presented in Section A.6 this case has not yet been observed during computational experiments.

A.5.3 Integer solutions

Define the 0-1 matrix $T = (T_{mp})$ with $|\mathcal{P}|$ columns and a row for each aircraft, i.e. the dimension of T is $|\mathcal{M}| \times |\mathcal{P}|$, where $T_{mp} = 1$ if column $p \in \mathcal{P}_m$ (each column of T contains just one nonzero element) and $T_{mp} = 0$ otherwise. Equivalently, define the 0-1 matrix $A = (A_{pst})$ with $|\mathcal{P}|$ columns and a row for each taxiway segment in each time interval, i.e. the dimension of A is $|\mathcal{S}| \times |\mathcal{I}| \times |\mathcal{P}|$, where $A_{pst} = 1$ if taxiway segment $s \in \mathcal{S}$ is claimed by path $p \in \mathcal{P}$ in time interval $t \in \mathcal{I}$ and $A_{pst} = 0$ otherwise.

As can be seen, the constraint matrix of TRAP can be partitioned into $|\mathcal{M}|$ *aircraft blocks*, where each aircraft block B_m, $m \in \mathcal{M}$

$$B_m = \begin{bmatrix} T_m \\ A_m \end{bmatrix}$$

contains all possible paths for a particular aircraft. Each aircraft block is a zero-one matrix with a generalized upper bound set partitioning constraint. Exactly as described in [Lusby et al., 2011b], we thus have that each aircraft block is a *perfect* matrix (see [Padberg, 1972]), and we therefore have a guarantee that the extreme points of the polytope $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{B_m x} \leq \mathbf{1}, \mathbf{x} \geq \mathbf{0}\}$ are integer (see [Rezanova and Ryan, 2010]). This ensures that fractions will only occur, if two or more aircraft are competing for the same taxiway segment in the same time interval.

The problem will thus by construction have good integer properties, and it is therefore likely that the solutions to R-TRAP are either integer or not that far from being integer. If the solution to R-TRAP is fractional, we can exploit the described properties of the constraint matrix to impose Ryan-Foster constraint branching when searching for an optimal integer solution (see [Ryan et al., 1981]). However, as the taxiway route allocation problem is a highly dynamic problem, being able to find a near-optimal integer solution efficiently is often preferred to finding an optimal integer solution, even at the strategical and tactical levels. As a consequence, we have decided to handle fractional solutions, simply by solving TRAP using the paths in $\mathcal{P}'$, i.e. when a valid but fractional solution is found, R-TRAP is "unrelaxed" and solved as an integer problem using a standard IP solver.

If it is not possible to find a valid integer solution to TRAP using only the paths in $\mathcal{P}'$, the algorithm will expand $\mathcal{P}'$ by adding more paths for each aircraft, and then try to re-solve the IP. In a worst case scenario, the algorithm will end up adding all paths from $\mathcal{P}$ in this process, however, as will be presented in Section A.6 this case has not yet been observed during the computational experiments.

By terminating the solution method the first time a valid solution is encountered, we have no guarantee that the found solution is optimal. However, since we are only adding the best paths for each aircraft in each iteration, it is likely that the objective value of the found solution is not that far from the objective value of the optimal solution. This claim still needs further investigation.

A.6 Case study: Changing start-up positions for selected stands at Copenhagen Airport

As a part of a larger project considering the general air quality in Copenhagen Airport, it was discovered that having start-ups on taxiway M, significantly increases the concentration of ultra fine particles in the air in the areas between Pier A and B (see Figure A.4). When present in high concentrations, ultra fine particles are harmful to the health and on a daily basis both handling agents, airline staff, airport staff, and passengers are regular visitors to the areas between Pier A and B. It was therefore suggested that the push-back options for the stands A7, A9, A11, B2, B4, B6, B8, and B10 were changed so that instead of using the start-up positions on taxiway M, the aircaft should be pushed out to the start-up positions at either taxiway Y or Z. During the month of September 2012, these new push-back options were tested and the impact on the air quality was analyzed. The analysis showed that the new push-back options improved the air quality in the areas between Pier A and B. However, as taxiway Z and Y are the two main taxiways connecting Terminal 1 and the southern and western parts of Pier A with the runways, both ATC and the airlines expressed concerns that having start-ups on these taxiways would lead to an increased amount of congestion in the network, and they therefore wanted to supplement the air quality analysis with an analysis of the potential operational consequences of the new start-up positions before making a final decision.

The analysis considered five different instances representing typical busy morning peaks at CPH. For each data set, the taxiway route allocation problem was then solved in two scenarios: Scenario A, where the push-back options were equivalent to the current practice, and Scenario B, where the push-back options had been changed for the selected stands. The idea of the analysis was then to compare the generated solutions for the two scenarios for each data set, considering the following indicators: the number of unallocated aircraft (Π), the total amount of stand delay (Δ^S), the total amount of taxiway delay (Δ^T), the total push-back time (Φ), and the total taxitime (Ψ). Furthermore, it was analyzed how the changed start-up positions affected the overall usage of the taxiways M, Y, and Z.

During the analysis, the performance of the developed solution method was also evaluated by considering the following indicators: the total solution time (T), the total time spent inspecting the solution and generating variables (T_v), the number of constraints, the number of variables added to the model, the objective value, and the integer properties in terms of whether or not the algorithm returned a valid integer solution to the R-TRAP.

Though being very detailed in its formulation, the developed mathematical model (TRAP) and solution method, will never fully encapsulate all operational aspects of the taxiway route allocation problem. The routing found as a solution to TRAP is thus very likely to be different from the real-life routing of the same aircraft, however, we assume that the solutions to TRAP are very close to the real-life routings. Nevertheless, it does not make sense to perform a detailed analysis of the routing of the aircraft and, as a consequence, the analysis focused more on the overall characteristics of the found solutions, and therefore the detailed routing of the individual aircraft was not reported as part of the results.

For analytical purposes, it was assumed that the entire taxiway network was available, i.e. closures due to maintenance were not taken into consideration. Moreover, only arriving and departing aircraft were assumed to be occupying the taxiway network, i.e. towings were not included and for each stand-runway pair, only the standard routes defined by ATC were made available. Finally, it was assumed that all aircraft taxi with a constant speed of 15km/h and that taxiway delays could only be imposed at either start-up positions or intersections between taxiways and that each aircraft could be delayed no more than three times and no more than ten minutes in total. To express that stand delay should be preferred to taxiway delay, and that departures should in general have a higher priority than arrivals, the following weights were used in the analysis: $\omega_1 = 1,000,000$, $\omega_2 = 600$, $\omega_3 = 450$, $\omega_4 = 2$, and $\omega_5 = 1.5$.

For each arriving aircraft, the estimated landing time, landing runway, runway exit point, and arrival stand were assumed to be known. Equivalently, for each departing aircraft, the departure stand, the estimated off-block time, and the departure runway entry point were assumed to be known. The characteristics of the instances considered are given in Table A.1.

Given the assumption on the speed of the aircraft and the planning horizon of the instances, a time discretization of 15 seconds was found to be suitable. The taxiway network consists of 349 taxiway segments, i.e. for a two hour period the model has 167,520 resource-time interval constraints.

Data set	**Weekday**	**Time interval**	**# Operations**	**# Arrivals**	**# Departures**
1207	Saturday	06:35—10:27	130	56	74
1209	Saturday	06:35—10:22	130	52	78
1210	Sunday	06:48—11:54	130	53	77
1301	Tuesday	06:57—15:45	130	56	74
1302	Thursday	06:31—09:17	130	57	73

Table A.1: *Instances used for testing.*

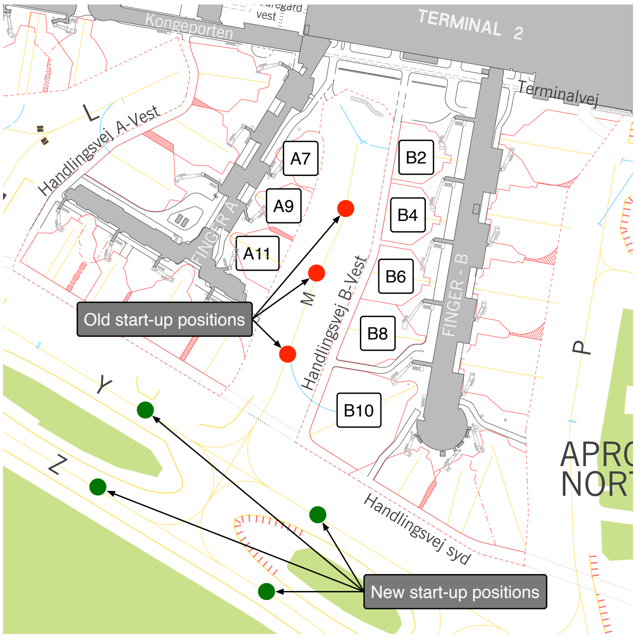

Figure A.4: *Illustration of the new and old start-up positions for the stands A7, A9, A11, B2, B4, B6, B8, and B10 at CPH.*

A.6.1 Computational results

The analysis was run on a Linux server running SuSE Linux Enterprise 11 64 bit with 8 GB RAM and a single Intel Merom single core processor. We used the Gurobi Optimizer version 5.5.0 as both the LP and IP solver and the algorithm was coded in C++. The computational results are shown in Table A.2, A.3, and A.4.

As can be seen from Table A.2 it was not always possible to route all departing aircraft using only the predefined standard routes. This indicates, that in a real-world context, deviations to the standards routes are used to ensure that all aircraft are routed through the network in a conflict free manner, i.e. the set of standard routes is too restricted in itself. The amount of taxiway delay either remains unchanged or is increased when introducing the new start-up position, but no general conclusions can be made for the amount of stand delay. For all instances, the total push-back time is increased, whereas the total taxitime is decreased in Scenario B. The increase in push-back time simply reflects that the new start-up positions used in Scenario B, are farther away from the stands than the start-up positions used in Scenario A. The decrease in total taxitime can be explained by two things. First of all, the new start-up positions are closer to the runways, i.e. aircraft departing from the stands A7, A9, A11, B2, B4, B6, B8, and B10 might spend more time doing their push-back and start-up, but the subsequent taxi-time is correspondingly reduced.

	Π		Δ^T (s)		Δ^S (s)		Φ (s)		Ψ (s)	
Data set	A	B	A	B	A	B	A	B	A	B
1207	0	1	0	90	5,865	5,115	44,115	46,920	86,955	85,485
1209	0	1	0	510	5,280	4,035	45,375	49,590	89,025	87,180
1210	0	0	0	0	2,415	2,595	46,050	51,135	82,815	80,520
1301	0	0	0	0	3,390	3,330	46,530	49,470	86,820	86,610
1302	2	2	0	135	4,395	5,430	42,345	45,465	90,090	89,820

Table A.2: *Quality of the solutions.* Π *is the number of unallocated aircraft,* Δ^S *the total amount of stand delay,* Δ^T *the total amount of taxiway delay,* Φ *the total push-back time, and* Ψ *the total taxitime.*

	M (s)		**Y** (s)		**Z** (s)	
Data set	A	B	A	B	A	B
1207	10,335	7,785	12,810	15,930	12,150	13,080
1209	13,485	10,695	13,695	18,090	10,110	11,175
1210	10,545	7,590	9,015	11,490	13,365	16,635
1301	5,655	4,515	16,005	18,120	5,400	7,080
1302	7,740	7,230	15,255	17,580	10,305	10,080

Table A.3: *Utilization of taxiway M, Y, and Z.*

	T(s)		T_v(s)		**Num cons.**		**Num vars.**		**LP int.**		**Obj. val**	
Data set	A	B	A	B	A	B	A	B	A	B	A	B
1207	42	1,307	8	1,169	379,101	379,101	4,137	7,960	0	0	4,194,505	4,645,653
1209	52	746	18	636	372,139	372,139	4,090	7,380	0	0	3,902,481	4,219,416
1210	42	55	5	14	482,108	482,108	4,070	4,712	0	0	1,763,908	1,997,926
1301	52	59	10	10	791,133	791,133	4,173	4,428	0	0	2,321,688	2,275,526
1302	367	1,901	302	1,797	287,227	287,227	5,713	6,799	0	0	5,424,922	6,307,494

Table A.4: *Performance of the solution algorithm. T is the total solution time, T_v is the total time spent inspecting the solution and generating variables, and the column LP int. shows whether or not the algorithm returned a valid integer solution to the R-TRAP.*

Second of all, by moving the start-up of departing aircraft away from taxiway M, arriving aircraft can more easily taxi directly to the stands without being delayed, because they no longer have to wait for departing aircraft performing the start-up procedure on taxiway M.

Table A.3 presents the total utilization (including push-back) of the taxiways M, Y, and Z. As expected, the new start-up positions lead to a reduced utilization of taxiway M, whereas the utilization of Y and Z is increased.

In general, the analysis revealed no reason not to move the start-up positions; the small increase in total push-back time is negated by the reduction in the subsequent taxitime and the small increase in the total delay is acceptable compared to the potential improvement of the air quality in the areas around taxiway M. As a consequence, the final decision was to permanently change the push-back options for the stands A7, A9, A11, B2, B4, B6, B8, and B10.

A.7 Conclusions and future work

In this paper, we have considered the tactical variant of the taxiway route allocation problem. We have demonstrated how the problem can be formulated as a set packing model with a resource-time interval based constraint system and demonstrated how a relatively simple LP based heuristic can be used to solve the problem. The quality of the solutions is considered good by the airport which is illustrated via a case study from Copenhagen Airport, where the model and solution method has been used to analyze the operational consequences of changing the start-up positions for a selected set of aircraft stands. To mimic the operational practice of ATC at CPH (and of many other airports), the solution algorithm was only allowed to use a pre-defined set of standard routes for each stand and runway pair.

An improvement of the algorithm is to include the possibility to let aircraft deviate from the standard routes. When considering a free routing through the network, several options may exist, and an obvious choice would be to always choose the shortest path. However, numerous practical restrictions might impose that the shortest path is not always the most attractive path. For instance, the number of *turns* of the different routes should also be taken into consideration. Besides the conventional meaning, a turn also marks the shift from one taxiway to another. The number of turns thus denotes the complexity of a route; the more turns, the more complex the route is, and the more, the pilots need to pay attention the taxiing, while still being in dialogue with ATC. Furthermore, the airport and ATC might still have preferences like the ones specified via the standard routes, i.e. they want the aircraft to follow a specific flow through the network.

Another problem when allowing a free routing, is that when using the conventional graph formulation of the network consisting of edges and vertices, nothing prevents the generation of *illegal* routes; if two or more taxiway segments intersect, it is not always the case that all possible turns at the intersection are allowed. It is therefore necessary to impose restrictions on the different taxiway intersections in the network. However, maintaining such a set of restrictions can be rather complicated and also just adds constraints to the model.

An alternative option is to model the network using a so-called *double vertex graph*. Double vertex graphs were introduced by [Montigel, 1992] to model railway networks and junctions and the idea is that each vertex v representing has a unique partner v'. The motivation for representing the taxiway network as a double vertex graph is the notion of a route in the network; by applying the rule that a route always follows the pattern *vertex-vertex-edge-vertex-vertex-edge-...*, it is prevented that impossible routes are explored. The *edges* represent taxiway segments. In Figure A.5 picture (a) illustrates an intersection between two taxiways, where the turn from A to B is not allowed. In (b) the traditional graph representation is used to represent the intersection. As can be seen, nothing prevents a route from going from A to B (A-C-B or A-D-B) unless specific constraints are formulated and added to the model. In (c) the same intersection is represented using the concept of a double vertex graph. Routes following the pattern *vertex-vertex-edge-vertex-vertex-edge-...* naturally cannot include the turn from A to B.

Another challenge with the model is that a rather fine time discretization is required to correctly encapsulate the dynamic nature of the problem. In a tactical setting, one typically considers a planning period of at least a couple of hours and the number of constraints may thus be very large. Finding the right time discretization is essentially a tradeoff between accuracy and complexity. A fine time discretization accurately models the problem, but the number of constraints increase the complexity of the model. On the other hand, a more coarse discretization, reduces the number of constraints, but it also needlessly decreases the capacity of the taxiway network. This problem could, however, be overcome by allowing that each taxiway segment can be claimed by more than one aircraft at a time, yet this aggregation would decrease the accuracy of the found solutions.

To reduce the number of constraints, multiple options exist. As described in Section A.2, one option is to split the considered planning period into multiple smaller, disjoint periods, and then solve these consecutively, assuming that routings covering more than one period are fixed in the first period they are encountered.

Furthermore, one could exploit what is termed *lazy constraints* in most modern LP and IP solvers (cf. e.g. GUROBI and CPLEX). The idea is to define a model with a core set of constraints, to keep it manageable, and then during execution additional constraints (the lazy constraints) are added based on the set of solutions found. Many resource-time intervals will be redundant anyway, and by using lazy constraint, only the required resource-time interval constraints will be added to the model.

The fact that many of the resource-time interval constraints are redundant can also be exploited

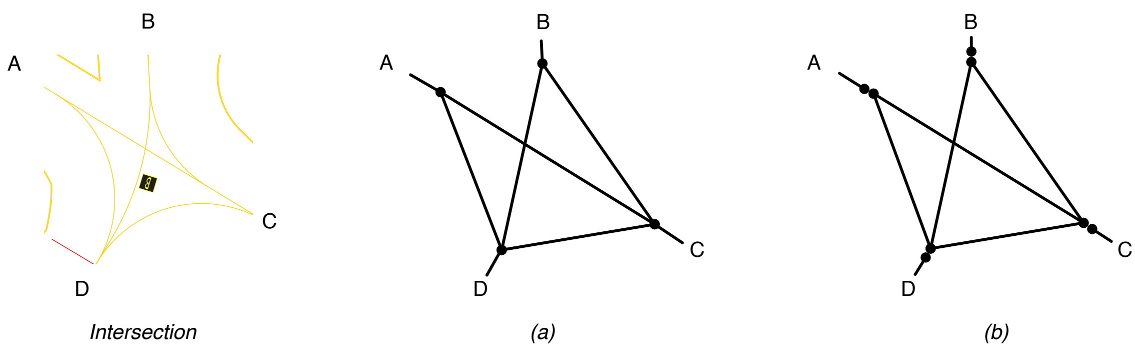

Figure A.5: *Illustration of the concept of a double vertex graph.*

in the solution algorithm. The formulation of the problem is equivalent to the formulation of the problem of routing trains through railway junctions presented in [Lusby et al., 2011b]. As described in [Lusby et al., 2011b], the dual representation of any basic feasible solution to the model, will only contain a subset of the possible aircraft path variables and has a much smaller basis than the full model. The idea is then to solve the dual representation of any basic feasible solution, and then in a branch-and-price framework exploit that an entering variable of the model is equivalent to a violated constraint in the dual problem.

Finally, an operational improvement to the model would be to add constraints ensuring that departures with restrictions on their take-off time also reach the runway within the assigned slot (CTOT).

Bibliography

Anderson, R., Milutinović, D., 2013. An approach to optimization of airport taxiway scheduling and traversal under uncertainty. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering 227 (2), 273–284.

Atkin, J. A. D., Burke, E. K., Ravizza, S., 06 2010. The airport ground movement problem: Past and current research and future directions. In: 4th International Conference on Research in Air Transportation. pp. 131–138.

Baik, H., Sherali, H. D., Trani, A. A., 2002. Time-dependent network assignment strategy for taxiway routing at airports. Transportation Research Record, 70–75.

Balakrishnan, H., Yoon, J., 2007. A framework for coordinated surface operations planning at dallas-fort worth international airport. Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference 2007 3, 2382–2400.

Cheng, Y., Sharma, Foyle, 2001. A study of aircraft taxi performance for enhancing airport surface traffic control. IEEE Transactions on Intelligent Transportation Systems 2 (2), 39–54.

Clare, G., Richards, A., 2011. Optimization of taxiway routing and runway scheduling. IEEE Transactions on Intelligent Transportation Systems 12 (4), 1000–1013.

Clare, G., Richards, A., Sharma, S., 2009. Receding horizon, iterative optimization of taxiway routing and runway scheduling. AIAA Guidance, Navigation, and Control Conference and Exhibit, 2009–5646.

D'Ariano, A., Pistelli, M., Pacciarelli, D., 2012. Aircraft retiming and rerouting in vicinity of airports. IET Intelligent Transport Systems 6 (4), 433–443.

EUROCONTROL, 2010. Long term forecast 2010-2030.

García, J., Berlanga, A., Molina, J. M., Casar, J. R., 2005. Optimization of airport ground operations integrating genetic and dynamic flow management algorithms. AI Communications 18 (2), 143–164.

Gotteland, J. B., Durand, N., 2003. Genetic algorithms applied to airport ground traffic optimization. Proceedings - CEC: 2003 Congres on evolutionary computation, 544–551.

Haagensen, K., 2012. Local regulations/appendix 1 - taxiway routing and guidelines for push-back at copenhagen airport.

Khadilkar, H., Balakrishnan, H., 2012. A network congestion control approach to airport departure management. Proceedings of the American Control Conference, 1682–1688.

Koeners, G. J. M., Rademaker, R. M., 2012. Ground control support functions to optimize surface traffic flow in a cdm environment.

Koeners, G. J. M., Stout, E. P., Rademaker, R. M., 2011. Improving taxi traffic flow by real-time runway sequence optimization using dynamic taxi route planning, 2C3–1–2C3–12.

Li, N., Zhao, Q., hao Xu, X., 2011. Research of taxing optimization for aircraft, 25–28.

Liu, C., Guo, K., 2010. Airport taxi scheduling optimization based on genetic algorithm.

Liu, C., Wang, Y., 2011. Aircraft taxi path optimization based on ant colony algorithm 2, 226–229.

Lusby, R., Larsen, J., Ehrgott, M., Ryan, D., 10 2011a. Railway track allocation: models and methods. OR Spectrum 33 (4), 843–883.
URL http://dx.doi.org/10.1007/s00291-009-0189-0

Lusby, R., Larsen, J., Ryan, D., Ehrgott, M., 2011b. Routing trains through railway junctions: A new set-packing approach. Transportation Science 45 (2), 228–245.

Marín, A., Codina, E., 2008. Network design: taxi planning. Annals of Operations Research 157 (1), 135–151.

Marín, Á., Salmerón, J., 2008. Taxi planner optimization: a management tool. Proceedings of the Institution of Mechanical Engineers – Part G – Journal of Aerospace Engineering (Professional Engineering Publishing) 222 (7), –.

Marín, Á. G., 2006. Airport management: taxi planning. Annals of Operations Research 143 (1), 191–202.

Mehta, V., Miller, M. E., Reynolds, T., Ishutkina, M., Jordan, R., Seater, R., Moser, W., 2011. Decision support tools for the tower flight data manager system, I4–1–I4–12.

Mitchell, M., 1998. An introduction to genetic algorithms.

Montigel, M., 1992. Formal representation of track topologies by double vertex graphs. Computers in Railways, 359–370.

Padberg, M. W., 1972. On the facial structure of set packing polyhedra. Mathematical Programming 5 (1), 199–215.

Pesic, B., Durand, N., Alliot, J. M., 2001. Aircraft ground traffic optimisation using a genetic algorithm. GECCO, 1397–1404.

Rathinam, S., Montoya, J., Jung, Y., 2008. An optimization model for reducing aircraft taxi times at the dallas fort worth international airport, 26th International Congress Of The Aeronautical Sciences.

Ravizza, S., Chen, J., Atkin, J. A. D., Burke, E. K., Stewart, P., 2012. The trade-off between taxi time and fuel consumption in airport ground movement. In: Proceedings of the Conference on Advanced Systems for Public Transport (CASPT12), Santiago, Chile.

Rezanova, N. J., Ryan, D., 2010. The train driver recovery problem—a set partitioning based model and solution method. Computers and Operations Research 37 (5), 845–856.

Roling, P. C., Visser, H. G., 2008. Optimal airport surface traffic planning using mixed-integer linear programming. International Journal of Aerospace Engineering.

Ryan, Foster, Wren, 1981. An integer programming approach to scheduling. In: Computer Scheduling of Public Transport. Urban Passenger Vehicle and Crew Scheduling. Proceedings of an International Workshop — 1981.

Smeltink, J. W., Soomer, M. J., 2008. An optimisation model for airport taxi scheduling.