TECHNISCHE UNIVERSITÄT MÜNCHEN

TUM School of Management

Essays on Optimization of Airport Infrastructure

Thomas Johannes Hagspihl

Vollständiger Abdruck der von der TUM School of Management der Technischen Universität München zur Erlangung eines

Doktors der Wirtschafts- und Sozialwissenschaften
(Dr. rer. pol.)

genehmigten Dissertation.

Vorsitz:		Prof. Dr. Stefan Minner
Prüfer der Dissertation:	1.	Prof. Dr. Rainer Kolisch
	2.	Prof. Dr. Florian Jaehn

Die Dissertation wurde am 23.11.2023 bei der Technischen Universität München eingereicht und durch die TUM School of Management am 15.02.2024 angenommen.

Acknowledgments

I would like to express my deep gratitude towards my doctoral supervisor Prof. Dr. Rainer Kolisch for his continuous support and constant motivation of my work as well as for our extraordinarily good relationship characterized by mutual trust. I always appreciated his openness towards my research interests, his pragmatism, and his flexibility whenever I asked for it. Furthermore, I would like to thank Prof. Dr. Florian Jaehn for being my second examiner and Prof. Dr. Stefan Minner for chairing the examination board.

I would also like to thank Prof. Dr. Pirmin Fontaine and Prof. Dr. Sebastian Schiffels for their mentoring and co-authorship of my publications. Together with Prof. Dr. Rainer Kolisch, they have enabled me to grow both as a researcher and as a person. Moreover, my special thanks go to my former colleague Dr. Christian Ruf, who initiated and co-authored my first publication and who motivated me to get involved in research. To my former and current colleagues, contacts, and friends at TUM School of Management: Pia Ammann, Carolin Bauerhenne, Baturhan Bayraktar, Giacomo Dall'Olio, Gregor Godbersen, Andreas Hagn, Christian Jost, Matias Kohn, Maximilian Kolter, Dr. Tobias Lieberum, Dr. Anulark Naber, Dr. Maximilian Pohl, Christine Steinberger, Dr. Stephen Starck, and Hendrik Weber, I would like to express my gratitude for the great time together and the fruitful exchange.

In addition, the research presented in this thesis has benefited considerably from exchange with external partners from research and industry. In this context, I would particularly like to thank Filippos Adamidis, Doris Anderl, Prof. Dr. Constantinos Antoniou, Carolin Aust, Ivonne Kuger, Dr. Malte Oehlmann, Dr. Kay Plötner, and Prof. Dr. Jutta Roosen for their support of my work.

Finally, I would like to thank my parents Birgit and Hans-Günter, and my partner Natalie for their unbounded love and support. They have always given me strength and motivation when I needed it most. Thank you.

Contents

List of Figures

List of Tables

Acronyms

AAM	Advanced Air Mobility
ADG	Airplane Design Group
AGLP	Airport Gate Layout Problem
CG	Column generation
DA	Decomposition approach
DGCP	Dynamic Gate Configuration Problem
eVTOL	Electric vertical take-off and landing aircraft
IIA	Independence of irrelevant alternatives
MARS	Multi Aircraft Ramping Stand
MP	Master Problem
OP	Original Problem
PBB	Passenger boarding bridge
RMP	Restricted Master Problem
SP	Subproblem

1 Introduction

1.1 Motivation

Global air traffic has grown significantly throughout the past decades. The number of global airline flights has increased from 23.8 million in 2004 to 38.9 million in 2019, resulting in a compound annual growth rate of 3.33 percent (International Air Transport Association 2023b). The number of globally boarded airline passengers has increased even faster in the same period, from 2.0 billion to 4.5 billion, corresponding to a compound annual growth rate of 5.56 percent (International Air Transport Association 2023c). Naturally, disruptive events such as the terrorist attacks of September 11, 2001 or the crisis caused by the SARS-CoV-2 virus have severely affected global aviation. However, demand recovered remarkably fast after these crises, and air traffic is currently approaching pre-SARS-CoV-2 levels (International Air Transport Association 2023a). Global air traffic can therefore be expected to continue its growth in the long term, despite recurring disruptive events.

Since the deregulation of the airline sector in the 1980s and the 1990s, most airlines operate hub-and-spoke networks to reduce their total network costs and make use of economies of scale (Brueckner 2004, Burghouwt and de Wit 2005, Tu et al. 2020). In these networks, passenger flows are consolidated at one or more hub airport(s) to allow larger, more efficient aircraft to operate between the hubs and to maximize aircraft load factors. Aircraft typically arrive and depart in waves at hub airports, as airlines aim to offer their passengers short connecting times and a large number of attractive connecting flights at their hubs (Burghouwt and de Wit 2005, Mirković and Tošić 2016, 2017). This causes considerable fluctuations of traffic volume and fleet mix over the course of each day.

The combination of traffic growth and traffic concentration during peak times imposes high demands on the capacity of hub airports. In particular, the airport infrastructure must be able to handle the traffic well during peak times. Airport infrastructure is usually divided into airside and landside infrastructure (e.g., International Civil Aviation Organization 1987, International Air Transport Association 2004, Ashford et al. 2011, Young and Wells 2011, de Neufville and Odoni 2013). The airside comprises the runway, taxiway, and apron systems, while the landside includes the terminal building (with areas for check-in, passport control, baggage claim, and security control), parking areas, transit stations, and ground access systems such as roads and rails. The interface between land- and airside infrastructure is formed by the gates where aircraft are parked during their time on the ground and passengers board and disembark (Horonjeff et al. 2010). The total capacity of an airport results from the capacities provided at the individual land- and airside infrastructure elements, as well as the relations between these elements (Katsigiannis and Zografos 2021). In order to successfully compete for new airlines, routes, and transfer passengers, hub airports must ensure that all infrastructure elements have sufficient capacity to allow for short transfer times as well as seamless day-to-day operations. Furthermore, in the competition for passengers who do not transfer but begin and/or end their flight journey at a hub airport, also the capacity and attractiveness of the feeder systems available to connect the airport to the surrounding region are of high importance. Therefore, in the course of this work, we will also include the infrastructure necessary for the operation of feeder systems (such as suburban train stations) in the airport infrastructure, even if it is not installed at the airport itself, but in the region surrounding the airport.

Prior to the crisis caused by the SARS-CoV-2 virus, many hub airports were operating at their capacity limits (Gelhausen et al. 2020). Given the general trend of air traffic growth, airports will reach their capacity limits again in the near future. To mitigate capacity shortages, airports can (a) use new space to build additional infrastructure such as runways or terminal buildings, (b) adapt existing infrastructure to increase throughput using the existing space, or (c) optimize operational processes to improve efficiency without touching the infrastructure. Of course, combinations of (a), (b), and (c) are also possible. While (a) provides the greatest leverage to increase capacity significantly, airport expansions that require additional space are not always feasible for financial, political, and environmental reasons, especially in Europe, or because there is no space available

in the vicinity of the existing airport. For example, London Heathrow Airport has been subject to numerous expansion plans but cannot easily be extended as it is surrounded by urban development. At Munich Airport, the construction of a third runway has been discussed for more than a decade, but the current Bavarian government has put these plans on hold until further notice (Munich Airport 2023b). Also in Munich, plans were pursued between the years 2000 and 2008 to connect the city center with the airport by a high-speed maglev train, significantly improving access to the airport. However, the plans were discontinued due to a considerable increase in the cost projections of the project. As a result, the current expansion plans of airports in the sense of (a) may not be sufficient to keep up with the growing demand for air transport (Eurocontrol 2018), increasing the importance of (b) and (c).

Existing literature addressing airport capacity, particularly in terms of airport infrastructure, can be divided into two branches. In the first branch, capacity is considered as decision variable on the strategic level. For instance, Sun and Schonfeld (2015) propose a model to strike a balance between capacity expansion costs and aircraft delay costs caused by capacity shortages assuming demand uncertainty. Xiao et al. (2013) investigate runway capacity choices for profit-maximizing, welfare-maximizing, as well as competing airports, also assuming demand uncertainty. While both Sun and Schonfeld (2015) and Xiao et al. (2013) consider airport capacity strategically in their models, they do not address how the capacities determined by their models can actually be provided in reality. In the second branch, operational problems associated with individual elements of airside infrastructure (which address what we earlier denoted by (c)) are considered. For example, in the runway scheduling problem, the optimal sequence of departing and/or arriving aircraft on one (or more) runway(s) is determined to maximize the throughput of a given runway system and/or to minimize deviations from the flight plan. A recent literature review on this problem is provided by Ikli et al. (2021). In the gate assignment problem, arriving aircraft are assigned to available and compatible gates. Numerous objectives have been proposed for this problem in the literature, and a recent survey is provided by Daş et al. (2020). Finally, in the aircraft ground routing problem, optimal taxi routes for aircraft between parking positions and runways are determined to minimize aircraft taxi times, fuel consumption, and/or delays (e.g., Guépet et al. 2016, Weiszer et al. 2020). However, in all three problems, the airport infrastructure is assumed to be given and cannot be changed. Thus, to the best of our knowledge, no literature exists to date that exam-

ines the extent to which the capacity of airport infrastructure can be optimized through (re)construction measures.

1.2 Scope and structure of the dissertation

In this dissertation, we address three planning problems related to the optimization of airport infrastructure. First, we consider the establishment of a shuttle network for a given airport using small, electrically powered, vertical takeoff and landing aircraft to improve airport accessibility. Such aircraft need special structures for landing, ground handling and take-off, also known as vertiports. We present a framework for determining the optimal locations for vertiports in the area surrounding an airport, with the objective to maximize the number of passengers who choose to travel to the airport by air taxi. In doing so, we explicitly take into account competing means of ground transport and the mode choice behavior of passengers.

In the second planning problem, we optimize the positions and orientations of aircraft parking positions close to a given airport terminal, considering two lexicographically ordered objectives. We first identify a layout that minimizes the overall number of aircraft weighted by size that cannot be parked near the terminal, incorporating the traffic characteristics at hub airports and the resulting variations in traffic volumes and fleet mix. Among all layouts that are optimal with respect to this objective, we then search for the layout that minimizes the number of parking positions that need to be built, weighted by the size of the largest aircraft that they are equipped for. Our model can be applied to both greenfield and brownfield instances, allowing it to be used whenever a new terminal is in planning or an existing one is to be rebuilt. The approach addresses the gap we identified above, as it enables airports to optimize the parking capacity of a given apron without using additional space.

This also applies to the third planning problem considered in this dissertation, which involves the recurring decision of which aircraft parking position at the facade of a given terminal should be equipped with which and how many passenger boarding bridge(s). We consider a planning horizon of up to 10 years and assume that the positions and orientations of the aircraft parking positions at the terminal facade are given. The objective is

to minimize the total cost for the airport over the time horizon, including investment and operating costs for passenger boarding bridges as well as penalty costs for aircraft that cannot be parked at a parking position close to the terminal. Using our approach allows airports to optimally adapt existing parking positions to the changing fleet mix over time in order to handle as many aircraft as possible at parking positions close to the terminal with existing apron and terminal space.

We apply a variety of modeling and solution methods in this dissertation. In the vertiport location problem, we incorporate a multinomial logit model into a hub location problem with multiple allocation and linearize the resulting model. In the apron layout planning problem, we first strengthen our original model formulation using the Bron-Kerbosch algorithm, and then apply a problem-specific decomposition approach as well as a number of acceleration techniques to solve the problem efficiently. Finally, we employ Dantzig-Wolfe decomposition to solve the third planning problem using a column generation heuristic. For all three planning problems, we provide extensive case studies for Munich Airport to showcase our approaches and to derive important managerial insights.

The remainder of this dissertation is structured as follows. In Chapter 2, we present our framework for determining the optimal locations for vertiports in the area surrounding an airport. This chapter is based on the working paper Hagspihl et al. (2023c), which is currently under review at OR Spectrum. Our approach to optimize the positions and orientations of aircraft parking positions close to an airport terminal is addressed in Chapter 3. This chapter is based on the working paper Hagspihl et al. (2023a), which is currently under review at Transportation Research Part B: Methodological after a major revision. Chapter 4 presents our work on the equipment of aircraft parking positions with passenger boarding bridges. In this chapter, we use several concepts already introduced in Chapter 3, but with a different notation. Chapter 4 is based on the paper Hagspihl et al. (2022), which was published in the European Journal of Operational Research in September 2022. The dissertation then comes to its end in Chapter 5 with some concluding remarks as well as an outlook on future research.

2 Planning an Airport Shuttle Network with Air Taxis

Airports and airlines share a common interest in providing passengers with fast and convenient access to airports. With the introduction of air taxis as a new means of transport, the opportunity will soon emerge to introduce air taxi shuttle services to airports. In this context, for each individual airport the question arises where the facilities for landing, ground handling and take-off, also called vertiports, should be located in the area surrounding the airport. We formulate the problem as a hub location problem with multiple allocation, with the objective to maximize the number of passengers who choose to reach the airport by air taxi. In our model, we explicitly incorporate passengers' choice behavior with respect to available means of transport using a multinomial logit model. We linearize the problem and provide guidance on the steps needed to use the model in practice. In an extensive case study, we apply the model to real data for Munich Airport, perform a stated preference study to parameterize the multinomial logit model, suggest locations for vertiports throughout Bavaria as a result, and investigate to what extent the introduction of air taxi shuttle flights can reduce travel times to Munich Airport.

2.1 Introduction

Current technological advances are paving the way for the imminent market launch of small, fully electric, vertical take-off and landing aircraft (eVTOL), also called air taxis in the following. For example, several manufacturers are planning the introduction of their first eVTOL models to the market for 2025 (e.g., Lilium 2022, SkyDrive 2023, Vertical Aerospace 2023). The concept of Advanced Air Mobility (AAM) envisions the use of air

taxis as a novel means of transport to move passengers and cargo safely and efficiently at low altitudes (Federal Aviation Administration 2022, Advanced Air Mobility Institute 2023). Under the umbrella of AAM, numerous potential applications for eVTOLs arise; for instance, the literature suggests that air taxis may be used for special purposes, in particular to create feeder networks for airports (e.g., Frej Vitalle et al. 2020, Straubinger et al. 2020b, Garrow et al. 2021, Schweiger and Preis 2022, Fu et al. 2022). Furthermore, the terms Urban Air Mobility and Regional Air Mobility refer to the use of air taxis in inner-city and inter-city point-to-point traffic, respectively, to relieve congested metropolitan areas (National Aeronautics and Space Administration 2021, Federal Aviation Administration 2022).

Regardless of the particular application, eVTOLs require appropriate ground infrastructure for take-off, landing, and all ground-based handling processes (Schweiger and Preis 2022). This infrastructure is commonly referred to as a vertiport (Rajendran and Shulman 2020). The design and operation of vertiports are not yet fully determined, which is why investment and operating costs of vertiports can currently only be roughly estimated. Depending on the size, location, design, and operation, we find estimates for the investment costs of a vertiport of up to €15 million, while predictions for annual operating costs per vertiport range range up to €17 million (Lilium 2020, McKinsey & Company 2020). As a result, the number of vertiports that can be opened at market launch is limited and locations must be chosen carefully. The selection of vertiport locations, as well as the construction of vertiports, will largely depend on the support of public authorities.

The deployment of Advanced Air Mobility faces a number of challenges. Passenger acceptance may suffer from initially high ticket prices, which both research and industry predict (Ploetner et al. 2020, Pukhova et al. 2021, Holden and Goel 2016). In addition, waiting times on the ground are to be expected before boarding and after deboarding, for example for security checks as well as loading and unloading processes, reducing the time savings that incentivize passengers to opt for an eVTOL flight. Moreover, Straubinger et al. (2020b) and Schweiger and Preis (2022) point to local safety regulations as well as political concerns, such as environmental issues, especially noise pollution, that could impede the implementation of AAM. In light of these challenges, we consider shuttle services to airports to be the most promising application area for air taxis at market launch. Airport passengers are likely to have a high willingness to pay for eVTOL flights, as timely arrival at the airport is particularly important (Frej Vitalle et al. 2020). Compared to

intra-city routes, trips to the airport can also be longer, as airports are often located remotely from the city center. As a result, the expected waiting times before boarding and after deboarding of air taxi flights are less problematic, as they will have a smaller relative impact on the overall travel time. Furthermore, safety and political concerns can be mitigated, because air taxis can begin their flights to the airport from the outskirts of cities, rather than from downtown.

While both public and research interest in AAM have grown significantly in the recent past, the majority of contributions to date have focused on aircraft technology and operational concepts, whereas the integration of AAM into existing transportation systems has received less attention (Garrow et al. 2021). We address this gap by proposing an approach to study optimal vertiport locations to create an air taxi feeder network for an airport, pursuing the objective to maximize the number of passengers who choose to reach the airport by air taxi.

We recognize that air taxis will compete with other means of transport from the moment they enter the market. Passengers are free to choose which means of transport they use to reach the airport, and it is also possible to change from one means of transport to another en route. When choosing a means of transport, passengers primarily take into account the travel times and travel costs of the available alternatives (Straubinger et al. 2020a,b). The travel times and travel costs associated with eVTOLs clearly depend on the locations of the vertiports. If, for example, from an individual passenger's point of view, a vertiport is located on the direct way from his/her home to the airport, the use of the air taxi is associated with shorter travel times and lower travel costs for this passenger than if the vertiport were located in the opposite direction. Thus, demand for eVTOL flights and optimal vertiport locations are interdependent.

Our contribution is threefold. To the best of our knowledge, we are the first to formulate this problem as an uncapacitated hub location problem with multiple allocation and integrated multinomial logit model to account for passengers' choice behavior with respect to competing means of transport. Second, we build on the ideas of Haase (2009) to linearize the model, prevent numerical instabilities, and eliminate redundant constraints. Third, we apply the resulting mixed-integer linear model to data from Munich Airport and provide managerial implications based on the results of an extensive case study. For this purpose, we conduct a stated preference study to parameterize the multinomial logit

model. We find optimal locations for vertiports and discuss the influence of variations in various parameters such as air taxi ticket prices and cruising speeds.

The remainder of this chapter is structured as follows. In Section 2.2, we review the related literature, before introducing and adjusting our model in Section 2.3. We discuss what data must be available and how this data can be obtained to apply the model to real life instances in Section 2.4. Section 2.5 presents the case study for Munich Airport including the stated preference survey, and we conclude the chapter in Section 2.6.

2.2 Related literature

2.2.1 Vertiport location problems

The selection of vertiport locations has been the subject of a number of papers in the recent past. However, most papers consider air taxis in the general context of Urban or Regional Air Mobility rather than as a new means of transport specifically for airport access. Furthermore, few papers explicitly consider the interdependence of location decisions and demand.

Existing quantitative work in the field can be divided into clustering-based analyses and approaches using mathematical programming (Sinha and Rajendran 2023). Clustering-based approaches are provided by Lim and Hwang (2019), Rajendran and Zack (2019), and Jeong et al. (2021). Sinha and Rajendran (2023) employ both a clustering algorithm and mathematical programming in a stepwise procedure. Wu and Zhang (2021) propose an integer program to identify optimal locations for vertiports in order to create an on-demand hub-and-spoke network using disaggregate demand data and apply their model to the Tampa Bay area in a detailed case study. While they do consider the interdependence of location decisions and demand for air taxis, they assume that travelers' mode choice results exclusively from the travel times and costs of available modes and the travelers' value of time. Willey and Salmon (2021) formulate an integer program to design an urban vertiport network and use subgraph isomorphism to ensure that the resulting network has a structure desired by the decision maker. They also take into account that vertiports can be used as transfer locations between air taxis, i.e., passengers can travel several

legs of their journey by air taxi. However, they use a comparatively simplistic model to estimate demand. Chen et al. (2022) present another mixed-integer program to identify optimal vertiport locations to create an urban hub-and-spoke system. In contrast to other contributions, they model the studied area as a grid. The demand between two cells is treated as a given, regardless of the vertiport locations. They develop a problem-specific variable neighborhood search to solve the problem efficiently, and in a case study apply their approach to the Beijing metropolitan region. Wang et al. (2022) model the interrelation of location decisions and passenger demand using a multinomial logit model, and develop an adaptive discretization scheme to solve the non-linear problem. However, instead of addressing the question of finding the optimal locations for vertiports from a purely strategic point of view, they take a holistic system perspective, from which they also consider tactical and operational aspects. Rath and Chow (2022) also propose an integer model for identifying optimal locations of vertiports. In contrast to all other contributions mentioned above, they consider air taxis as a new alternative means of transport specifically for airport access. Vertiports are located with the alternative goals to either maximize the number of passengers choosing air taxis to reach the airport, or to maximize the revenue generated by the air taxi feeder network. Similar to our approach, they employ a multinomial logit model to estimate the number of passengers who will use each vertiport depending on the location decisions taken. However, while we assume multiple allocation, i.e., passengers from one origin can make different choices, they assume single allocation, which simplifies the problem but is not realistic.

2.2.2 Location problems with endogenous demand

Determining optimal vertiport locations is a specific application of hub location problems, which in turn belong to the class of location problems. For an overview of location problems, we refer the reader to Owen and Daskin (1998), ReVelle and Eiselt (2005), Farahani and Hekmatfar (2009), Daskin (2011), and Laporte et al. (2019). Recent surveys focusing on hub location problems are provided by Alumur and Kara (2008) and Farahani et al. (2013). While many hub location problems consider systems in which the number of passengers between origin-destination pairs are given as fixed values (e.g., because only one transportation mode or service provider exists), we explicitly take into account that passengers have the freedom of choice to reach their destination with or without

using air taxis. That is, passengers compare the means of transport available to them and choose the one that provides them with the highest utility. Thus, assuming that travel time and travel costs significantly influence passengers' decisions (see, e.g., Fu et al. 2019), demand depends on the vertiport location decisions. Existing work on hub location problems under competition can be divided into approaches where gravity models are used to reflect the dependence of demand on location decisions (e.g., Eiselt and Marianov 2009), and approaches where discrete choice models are applied for that purpose (e.g., Aros-Vera et al. 2013, Lüer-Villagra and Marianov 2013, Kim and Shim 2021, Rezaei et al. 2022).

2.2.3 Multinomial logit models

Discrete choice models have been employed with great success on various occasions to model humans selecting one alternative from a finite set of alternatives. A comprehensive introduction to the field is provided by Train (2009). Multinomial logit models are predominantly used. Due to rigid assumptions, the predictions of these models can be less realistic than those of more sophisticated approaches such as mixed logit models, but they offer methodological advantages in terms of their integrability into optimization models. In particular, they exhibit the independence of irrelevant alternatives (IIA) property, which states that the relative ratio of the choice probabilities of two available alternatives remains constant as further alternatives are added to or removed from the choice set.

When using a multinomial logit model to depict the interrelation of demand and location decisions, the hub location problem becomes non-linear (Aros-Vera et al. 2013). Haase and Müller (2014) compare three different approaches from the existing literature to linearize the problem, finding that the approach developed independently by Haase (2009) and Aros-Vera et al. (2013) is the most efficient. In Section 2.3, we will therefore adapt their approach, which is based on the IIA property, to our model and extend it.

Different people typically have different preferences, and depending on the circumstances of the decision to be made, even a single person can arrive at different decisions when facing the same choice set. For example, we might expect lower-income individuals to be more price-sensitive than higher-income individuals, resulting in different choices. Furthermore, a single person might choose different means of transport depending on the purpose

of the trip. In consequence, multinomial logit models must be specified and estimated on the basis of empirical data appropriate to the particular application. Some papers specify and estimate multinomial logit models in the context of the introduction of air taxis. Fu et al. (2019) investigate the choice behavior of commuting passengers in the Munich region. They consider relatively short trips of 15km, and respondents can choose between car, public transport, autonomous ground taxi, and autonomous flight taxi. Their analysis focuses on the impact of travel time, travel costs, and the level of safety, as well as sociodemographic variables of the respondents on the choice behavior. Ilahi et al. (2020) provide a similar analysis for the Greater Jakarta area. Finally, Rimjha et al. (2021a) employ a mixed logit model to investigate mode choice behavior of commuting passengers in Northern California. Compared to Fu et al. (2019), the latter consider larger distances, as they take into account origin-destination pairs from an area with 150km radius. Unfortunately, we cannot use the findings from these papers for our Munich Airport case study, as the data were either collected in other areas of the world (Ilahi et al. 2020, Rimjha et al. 2021a), and/or the trip purpose was different (none of them considered airport shuttle services). Thus, we conduct our own survey for our case study (details will be discussed in Section 2.5).

2.3 Model

We identify optimal locations for vertiports by solving an uncapacitated hub location problem with multiple allocation that explicitly accounts for the interdependence of demand for air taxis and location decisions. In the following, we first introduce a discrete choice model, which we use to depict the interrelation between vertiport locations and demand in Section 2.3.1, before addressing the hub location problem itself in Section 2.3.2. The notation of both the discrete choice model and the hub location problem is summarized in Appendix A.1.

2.3.1 Modeling passenger demand

The demand for air taxis as a means of transport for airport access results from the number of passengers who choose to make use of an air taxi on their way to the airport.

We model passenger choice behavior using a discrete choice model. Passengers from each origin $o \in \mathcal{O}$ can choose between a finite number of itineraries $i \in \mathcal{I}$ to reach the airport. An itinerary $i \in \mathcal{I}$ defines which means of transport is/are used and where, if applicable, passengers change means of transport. For example, reaching the airport directly by car, without changing to other means of transport en route, could be one itinerary $i \in \mathcal{I}$. Another itinerary could be to reach a certain vertiport by public transport, and then fly to the airport by air taxi. We assume that passengers use air taxis for at most one leg of an itinerary, and that this leg is always the last on the way to the airport. In addition, we only consider itineraries with at most two legs. Note that our definition of an itinerary does not specify the starting point of the trip, so in principle all itineraries are available to passengers from all origins. However, as not all means of transport may be available at each origin, $\mathcal{I}_o \subseteq \mathcal{I}$ denotes the set of itineraries which passengers from origin $o \in \mathcal{O}$ can choose from. Furthermore, all purely ground-based itineraries are contained in set $\mathcal{I}^{\text{gnd}} \subset \mathcal{I}$, and all itineraries where passengers reach the airport by air taxi are contained in set $\mathcal{I}^{\text{air}} \subset \mathcal{I}$. Thus, $\mathcal{I} = \mathcal{I}^{\text{gnd}} \cup \mathcal{I}^{\text{air}}$. Note that the size of set $\mathcal{I}^{\text{air}}$ depends on the number of opened vertiports, as each additional vertiport adds at least one additional itinerary. Finally, let $\mathcal{I}_o^{\text{gnd}} = \mathcal{I}^{\text{gnd}} \cap \mathcal{I}_o$ and let $\mathcal{I}_o^{\text{air}} = \mathcal{I}^{\text{air}} \cap \mathcal{I}_o$.

In the following, we derive the probability p_{oi} that passengers from origin $o \in \mathcal{O}$ will choose itinerary $i \in \mathcal{I}_o$ based on Train (2009). For passengers from origin o, itinerary $i \in \mathcal{I}_o$ is associated with utility U_{oi}, and we assume that passengers exhibit a utility-maximizing behavior when choosing the transportation mode, i.e., choose itinerary $\hat{i} \in \mathcal{I}_o$ for which $U_{o\hat{i}} \geq U_{oi}$ holds for all $i \in \mathcal{I}_o \setminus \left\{\hat{i}\right\}$. While we cannot directly observe the values of U_{oi}, we can decompose U_{oi} into a deterministic component V_{oi}, which can be estimated based on passenger choices in empirical studies and which we will refer to as the representative utility, and a random component ϵ_{oi}, which denotes the difference between the estimated representative utility V_{oi} and the true utility U_{oi}:

$$U_{oi} = V_{oi} + \epsilon_{oi} \quad \forall o \in \mathcal{O}; i \in \mathcal{I}_o \tag{2.1}$$

The choice probability p_{oi} is then given as

$$\begin{aligned} p_{oi} &= p\left(U_{oi} \geq U_{oj} \quad \forall j \in \mathcal{I}_o \setminus \{i\}\right) \\ &= p\left(V_{oi} + \epsilon_{oi} \geq V_{oj} + \epsilon_{oj} \quad \forall j \in \mathcal{I}_o \setminus \{i\}\right) \\ &= p\left(\epsilon_{oj} - \epsilon_{oi} \leq V_{oi} - V_{oj} \quad \forall j \in \mathcal{I}_o \setminus \{i\}\right). \end{aligned} \tag{2.2}$$

Assuming that the error terms ϵ_{oi} are independently and identically Gumbel distributed, we obtain a multinomial logit model where p_{oi} is calculated (see, e.g., Train 2009, p. 36) as

$$p_{oi} = \frac{e^{V_{oi}}}{\sum\limits_{j \in \mathcal{I}_o} e^{V_{oj}}} \quad \forall o \in \mathcal{O}; i \in \mathcal{I}_o. \tag{2.3}$$

2.3.2 Hub location problem

Model formulation In addition to the notation already introduced, we denote the demand at origin $o \in \mathcal{O}$ by n_o. Furthermore, let $\mathcal{K} = \{1, \ldots, K\}$ be the set of potential vertiport locations, and let binary decision variable y_k be 1 if vertiport $k \in \mathcal{K}$ is opened, and 0, otherwise. For each itinerary $i \in \mathcal{I}^{\text{air}}$, we denote the vertiport $k \in \mathcal{K}$ where passengers transfer from any ground-based means of transport to air taxi by k_i. Then, we formulate the problem of optimally locating q vertiport locations as follows:

$$\max z = \sum_{o \in \mathcal{O}} \sum_{i \in \mathcal{I}_o^{\text{air}}} n_o \cdot p_{oi} \tag{2.4a}$$

subject to

$$\sum_{k \in \mathcal{K}} y_k = q \tag{2.4b}$$

$$p_{oi} = \frac{e^{V_{oi}} \cdot y_{k_i}}{\sum\limits_{j \in \mathcal{I}_o^{\text{gnd}}} e^{V_{oj}} + \sum\limits_{j \in \mathcal{I}_o^{\text{air}}} e^{V_{oj}} \cdot y_{k_j}} \qquad \forall o \in \mathcal{O}; i \in \mathcal{I}_o^{\text{air}} \tag{2.4c}$$

$$y_k \in \{0, 1\} \qquad \forall k \in \mathcal{K} \tag{2.4d}$$

$$0 \leq p_{oi} \leq 1 \qquad \forall o \in \mathcal{O}; i \in \mathcal{I}_o^{\text{air}} \tag{2.4e}$$

Objective Function (2.4a) maximizes the expected number of passengers who choose an itinerary $i \in \mathcal{I}^{\text{air}}$. Constraint (2.4b) sets the number of vertiports to be opened to the value of parameter q. Constraints (2.4c) are based on Eqs. (2.3) and determine the choice probabilities for each origin and itinenary combination, depending on which vertiports are opened. Finally, Constraints (2.4d) and (2.4e) define the variable domains.

Linearization Constraints (2.4c) are non-linear and thus have to be linearized. In view of the findings by Haase and Müller (2014), we linearize Constraints (2.4c) based on the approach proposed by Haase (2009) and Aros-Vera et al. (2013). The approach makes use of the fact that multinomial logit models exhibit the IIA characteristic. That is, the fraction of two choice probabilities p_{oi} and p_{oj} remains constant regardless of the number of options available in the choice set. Let r_{oi} denote the probability that passengers from origin $o \in \mathcal{O}$ use itinerary $i \in \mathcal{I}_o$ assuming all vertiports $k \in \mathcal{K}$ are opened (i.e., $y_k = 1$ for all $k \in \mathcal{K}$). Note that the values of r_{oi} can be computed in a preprocessing step, making r_{oi} a parameter. Then, we substitute Constraints (2.4c) by the following sets of constraints:

$$p_{oi} \leq y_{k_i} \qquad \forall o \in \mathcal{O}; i \in \mathcal{I}_o^{\text{air}} \tag{2.5a}$$

$$\sum_{i \in \mathcal{I}_o} p_{oi} = 1 \qquad \forall o \in \mathcal{O} \tag{2.5b}$$

$$r_{oj} \cdot p_{oi} \leq r_{oi} \cdot p_{oj} + \mathbb{1}\left(j \in \mathcal{I}^{\text{air}}\right)\left(1 - y_{k_j}\right) \qquad \forall o \in \mathcal{O}; i, j \in \mathcal{I}_o : i \neq j \tag{2.5c}$$

Constraints (2.5a) ensure for each origin $o \in \mathcal{O}$ and itinerary $i \in \mathcal{I}_o^{\text{air}}$ that the choice probability p_{oi} is 0 if vertiport k_i is not opened. Constraints (2.5b) make sure that all passengers from each origin have to select exactly one itinerary to get to the airport. Finally, Constraints (2.5c) ensure that the IIA property (i.e., $\frac{p_{oi}}{p_{oj}} = \frac{r_{oi}}{r_{oj}}$) is adhered to if both itineraries $i, j \in \mathcal{I}_o$ are available, where $\mathbb{1}(\cdot)$ represents the indicator function. That is, if $j \in \mathcal{I}_o^{\text{air}}$ and vertiport y_{k_j} is not opened, the $\left(1 - y_{k_j}\right)$ term ensures that the respective constraint is not binding.

Numerical instabilities Utilization probabilities p_{oi} may assume very small values for certain origins $o \in \mathcal{O}$ and itineraries $i \in \mathcal{I}_o$, resulting in numerical instabilities when solving the model. Following Haase (2009), we define a small value γ (e.g., $\gamma = 0.0001$) and reformulate Constraints (2.5c) as follows to prevent numerical instabilities:

$$r_{oj} \cdot p_{oi} \leq r_{oi} \cdot p_{oj} + \mathbb{1}\left(j \in \mathcal{I}^{\text{air}}\right)\left(1 - y_{k_j}\right) \quad \forall o \in \mathcal{O}; i, j \in \mathcal{I}_o : r_{oi}, r_{oj} \geq \gamma; i \neq j \tag{2.6a}$$

$$p_{oi} = 0 \quad \forall o \in \mathcal{O}; i \in \mathcal{I}_o : r_{oi} < \gamma \tag{2.6b}$$

Constraints (2.6a) are now only defined iff $r_{oi}, r_{oj} \geq \gamma$. Otherwise, if $r_{oi} < \gamma$, p_{oi} is set to 0 in Constraints (2.6b). As described by Haase (2009), this potentially leads to slight deviations of p_{oi} from the actual utilization probabilities; however, for sufficiently small values of γ, we expect these deviations to be negligible.

Redundant constraints Constraints (2.6a) contain redundant constraints due to transitivity (Haase 2009). For each origin $o \in \mathcal{O}$, let $\hat{i}_o$ denote the itinerary from set $\mathcal{I}_o^{\text{gnd}}$ with the highest choice probability given that all vertiports are opened, i.e., $r_{o\hat{i}_o} \geq r_{oi}$ for all $i \in \mathcal{I}_o^{\text{gnd}}$. Then, we substitute Constraints (2.6a) as follows:

$$r_{o\hat{i}_o} \cdot p_{oi} \leq r_{oi} \cdot p_{o\hat{i}_o} \quad \forall o \in \mathcal{O}; i \in \mathcal{I}_o : r_{oi} \geq \gamma; i \neq \hat{i}_o \tag{2.7a}$$

Note that the $\left(1 - y_{k_j}\right)$ term in Constraints (2.5c) and (2.6a) is no longer needed in Constraints (2.7a), as we chose itinerary $\hat{i}_o$ from set $\mathcal{I}_o^{\text{gnd}}$, i.e., $\hat{i}_o$ is always available regardless of the values of the y_k variables. While Constraints (2.6a) are defined for all tuples $i, j \in \mathcal{I}_o : r_{oi}, r_{oj} \geq \gamma; i \neq j$, only one constraint per itinerary $i \in \mathcal{I}_o : r_{oi} \geq \gamma; i \neq \hat{i}_o$ exists in Constraints (2.7a). However, we find that Constraints (2.7a) alone do not ensure that the IIA property is adhered to. For example, consider a situation with only one origin (i.e., we can drop the index o) and four itineraries $\mathcal{I} = \{1, 2, 3, 4\}$. Furthermore, let $r_i = (0.4, 0.2, 0.1, 0.3)$, where $r_i > \gamma \;\; \forall i \in \mathcal{I}$, and let $\mathcal{I}^{\text{gnd}} = \{1\}$. Now assume that the vertiport used on itinerary $i = 2$ is not opened, i.e., $p_2 = 0$ due to Constraints (2.5a),

but all other vertiports are opened. Due to the IIA property, the p_i variables should then take the values $(0.5, 0, 0.125, 0.375)$. However, with Constraints (2.5a), (2.5b), (2.6b), and (2.7a) only, there are infinitely many feasible combinations for the values of the p_i variables (for example, $p_i = (0.6, 0, 0.15, 0.25)$). In contrast to Haase (2009), we therefore add Constraints (2.8a) to the model.

$$r_{o\hat{i}_o} \cdot p_{oi} + \mathbb{1}\left(i \in \mathcal{I}^{\text{air}}\right)(1 - y_{k_i}) \geq r_{oi} \cdot p_{o\hat{i}_o} \qquad \forall o \in \mathcal{O}; i \in \mathcal{I}_o : r_{oi} \geq \gamma; i \neq \hat{i}_o \qquad (2.8a)$$

Together with Constraints (2.7a), Constraints (2.8a) ensure that the IIA property is always adhered to. However, the total number of Constraints (2.7a) and (2.8a) still is considerably smaller than the number of Constraints (2.6a).

2.4 Model application

To apply Model (2.4a)-(2.4b), (2.4d)-(2.4e), (2.5a)-(2.5b), (2.6b), (2.7a)-(2.8a), several preparation steps are necessary. In particular, some modeling decisions have to be made and appropriate data has to be acquired. Often, the lack of available data severely limits the modeling possibilities, so both processes should be considered simultaneously. The notation introduced in this section is summarized in Appendix A.1.

2.4.1 Modeling decisions

Scope and granularity First, the planning territory and the planning granularity must be defined to obtain the set of origins $\mathcal{O}$. Both can be based on geometric as well as administrative considerations. For example, the territory can be defined by a circle with a certain radius around the airport, or by (federal) state borders surrounding the airport, and the territory can be divided into origin cells based on regional administrative boundaries (such as postal codes or municipal boundaries) or based on a custom grid structure. In line with expert interviews that we conducted, we expect that administrative concerns will usually be the determining factor, as the selection of vertiport locations and the construction of vertiports will most probably be the responsibility of public authorities.

Vertiport locations Subsequently, the set of potential vertiport locations $\mathcal{K}$ has to be defined. Again, the selection of potential vertiport locations can be based on diverse criteria, such as legal, political, or technological concerns. For example, constructing a vertiport may not be feasible and/or desired in densely populated areas, and the expected operating range of air taxis may prohibit vertiports located too far from the airport. In addition, regional circumstances can affect the selection of potential vertiport sites. For example, in highly regulated areas of the world (such as the European Union), vertiports might naturally be established at existing small airfields where take-offs and landings are already authorized. Expert interviews and/or the use of focus groups can help identify the set of potential vertiport locations. $\mathcal{K}$ does not have to be a subset of $\mathcal{O}$, i.e., potential vertiport locations can be selected regardless of how the set of origins is defined.

Itineraries The set of itineraries $\mathcal{I}$ should reflect all options passengers can choose from to reach the airport. In this work, we consider four types of itineraries: Passengers can reach the airport

- directly by car,
- directly by public transport (denoted by PT in the following),
- by air taxi, reaching the vertiport by car (car_AT), and
- by air taxi, reaching the vertiport by public transport (PT_AT).

In the following, we denote these sets of itineraries by $\mathcal{I}^{\text{car}}$, $\mathcal{I}^{\text{PT}}$, $\mathcal{I}^{\text{car\_AT}}$, and $\mathcal{I}^{\text{PT\_AT}}$, respectively. As not all itineraries may be available at all origins, let $\mathcal{I}_o^{\text{car}}$, $\mathcal{I}_o^{\text{PT}}$, $\mathcal{I}_o^{\text{car\_AT}}$, and $\mathcal{I}_o^{\text{PT\_AT}}$ be the sets of itineraries available to passengers at origin $o \in \mathcal{O}$. Furthermore, we define $\mathcal{I}_o^{\text{gnd}} = \mathcal{I}_o^{\text{car}} \cup \mathcal{I}_o^{\text{PT}}$ and $\mathcal{I}_o^{\text{air}} = \mathcal{I}_o^{\text{car\_AT}} \cup \mathcal{I}_o^{\text{PT\_AT}}$. Additional types of itineraries (e.g., car sharing) can be added to $\mathcal{I}$ and all subsets we introduced accordingly.

Representative utilities Next, the representative utility V_{oi} that passengers from origin o have when choosing itinerary i needs to be determined for each origin $o \in \mathcal{O}$ and itinerary $i \in \mathcal{I}_o$. We calculate the values of V_{oi} as

$$V_{oi} = \beta_i^{\text{asc}} + \sum_{q \in \mathcal{Q}} \beta_i^q \cdot \eta_{oi}^q \quad \forall o \in \mathcal{O}; i \in \mathcal{I}_o, \tag{2.9}$$

where

- β_i^{asc} represents the alternative-specific constant of itinerary $i \in \mathcal{I}$,
- $\mathcal{Q}$ denotes the set of criteria considered by the passengers when evaluating alternative itineraries,
- β_i^q is a weighting parameter for itinerary $i \in \mathcal{I}$ with respect to criterion $q \in \mathcal{Q}$, and
- η_{oi}^q is the performance of itinerary $i \in \mathcal{I}_o$ for passengers from origin $o \in \mathcal{O}$ with respect to criterion $q \in \mathcal{Q}$.

The most important criteria passengers consider when choosing an itinerary are travel times and travel costs (Straubinger et al. 2020b,a). Travel times include waiting times such as transfer times at vertiports, which can account for a significant portion of travel times on itineraries $i \in \mathcal{I}^{\text{air}}$. Let $\beta^{\text{car,asc}}$, $\beta^{\text{PT,asc}}$, $\beta^{\text{car\_AT,asc}}$, and $\beta^{\text{PT\_AT,asc}}$ denote the alternative-specific constants for the respective itinerary types. Similarly, let $\beta^{\text{car},tt}$, $\beta^{\text{PT},tt}$, $\beta^{\text{car\_AT},tt}$, and $\beta^{\text{PT\_AT},tt}$ denote the itinerary type specific weighting factors for travel time, and let $\beta^{\text{car},tc}$, $\beta^{\text{PT},tc}$, $\beta^{\text{car\_AT},tc}$, and $\beta^{\text{PT\_AT},tc}$ be the itinerary type specific weighting factors for travel costs. Finally, let η_{oi}^{tt} and η_{oi}^{tc} denote the travel time and travel costs of itinerary $i \in \mathcal{I}$ for passengers at origin $o \in \mathcal{O}$. Then, we determine the representative utilities for all $o \in \mathcal{O}$ and $i \in \mathcal{I}_o$ as:

$$V_{oi} = \begin{cases} \beta^{\text{car,asc}} + \beta^{\text{car},tt} \cdot \eta_{oi}^{tt} + \beta^{\text{car},tc} \cdot \eta_{oi}^{tc} & \forall i \in \mathcal{I}_o^{\text{car}} \\ \beta^{\text{PT,asc}} + \beta^{\text{PT},tt} \cdot \eta_{oi}^{tt} + \beta^{\text{PT},tc} \cdot \eta_{oi}^{tc} & \forall i \in \mathcal{I}_o^{\text{PT}} \\ \beta^{\text{car\_AT,asc}} + \beta^{\text{car\_AT},tt} \cdot \eta_{oi}^{tt} + \beta^{\text{car\_AT},tc} \cdot \eta_{oi}^{tc} & \forall i \in \mathcal{I}_o^{\text{car\_AT}} \\ \beta^{\text{PT\_AT,asc}} + \beta^{\text{PT\_AT},tt} \cdot \eta_{oi}^{tt} + \beta^{\text{PT\_AT},tc} \cdot \eta_{oi}^{tc} & \forall i \in \mathcal{I}_o^{\text{PT\_AT}} \end{cases} \tag{2.10}$$

2.4.2 Data acquisition

Passenger data For each origin cell $o \in \mathcal{O}$, the number of airport passengers n_o needs to be known. The values of n_o can be obtained from ticket sales data or passenger surveys. Alternatively, n_o can also be estimated for each origin $o \in \mathcal{O}$ based on population data, income data, and the distances to the airport considered as well as adjacent airports. Of

course, n_o should only be estimated if no true passenger data is available. As a result, the modeling decisions about what area to consider and how to divide the area into origin cells will often depend on the availability of adequate passenger data.

Itinerary data To calculate the representative utility V_{oi} for each origin $o \in \mathcal{O}$ and itinerary $i \in \mathcal{I}_o$ as defined in Eqs. (2.10), two sets of data must be acquired. First, the values of all weighting factors and alternative-specific constants β need to be estimated based on empirical data, which we will showcase in our case study in Section 2.5. Typically, a stated preference survey has to be performed.

Second, travel times and travel costs for each origin $o \in \mathcal{O}$ and itinerary $i \in \mathcal{I}_o$, denoted by η_{oi}^{tt} and η_{oi}^{tc} in Eqs. (2.10), must be determined. For this purpose, various sources and methods can be applied. We use Google Maps to determine travel times from all origins $o \in \mathcal{O}$ to all vertiports $k \in \mathcal{K}$ and to the airport (abbreviated as "apt" in the following), for both car and public transport. Let $t_{o,k}^{\text{car}}$, $t_{o,\text{apt}}^{\text{car}}$, $t_{o,k}^{\text{PT}}$, and $t_{o,\text{apt}}^{\text{PT}}$ denote the respective travel times. For each origin $o \in \mathcal{O}$, we determine the travel times for all itineraries $i \in \mathcal{I}_o^{\text{car}}$ as the sum of $t_{o,\text{apt}}^{\text{car}}$ and $t^{\text{car+}}$, where $t^{\text{car+}}$ denotes the time needed to park the car. Similarly, the travel times for all itineraries $i \in \mathcal{I}_o^{\text{PT}}$ are calculated as the sum of $t_{o,\text{apt}}^{\text{PT}}$ and $t^{\text{PT+}}$, where $t^{\text{PT+}}$ represents the waiting time at the departure station of the public transport service. Similar to travel times, we denote the ground (i.e., street) distances between origin $o \in \mathcal{O}$, vertiport $k \in \mathcal{K}$, and the airport by $d_{o,k}$ and $d_{o,\text{apt}}$, and the direct (i.e., Euclidean) distance between vertiport $k \in \mathcal{K}$ and the airport by $d_{k,\text{apt}}$. While numerous sources and methods can be used to determine all distances, we use Google Maps. We approximate the travel times of air taxis from each vertiport $k \in \mathcal{K}$ to the airport, denoted by $t_{k,\text{apt}}^{\text{AT}}$, by dividing $d_{k,\text{apt}}$ by the air taxi cruising speed v^{AT}. For each origin $o \in \mathcal{O}$, the travel times for itineraries $i \in \mathcal{I}_o^{\text{car\_AT}}$ are calculated as the sum of t_{o,k_i}^{car}, $t_{k_i,\text{apt}}^{\text{AT}}$, $t^{\text{car+}}$, and $t^{\text{AT+}}$, which denotes the transfer time at the vertiport. The travel times for itineraries $i \in \mathcal{I}_o^{\text{PT\_AT}}$ are determined analogously.

The travel costs for itineraries $i \in \mathcal{I}_o^{\text{car}}$ are computed by adding a fixed component $c^{\text{car,fix}}$, which accounts for the parking cost at the airport, to a distance dependent component $c^{\text{car,var}} \cdot d_{o,\text{apt}}$. For itineraries $i \in \mathcal{I}_o^{\text{PT}}$ we consider three cost levels $c^{\text{PT,low}}, c^{\text{PT,mid}}, c^{\text{PT,high}}$, depending on the location of origin o and the travel time $t_{o,\text{apt}}^{\text{PT}}$. If o is located in the same local transport network as the airport or if $t_{o,\text{apt}}^{\text{PT}} \leq 90\,\text{min}$, we assume travel costs

of $c^{\text{PT,low}}$. Otherwise, if $90\,\text{min} < t^{\text{PT}}_{o,\text{apt}} < 180\,\text{min}$, we set the travel costs to $c^{\text{PT,mid}}$, and if $t^{\text{PT}}_{o,\text{apt}} \geq 180\,\text{min}$, we set the travel costs to $c^{\text{PT,high}}$. In line with Rimjha et al. (2021b), we model air taxi ticket prices as the sum of a fixed cost component $c^{\text{AT,fix}}$ and a distance dependent component $c^{\text{AT,var}} \cdot d_{k_i,\text{apt}}$. For itineraries $i \in \mathcal{I}_o^{\text{car\_AT}}$, the travel costs then equal the sum of the costs of reaching vertiport $k_i \in \mathcal{K}$ by car (including the fixed costs for parking the car at the vertiport) and the air taxi ticket price from vertiport k_i to the airport. Similarly, the travel costs for itineraries $i \in \mathcal{I}_o^{\text{PT\_AT}}$ are determined as the sum of the respective costs for public transport and air taxi.

We summarize how we determine travel times and travel costs for all itineraries in Eqs. (2.11) and (2.12).

$$\eta^{tt}_{oi} = \begin{cases} t^{\text{car}}_{o,\text{apt}} + t^{\text{car+}} & \forall i \in \mathcal{I}_o^{\text{car}} \\ t^{\text{PT}}_{o,\text{apt}} + t^{\text{PT+}} & \forall i \in \mathcal{I}_o^{\text{PT}} \\ t^{\text{car}}_{o,k_i} + t^{\text{AT}}_{k_i,\text{apt}} + t^{\text{car+}} + t^{\text{AT+}} & \forall i \in \mathcal{I}_o^{\text{car\_AT}} \\ t^{\text{PT}}_{o,k_i} + t^{\text{AT}}_{k_i,\text{apt}} + t^{\text{PT+}} + t^{\text{AT+}} & \forall i \in \mathcal{I}_o^{\text{PT\_AT}} \end{cases} \tag{2.11}$$

$$\eta^{tc}_{oi} = \begin{cases} c^{\text{car,fix}} + c^{\text{car,var}} \cdot d_{o,\text{apt}} & \forall i \in \mathcal{I}_o^{\text{car}} \\ \in \left\{ c^{\text{PT,low}}, c^{\text{PT,mid}}, c^{\text{PT,high}} \right\} & \forall i \in \mathcal{I}_o^{\text{PT}} \\ c^{\text{car,fix}} + c^{\text{car,var}} \cdot d_{o,k_i} + c^{\text{AT,fix}} + c^{\text{AT,var}} \cdot d_{k_i,\text{apt}} & \forall i \in \mathcal{I}_o^{\text{car\_AT}} \\ \in \left\{ c^{\text{PT,low}}, c^{\text{PT,mid}}, c^{\text{PT,high}} \right\} + c^{\text{AT,fix}} + c^{\text{AT,var}} \cdot d_{k_i,\text{apt}} & \forall i \in \mathcal{I}_o^{\text{PT\_AT}} \end{cases} \tag{2.12}$$

Once all values of V_{oi} have been determined, Eqs. (2.3) can be used to calculate choice probabilities r_{oi} for each origin $o \in \mathcal{O}$ and itinerary $i \in \mathcal{I}_o$, assuming that all vertiports are opened. Then, all sets and parameters are defined and the model can be solved to identify the optimal q vertiport locations from set $\mathcal{K}$.

2.5 Case study

In our case study we consider the introduction of an air taxi shuttle network for Munich Airport. We first explain the general setting in Section 2.5.1; then, in Section 2.5.2, we elaborate on the stated preference survey we conducted to estimate all weighting factors β in Eqs. (2.10); finally, we present our computational experiments in Section 2.5.3.

2.5.1 General setting

Motivation With 47.9 million passengers handled in 2019, Munich Airport is among the ten largest airports in Europe (Munich Airport 2020). In 2015, the airport was the first in Europe to be awarded a five-star rating by the renowned rating organization Skytrax (Munich Airport 2023a). The Skytrax airport rating considers key aspects of passenger perception, such as service quality, ambience, cleanliness, and comfort in all areas of the airport. The five-star rating was reconfirmed by Skytrax in 2017 and 2020, validating the high level of service offered to passengers at Munich Airport. However, compared to similar airports, Munich Airport is less accessible, both from the point of view of its immediate surroundings including the city of Munich, and from a supra-regional perspective. The airport is located about 30 kilometers northeast of Munich and is connected to the city only by a highway and two suburban train lines. It takes about 50 minutes to reach the airport from Munich city center by suburban train, and the suburban train lines suffer from poor reliability (Schubert 2023). From a supra-regional perspective, the airport lacks regional and long-distance train connections, which, in contrast, are available at Frankfurt Airport or Amsterdam Airport, for example. As a result, Munich Airport in particular can benefit from the introduction of an air taxi shuttle network, providing passengers with a fast and convenient access option.

Origins and vertiport locations We consider the entire territory of the Free State of Bavaria as the catchment area of Munich Airport. To determine the set of origins $\mathcal{O}$ as well as the number of passengers n_o for each origin $o \in \mathcal{O}$, we employ the results of a passenger survey that was conducted at Munich Airport in 2019, where approximately $30,000$ passengers were asked for the postal code from which they started their journey to

the airport, as well as their residential postal code. The results are extrapolated to annual traffic. The survey distinguishes between passengers traveling for business and leisure. In addition, the survey differentiates between passengers departing from Munich Airport for their outbound flight (source traffic) and passengers departing from Munich Airport for their return flight (visitor traffic). In the following, we consider source traffic only, since visiting passengers might adjust their travel plans when air taxis are introduced, whereas we do not expect people to change residence based on vertiport locations. Furthermore, we only consider respondents who started their journey to the airport from their residential postal code (about 95% of the source traffic). We define set $\mathcal{O}$ as the set of postal code areas in Bavaria that were specified by at least one respondent. In consequence, set $\mathcal{O}$ consists of $1,125$ postal codes and n_o has values between 49 and $223,535$. In total, $7,376,700$ passengers are considered.

We define the set of potential vertiport locations $\mathcal{K}$ based on interviews we conducted with planning experts from Munich Airport. To account for the maximum range of air taxis, all locations are within a maximum range of 150 kilometers (Euclidean distance) from Munich Airport. In addition, either an airport, airfield, or heliport already exists at all locations. In collaboration with planning experts from Munich Airport, we identified locations which are distributed fairly evenly across Bavaria (except northern Bavaria; this is due to the maximum distance between vertiport and airport of 150 kilometers). In total, this yields a set of 24 potential vertiport locations (12 airports/airfields and 12 helipads). Figure 2.1 shows a map of Bavaria divided into postal codes, the location of Munich Airport, and the potential vertiport locations. Postal code areas $o \in \mathcal{O}$ are colored depending on n_o, with darker colors (as in and around Munich, for example) representing higher values.

2.5.2 Stated preference survey

We estimate the values of all β parameters based on the results of a stated preference survey that we conducted online in May 2023. The survey consisted of four parts. First, participants were introduced to Advanced Air Mobility and provided with relevant information on eVTOLs. Second, respondents were asked several questions regarding their current transportation mode choice behavior when travelling to Munich Airport. For the third part, participants were asked to imagine themselves in the scenario of wanting

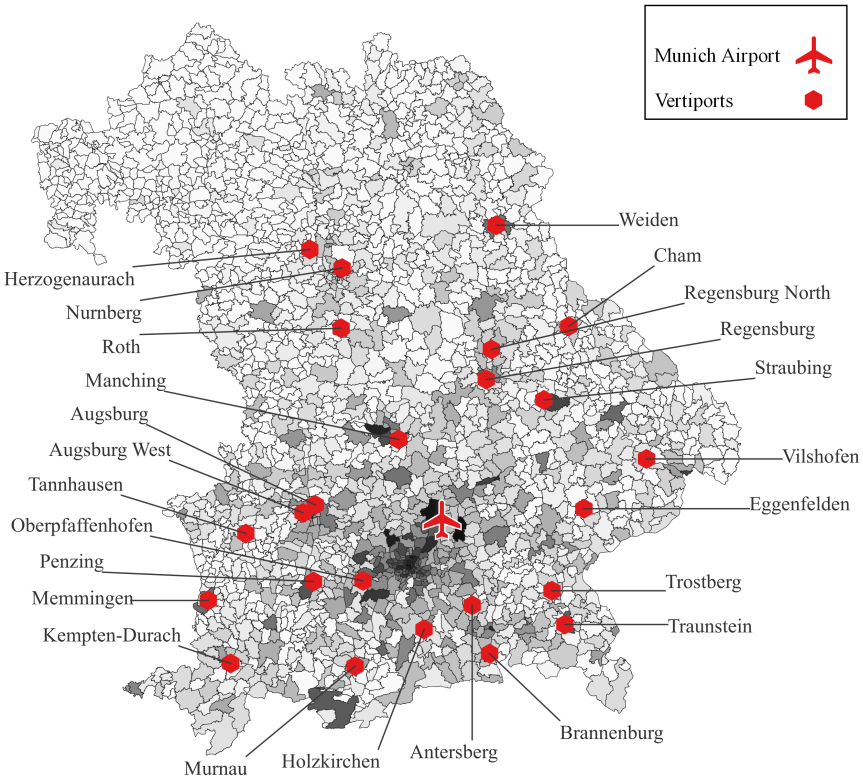


Figure 2.1: Map of postal codes in Bavaria and potential vertiport locations

to travel to Munich Airport, with the distance between the starting point of their trip and Munich Airport provided. Then, respondents were presented with twelve consecutive single-choice tasks, each asking them to choose among a set of itineraries to reach the airport. In each choice task, participants were offered four labeled itineraries (one itinerary of each type car, PT, car_AT, and PT_AT), and they were provided with travel times and travel costs for each itinerary. In addition, participants could indicate that they would not choose any of the provided options in each choice task. The choice tasks differed in terms of the values of travel times and travel costs of the itineraries offered. In the fourth and final part of the survey, participants were asked for socio-demographic variables like age, gender, postal code, and income.

Since passengers live at different distances from Munich Airport and the results of our survey should be valid for all passengers independent of the distance from their origin to the airport, respondents were assigned different distances at the beginning of the third part of the survey (either 50, 100, or 150 kilometers) and the travel times and travel costs of all itineraries offered in the choice tasks depended on the distance. Each respondent was assigned only one distance - that is, the scenario did not change across the twelve choice tasks - and the assignment of distances to participants was random. Furthermore, respondents were not provided with the total travel time per itinerary, but with the net travel time (i.e., time spent moving in the vehicle(s)) separate from the waiting time. However, the impact of waiting time on the participants' choices was insignificant, so we calculate travel time values in Eqs. (2.10) as the sum of the values of net travel time and waiting time from the survey. For each distance, the attribute values for travel costs, net travel times, and waiting times used in the survey are provided in Table 2.1.

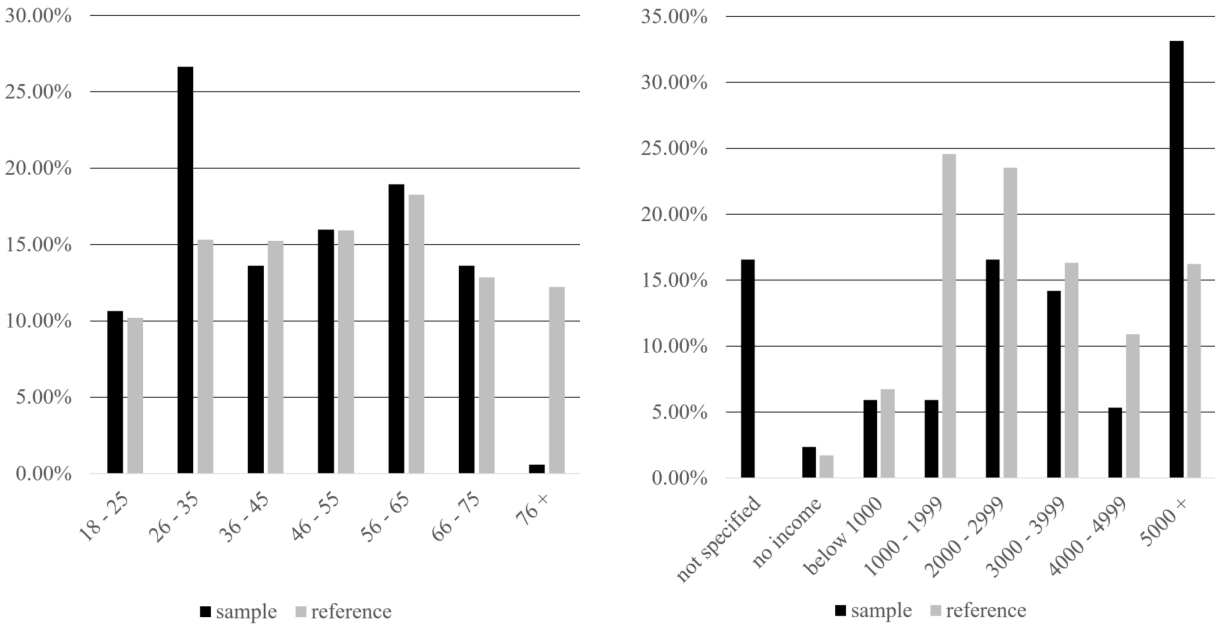
Table 2.1: Attribute values used in the stated preference survey

Attribute	Itinerary type	Distance		
		50 km	100 km	150 km
Travel costs [€]	car	$40, 55, 70$	$70, 90, 110$	$90, 120, 150$
	PT	$20, 25, 30$	$30, 40, 50$	$30, 60, 90$
	car_AT	$50, 115, 180$	$80, 210, 340$	$110, 305, 500$
	PT_AT	$35, 105, 175$	$50, 190, 330$	$65, 275, 485$
Net travel time [min]	car	$30, 40, 50$	$60, 80, 100$	$75, 100, 125$
	PT	$55, 75, 95$	$80, 110, 140$	$125, 170, 215$
	car_AT	$15, 30, 45$	$35, 60, 85$	$45, 75, 105$
	PT_AT	$25, 50, 75$	$40, 75, 110$	$55, 115, 175$
Waiting time [min]	car	5	5	5
	PT	10	10	10
	car_AT	$10, 15, 20$	$10, 15, 20$	$10, 15, 20$
	PT_AT	$15, 20, 25$	$15, 20, 25$	$15, 20, 25$

Based on the results of a pilot study, we created an efficient design for the choice tasks, using the commercial software Ngene. We ensured that no choice task had a dominant

alternative, i.e., an alternative with lower travel costs, shorter net travel time, and shorter waiting time than all other alternatives.

Sample data The survey was distributed on various social networks, inviting a wide range of people to participate. A total of 251 people participated in the survey, 169 of whom remained in the sample after filtering out respondents who were not from Bavaria, were under 18 years old, and/or completed the survey remarkably fast (i.e., in less than a third of the average completion time that we observed in the pilot study). Of all respondents, 43% reported being female, 56% male, and 1% diverse. In Figures 2.2a and 2.2b, we compare the distribution of age and net monthly household income among our sample with reference data available for Germany (Statistisches Bundesamt 2023a,b).



(a) Age

(b) Net monthly houshold income

Figure 2.2: Distribution of socio-demographic data in the sample compared to reference data

We observe a reasonable fit between the age distributions of the participants in our survey and the German population. The overrepresentation of participants between the ages of 26 and 35 and the underrepresentation of respondents over the age of 76 can be explained

by the fact that our survey was only distributed online and that young people are more likely to use social networks than older people. As for the net monthly houshold income data, low and medium incomes between $1,000€$ and $4,999€$ are underrepresented in our sample, while high incomes of more than $5,000€$ are overrepresented. We identify the following reasons for these deviations: First, many participants ($> 15\%$) preferred not to report their income. We cannot examine how this actually affects the distribution, but we assume that primarily participants with low and medium incomes did not want to disclose their income, even though the survey was conducted anonymously. Second, the reference data covers Germany as a whole, whereas our sample is restricted to participants from Bavaria, where incomes are higher on average than in Germany as a whole (Statistisches Bundesamt 2022). Finally, we shared the survey through our own social networks, reaching many colleagues in academia and industry with high levels of education and high average incomes.

Estimating the multinomial logit model The participants chose the car itinerary in 24.40%, the PT itinerary in 43.76%, the car_AT itinerary in 12.47%, and the PT_AT itinerary in 19.37% of all choice tasks, respectively. To estimate the multinomial logit model based on the survey results, we used the Apollo package for R (Hess and Palma 2019). We selected $\mathcal{I}^{\text{PT}}$ as reference category, and thus set the value of $\beta_{\text{PT},0}$ to 0. The resulting estimates and performance statistics of the model are provided in Table 2.2.

Table 2.2: Estimated coefficients and model statistics

Attribute	Itinerary type			
	car	PT	car_AT	PT_AT
Alternative-specific constant	-0.92993^{*}	–	-0.24990	0.15572
Travel time	-0.01191^{*}	-0.01592^{*}	-0.01430^{*}	-0.01643^{*}
Travel costs	-0.01951^{*}	-0.02259^{*}	-0.02384^{*}	-0.02245^{*}
Initial log-likelihood	$-2,755.95$			
Final log-likelihood	$-1,901.73$			
Adjusted Rho-squared vs. equal shares	0.31			

* highly significant ($p < 0.01$)

Besides $\beta_{\text{car\_AT,asc}}$ and $\beta_{\text{PT\_AT,asc}}$, all estimated coefficients are highly significant ($p < 0.01$). For itinerary types car, PT, car_AT, and PT_AT the resulting values of time are $36.62\frac{€}{\text{hr}}$, $42.27\frac{€}{\text{hr}}$, $35.99\frac{€}{\text{hr}}$, and $43.92\frac{€}{\text{hr}}$, respectively.

2.5.3 Computational experiments

We first define a base case for our experiments in Section 2.5.3.1. After inspecting the optimal solution for this base case, we create a series of scenarios in the following sections.

2.5.3.1 Base case

We use the parameter values provided in Table 2.3 to calculate travel times and travel costs for all origins $o \in \mathcal{O}$ and itineraries $i \in \mathcal{I}_o$ as stated in Eqs. (2.11) and (2.12).

Table 2.3: Parameter values used to calculate travel time and travel costs in the base case

Parameter	Value
$t^{\text{car+}}$	5 min
$t^{\text{PT+}}$	10 min
$t^{\text{AT+}}$	15 min
v^{AT}	$200\frac{\text{km}}{\text{h}}$
$c^{\text{car,fix}}$	20€
$c^{\text{car,var}}$	$0.68\frac{€}{\text{km}}$
$\{c^{\text{PT,low}}, c^{\text{PT,mid}}, c^{\text{PT,high}}\}$	{15€, 25€,50€}
$c^{\text{AT,fix}}$	50€
$c^{\text{AT,var}}$	$2.50\frac{€}{\text{km}}$

The value of the transfer time $t^{\text{AT+}}$ results from expert interviews. We set the air taxi cruising speed v^{AT} to $200\frac{\text{km}}{\text{h}}$ based on available information from air taxi manufacturers (e.g., Lilium 2022, Vertical Aerospace 2023). The cost per kilometer of driving $c^{\text{car,var}}$ originates from information provided by the German automobile club ADAC, which regularly calculates and publishes the total cost per kilometer for a variety of car models with different specifications. We use the median value from the data for the 2021/2022

winter season (Allgemeiner Deutscher Automobil-Club e. V. 2021). The public transport fares are derived from different tariffs available in the Munich region. As for ticket prices for air taxi flights, we find a wide variety of estimates in the literature. Clearly, ticket prices will be high upon market introduction and decrease over time. Furthermore, they also depend on the seating capacity of the air taxi due to economies of scale. Ticket prices per passenger are often quoted in U.S. dollars per mile ($\frac{\$}{\text{mi}}$). Overall, we find estimates between $0.50\frac{\$}{\text{mi}}$ and $9.50\frac{\$}{\text{mi}}$ per passenger (Holden and Goel 2016, Goyal et al. 2021). Finally, for the base case we assume that upon market entry five vertiports are to be opened, i.e., $q = 5$.

Computational results With the given objective and parameter values, vertiports are opened at the five locations given in Figure 2.3.

In total, 6.22% of all passengers (458, 560 passengers) choose to reach the airport by air taxi, where 48.26% of these passengers reach the vertiport by car, and the remaining 51.74% by public transport. 32.14% of all passengers choose the car to reach the airport directly, and 61.64% choose public transport. The opened vertiports are located comparatively close to Munich Airport, for which we identify two reasons: First, the farther a vertiport is located from the airport, the greater the absolute difference between air taxi costs and the costs of purely ground-based transportation. As passengers weigh costs more heavily than time (see Table 2.2), the utilities V_{oi} of the air taxi itineraries $\mathcal{I}^{\text{air}}$ generally decrease relative to the utilities of the ground based alternatives $\mathcal{I}^{\text{gnd}}$ for longer distances from vertiports to Munich Airport. Second, the average number of passengers per postal code decreases with increasing distance from Munich Airport, so that more distant vertiports are viable for fewer passengers.

2.5.3.2 Varying the number of opened vertiports

We now vary the number of vertiports to be opened. For this purpose, we solve the problem for $q \in \{1, 3, 5, 7, 9\}$. For each value of q and each vertiport k that is opened for the respective value of q, Table 2.4 gives the share of passengers who choose to reach Munich Airport by air taxi via vertiport k (regardless of the means of transport used to reach the vertiport). Across all values of q, between 2.27% and 7.36% of all passengers

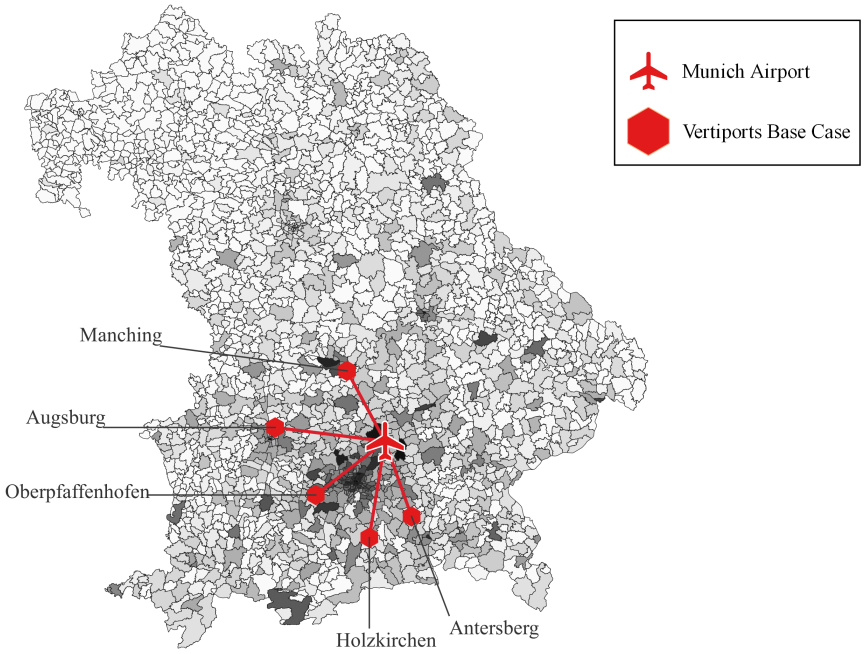


Figure 2.3: Vertiport locations opened in the base case

choose to reach Munich Airport by air taxi. The optimal solutions which vertiport(s) to open are consistent across different values of q, i.e., if a vertiport is opened for a given value of q, it remains opened when q is increased to $q+2, q+4, \ldots, 9$. Furthermore, opening an additional vertiport has little impact on the passenger numbers at vertiports that are already opened. E.g., the utilization of the vertiport in Oberpfaffenhofen reduces from 2.27% ($q = 1$) to 2.22% ($q = 3$) when two more vertiports are opened in Manching and Holzkirchen. The reasons for this are, on the one hand, the large distances between most of the opened vertiports, in particular at small values of q, and, on the other hand, the IIA characteristic of the discrete choice model.

Table 2.4: Air taxi utilization per vertiport and in total when q is varied

q	Vertiport*									$\sum$
	I	II	III	IV	V	VI	VII	VIII	IX	
1	2.27%	–	–	–	–	–	–	–	–	2.27%
3	2.22%	1.52%	0.96%	–	–	–	–	–	–	4.70%
5	2.18%	1.50%	0.94%	0.85%	0.75%	–	–	–	–	6.22%
7	2.15%	1.48%	0.93%	0.83%	0.75%	0.50%	0.35%	–	–	6.99%
9	2.14%	1.48%	0.93%	0.82%	0.75%	0.50%	0.35%	0.20%	0.19%	7.36%

* The Roman numerals represent the following vertiports: I = Oberpfaffenhofen, II = Manching, III = Holzkirchen, IV = Augsburg, V = Antersberg, VI = Augsburg West, VII = Penzing, VIII = Brannenburg, IX = Regensburg

2.5.3.3 Varying air taxi travel costs and travel times

We now vary both travel costs and travel times of air taxi flights within realistic ranges and examine the impact on air taxi utilization. For this purpose, we vary the values of the fix costs per air taxi ticket $c^{\text{AT,fix}}$ (0€, 25€, 50€, 75€, 100€), the variable cost per air taxi passenger and kilometer $c^{\text{AT,var}}$ ($0.5\frac{€}{\text{km}}$, $1.5\frac{€}{\text{km}}$, $2.5\frac{€}{\text{km}}$, $3.5\frac{€}{\text{km}}$, $4.5\frac{€}{\text{km}}$), the transfer time at the vertiports $t^{\text{AT+}}$ (5 min, 10 min, 15 min, 20 min, 35 min), and the air taxi cruising speed v^{AT} ($100\frac{\text{km}}{\text{h}}$, $150\frac{\text{km}}{\text{h}}$, $200\frac{\text{km}}{\text{h}}$, $250\frac{\text{km}}{\text{h}}$, $300\frac{\text{km}}{\text{h}}$). We use a ceteris paribus design, i.e., only one parameter is varied at a time. The results are shown in Figures 2.4a-2.4d.

The same vertiports are opened in all cases. That is, regardless of the values of $c^{\text{AT,fix}}$, $c^{\text{AT,var}}$, $t^{\text{AT+}}$, and v^{AT}, vertiports are opened in Oberpfaffenhofen, Manching, Holzkirchen, Augsburg, and Antersberg. Hence, the solution is robust to variations in ticket prices and travel times when considering the objective to maximize the number of air taxi passengers. Variations in the cost parameters $c^{\text{AT,fix}}$ and $c^{\text{AT,var}}$ have a stronger impact on the utilization of air taxis than variations in the time parameters $t^{\text{AT+}}$ and v^{AT}. For example, doubling the air taxi fix cost $c^{\text{AT,fix}}$ from 50€ to 100€ results in a 66.88% reduction in air taxi utilization, while doubling the air taxi speed from $100\frac{\text{km}}{\text{h}}$ to $200\frac{\text{km}}{\text{h}}$ increases air taxi utilization by only 22.20%.

There are two reasons for this discrepancy: First, passengers weigh travel cost per € more

heavily than travel time per minute (Table 2.2), and second, a doubling of $c^{\text{AT,fix}}$ from 50€ to 100€ has a larger absolute impact in Eqs. (2.10) than a doubling of v^{AT} from $100\frac{\text{km}}{\text{h}}$ to $200\frac{\text{km}}{\text{h}}$ if the distance traveled by air taxi is shorter than 100 kilometers. This is the case here, since the opened vertiports are located 49.29 kilometers from Munich Airport on average.

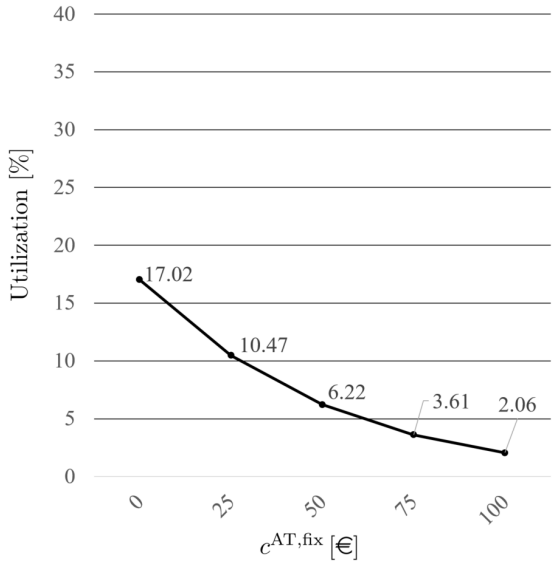
2.5.3.4 Maximizing travel time savings

Thus far, we have solved the problem to maximize the number of passengers who use the air taxi shuttle network. Since it is in the interest of public authorities to improve the accessibility to Munich Airport, we now consider the alternative objective to maximize the total time savings of all passengers traveling to Munich Airport compared to the status quo. Let z^{gnd} denote the total travel time of all passengers when no vertiport is opened, i.e., when $\mathcal{I}^{\text{air}} = \emptyset$ and passengers can only choose between ground-based means of transport. Then, we define the new objective as follows:

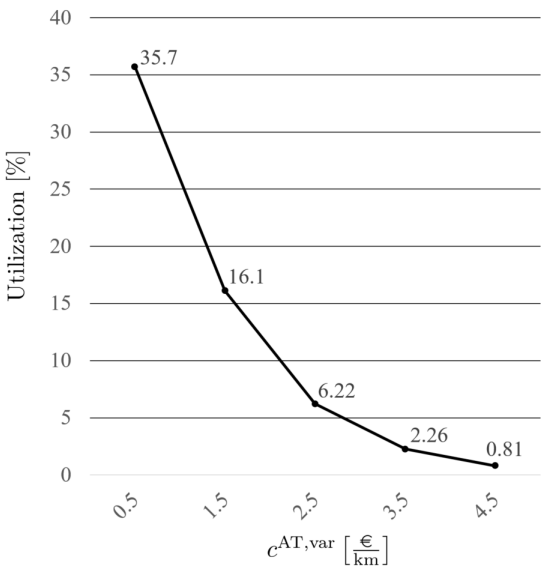
$$\max z = z^{\text{gnd}} - \left(\sum_{o \in \mathcal{O}} \sum_{i \in \mathcal{I}_o} n_o \cdot \eta_{oi}^{tt} \cdot p_{oi} \right) \tag{2.13}$$

We choose the same experimental setting as in Section 2.5.3.3. In contrast to the results obtained in Sections 2.5.3.2 and 2.5.3.3, the decisions about which vertiports are opened are no longer identical for different values of c^{fix} and c^{var}. However, the average distance between the vertiports that are opened and Munich Airport is consistently larger when Objective Function (2.13) is used rather than Objective Function (2.4a). For example, for the parameter values we assumed in the base case ($c^{\text{AT,fix}} = 50$€, $c^{\text{AT,var}} = 2.5$€, $t^{\text{AT+}} = 15$ min, $v^{\text{AT}} = 200\frac{\text{km}}{\text{h}}$), the average (air) distance from the opened vertiports to Munich Airport increases from 49.29 kilometers to 88.50 kilometers. The larger average distance between vertiports and Munich Airport results from the fact that the transfer time at the vertiport can only be compensated for on longer routes by the higher cruising speed of air taxis compared to cars or public transport.

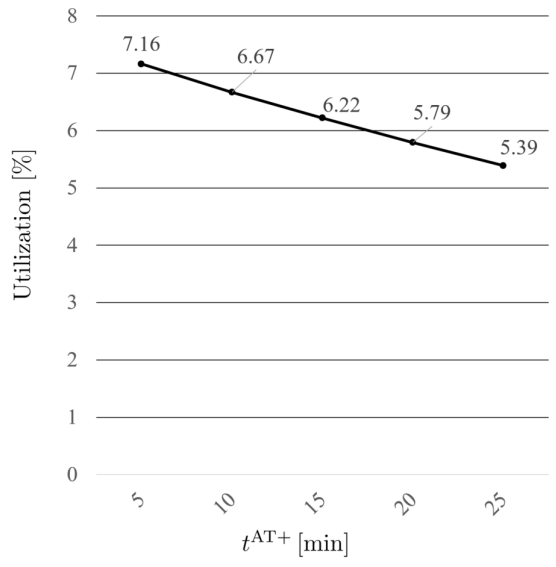
We provide the air taxi utilization and the total number of hours saved by passengers traveling to Munich Airport for all values of $c^{\text{AT,fix}}$, $c^{\text{AT,var}}$, $t^{\text{AT+}}$, and v^{AT} in Tables



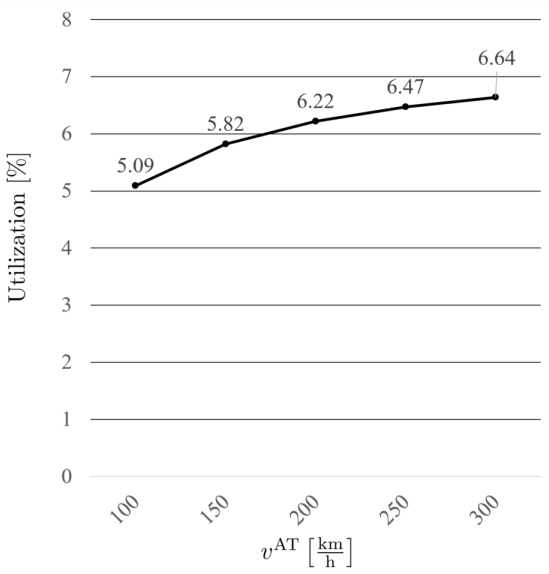
(a) Variation in $c^{\mathrm{AT,fix}}$



(b) Variation in $c^{\mathrm{AT,var}}$



(c) Variation in $t^{\mathrm{AT+}}$



(d) Variation in v^{AT}

Figure 2.4: Impact of variation in air taxi travel costs and travel times on air taxi utilization

2.5a-2.5d. Air taxi utilization ranges between 0.04% and 11.91%, and the total time saved by passengers ranges between 730 hours and 40.330 hours. The air taxi utilization figures are significantly lower than those observed in Section 2.5.3.3, as travel costs for itineraries $i \in \mathcal{I}^{\text{air}}$ increase disproportionately when distance increases compared to travel costs for itineraries $i \in \mathcal{I}^{\text{gnd}}$. In consequence, we find that the average time saving per passenger is rather low. For example, with the parameter values from the base case, passengers reach the airport only about 0.08 minutes faster on average. Thus, to reduce travel times to Munich Airport significantly, air taxi travel costs and/or times must be significantly smaller than in our base case.

Variations in the cost parameters $c^{\text{AT,fix}}$ and $c^{\text{AT,var}}$ still have a stronger impact on the utilization of air taxis than variations in the time parameters $t^{\text{AT+}}$ and v^{AT}. However, this is not true for the time saved by passengers: In Tables 2.5a-2.5d, the time saved falls within similar ranges. Thus, a higher air taxi utilization does not always translate into greater average time savings for passengers (for example, check the time saved for small values of $c^{\text{AT,var}}$ in Table 2.5b). In particular, lower air taxi travel costs also attract passengers who save little time, or even lose time, by traveling to the airport by air taxi.

Table 2.5: Air taxi utilization and time savings for different air taxi travel costs and travel times when maximizing passenger time savings

$c^{\text{AT,fix}}$ [€]	Utilization [%]	Time saved [1,000 hours]
0	1.72	24.44
25	1.03	14.87
50	0.57	9.55
75	0.36	6.17
100	0.19	4.31

(a) Variation in $c^{\text{AT,fix}}$

$c^{\text{AT,var}}$ $\left[\frac{€}{\text{km}}\right]$	Utilization [%]	Time saved [1,000 hours]
0.5	11.91	17.12
1.5	2.42	40.33
2.5	0.57	9.55
3.5	0.16	3.26
4.5	0.04	0.73

(b) Variation in $c^{\text{AT,var}}$

$t^{\text{AT+}}$ [min]	Utilization [%]	Time saved [1,000 hours]
5	0.76	19.08
10	0.62	13.77
15	0.57	9.55
20	0.51	5.82
25	0.10	3.53

(c) Variation in $t^{\text{AT+}}$

v^{AT} $\left[\frac{\text{km}}{\text{h}}\right]$	Utilization [%]	Time saved [1,000 hours]
100	0.05	0.96
150	0.45	4.58
200	0.57	9.55
250	0.61	12.94
300	0.73	15.60

(d) Variation in v^{AT}

2.6 Conclusion

In this chapter, we investigated the potential of introducing an air taxi shuttle network to improve airport access. To the best of our knowledge, we are the first to formulate this problem as a hub location problem with multiple allocation and integrated multinomial logit model to represent passengers' itinerary choices. Based on the approach by Haase (2009), we linearized the problem, reformulated the model to avoid numerical instabilities, and eliminated redundant constraints. Furthermore, we extended the resulting model by additional constraints to ensure that the IIA property is always adhered to. After discussing what modeling decisions need to be taken and what data must be collected to use the model for solving real life instances in general, we applied it to Munich Airport in a detailed case study. For this purpose, we collected large amounts of data and, in particular, conducted a stated preference survey to parameterize the multinomial logit model.

The results of our computational experiments show that air taxi utilization will mostly depend on the number and location(s) of opened vertiports as well as the ticket prices for air taxi flights, while the transfer times at the vertiports and the air taxi cruising speed had less of an impact in our experiments. In order to maximize the number of air taxi passengers, vertiports should not be opened too far from the airport, especially when (distance dependent) ticket prices are high, since high ticket prices impose significant limitations on demand. However, the selection of vertiport locations too close to the airport prevents passengers from saving travel time due to the transfer times at vertiports and the short travel times of competing (i.e., ground-based) means of transport. On the other hand, while the time saved per passenger increases the farther vertiports are located from the airport, high ticket prices result in low choice probabilities for air taxi itineraries when vertiports are located far from the airport. Thus, the results of our case study suggest that air taxis can only reduce the average travel time to the airport significantly in the medium to long term, when ticket prices decrease on longer air taxi flights. Accordingly, our results suggest establishing vertiports near the airport first to develop the market, and only later, when air taxi tickets can be offered at significantly lower prices, integrating more distant vertiports into the system to reduce average travel times of passengers to the airport.

For future research, we suggest considering tactical and/or operational constraints in our strategic problem. In particular, potential capacity restrictions as well as the resulting challenges should be adressed. If, for instance, many passengers choose to use a certain vertiport, capacity restrictions at this vertiport may cause additional waiting times in queues. In return, additional waiting times may reduce the demand at the vertiport. Similarly, capacity constraints can worsen travel times of ground-based means of transport. For example, if many passengers travel to the airport directly by car, the streets to the airport may be congested and it may be difficult to find a parking spot at the airport. Furthermore, the waiting times at vertiports also depend on whether air taxi shuttle flights are operated on a schedule or an on-demand basis. If air taxis are operated on a schedule, passengers can be expected to arrive at the vertiport with a time buffer to avoid missing the air taxi, resulting in additional waiting times at the vertiport. In addition, the accuracy of the multinomial logit model may be improved by including additional independent variables besides travel times and travel costs to estimate choice probabilities. For instance, average household income, political voting data, and/or age distribution could be considered. However, only variables whose values are known for each origin (e.g., for each postal code area) can be included. Finally, the travel costs of itineraries where at least one leg of the trip is covered by car should be examined in more detail. First, the travel costs per passenger depend on the number of passengers in a car, as fuel and parking costs are incurred only once per car. This is different compared to public transport and air taxi trips, where travel costs per passenger are mostly independent of the number of passengers travelling together. Second, the parking fees depend on the length of time the car is parked, and the parking fees at vertiports may also differ from those at the airport. Lower parking fees at the vertiports may attract more passengers to choose the air taxi to reach the airport.

We hope this work motivates further research in this up-and-coming field. We also encourage practitioners to make use of our approach to identify optimal locations for vertiports in order to make the introduction of air taxis as a new means of transport a success.

3 Optimal Positioning of Aircraft Stands

At many airports, space on the apron is scarce and has to be used efficiently. To that end, we optimize the layout of aircraft parking positions adjacent to the airport terminal, pursuing two lexicographically ordered objectives. First, we minimize the number of aircraft that have to be diverted to remote parking positions, because positions adjacent to the terminal are not available. Second, we minimize the construction effort required for gate infrastructure. Aircraft collisions must be prevented at all times, and we consider various traffic situations, as traffic volume and fleet mix are not constant in time. We introduce the Airport Gate Layout Problem and formulate it as a mixed-integer model, which considers both greenfield and brownfield scenarios. To solve the problem efficiently, we introduce a decomposition framework that exploits the structure of the problem and employ various acceleration techniques. Our approach reduces computation times substantially, allowing us to solve instances that are intractable for CPLEX. Based on a case study for Munich Airport, we demonstrate how airports can gain valuable insights from solving the problem.

3.1 Introduction

At many airports, space on the apron is scarce and should therefore be used as efficiently as possible (see, e.g., Caves 1994). We consider the situation where an airport terminal is to be either constructed, extended, or refurbished and the arrangement of aircraft parking positions adjacent to the terminal building needs to be determined. Compared to remote parking positions, the use of parking positions adjacent to a terminal building is preferred

by airlines, airports, and passengers as operations are safer, more efficient, and more comfortable. Passengers and their baggage do not need to be transported over longer distances and turnaround times are shorter. Consequently, the apron layout should be designed to minimize the number of aircraft that have to be handled at remote parking positions for space reasons.

When planning the layout, minimum safety distances between adjacent aircraft as well as aircraft and airport infrastructure must be adhered to in order to prevent collisions (International Civil Aviation Organization 2018, European Aviation Safety Agency 2017). Furthermore, there is a wide variety of aircraft types, which differ considerably with regard to their dimensions, minimum safety distances, and requirements for parking position equipment. Hence, the equipment to be installed at a parking position as well as the distances to adjacent parking positions are based on the aircraft to be handled at the position, and the optimal overall layout depends on the expected fleet mix at the airport. Moreover, traffic volume and fleet mix are subject to considerable fluctuations at most airports, especially at large hub airports. Thus, airports are keen to identify the layout that minimizes the number of aircraft having to divert to remote parking positions across all expected traffic situations. On the other hand, the (re-)construction of a parking position adjacent to a terminal building is associated with high costs. For example, investment costs of 450.000€ are to be expected per passenger boarding bridge (see, e.g., Airport Improvement Magazine 2010, Travel PR News 2019). Consequently, among all alternative layouts that minimize the number of aircraft that have to be handled at remote parking positions, airports are particularly interested in determining the layout that minimizes the number of parking positions that need to be built.

Currently, the planning problem described is mostly approached manually, with CAD programs and simulation tools supporting the decision makers. However, while different layout proposals can be evaluated relative to each other using simulation, there is currently a lack of possibility to assess a proposal on its own, since the optimal layout is not known. Hence, the quality of the resulting layout strongly depends on the experience and skills of the planner. We address this gap by finding the best solution for a particular apron.

Layout planning problems in general have been considered from various perspectives in the literature. Most prominently, existing work on the facility layout problem addresses the general question of how individual facilities should be arranged within a given area,

usually a factory floor, in order to minimize the overall transportation costs of goods. Recent surveys on this class of problems and its variations are provided by Drira et al. (2007) and Anjos and Vieira (2017). More application-specific layout planning problems are considered by Briskorn and Dienstknecht (2019) and Huang et al. (2011), who investigate the optimal positioning of tower cranes at construction sites, as well as Stephan et al. (2021), who maximize the capacity of parking lots. However, due to the complex geometries of aircraft, the heterogeneity of different aircraft types, and the requirement that the layout be optimal with respect to the totality of different traffic situations, none these approaches can be applied to our problem.

In contrast, complex geometries and heterogeneity of objects are considered in two-dimensional irregular object packing problems. There, objects with complex geometries have to be positioned in a (potentially irregularly shaped) area such that the objects do not collide and the total value of the objects that cannot be positioned is minimized. Detailed reviews of this class of problems are provided by Bennell and Oliveira(2008, 2009) and Leao et al. (2020). There is a long history of heuristics for solving these packing problems (see, e.g., Martinez-Sykora et al. 2017, Chehrazad et al. 2022, Umetani and Murakami 2022, Zhang et al. 2022, Luo and Rao 2023), but recently exact solution methods have also been proposed (see, e.g., Alvarez-Valdes et al. 2013, de Souza Queiroz and Andretta 2022). With respect to aircraft parking, Qin et al. (2018) use no-fit polygons to place aircraft in a service hangar, minimizing the unused space. While two-dimensional irregular object packing problems are related to our objective to minimize the number of aircraft that have to be processed at remote parking positions, they do not consider our second objective to minimize the number of parking positions that need to be built. Additionally, we incorporate different traffic situations in our problem, and the parking positions built need to be optimal across all situations. Furthermore, aircraft parking positions cannot be placed freely as objects in general packing problems, as we will explain in more detail in Section 3.2.

Thus far, no studies have been published on optimizing the layout of parking positions at airport aprons. There are several studies investigating apron capacity, which either assume the layout is given or disregard it. For example, Mirković and Tošić (2014, 2016, and 2017) provide a mathematical framework to describe the capacity of a given apron, and Steuart (1974), Bandara and Wirasinghe (1989), Wirasinghe and Bandara (1990), Hassounah and Steuart (1993), and Narciso and Piera (2015) present approaches to calculate the number

of aircraft parking positions a terminal should be equipped with.

A related problem on the operative level of decision making that considers airport gates is the Gate Assignment Problem, in which arriving aircraft must be assigned to aircraft stands. Typical objectives are to minimize the total passenger walking distance or to minimize the total aircraft taxi time. Daş et al. (2020) find that recent work tends to consider multiple objectives simultaneously. Extensive reviews on the gate assignment problem are provided by Dorndorf et al. (2007), Cheng et al. (2012), Guépet et al. (2015), and Daş et al. (2020). Dorndorf et al. (2008, 2012, and 2017) also consider the prevention of collisions between parked aircraft in their models. However, in contrast to the recurrent operative task to assign arriving aircraft to available gates, defining the layout represents a strategic problem.

In conclusion, no literature exists to date in which the layout of aircraft parking positions is optimized by means of mathematical programming, nor can other existing modeling approaches be directly applied to this problem. In the following, we introduce all aspects of the Airport Gate Layout Problem (AGLP) in detail and provide a mixed-integer formulation that can be applied to both greenfield and brownfield instances. We develop an exact procedure for solving the problem, in which we combine a decomposition approach with a bounding algorithm and a relaxation scheme. In a case study, we demonstrate the superiority of our solution procedure over CPLEX and show how the results of our work can support decision makers in practice.

The remainder of this chapter is structured as follows: In Section 3.2, we provide a detailed problem description, and our modeling approach is described in Section 3.3. The resulting mathematical model and our solution methodology are provided in Sections 3.4 and 3.5, respectively. In Section 3.6, we present our computational experiments based on real world data, and we provide our conclusions in Section 3.7.

3.2 Problem description

Planning problem We consider the situation where an airport terminal is to be constructed, extended, or refurbished and the aircraft parking positions adjacent to the terminal building must be planned. The position, shape, and dimensions of the terminal

building and surrounding taxiways are given and define the areas where parking positions can be planned. Furthermore, a forecast for the traffic that needs to be accommodated at the apron is available. In this planning problem, we pursue two objectives, which are ordered lexicographically. The primary goal is to minimize the number of aircraft that cannot be parked in close proximity to the terminal building; the secondary goal is the minimization of the implementation costs of the layout. That is, if a new terminal or a terminal extension is in planning, the number of parking positions to be built should be minimized. If an existing terminal is to be refurbished, as few changes as necessary should be made to the existing layout.

Aircraft parking positions We will use the following terminology with respect to aircraft parking. The terms gate and aircraft stand refer to a subarea of the terminal apron utilized to accommodate an aircraft and are used synonymously[1]. Aircraft stands in the immediate vicinity of the terminal building are called contact stands, while stands at a greater distance from the terminal are referred to as remote stands.

Gates need to be equipped to handle aircraft, with equipment specifications depending on the aircraft types to be accommodated at the gate. For example, a gate at which small aircraft are to be handled must be equipped with a single passenger boarding bridge. At gates where large aircraft are to be handled, however, a second (or even a third) passenger boarding bridge must be installed to guarantee appropriate passenger boarding and deboarding times. The equipment installed at a gate is downward compatible. In other words, if an aircraft of a certain size can be handled at a gate, other aircraft of the same size and all smaller aircraft can also be handled. Furthermore, gates can also be equipped to handle either one large aircraft or two small aircraft simultaneously. Such gates are referred to as Multi Aircraft Ramping Stands (MARS) (see, e.g., International Air Transport Association 2004, National Academies of Sciences, Engineering, and Medicine 2010).

Due to the large number of infrastructure elements to be procured (for example, passenger boarding bridges, fuel, power and fresh air connections, ground markings, equipment inside the terminal), the construction of a new gate is expensive. In addition, since more or larger equipment is needed to handle larger aircraft, the investment costs per gate increase with the size of the aircraft to be handled. As a result, the number of gates to be

[1]This is in line with Dorndorf et al. (2007) and Ashford et al. (2011)

built or rebuilt, especially those capable of handling large aircraft, should be minimized as long as this does not reduce the airport's ability to handle air traffic.

Aircraft handled at contact stands are commonly parked in the nose-in orientation, in which the noses of parked aircraft point toward the terminal facade. The exact parking positions of aircraft at gates are defined by lead-in lines and stop lines: When parked, the fuselage of the aircraft is aligned collinearly with the lead-in line and the nose wheel is at the stop line belonging to the aircraft type. Each lead-in line is assigned to exactly one gate, whereas each gate must have at least one lead-in line. Lead-in lines can be placed at any angle to the terminal facade. Although a greater number of lines per gate increases flexibility in terms of parking aircraft at gates, as few lead-in lines as possible should be used to minimize complexity in daily operations[2]. Lead-in lines are typically linear and can therefore be defined by a starting point and an ending point. Figure 3.1 shows the ground layout of three gates at Nice Airport, where gate 46 is a MARS position.

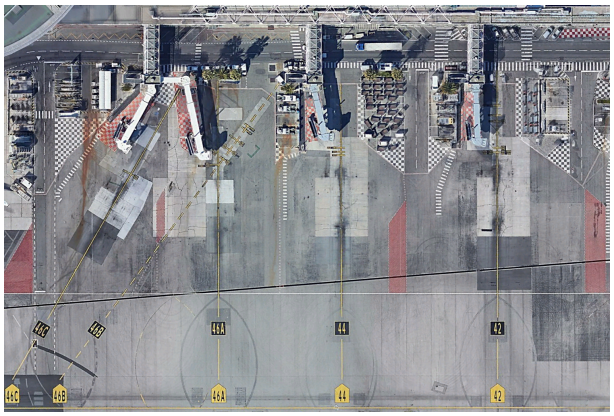


Figure 3.1: Gates 42, 44, and 46 at Nice Airport (Source: Google Maps)

Aircraft classes To ensure collision-free operations on the apron, the geometries of the aircraft to be handled must be taken into account in the planning process. Because of the large number of aircraft types, planning is simplified by grouping aircraft into size classes. We will use the Airplane Design Group (ADG) for that purpose, which was introduced by Federal Aviation Administration (2012) and classifies aircraft according to wingspan,

[2]MARS positions must have at least two lead-in lines to accommodate two small aircraft simultaneously.

see Table 3.1[3].

Table 3.1: Classification of aircraft (Federal Aviation Administration 2012)

ADG	Wingspan [m]	Safety clearance [m]	Reduced safety clearance [m]
I	< 15	3	3
II	15 - 24	3	3
III	24 - 36	4.5	3
IV	36 - 52	7.5	4.5
V	52 - 65	7.5	4.5
VI	65 - 80	7.5	4.5

Aircraft parking restrictions There are few regulations regarding the design of aircraft stands. Most importantly, International Civil Aviation Organization (2018) and European Aviation Safety Agency (2017) consistently define minimum safety clearances that must always be maintained between an aircraft and any other aircraft as well as airport structures. The minimum safety clearances depend on the ADG of the particular aircraft, and are reduced at aircraft stands equipped with a visual guidance docking system or when other special cases apply. General and reduced safety clearances per ADG are provided in Table 3.1.

Parking an aircraft of a given class at a specific lead-in line may be prohibited for three reasons. First, certain areas of the apron may be inaccessible to aircraft of a certain class and above. Second, the location of a lead-in line, in combination with the layout of the surrounding infrastructure, might result in only aircraft up to a certain class being allowed to park on the line, as otherwise minimum safety clearances between aircraft and infrastructure would be violated or the aircraft would collide with the infrastructure. Third, the parking of an aircraft of a given class on a particular lead-in line could be temporarily prohibited if another aircraft of the same or a different class is being handled simultaneously at an adjacent position. The reason for this restriction is again to avoid violations of minimum safety clearances as well as collisions.

[3]The same classification is reproduced by International Civil Aviation Organization (2018) and European Aviation Safety Agency (2017) using a different notation.

Air traffic characteristics at hub airports Traffic volumes and fleet mix are typically subject to significant fluctuations during the course of the day at hub airports. Traffic arrives and departs in waves, so airlines can offer their passengers comparatively short transfer times and a large number of transfer connections. Arrival and departure waves directly propagate to the situation on the apron, where the number and composition of the aircraft to be handled simultaneously can fluctuate considerably. To avoid recurring congestion, the apron should therefore be designed to handle as much of the expected peak-time traffic as possible at contact gates.

Moreover, the apron should not only be designed for current traffic peaks, but in anticipation of expected future traffic developments due to the high costs associated with gate infrastructure as well as the high operational effort and financial burden in case of future changes to the layout. For example, the proportion of very large ADG VI aircraft is likely to decline in the medium term, as both Airbus and Boeing recently ceased production of their only ADG VI aircraft, the A380 and 747-8, respectively, due to lack of demand. In contrast, the proportion of slightly smaller ADG V aircraft can be expected to trend upward due to their better fuel efficiency and greater operational flexibility.

3.3 Modeling approach

Representation of air traffic We represent air traffic on the apron as a collection of snapshots, each containing the number of aircraft per class that must be handled simultaneously at one particular point in time where traffic volume reaches a peak. We refer to these snapshots as demand patterns, denoted by set $\mathcal{K} = \{1, \ldots, K\}$.

We classify aircraft according to the ADG classification, resulting in the set of aircraft classes $\mathcal{A} = \{1, \ldots, A\}$, where classes are sorted by ascending aircraft size. For each aircraft class $a \in \mathcal{A}$, let D_{ak} be the number of aircraft that are to be parked simultaneously for demand pattern $k \in \mathcal{K}$. We identify the set of demand patterns and the values of D_{ak} for all $a \in \mathcal{A}$ and $k \in \mathcal{K}$ from a given flight plan in a two-step procedure. First, we add one demand pattern to set $\mathcal{K}$ for each point in time provided in the flight plan (e.g., for each five-minute interval within a week) and determine D_{ak} for all aircraft classes $a \in \mathcal{A}$ based on the flight plan data. Then, we eliminate all demand patterns $k \in \mathcal{K}$ that are

dominated by at least one other demand pattern $\hat{k} \in \mathcal{K}$, i.e., for which we can find a demand pattern $\hat{k} \in \mathcal{K}$ where $D_{ak} \leq D_{a\hat{k}}$ $\forall a \in \mathcal{A}$ and $\exists a \in \mathcal{A} : D_{ak} < D_{a\hat{k}}$ holds. For hub airports, the remaining, non-dominated demand patterns represent the traffic peaks which occur as traffic typically arrives and departs in waves.

Each value of D_{ak} is associated with a weighting factor $W_{ak} \in [0, 1]$, which indicates the relative importance of accommodating aircraft of class $a \in \mathcal{A}$ for demand pattern $k \in \mathcal{K}$. The value of W_{ak} depends on the size of the aircraft in class $a \in \mathcal{A}$ and the frequency of occurrence of demand pattern $k \in \mathcal{K}$. The larger an aircraft, the more passengers it can carry and the more important it is that it can be parked at a contact gate. Hence, larger aircraft classes are associated with higher values of W_{ak}. The more frequently a demand pattern is expected to occur, the more relevant it is for the planning process. Thus, demand patterns with a higher expected frequency of occurrence are associated with higher values of W_{ak} as well.

Aircraft parking positions Aircraft must be parked on lead-in lines, and each lead-in line used to park an aircraft must be assigned to a gate. We employ the given layout of the terminal building and the surrounding taxiways to derive sets of feasible positions for gates and lead-in lines in the following.

In the first step, we identify polygons as subareas of the apron inside which lead-in lines can be placed. The edges of these polygons are given either by airport infrastructure that must not be infringed upon by aircraft (for instance, the terminal facade or edges of the apron), or by taxiways. Second, in each polygon we define a set of starting points for lead-in lines; by starting point we mean the point of the lead-in line at which the nose of a parked aircraft is located. The starting points are placed at a constant distance $\Delta > 0$ from each other and at a constant safety distance from the terminal facade.

Third, we create lead-in lines by drawing straight lines from the starting points until they intersect with an edge of the surrounding polygon. These intersections define the ending points of the lead-in lines. We generate multiple lead-in lines from each starting point by varying the direction of the line. Therefore, parameter κ denotes the rotation angle between the lead-in lines that share the same starting point, $0 < \kappa < 360°$ and $360°$ should be an integer multiple of κ. Both Δ and κ are parameters of our approach that determine the planning granularity and must be specified up-front. Lead-in lines whose

ending point is not on a taxiway cannot be reached by aircraft and are therefore discarded. All remaining lead-in lines resulting from this process are included in set $\mathcal{L} = \{1, \ldots, L\}$. Figure 3.2 shows an example of a terminal building with possible lead-in lines.

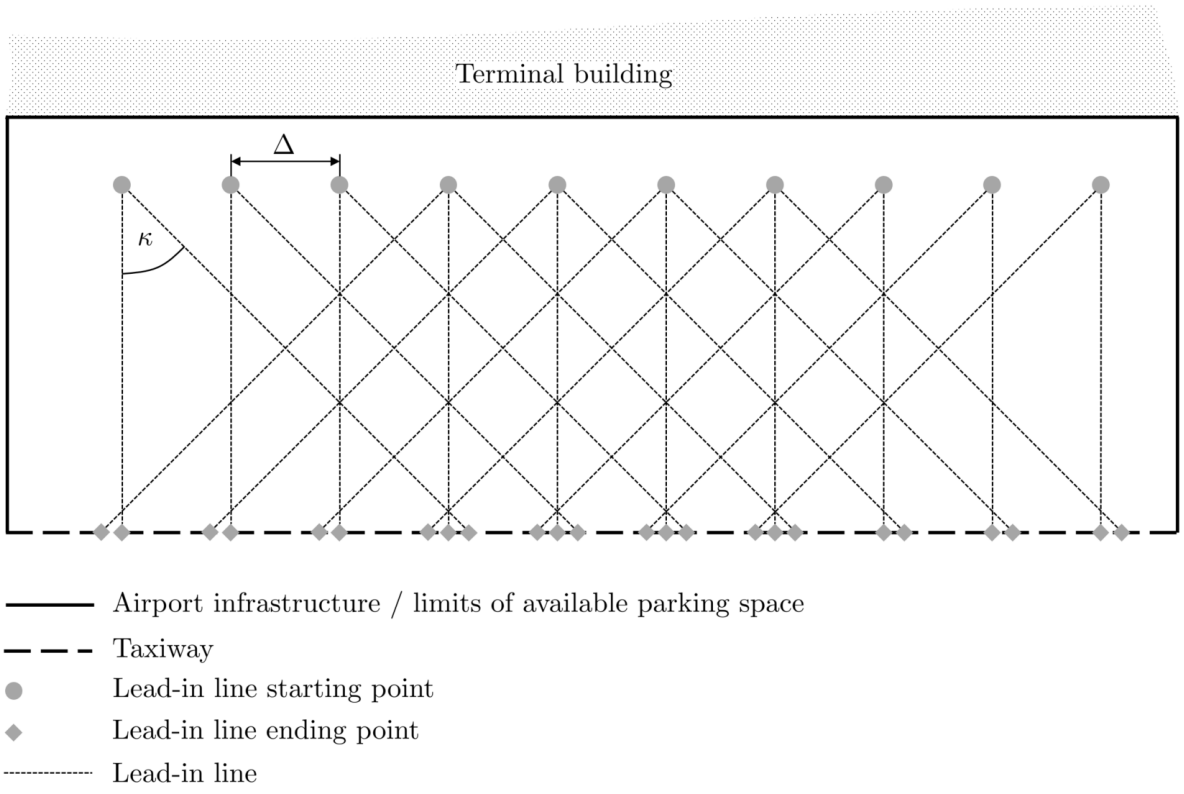


Figure 3.2: Example for generation of lead-in lines

Each lead-in line used to park an aircraft has to be assigned to a gate. Thus, we define potential gate positions and introduce a parameter that indicates whether a lead-in line can be assigned to a gate or not. We consider the set of lead-in line starting points as the set of possible gate positions. All potential gates are contained in set $\mathcal{G} = \{1, \ldots, G\}$. For each pair of lead-in line $l \in \mathcal{L}$ and gate $g \in \mathcal{G}$, binary parameter F_{lg} equals 1, if the direct path between lead-in line l and gate g is unobstructed and smaller than a given threshold, and lead-in line l can therefore be assigned to gate g. Otherwise, F_{lg} equals 0.

Aircraft parking restrictions Parking an aircraft of a particular class on a particular lead-in line could be either permanently or temporarily prohibited to prevent collisions

with airport infrastructure or other aircraft parked at adjacent positions. To identify all such infeasibilities, we apply a two-step procedure.

First, we compute a so-called safety envelope for each aircraft class. The safety envelope of an aircraft class $a \in \mathcal{A}$ is defined by the smallest possible polygon with the shape shown in Figure 3.3 inside which all aircraft of the class can be parked in a given orientation, including the minimum safety distances required according to Table 3.1. We compute the safety envelope for each aircraft class based on the geometry data of all aircraft types belonging to the class. The safety envelopes of all classes have the shape as shown in Figure 3.3 and differ only in the dimensions. Details are provided in Appendix B.2.

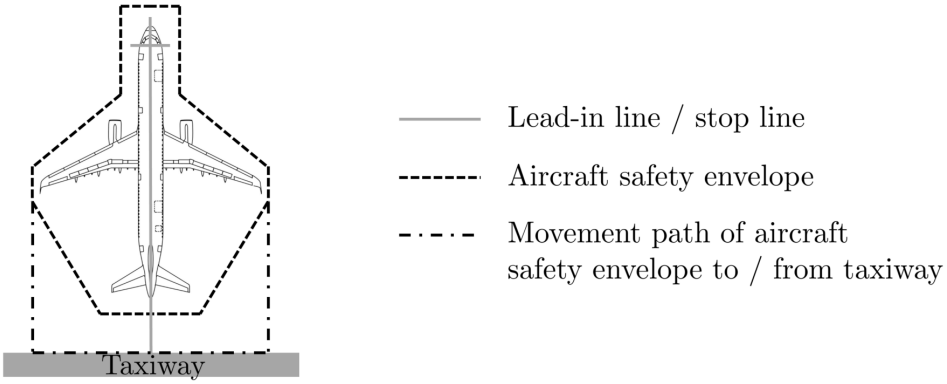


Figure 3.3: Shape of safety envelopes (Source of aircraft model: Airbus 2022)

Second, for each aircraft class $a \in \mathcal{A}$ and lead-in line $l \in \mathcal{L}$, we use the safety envelopes to determine whether or not the aircraft can generally be parked on the line, and whether or not the aircraft can be parked on the line if another aircraft of class $b \in \mathcal{A}$ is parked on another lead-in line $m \in \mathcal{L} \setminus \{l\}$ at the same time. If we place the safety envelope of an aircraft class on a lead-in line and the safety envelope then touches or intersects an edge of the surrounding polygon, i.e., collides with airport infrastructure, no aircraft of that class can be parked on the lead-in line. As a result, the binary parameter C_{al} equals 1 if aircraft of class $a \in \mathcal{A}$ can be accommodated on lead-in line $l \in \mathcal{L}$, and 0, otherwise. C_{al} also equals 0 if lead-in line l is located in an area of the apron that cannot be reached by class a aircraft. If $C_{al} = 0$ for a given lead-in line $l \in \mathcal{L}$ and all aircraft classes $a \in \mathcal{A}$, line l is discarded. Similarly, for all combinations of two lead-in lines $l, m \in \mathcal{L} : l \neq m$ and two aircraft classes $a, b \in \mathcal{A}$, we place the safety envelopes of classes a and b on lines l and m, respectively, and then examine whether the safety envelopes collide with each

other (for details, we refer to Appendix B.3). Binary parameter E_{lmab} is 1 if aircraft of classes a and b cannot be handled simultaneously without collision at lead-in lines l and m, and 0, otherwise.

3.4 Model

In this section, we present a mixed-integer linear program for the AGLP. The notation is summarized in Table B.1 in Appendix B.1.

Decision variables First, the binary decision variable v_{ga} indicates whether or not gate $g \in \mathcal{G}$ is equipped to handle aircraft of class $a \in \mathcal{A}$. If $v_{ga} = 0$ for all aircraft classes $a \in \mathcal{A}$, gate $g \in \mathcal{G}$ cannot handle aircraft of any class and hence, is not built. Furthermore, from the downward compatibility of gate equipment it follows that if $v_{ga} = 1$, then $v_{gb} = 1 \; \forall b \in \mathcal{A} : b < a$.

Second, the binary decision variable y_l equals 1 if lead-in line $l \in \mathcal{L}$ is used to park an aircraft for any demand pattern $k \in \mathcal{K}$, and 0, otherwise.

Third, the binary decision variable u_{gl} is 1 if lead-in line $l \in \mathcal{L}$ is assigned to gate $g \in \mathcal{G}$, and 0, otherwise.

Fourth, the binary decision variable x_{glak} equals 1 if for demand pattern $k \in \mathcal{K}$ an aircraft of class $a \in \mathcal{A}$ is parked at lead-in line $l \in \mathcal{L}$ and lead-in line l is assigned to gate $g \in \mathcal{G}$, and 0, otherwise.

Finally, the non-negative decision variable q_{ak} yields for each demand pattern $k \in \mathcal{K}$ the number of aircraft of class $a \in \mathcal{A}$ that cannot be accommodated at any of the lead-in lines in set $\mathcal{L}$, and hence, need to deviate to remote stands.

Model formulation We formulate the AGLP as follows:

$$\min \; z_1 = \sum_{a \in \mathcal{A}} \sum_{k \in \mathcal{K}} W_{ak} \cdot q_{ak} \tag{3.1a}$$

$$\min \; z_2 = \sum_{g \in \mathcal{G}} \sum_{a \in \mathcal{A}} v_{ga} \tag{3.1b}$$

subject to

$$\sum_{g \in \mathcal{G}} \sum_{l \in \mathcal{L}: C_{al}=1 \cap F_{lg}=1} x_{glak} + q_{ak} \geq D_{ak} \qquad \forall a \in \mathcal{A}; k \in \mathcal{K} \tag{3.1c}$$

$$\sum_{g \in \mathcal{G}: F_{lg}=1} \sum_{a \in \mathcal{A}: C_{al}=1} x_{glak} \leq y_l \qquad \forall l \in \mathcal{L}; k \in \mathcal{K} \tag{3.1d}$$

$$\sum_{g \in \mathcal{G}: F_{lg}=1} u_{gl} = y_l \qquad \forall l \in \mathcal{L} \tag{3.1e}$$

$$x_{glak} \leq u_{gl} \qquad \forall g \in \mathcal{G}; l \in \mathcal{L}; a \in \mathcal{A}; k \in \mathcal{K} : F_{lg} = 1; C_{al} = 1 \tag{3.1f}$$

$$\sum_{l \in \mathcal{L}: F_{lg}=1} \sum_{a \in \mathcal{A}: C_{al}=1} a \cdot x_{glak} \leq \sum_{a \in \mathcal{A}} v_{ga} \qquad \forall g \in \mathcal{G}; k \in \mathcal{K} \tag{3.1g}$$

$$v_{ga} \leq v_{gb} \qquad \forall g \in \mathcal{G}; a \in \{2, \ldots, A\}; b = a - 1 \tag{3.1h}$$

$$\sum_{l \in \mathcal{L}: F_{lg}=1} \left(\sum_{a \in \mathcal{A}^{\text{small}}: C_{al}=1} x_{glak} + \sum_{a \in \mathcal{A}^{\text{large}}: C_{al}=1} 2 \cdot x_{glak} \right) \leq 2 \quad \forall g \in \mathcal{G}; k \in \mathcal{K} \tag{3.1i}$$

$$\sum_{g \in \mathcal{G}: F_{lg}=1} x_{glak} + \sum_{h \in \mathcal{G}: F_{mh}=1} x_{hmbk} \leq 1 \qquad \forall l, m \in \mathcal{L}; a, b \in \mathcal{A}; k \in \mathcal{K} : E_{lmab} = C_{al} = C_{bm} = 1; l < m \tag{3.1j}$$

$$q_{ak} \geq 0 \qquad \forall a \in \mathcal{A}; k \in \mathcal{K} \tag{3.1k}$$

$$v_{ga} \in \{0,1\} \qquad \forall g \in \mathcal{G}; a \in \mathcal{A} \tag{3.1l}$$

$$x_{glak} \in \{0,1\} \qquad \forall g \in \mathcal{G}; l \in \mathcal{L}; a \in \mathcal{A}; k \in \mathcal{K} : F_{lg} = 1; C_{al} = 1 \tag{3.1m}$$

$$y_l \in \{0,1\} \qquad \forall l \in \mathcal{L} \tag{3.1n}$$

$$u_{gl} \in \{0,1\} \qquad \forall g \in \mathcal{G}; l \in \mathcal{L} : F_{lg} = 1 \tag{3.1o}$$

The two lexicographically ordered objective functions are given in (3.1a) and (3.1b). Objective Function (3.1a) minimizes the number of aircraft that cannot be processed at any of the lead-in lines in set $\mathcal{L}$ over all demand patterns $k \in \mathcal{K}$, where the values of q_{ak} are weighted with weighting factors W_{ak}. Objective Function (3.1b) minimizes the weighted sum of gates built, where each gate is weighted with the largest aircraft class it is equipped for. For example, if the largest aircraft that can be handled at a gate belongs to class 3, the gate is weighted with a factor of 3. This is ensured through Constraints (3.1h) in Objective Function (3.1b).

Demand Constraints (3.1c) ensure for each aircraft class and demand pattern that either all aircraft are handled at contact gates or q_{ak} is increased accordingly. Constraints (3.1d)-(3.1h) determine the correct infrastructure decisions: Constraints (3.1d) make sure that the value of y_l equals 1 once lead-in line $l \in \mathcal{L}$ is used to park an aircraft. Constraints (3.1e) enforce that each lead-in line that is used to park an aircraft is assigned to exactly one gate. That is, while a lead-in line can be used to park an aircraft for more than one demand pattern, it has to be assigned to the same gate for all demand patterns. Constraints (3.1f) align the values of variables x_{glak} and u_{gl}, and Constraints (3.1g) ensure that gates are built and equipped for the aircraft classes they are supposed to handle. Note that when two small aircraft are handled simultaneously at one gate (MARS), the left side of the constraint takes both aircraft into account. For example, if two aircraft of class 1 are handled at a gate simultaneously, the left side of Constraints (3.1g) equals 2. Hence, the constraint necessitates that the gate be equipped for aircraft of class 2. This takes into account that gates at which two aircraft of a class are handled simultaneously require more equipment than gates at which only one aircraft of the same class is handled. Most obviously, two passenger boarding bridges need to be installed to handle two

aircraft simultaneously. Constraints (3.1h) make sure that gate equipment is downward compatible, and Constraints (3.1i) enforce that a maximum of two aircraft (either two small aircraft or one large aircraft) can be handled at one gate simultaneously. For that purpose, we divide all aircraft classes from set $\mathcal{A}$ into two subsets $\mathcal{A}^{\text{small}}$ and $\mathcal{A}^{\text{large}}$, each containing the aircraft classes of which two and one aircraft, respectively, can be handled simultaneously at a gate. $\mathcal{A}^{\text{small}}$ contains ADG III aircraft and all smaller aircraft, while $\mathcal{A}^{\text{large}}$ contains all aircraft belonging to ADG IV or larger. Safety Constraints (3.1j) prevent minimum safety clearances from being violated or aircraft from colliding with each other. Finally, Constraints (3.1k)-(3.1o) define the domains of all decision variables.

Brownfield scenarios If an existing terminal is to be renovated or extended and existing parking positions can be relocated in the process, deviations from the existing gate layout should be kept to a minimum to minimize investment costs. More specifically, additional gates should only be built if they yield an improvement of z_1.

In a brownfield scenario, the set $\mathcal{G}$ is extended by the gates that already exist in reality. Additionally, we introduce the binary parameter H_{ga}, whose value is 1, if existing gate $g \in \mathcal{G}$ is equipped to handle an aircraft of class $a \in \mathcal{A}$, and 0, otherwise. To account for the downward compatibility of gates, $H_{g\hat{a}} = 1$ for all $\hat{a} \in \{0, \ldots, a-1\}$ if $H_{ga} = 1$. For all gates in $\mathcal{G}$ which do not yet exist in reality, H_{ga} equals 0 for all $a \in \mathcal{A}$.

Then, Model (3.1a)-(3.1o) is extended by the following constraints to ensure that existing gates are considered in the optimization process.

$$v_{ga} = H_{ga} \qquad \forall g \in \mathcal{G}; a \in \mathcal{A} : \sum_{\hat{a} \in \mathcal{A}} H_{g\hat{a}} > 0 \tag{3.1p}$$

Aggregating constraints For small distances Δ and rotation angles κ between adjacent lead-in lines, the number of Safety Constraints (3.1j) becomes very large. We mitigate this problem by aggregating Constraints (3.1j). For example, omitting the indices of demand patterns and gates for simplicity, three individual Constraints (3.1j) $x_{la} + x_{mb} \leq 1$, $x_{la} + x_{nc} \leq 1$, and $x_{mb} + x_{nc} \leq 1$ can be substituted by one single, equivalent constraint $x_{la} + x_{mb} + x_{nc} \leq 1$, where $l, m, n \in \mathcal{L}$ and $a, b, c \in \mathcal{A}$. Let $\mathcal{X}_i$ denote the set of x variables

contained in the sum on the left hand side of one aggregated constraint i. Our goal is to minimize the number of aggregated constraints needed to ensure that no minimum safety clearances are violated. However, a particular x variable x_{la} can only be added to the left hand side of an aggregated constraint i iff for each variable $x_{mb} \in \mathcal{X}_i$ there exists one Constraint (3.1j) $x_{la} + x_{mb} \leq 1$. Otherwise, the aggregated constraints and original Constraints (3.1j) would not be equivalent.

The problem of determining the set of aggregated constraints can be depicted as a clique separation problem in a graph $G = (V, E)$, where each vertex represents one tuple $(l \in \mathcal{L}, a \in \mathcal{A} : C_{al} = 1)$ and two nodes (l, a) and (m, b) are connected iff $E_{lmab} = 1$. Using the Bron-Kerbosch algorithm (Tomita et al. 2006), we determine the list of maximal cliques in this graph. We then map the optimal solution of the clique separation problem onto the set of aggregated constraints as follows: For each maximal clique in this graph, we create one aggregated constraint i that contains decision variable x_{la} in $\mathcal{X}_i$ for each node (l, a) in the clique.

In addition to reducing the number of constraints, the aggregated constraints strengthen the LP relaxation of the problem, which reduces computation times considerably.

3.5 Solution methodology

The results of our computational experiments in Section 3.6 will demonstrate that Model (3.1a)-(3.1p) quickly becomes intractable for decreasing values of Δ and κ. Therefore, we introduce a decomposition approach, which allows solving instances with high planning granularity within reasonable computation time to optimality. We will first give an overview in Section 3.5.1 before providing detailed insights into individual components in Sections 3.5.2 to 3.5.4. Finally, we will present the acceleration techniques that we use to support our approach in Section 3.5.5. The notation used in this section is summarized in Table B.2 in Appendix B.1.

3.5.1 Overview

We provide a summary of our solution approach as pseudo-code in Algorithm 3.1.

Algorithm 3.1 Overview of our solution approach

```
1: Find set of areas 𝒮                                  ▷ See Section 3.5.2
2: Find set of demand decompositions 𝒞                  ▷ See Section 3.5.3
3: Find initial value of LB_c for all c ∈ 𝒞              ▷ See Section 3.5.4
4: i ← 1
5: while |𝒞| > 1 do
6:    Select ĉ ∈ 𝒞 where LB_ĉ ≤ LB_c for all c ∈ 𝒞 \ {ĉ}
7:    Solve subproblems of ĉ                            ▷ See Section 3.5.5
8:    if i = 1 then
9:       Determine UB based on the solution obtained
10:   else
11:      If possible, update UB and LB_c for all c ∈ 𝒞
12:   end if
13:   Eliminate all c ∈ 𝒞 where LB_c ≥ UB
14:   i ← i + 1
15: end while
```

First, we decompose the apron into a set of areas $\mathcal{S} = \{1, \ldots, S\}$ that are independent with respect to Safety Constraints (3.1j), intending to solve the AGLP for each area separately (line 1 in Algorithm 3.1, see Section 3.5.2); however, Demand Constraints (3.1c) consider the apron as a whole. Thus, in order to solve the problem for each area independently, we must first decide for each demand pattern how many aircraft of which class(es) are to be assigned to which area. Among the many possible ways to decompose demand patterns, we only consider those where (i) all aircraft assigned to an area can be parked at contact gates in the particular area and where (ii) the total number of aircraft (weighted by W_{ak}) that cannot be assigned to any area due to (i) is minimized. We call each of the resulting assignments a demand decomposition, and the set of demand decompositions is denoted by $\mathcal{C} = \{1, \ldots, C\}$ (line 2, see Section 3.5.3). We generate all demand decompositions that satisfy (i) and (ii).

Let z_1^* and z_2^* denote the optimal values of Objective Functions (3.1a) and (3.1b), respectively. Similarly, let z_{1c}^* and z_{2c}^* be the optimal values of Objective Functions (3.1a) and (3.1b) when aircraft are assigned to areas according to demand decomposition $c \in \mathcal{C}$

($z^*_{1c} \geq z^*_1$ and $z^*_{2c} \geq z^*_2$). As we will show in Section 3.5.3, ensuring (i) and (ii) is equivalent to minimizing Objective Function (3.1a), and thus all demand decompositions in $c \in \mathcal{C}$ lead to z^*_1, i.e., $z^*_{1c} = z^*_1 \; \forall c \in \mathcal{C}$. In order to solve the AGLP to optimality, we must identify a demand decomposition $c \in \mathcal{C}$ that also leads to z^*_2, i.e., where $z^*_{2c} = z^*_2$. We reduce the number of demand decompositions that need to be examined to find z^*_2 and to prove optimality of z^*_2 by means of a bounding algorithm (lines 5 to 15). First, we compute a lower bound LB_c on z^*_{2c} for each demand decomposition $c \in \mathcal{C}$ (line 3, see Section 3.5.4). Then, we select the demand decomposition with the lowest value of LB_c and determine z^*_{2c}. The result poses an upper bound UB on z^*_2. Thus, we can remove all demand decompositions from $\mathcal{C}$ for which $LB_c \geq UB$ holds (line 13). Furthermore, we will show in Section 3.5.4 that the solution may be used to update $LB_{c'}$ for other demand decompositions $c' \in \mathcal{C} \setminus \{c\}$, potentially leading to their removal from $\mathcal{C}$ as well. In the next iteration, we again select the demand decomposition $c \in \mathcal{C}$ with lowest value of LB_c. We repeat this process until no demand decomposition remains in $\mathcal{C}$. Then, the value of UB equals z^*_2.

To determine z^*_{2c} for one demand decomposition $c \in \mathcal{C}$, we solve Model (3.1b)-(3.1p) separately for each area $s \in \mathcal{S}$ (line 7, see Section 3.5.5). We call the resulting models the subproblems of the AGLP, and we denote the optimal objective function value of the subproblem associated with demand decomposition $c \in \mathcal{C}$ and area $s \in \mathcal{S}$ with z^*_{2cs}. When the subproblems for all areas $s \in \mathcal{S}$ have been solved for demand decomposition $c \in \mathcal{C}$, we calculate z^*_{2c} as $\sum_{s \in \mathcal{S}} z^*_{2cs}$. We provide the mathematical formulation of a subproblem in Appendix B.4.

3.5.2 Decomposing the apron into independent areas

We partition the apron into independent areas $\mathcal{S}$ in an iterative process. A terminal facade can be described as a polygonal chain, consisting of a sequence of connected line segments. We initially assume that all gates and lead-in lines located before the same line segment are part of the same area, and that gates and lead-in lines located in front of different line segments belong to different areas. Let $\mathcal{G}_s$ and $\mathcal{L}_s$ be the resulting sets of lead-in lines and gates belonging to area $s \in \mathcal{S}$.

Definition 3.1 (**Independence of areas**). Two areas $s, t \in \mathcal{S}, s \neq t$ are independent iff $E_{lmab} = 0 \ \forall l \in \mathcal{L}_s; m \in \mathcal{L}_t; a, b \in \mathcal{A}$.

If two areas $s, t \in \mathcal{S}, s \neq t$ are not independent according to Definition 3.1, we combine both areas into a new area u ($\mathcal{L}_u = \mathcal{L}_s \cup \mathcal{L}_t$, $\mathcal{G}_u = \mathcal{G}_s \cup \mathcal{G}_t$), add u to $\mathcal{S}$, remove s and t from $\mathcal{S}$, and check again whether all areas $s \in \mathcal{S}$ are mutually independent according to Definition 3.1. The process is finished when all areas in $\mathcal{S}$ are confirmed to be independent[4]. As an example, Figure 3.4 shows how London Heathrow Airport Terminal 4 is separated into four independent areas following this procedure.

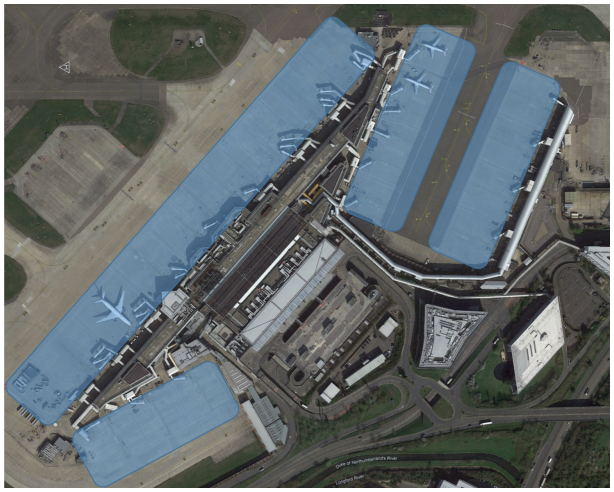


Figure 3.4: Independent areas of London Heathrow Airport Terminal 4 (Source: Google Maps)

3.5.3 Determining the set of demand decompositions

We identify demand decompositions using a two-step procedure: First, we determine for each area $s \in \mathcal{S}$ how many aircraft of which classes *can* be parked simultaneously at lead-in lines $\mathcal{L}_s$.

[4]In a worst-case scenario, $|\mathcal{S}| = 1$. In that case, our approach has no advantage over solving Model (3.1a)-(3.1p) directly, as the problem cannot be decomposed into independent areas. However, in reality $|\mathcal{S}| \geq 2$ is true for most terminals, especially when the terminal is large. Furthermore, if $|\mathcal{S}| = 1$, our approach can still be used as a heuristic by manually removing individual lines from the set of lead-in lines $\mathcal{L}$, resulting in independent areas according to Definition 3.1.

Definition 3.2 (**Parking patterns**). Let $r_a \in \mathbb{N}_0$ represent a number of class $a \in \mathcal{A}$ aircraft. We call $(r_1, r_2, \ldots, r_A)$ a feasible parking pattern for an area if all aircraft $r_1, r_2, \ldots, r_A$ can be parked at contact gates of the area simultaneously. Let $r_{ap} \in \mathbb{N}_0$ denote the number of aircraft of class $a \in \mathcal{A}$ that are parked simultaneously in a parking pattern p. We call $p = (r_{1p}, r_{2p}, \ldots, r_{Ap})$ an efficient parking pattern for an area if r_{ap} cannot be increased for any $a \in \mathcal{A}$ without leading to infeasibility with respect to Safety Constraints (3.1j). The set of efficient parking patterns associated with area $s \in \mathcal{S}$ is denoted by $\mathcal{P}_s$.

Example 3.1. Let $\mathcal{A} = \{1, 2\}$, with $a = 1$ and $a = 2$ representing small and large aircraft, respectively. Assume that in a given area it is possible to park a maximum of four small aircraft when zero large aircraft are parked, and that a maximum of two large aircraft can be parked when zero small aircraft are parked. Furthermore, assume that two small aircraft can be parked when the number of large aircraft that are parked is one. Then, the set of efficient parking patterns for area $s \in \mathcal{S}$ is given as $\mathcal{P}_s = \{(4, 0), (2, 1), (0, 2)\}$.

Second, once $\mathcal{P}_s$ has been determined for all areas $s \in \mathcal{S}$, we create demand decompositions by selecting one parking pattern $p \in \mathcal{P}_s$ for each demand pattern $k \in \mathcal{K}$ and area $s \in \mathcal{S}$.

Definition 3.3 (**Demand decompositions**). Let the function $p(c, s, k)$ return the parking pattern p that is selected from $\mathcal{P}_s$ for area $s \in \mathcal{S}$ and demand pattern $k \in \mathcal{K}$ in demand decomposition $c \in \mathcal{C}$. Then, demand decomposition $c \in \mathcal{C}$ is defined as

$$\begin{pmatrix} p(c,1,1) & p(c,1,2) & \ldots & p(c,1,K) \\ p(c,2,1) & p(c,2,2) & \ldots & p(c,2,K) \\ \vdots & \vdots & \vdots & \vdots \\ p(c,S,1) & p(c,S,2) & \ldots & p(c,S,K) \end{pmatrix}.$$

Patterns are selected such that $\sum_{a \in \mathcal{A}} \sum_{k \in \mathcal{K}} W_{ak} \cdot \left(D_{ak} - \sum_{s \in \mathcal{S}} r_{ap(c,s,k)} \right)$ is minimized, which is equivalent to Objective Function (3.1a).

In the following, let $\mathcal{P}_{cs} = \{p(c, s, 1), p(c, s, 2), \ldots, p(c, s, K)\}$ contain the parking patterns from the s-th row of demand decomposition $c \in \mathcal{C}$.

3.5.3.1 Identifying efficient parking patterns

To determine the set of efficient parking patterns $\mathcal{P}_s$ for area $s \in \mathcal{S}$, we employ a graph-based approach, which is inspired by existing work on identifying the set of pareto-efficient paths through a network (see, e.g., Martins 1984, Tung Tung and Lin Chew 1992) and dynamic programming.

We create a directed graph in which each feasible combination of lead-in line $l \in \mathcal{L}_s$ and aircraft class $a \in \mathcal{A}$ ($C_{al} = 1$) is represented by a node (l, a). Arcs between nodes represent feasible combinations of two parked aircraft, i.e., two nodes (l, a) and (m, b) are connected if $E_{lmab} = 0$. Arcs are directed from lead-in lines with lower indices to lead-in lines with higher indices. We add two dummy nodes to the network, one representing a single source and the other representing a single sink. For each non-dummy node, an in-arc is added from the source node and an out-arc is added to the sink node. Each path through this network corresponds to one feasible, yet not necessarily efficient parking pattern for the respective area. We construct paths incrementally by iterating over all nodes, starting at the source node. Each iteration consists of two steps: First, we compare all paths that lead into the particular node and prune paths if possible. Second, we extend all non-dominated paths leading into the node to all nodes, which can be reached from that node.

A path is pruned if it is dominated by another path (see Definition 3.4) or it becomes apparent that the parking pattern associated with it is not needed given the demand patterns (see Lemma 3.1). Let $\sigma_{a\pi}^{(i)}$ denote the number of class $a \in \mathcal{A}$ aircraft parked in path π leading from the source node to node i.

Definition 3.4 (**Path dominance**)**.** A path π_1 dominates another path π_2 if $\sigma_{a\pi_1}^{(i)} \geq \sigma_{a\pi_2}^{(i)}$ $\forall a \in \mathcal{A}$ and $\sigma_{a\pi_1}^{(i)} > \sigma_{a\pi_2}^{(i)}$ $\exists a \in \mathcal{A}$.

This directly follows from Definition 3.2. If paths $\pi_1, \pi_2, \ldots, \pi_n$ are equivalent, i.e., $\sigma_{a\pi_1}^{(i)} = \sigma_{a\pi_2}^{(i)} = \ldots = \sigma_{a\pi_n}^{(i)}$ $\forall a \in \mathcal{A}$, one path is selected arbitrarily and all other paths are pruned.

Lemma 3.1 (**Paths and demand patterns**)**.** *A path π can be pruned if $\sigma_{a\pi}^{(i)} > max_{k \in \mathcal{K}} \{D_{ak}\}$ $\exists a \in \mathcal{A}$ at a given node i.*

We consider situations where space is scarce[5], i.e., the optimal value of Objective Function (3.1a) is larger than 0 with respect to each demand pattern $k \in \mathcal{K}$. Thus, parking more class $a \in \mathcal{A}$ aircraft than is required for any of the demand patterns in $\mathcal{K}$ would translate to fewer aircraft of another class $b \in \mathcal{A}$ that can be parked and hence, would result in a worse outcome with respect to Objective Function (3.1a). □

Example 3.2. Assume there are $A = 2$ aircraft classes and demand patterns $(15, 0)$, $(13, 1)$, and $(10, 2)$. Then, a path π with $\sigma_{2\pi} = 3$ can be pruned, since no more than two aircraft of class 2 need to be accommodated for any demand pattern.

Before we extend a path to node (m, b), we verify that parking a class b aircraft at lead-in line m does not lead to collisions with any other aircraft already considered in the path, i.e., that $E_{lmab} = 0$ for all nodes (l, a) already included in the path. While the definition of arcs ensures that this is always true for the node that was most recently added to the path, it may not be true for nodes added to the path earlier. Whenever we find that $E_{lmab} = 1$ for any node (l, a) already included in the path, we do not extend the path to node (m, b). We observe this phenomenon when the value of κ is small and/or when the terminal facade has a corner.

After the last iteration, i.e., when all paths have reached the sink node, none of the remaining paths is dominated by another path. Thus, each path that reaches the sink node is associated with a unique and efficient parking pattern for area s. Then, $\mathcal{P}_s$ contains all existing efficient parking patterns for area $s \in \mathcal{S}$ except of those which cannot be part of the optimal solution due to Lemma 3.1.

3.5.3.2 Creation of demand decompositions

A demand decomposition $c \in \mathcal{C}$ is created by selecting one parking pattern $p \in \mathcal{P}_s$ for each area $s \in \mathcal{S}$ and demand pattern $k \in \mathcal{K}$. Depending on the cardinalities of $\mathcal{P}_s$, $\mathcal{S}$, and $\mathcal{K}$, the number of possible demand decompositions may be very large. However, we are only interested in those demand decompositions that lead to z_1^*. Objective Function

[5]Whether this condition is satisfied can be checked in practice, for example, by first solving the AGLP for each demand pattern separately using CPLEX. If $z_1^* = 0$ and slack space exists, space is not scarce and our decomposition approach should not be applied to that particular instance.

(3.1a) as well as demand decompositions can be decomposed with respect to demand patterns $k \in \mathcal{K}$. Let z^*_{1k} be the optimal value of Objective Function (3.1a) with respect to demand pattern $k \in \mathcal{K}$, i.e., $z^*_1 = \sum_{k \in \mathcal{K}} z^*_{1k}$. In order to lead to z^*_1, a demand decomposition must lead to z^*_{1k} for each $k \in \mathcal{K}$; we use this by dividing the process to create demand decompositions into $K + 1$ stages. In the k-th stage, we pretend that demand pattern k is the only demand pattern that exists and create the set of all possible demand decompositions, which we denote as $\mathcal{C}_k$. Following the matrix representation of demand decompositions as given in Definition 3.3, each demand decomposition $c \in \mathcal{C}_k$ is a column vector $(p(c,1,k), p(c,2,k), \ldots, p(c,S,k))^T$. Next, for each $c \in \mathcal{C}_k$ we calculate the associated value of Objective Function (3.1a) with respect to demand pattern k as $\sum_{a \in \mathcal{A}} \left(W_{ak} \cdot \max \left\{ \left(D_{ak} - \sum_{s \in \mathcal{S}} r_{ap(c,s,k)} \right), 0 \right\} \right)$. Then, we determine z^*_{1k} as the minimum of the results and remove all $c \in \mathcal{C}_k$ not leading to z^*_{1k}. Thus, in each stage $k \in \mathcal{K}$ we obtain the set of vectors that can be chosen for column k in demand decompositions $c \in \mathcal{C}$. In stage $K + 1$, $\mathcal{C}_k$ has been determined for all $k \in \mathcal{K}$ and we can create the set of demand decompositions $\mathcal{C}$. That is, $\mathcal{C}$ is obtained as the Cartesian product $\mathcal{C}_1 \times \mathcal{C}_2 \times \ldots \times \mathcal{C}_K$.

3.5.4 Computing lower bounds for demand decompositions

The value of the lower bound LB_c for a demand decomposition $c \in \mathcal{C}$ is derived from the minimum number of gates needed to handle all aircraft contained in demand decomposition c and the required equipment for these gates. Since each demand decomposition $c \in \mathcal{C}$ defines how many aircraft of which class are assigned to which area $s \in \mathcal{S}$ and the gates of the areas can be planned independently, we first compute area-specific partial bounds LB_{cs} and then compute LB_c as $LB_c = \sum_{s \in \mathcal{S}} LB_{cs}$.

This division into area-specific bounds LB_{cs} has two advantages: First, calculating LB_c as the sum of the area-specific bounds LB_{cs} leads to a tighter bound on the value of Objective Function (3.1b), because the assignment of aircraft to areas is incorporated in the values of LB_{cs}. Second, depending on the particular instance, different demand decompositions $c_1, c_2 \in \mathcal{C}$ are often equivalent with respect to individual areas $s \in \mathcal{S}$, i.e., $\mathcal{P}_{c_1 s} = \mathcal{P}_{c_2 s}$. From this follows $LB_{c_1 s} = LB_{c_2 s}$. Thus, once the subproblem of demand decomposition c_1 is solved to optimality for area s, not only can $LB_{c_1 s}$ be updated, but

also LB_{c_2s} and hence, LB_{c_2}. This potentially allows eliminating c_2 if $LB_{c_2} > UB$ after the update.

When computing LB_{cs} for a pattern configuration $c \in \mathcal{C}$ and an area $s \in \mathcal{S}$, we do not specify which of the gates from $\mathcal{G}_s$ are used but we only consider the minimum number of gates required per aircraft class. However, we take downward compatibility of gates as well as MARS mode into account. Let $\omega_{csa} \in \mathbb{N}$ be the minimum number of gates equipped for class $a \in \mathcal{A}$ aircraft required in area $s \in \mathcal{S}$ to park all aircraft contained in $\mathcal{P}_{cs}$. Furthermore, let $\psi_{csak} \in \mathbb{N}$ be the number of class $a \in \mathcal{A}$ aircraft that can be parked at gates which are equipped for aircraft of class $a' > a$ in area $s \in \mathcal{S}$ for demand pattern $k \in \mathcal{K}$ when aircraft are assigned to areas according to demand decomposition $c \in \mathcal{C}$. Then, Algorithm 3.2 describes our procedure to calculate LB_{cs} in detail.

Algorithm 3.2 Algorithm to determine the value of LB_{cs}

Initialize LB_{cs}, ω_{csa}, and ψ_{csak}
for all aircraft classes $a \in \mathcal{A}$ (in descending order) **do**
  $\omega_{csa} \leftarrow 0$
  for all demand patterns $k \in \mathcal{K}$ **do**
    if $a = A$ **then**
      $\omega_{csa} \leftarrow \max\left\{\omega_{csa}, r_{ap(c,s,k)}\right\}$
    else
      if $r_{ap(c,s,k)} > \omega_{csa}$ **then**
        if $a \in \mathcal{A}^{\text{large}} \setminus \{A\}$ **then**
          $\psi_{csak} \leftarrow \sum_{a' \in \mathcal{A}: a' > a} \left(\omega_{csa'} - r_{a'p(c,s,k)}\right)$
        else if $a \in \mathcal{A}^{\text{small}}$ **then**
          $\psi_{csak} \leftarrow \sum_{a' \in \mathcal{A}: a' > a, a' \in \mathcal{A}^{\text{large}}} 2 \cdot \left(\omega_{csa'} - r_{a'p(c,s,k)}\right)$
            $+ \sum_{a' \in \mathcal{A}: a' > a, a' \in \mathcal{A}^{\text{small}}} \left(\omega_{csa'} - r_{a'p(c,s,k)}\right)$
        end if
        $\omega_{csa} \leftarrow \left(r_{ap(c,s,k)} - \psi_{csak}\right)$
      end if
    end if
  end for
end for
$LB_{cs} \leftarrow \sum_{a \in \mathcal{A}} a \cdot \omega_{csa}$
return LB_{cs}

First, we determine the minimum number of gates required for aircraft of the largest class A as $\max_{k \in \mathcal{K}} \left\{ r_{Ap(c,s,k)} \right\}$ in line 6. Next, we consider the next smaller aircraft class $A-1$. Here, for each demand pattern $k \in \mathcal{K}$, we first compute how many of the class $A-1$ aircraft contained in $p(c,s,k)$ can be parked at the gates equipped for class A aircraft, but not occupied by a class A aircraft for demand pattern k (lines 9 to 14). For the remaining aircraft, additional gates for class $A-1$ aircraft must be provided (line 15). We continue according to this scheme until all aircraft classes have been considered. As soon as the aircraft class under consideration is no longer included in $\mathcal{A}^{\text{large}}$ but in $\mathcal{A}^{\text{small}}$, we assume that the MARS mode can always be used, i.e., that two aircraft of class $a \in \mathcal{A}^{\text{small}}$ can be parked simultaneously at gates which are equipped for aircraft of the classes belonging to $\mathcal{A}^{\text{large}}$ (line 12).

In brownfield settings, we compare the results from Algorithm 3.2 with the already existing gates and adjust gate quantities ω_{csa} accordingly if necessary. We provide our algorithm for that task in Appendix B.5.

3.5.5 Acceleration techniques

We apply a number of problem-specific acceleration techniques to improve the performance of our approach.

Solution pool Subproblems for a given area $s \in \mathcal{S}$ are often identical for different demand decompositions, i.e., the same number of aircraft per class are assigned to a given area in different demand decompositions. Therefore, we make use of a solution pool, in which we store the solutions to all subproblems already solved and which we inspect each time before solving another subproblem.

Solving relaxed subproblems Each time we search for the optimal solution of a subproblem, we iteratively solve a relaxation of the problem and try to show that the solution satisfies all constraints that were relaxed. If the latter fails, we add the violated constraints to the relaxation and re-solve it in the next iteration. Otherwise, the solution is feasible for the original subproblem. Similar approaches exist in literature on the

vehicle routing problem, where subtour elimination constraints are relaxed and added to the problem only when they are violated by the solution (see, e.g., Laporte et al. 1984). The solutions to the relaxed problems are also added to the solution pool and reused if possible.

Consider the subproblem for a demand decomposition $c \in \mathcal{C}$ and an area $s \in \mathcal{S}$. In the relaxed subproblem, we first consider only one demand pattern $\hat{k}$, and all constraints of demand patterns $\mathcal{K} \setminus \left\{\hat{k}\right\}$ are removed. Once the optimal solution for the relaxed subproblem is found, we determine for each of the remaining demand patterns $k \in \mathcal{K} \setminus \left\{\hat{k}\right\}$ separately whether the solution is feasible or not, i.e., whether all aircraft contained in parking pattern $p\,(c, s, k)$ can be parked at contact gates in area s given the positions of and equipment installed at the gates in the solution. If we fail to show that the solution is feasible for one demand pattern $\tilde{k}$, the constraints associated with demand pattern $\tilde{k}$ are added to the relaxed problem and the next iteration is started by solving the relaxed problem again.

The behavior of the algorithm can be influenced by the sequence in which demand patterns are considered. We first sort the parking patterns $\mathcal{P}_{cs}$ by decreasing values of $r_{Ap(c,s,k)}$, i.e., by decreasing number of aircraft of the largest class. When two parking patterns have equal values of $r_{Ap(c,s,k)}$, we sort them by decreasing number of aircraft of the second largest aircraft class $A-1$, etc. The demand pattern we consider first in the subproblem is determined by the value of k of the first parking pattern in the resulting list, i.e., we select the demand pattern where the number of aircraft belonging to the largest class that occurs in $\mathcal{P}_{cs}$ is the highest. This ensures that already in the first iteration enough gates are equipped for the aircraft of that class.

Definition 3.5 (**Bottom-up and top-down strategies**)**.** We introduce two alternative strategies defining the sequence in which the remaining demand patterns are checked for feasibility.

- Top-down: We consider the demand patterns in the same order in which their associated parking patterns $p\,(c, s, k)$ appear in the sorted list.
- Bottom-up: We check the demand patterns in the opposite order. That is, we begin with the demand pattern whose associated parking pattern $p\,(c, s, k)$ accommodates

the smallest number of large aircraft. Typically, this parking pattern has the highest total number of aircraft, which is the motivation for first examining the demand pattern associated with this parking pattern.

Example 3.3. Consider a situation where $K = 3$, $A = 2$, $p(c, s, 1) = (6, 0)$, $p(c, s, 2) = (4, 1)$, and $p(c, s, 3) = (2, 2)$ for a given demand decomposition $c \in \mathcal{C}$ and area $s \in \mathcal{S}$. We select demand pattern 3 with parking pattern $(2, 2)$ in the first iteration, which ensures that two gates for class 2 aircraft are built in the solution of the relaxed subproblem in the first iteration. The solution may also be feasible for the remaining demand patterns 1 and 2, as gates are downward compatible and the MARS mode can be used. To verify that the solution is indeed feasible for demand patterns 1 and 2 without violating any Safety Constraints (3.1j), we need to perform a feasibility check for both demand patterns individually. If we apply the bottom-up strategy, we first check demand pattern 1 with parking pattern $(6, 0)$. In contrast, following the top-down strategy we begin with demand pattern 2 associated with parking pattern $(4, 1)$.

Checking for feasibility To determine if a solution of the relaxed subproblem is feasible for a demand pattern not considered in the relaxation, we use an adapted version of the network approach presented in Section 3.5.3. Let $\hat{v}_{ga}$ be the value of the variable v_{ga} in the current solution of the relaxed problem, and let k be the demand pattern for which the solution is checked. Before we create paths through the network, we delete all nodes representing combinations of lead-in lines and aircraft classes for which no feasible gate $g \in \mathcal{G}_s : F_{lg} = 1, \hat{v}_{ga} = 1$ exists. Again, we use an iterative process in which paths are constructed, checked for reasonableness, and pruned if possible. Beyond the pruning rules defined in Section 3.5.3, a path is truncated at node i if it meets any of the following conditions.

Lemma 3.2 (**Paths and parking patterns**). *A path π can be pruned if $\sigma_{a\pi}^{(i)} > r_{ap(c,s,k)}$ $\exists a \in \mathcal{A}$.*

If the number of class $a \in \mathcal{A}$ aircraft parked in path π leading from the source node to node i $\sigma_{a\pi}^{(i)}$ is greater than the number of class a aircraft contained in parking pattern $p(c, s, k)$, $\sigma_{a'\pi}^{(\text{sink})}$ must be smaller than $r_{a'p(c,s,k)}$ for another aircraft class $a' \in \mathcal{A} \setminus \{a\}$,

because parking pattern $p(c, s, k)$ is efficient according to Definition 3.2. Hence, path π is not suitable to show that the solution is feasible for demand pattern k. $\square$

In addition, we prune a path if it violates Constraints (B.1d), (B.1f), or (B.1h). That is, (i) each lead-in line that is used repeatedly for different demand patterns has to be assigned to the same gate for all demand patterns, (ii) aircraft can only be parked at lead-in lines which are assigned to gates equipped to handle aircraft of the respective class, and (iii) a gate can simultaneously accommodate at most one aircraft belonging to $\mathcal{A}^{\text{large}}$ or two aircraft belonging to $\mathcal{A}^{\text{small}}$. Let $\hat{y}_l$ and $\hat{u}_{gl}$ denote the values of variables y_l and u_{gl} in the current solution of the relaxed subproblem, respectively. Furthermore, let $\beta_{ga} \in \mathbb{N}$ denote the number of class $a \in \mathcal{A}$ aircraft that are handled at gate $g \in \mathcal{G}_s$ for demand pattern k simultaneously. Then, we check compliance of a path π with Constraints (B.1d), (B.1f), and (B.1h) as shown in Algorithm 3.3.

Algorithm 3.3 Heuristic algorithm to check compliance of a path with Constraints (B.1d), (B.1f), and (B.1h)

```
1: Initialize β_ga and boolean variable ς
2: for all nodes in path π do
3:    ς ← false
4:    Get l ∈ L_s and a ∈ A represented by the node
5:    for all g ∈ G_s if F_lg = 1 and v̂_ga = 1 do  ▷ Constraints (B.1f)
6:       if not (ŷ_l = 1 ∧ û_gl = 0) then          ▷ Constraints (B.1d)
7:          if Σ_{a∈A^small} β_ga + 2 · Σ_{a∈A^large} β_ga < 2 then   ▷ Constraints
   (B.1h)
8:             β_ga ← β_ga + 1
9:             ς ← true
10:            break
11:         end if
12:      end if
13:   end for
14:   if ς = false then
15:      return false                          ▷ solution is infeasible
16:   end if
17: end for
18: return true                                 ▷ solution is feasible
```

Algorithm 3.3 can be described as a greedy heuristic that tries to assign parked aircraft

to the first available gate. Hence, there is no guarantee that Algorithm 3.3 returns that a given solution is feasible if it actually *is* feasible; however, if the solution is actually *in*feasible, Algorithm 3.3 will always return that it is infeasible.

Monitoring the upper bound Each time we have solved a (relaxed) subproblem for a demand decomposition $c \in \mathcal{C}$, we check whether z^*_{2c} can still be smaller than the current value of UB given the solution(s) found so far. If not, the solution process for demand decomposition c can be stopped, and c can be removed from $\mathcal{C}$. As we have shown in Section 3.5.4, it may still be possible to update $LB_{\tilde{c}}$ for other demand decompositions $\tilde{c} \in \mathcal{C} \setminus \{c\}$ based on the solutions determined until abortion, which may also lead to their removal from $\mathcal{C}$.

Consistency of demand decompositions In the bounding algorithm, the set of subproblems to be solved next is found by selecting the demand decomposition from $\mathcal{C}$ that has the smallest LB_c value, ensuring that in each iteration the demand decomposition with the best potential to lead to a new best solution is explored. However, due to the large number of demand decompositions, we often observe several with the same LB_c values. We then sort the affected demand decompositions according to what we call the consistency criterion.

Definition 3.6 (**Consistency**)**.** Let ζ_{cak} denote the number of aircraft belonging to classes $\{a, \dots, A\}$ that are parked at all areas for demand pattern $k \in \mathcal{K}$ when aircraft are assigned to areas according to demand decomposition $c \in \mathcal{C}$, i.e., $\zeta_{cak} = \sum_{s \in \mathcal{S}} \sum_{a' \in \{a,\dots,A\}} r_{a'p(c,s,k)}$. We call demand decomposition c consistent if there is no area $s \in \mathcal{S}$ for which $\sum_{a' \in \{a,\dots,A\}} r_{a'p(c,s,k_1)} < \sum_{a' \in \{a,\dots,A\}} r_{a'p(c,s,k_2)}$ holds for any aircraft class $a \in \mathcal{A}^{\text{large}}$ and two demand patterns $k_1, k_2 \in \mathcal{K} : k_1 \neq k_2$, given that $\zeta_{cak_1} \geq \zeta_{cak_2}$. A demand decomposition thus is consistent if large aircraft are repeatedly assigned to the same areas for different demand patterns, rather than being assigned to different areas. According to the consistency criterion, preference should be given to consistent demand decompositions over non-consistent demand decompositions with same LB_c value.

Example 3.4. Consider a greenfield situation with $A = 2$, where $a = 1$ and $a = 2$ represent classes of small and large aircraft, respectively. Assume $K = 2$, with the

demand patterns given as $D_{ak} = (10, 1)$ and $(8, 2)$, and assume there are $S = 2$ areas. Furthermore, consider two distinct demand decompositions $c_1 = \begin{pmatrix} (3,1) & (3,1) \\ (6,0) & (4,1) \end{pmatrix}$ and $c_2 = \begin{pmatrix} (3,1) & (5,0) \\ (6,0) & (2,2) \end{pmatrix}$. According to Algorithm 3.2, $LB_{c_1} = LB_{c_2}$. However, c_1 is a consistent demand decomposition, whereas c_2 is not, and hence, demand decomposition c_1 is considered first in the bounding algorithm following the consistency criterion.

We provide the algorithm we utilize to determine whether a demand decomposition is consistent or not in Appendix B.6.

3.6 Computational experiments

In the following, we examine the performance of our approach and analyze the solutions of individual instances. In Section 3.6.1, we describe the instances we used in our experiments. Section 3.6.2 compares the performance of our approach to CPLEX solving Model (3.1a)-(3.1p) directly. In Section 3.6.3, we examine how the performance of our approach behaves for larger instances, and in Section 3.6.4, we investigate the impact of the acceleration techniques introduced in Section 3.5.5. Finally, we discuss the optimal solutions for individual instances in detail in Section 3.6.5. All experiments were performed on a computer equipped with an Intel Xeon E3-1225 v3 @3.20 GHz processor and 12 GB of working memory. The implementation was done in Java and CPLEX version 20.1.0 was used. To facilitate the reproducibility of our results and to encourage further research on the AGLP, all our instances are available in a public repository at Mendeley Data (Hagspihl et al. 2023b).

3.6.1 Instances

All instances are based on data from Munich Airport Terminal 1. With 47.9 million passengers in 2019, Munich Airport is the second largest airport in Germany (Munich Airport 2020). Before the crisis caused by the Corona virus in 2020, 101 airlines were

active at Munich Airport, connecting the airport to 254 destinations in 75 countries. Transfer passengers accounted for 38% of total passenger traffic in 2019, so Munich Airport is considered a hub airport. The airport has two terminals, Terminal 1 being the smaller of the two providing about a third of the airport's total capacity. The terminal is currently being renovated and extended by a new section. Figure 3.5 shows the apron layout for Terminal 1, with the new extension highlighted in red.

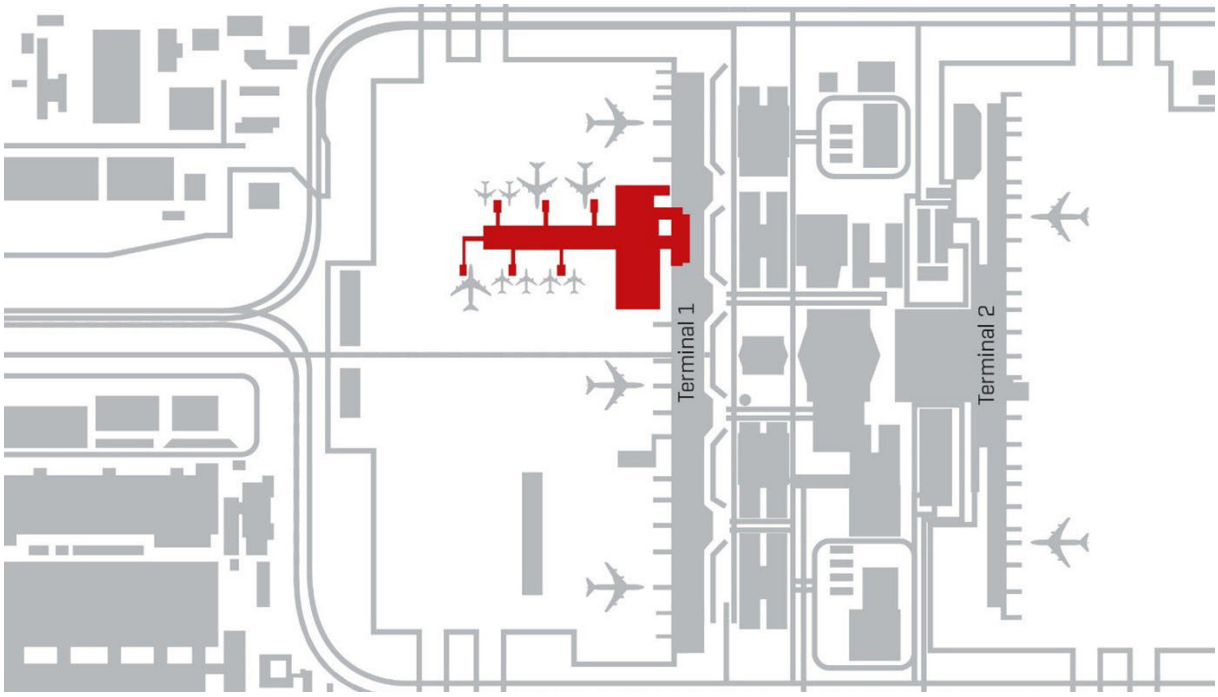


Figure 3.5: Terminal layout of Munich Airport and extension of Terminal 1 (Source: Munich Airport)

As part of the expansion and reconstruction activities, the positions of gates and lead-in lines have to be determined for the new section of the terminal, and changes may also be made in existing sections of Terminal 1. We consider three planning scenarios, resulting in three separate sets of instances:

- Greenfield: Existing gates and lead-in lines are not considered.
- Soft brownfield: Existing gates and lead-in lines south of the extension are considered. Existing gates cannot be changed, but additional lead-in lines and gates can be added to the layout of these sections.

- True brownfield: In sections south of the extension, the gate layout remains unchanged, i.e., only existing gates and lead-in lines can be used.

As demand patterns are defined for the terminal building as a whole, we solve the AGLP for the entire terminal, including the sections south of the extension regardless of the planning scenario. To determine the sets of lead-in lines $\mathcal{L}$ and gates $\mathcal{G}$ for all instances, as well as the values of the parameters C_{al}, E_{lmab}, F_{lg}, and H_{ga}, which determine the relationships among the gates and lead-in lines, we applied the procedure described in Section 3.3 and validated the results with planning experts from Munich Airport. We employed satellite images as well as floor plan drawings provided by Munich Airport to determine the relevant coordinates of the terminal building and surrounding taxiways. In all brownfield instances, 11 gates and 18 lead-in lines that already exist south of the extension were added to sets $\mathcal{G}$ and $\mathcal{L}$, respectively, based on their coordinates. Furthermore, all other gates and lead-in lines south of the new extension were removed from sets $\mathcal{G}$ and $\mathcal{L}$ in the true brownfield instances. To investigate the computational performance of our approach, we created instances of different complexity for each planning scenario by varying the distance between adjacent starting points Δ as well as the rotation angle κ between the lead-in lines that share the same starting point, where $\Delta \in \{10\text{ m}, 7.5\text{ m}, 5\text{ m}, 2.5\text{ m}\}$ and $\kappa \in \{90°, 45°, 30°, 22.5°, 15°\}$. Depending on the values of Δ and κ, the number of gates G considered in the model is between 136 and 507, and the number of lead-in lines L considered is between 119 and 2,400. We ensured that 90° is an integer multiple of each value of κ, since we concluded from the exchange with Munich Airport that lead-in lines perpendicular to the terminal facade are the easiest to operate in practice. However, the use of non-perpendicular lead-in lines may still considerably improve z_1^*, as they may allow accommodation of additional aircraft in corners or at the ends of the terminal building. We visualize sets $\mathcal{G}$ and $\mathcal{L}$ for $\Delta = 5$ m and $\kappa = 22.5°$ in Appendix B.7 for each planning scenario. In summary, we created instances for three scenarios, four values of Δ, and five values of κ, resulting in $3 \cdot 4 \cdot 5 = 60$ instances in total.

The smaller the values of Δ and κ, the more flexibly aircraft can be parked on the apron, and the better solutions we expect for Objective Function (3.1a). Better solutions for Objective Function (3.1a) might suggest worse solutions for Objective Function (3.1b), as accommodating more aircraft requires more gates. However, the smaller the values of Δ and κ, the more precisely gates can be placed, potentially reducing the number of gates needed to serve a given number of aircraft simultaneously.

Table 3.2: Values of D_{ak} derived from the flight plan forecast for the year 2030

	k				
a	1	2	3	4	5
1	39	31	29	25	23
2	0	2	3	5	6
3	0	0	0	1	1

In all instances, the set of aircraft classes is given as $\mathcal{A} = \{1, 2, 3\}$, where aircraft classes 1 , 2, and 3 correspond to ADGs III, V, and VI as introduced in Table 3.1. Hence, $\mathcal{A}^{\text{small}} = \{1\}$ and $\mathcal{A}^{\text{large}} = \{2, 3\}$. Based on the interviews with experts from Munich Airport, we set C_{al} to 1 for all $a \in \mathcal{A}$ and $l \in \mathcal{L}$. Furthermore, F_{lg} equals 1 if the path between gate g and lead-in line l is unobstructed, and gate g is located at a maximum distance of 40.5 meters to the left of lead-in line l from the point of view of the parked aircraft. Finally, the values of E_{lmab} were determined as stated in Section 3.3.

Demand patterns $k \in \mathcal{K}$ and associated traffic volumes D_{ak} were obtained from the flight plan forecast of Munich Airport for the year 2030. We identified five demand patterns for Terminal 1, which are given in Table 3.2.

Aircraft of ADGs I, II, and IV are rarely seen at Munich Airport and are therefore not present in any demand pattern. Finally, the weights W_{ak} for Objective Function (3.1a) are calculated as follows: We assume that all demand patterns given in Table 3.2 have equal likelihood, and hence the values of W_{ak} are independent of k. In contrast, larger aircraft are associated with larger weights. Thus, for each $a \in \mathcal{A}$ and $k \in \mathcal{K}$, we set W_{ak} to $0.2 \cdot a^2$.

3.6.2 Computational performance compared to commercial solver

To evaluate the performance of our approach, we applied it to all 20 soft brownfield instances. We selected the soft brownfield scenario, because the sets $\mathcal{G}$ and $\mathcal{L}$ are the largest for given values of Δ and κ compared to the greenfield or true brownfield counterparts.

Table 3.3: Computational results for soft brownfield instances

Instance					Objectives				Runtimes [min]				Info on DA				
					Objective (3.1a)		Objective (3.1b)		Objective (3.1a)		Objective (3.1b)						
Nr.	Δ [m]	κ [°]	G	L	CPLEX*	DA	CPLEX*	DA*	CPLEX	DA	CPLEX	DA	C	C_{con}	C_{exa}	SP_{exa}	$\bar{K}_{SP}$
1	10	90	136	119	6.8	6.8	34	34	0	0	0	0	2625	69	4	6	1.86
2		45		235	6.2	6.2	35	35	0	0	1	0	798	36	4	6	1.86
3		30		297	4.6	4.6	43 [12]	43	0	0	> 1440	0	630	20	2	4	2
4		22.5		418	4.0	4.0	37	37	7	0	36	1	416	48	4	6	1.5
5		15		613	3.4	3.4	–	39	139	2	> 1440	2	396	16	2	5	1.4
6	7.5	90	178	154	8.6	8.6	34	34	0	0	3	1	840	80	6	12	1.5
7		45		310	7.4	7.4	34	34	3	0	65	0	2750	112	2	7	1.14
8		30		392	5.6	5.6	36	36	3	0	14	1	80	18	4	5	1.57
9		22.5		556	4.4	4.4	40	40	504	1	12	1	198	24	2	4	1
10		15		815	4.6 [43]	4.6	–	36	> 1440	5	> 1440	7	384	48	2	4	1.25
11	5.0	90	260	222	6.2	6.2	35	35	0	0	2	0	315	18	2	4	1.5
12		45		454	5.6	5.6	–	39	82	1	> 1440	1	630	20	2	3	1.5
13		30		578	4.2	4.2	37 [1]	37	787	2	> 1440	4	980	36	8	8	1.2
14		22.5		823	3.2 [50]	3.2	–	37	> 1440	5	> 1440	189	1400	48	3	5	1.5
15		15		1210	6.8 [91]	3.2	–	38	> 1440	17	> 1440	334	770	30	2	4	1.75
16	2.5	90	507	428	5.6	5.6	38	38	2	1	138	2	216	42	4	7	1.43
17		45		891	5.2 [77]	4.2	–	41	> 1440	5	> 1440	511	24	6	2	4	2
18		30		1139	32 [99]	3.6	14	40	> 1440	14	15	313	504	63	6	8	1.38
19		22.5		1626	38.2 [100]	2.8	14	39 [3]	> 1440	25	32	> 1440	96	12	≥ 2	≥ 5	1.2
20		15		2400	45.8 [100]	2.2	14	≥ 38**	> 1440	172	11	> 1440	132	9	≥ 1	≥ 3	–
av.			270	684	10.4	5.1	–	37.5	581	12	652	212	709	37.75	3.2	5.5	1.48

* Values in parentheses indicate optimality gaps in percent
** Lower bound of first demand decomposition examined in the bounding algorithm

We used all acceleration techniques introduced in Section 3.5.5 and applied the top-down strategy. As a benchmark, we solved Model (3.1a)-(3.1p) using CPLEX, with a time limit of 24 hours per objective function. In Table 3.3, we provide for each soft brownfield instance the resulting objective function values, optimality gaps, and runtimes for both CPLEX and our decomposition approach (DA). Furthermore, we provide relevant performance indicators for our approach, where C_{con} denotes the number of consistent demand decompositions of an instance; C_{exa} and SP_{exa} represent the number of demand decompositions and subproblems, respectively, examined in the bounding algorithm; and $\bar{K}_{SP}$ gives the average number of demand patterns that had to be included in the relaxed subproblems until either feasibility of the solution for the remaining demand patterns was proven or the optimization process was aborted.

Our results demonstrate that Model (3.1a)-(3.1p) cannot be solved to optimality by CPLEX for small values of Δ and κ within a reasonable amount of time. For five instances, z_1^* was not found within 24 hours. Furthermore, for two instances z_1^* was found but not proven to be optimal. Only for 11 of the 15 instances, for which z_1^* could be determined, could z_2^* also be found within the time limit. Again, for two of these 11 instances,

z_2^* was found but not proven to be optimal. In summary, CPLEX could solve only 11 of 20 instances to optimality, thereby proving optimality for only nine instances. In particular, no instance with $\kappa = 15°$ could be solved to optimality, and for $\Delta \in \{5 \text{ m}, 2.5 \text{ m}\}$ only the instances with $\kappa = 90°$ could be solved to optimality with no remaining optimality gap.

In contrast, with the decomposition approach we can solve 18 of 20 instances to optimality within 24 hours, and we find a feasible solution for one additional instance with an optimality gap of 2.63%. Our approach finds z_1^* for all instances within less than three hours. Averaged over all instances where z_1^* was found and proven to be optimal by CPLEX within the time limit, our approach reduces the runtime per instance for Objective Function (3.1a) by 84.14%. As for Objective Function (3.1b), the decomposition approach reduces the average computation time per instance by 84.06% compared to CPLEX. Here, we only consider the instances for which both CPLEX and our algorithm found z_1^* and z_2^* and proved optimality within the time limit. Averaged over the same instances, our approach reduces the total computation time per instance, i.e., the time needed to solve both objective functions to optimality, by 82.70%.

While the runtimes provided in Table 3.3 suggest that CPLEX can find z_2^* much faster than our decomposition approach for Instances 18, 19, and 20, these short CPLEX runtimes result from the fact that CPLEX cannot find z_1^* for these instances within the time limit but only a (weak) upper bound. As a result, for Instances 18, 19, and 20, CPLEX can solve Objective Function (3.1b) faster than our decomposition approach because only the upper bound is added as a constraint in the lexicographic approach, not z_1^*. Thus, these runtimes cannot be compared to those of our decomposition approach.

The number of demand decompositions found varies between 24 and $2,750$. In comparison, the number of demand decompositions examined in the bounding algorithm is significantly smaller. For ten instances, only two demand decompositions need to be examined in the bounding algorithm until it is proven that the optimal solution has been found. The maximum number of demand decompositions examined across all instances is eight, highlighting the efficiency of the bounding algorithm. The number of subproblems examined using CPLEX is also low, ranging between three and 12. The number of independent areas found is four for Instances 6 and 7, and three for all other instances.

The objective function values meet our expectations. For a given value of κ, z_1^* improves

when Δ is reduced from 10 m to 5 m or from 5 m to 2.5 m. This is inevitable, because instances with $\Delta = 2.5$ m include all lead-in lines and gates of instances with $\Delta = 5$ m, and instances with $\Delta = 5$ m include all lead-in lines and gates of instances with $\Delta = 10$ m. Instances with $\Delta = 7.5$ m do not fit into this scheme, which is why the z_1^* values of these instances can be and in fact are worse than those of the instances with $\Delta = 10$ m. Similarly, smaller values of κ generally yield better solutions with respect to Objective Function (3.1a) for a given value of Δ. Regarding Objective Function (3.1b), we do not see any systematic impact of the values of Δ and κ on z_2^*. Nevertheless, the range of solutions with values between 34 and 43 indicates that the choice of Δ and κ exerts considerable influence on z_2^*.

3.6.3 Computational performance for larger instances

In the following, we create larger instances with respect to three criteria. First, we generate an instance with one additional independent area, which increases the total parking space of the terminal by a third. To ensure that apron space remains scarce, we also increase the value of D_{ak} by 33% for all $a \in \mathcal{A}$ and $k \in \mathcal{K}$ (rounded to the next integer value). Second, we create an instance with an additional demand pattern $k = 6$, where $D_{16} = 21$, $D_{26} = 7$, and $D_{36} = 1$. Finally, we generate an instance where ADG IV aircraft also exist (with demand 3, 3, 3, 2, 2 for demand patterns $k \in \mathcal{K}$). All three instances are created based on medium-sized Instance 13 from Table 3.3 ($\Delta = 5$ m and $\kappa = 30°$). We solve all three instances with our decomposition approach and compare the performance to that in Instance 13 in Table 3.3. The results are provided in Table 3.4, where $\Delta^+_{\text{time},a}$ and $\Delta^+_{\text{time},b}$ denote the percentage runtime increase compared to the reference instance for Objective Functions (3.1a) and (3.1b), respectively.

Table 3.4: Performance for larger instances

Instance description	$\Delta^+_{\text{time},a}$ [%]	$\Delta^+_{\text{time},b}$ [%]	Info on DA				
			C	C_{con}	C_{exa}	SP_{exa}	$\bar{K}_{SP}$
Reference (Instance 13 from Table 3.3)	–	–	980	36	8	8	1.2
Additional area	270.59	15.42	81,312	420	8	12	1.2
Additional demand pattern	2.61	−7.03	5,880	22	2	4	1.5
Additional aircraft class	1,050.33	693.88	19,584	120	8	10	1.2

While we observe a significant increase in runtime when an independent area or aircraft class is added, the runtime does not change considerably when a demand pattern is

added. Furthermore, when the instance grows by an independent area or aircraft class, the increase in runtime is higher for Objective Function (3.1a) compared to Objective Function (3.1b). We observe that the number of demand decompositions grows exponentially, in particular when an independent area or aircraft class is added. However, the number of demand decompositions and subproblems examined in the bounding algorithm does not change drastically. When a demand pattern is added, the values of C_{exa} and SP_{exa} are even smaller compared to the reference instance, which explains the shorter runtime for Objective Function (3.1b). The high value of $\Delta^{+}_{\text{time},b}$ observed when adding an aircraft class can be explained by the fact that the subproblems themselves become more complex to solve. Due to our acceleration strategy to solve relaxations of the subproblems where demand patterns are ignored, this is not necessarily true when a demand pattern is added.

3.6.4 Effectiveness of acceleration techniques

We introduced multiple acceleration techniques in Section 3.5.5. In the following, we examine the impact of each of these acceleration techniques on the performance of our approach. For this purpose, we solved all soft brownfield instances from Table 3.3 multiple times, each time removing or exchanging exactly one of the acceleration techniques.

For each configuration of our approach with respect to acceleration techniques, Table 3.5 depicts the number of instances (out of 20) for which a feasible solution was found within the time limit (n_{fea}) and the number of instances that were solved to optimality within the time limit (n_{opt}). Furthermore, averaged over all instances counted in n_{opt}, Table 3.5 provides the percentage runtime increase per instance (Δ^{+}_{time}) compared to the runtimes obtained in Table 3.3, where all acceleration techniques and the top-down strategy were used. As the acceleration techniques have no impact on the part of our approach dedicated to finding the best solution for Objective Function (3.1a), we concentrate on the computation times for Objective Function (3.1b). If not stated differently, the top-down strategy was applied to sort demand patterns.

Removing any acceleration technique leads to a degradation of the performance of our algorithm. Most importantly, the average runtime per instance increases by more than 800% when subproblems are solved directly without relaxations, and fewer instances can be solved in that case. When the optimization process of the subproblems associated

Table 3.5: Performance results when individual acceleration techniques are deactivated

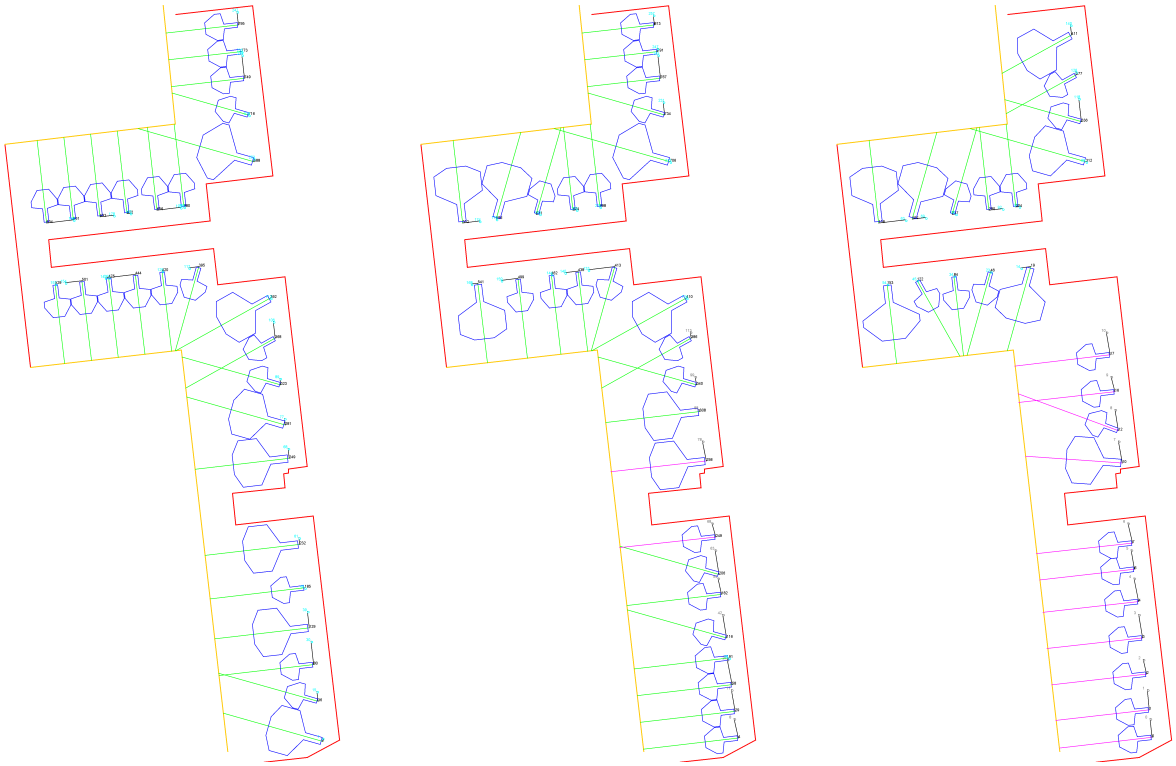
Description	n_{fea}	n_{opt}	Δ^{+}_{time} [%]
No solution pool	19	18	9.81
No relaxation of subproblems	15	14	859.21
Bottom-up strategy	19	18	58.17
No monitoring of UB	19	17	197.61
Consistency not considered	18	18	22.21

with demand decomposition $c \in \mathcal{C}$ is not terminated as soon as it becomes apparent that z^*_{2c} cannot be smaller than the current upper bound UB, we observe that the average runtime per instance increases by almost 200%. The remaining acceleration techniques also improve the performance of our algorithm, with average runtime reductions per instance and acceleration technique ranging between 10% and 60%.

3.6.5 Analysis of optimal layouts

We now address the resulting gate layout for Munich Airport Terminal 1. We have solved the greenfield, soft brownfield, and true brownfield instances for $\Delta = 5$ m and $\kappa = 22.5°$, and we compare the optimal layouts in the following. For each planning scenario, Figure 3.6 shows which gates and lead-in lines are used in the optimal solution, which lead-in lines are assigned to which gates, and the safety envelope of the aircraft parked at each lead-in line, indicating the class of each parked aircraft. The terminal facade and other airport structures are shown by red lines, and taxiways are illustrated by yellow lines. Existing gates are represented by gray circles in front of the terminal facade, whereas new gates are colored cyan. Similarly, existing lead-in lines are depicted in purple, while new lead-in lines are colored green. Black lines connect lead-in lines to the gates they are assigned to. Finally, aircraft safety envelopes are displayed as blue polygons. In order not to overload the figures, they show only lead-in lines and aircraft safety envelopes for demand pattern 5. Which lead-in lines are used for the remaining demand patterns is illustrated in Appendix B.8. Furthermore, for all planning scenarios, Table 3.6 provides the optimal values of Objective Functions (3.1a) and (3.1b), the optimal values of Objective Function

(3.1b) when already existing gates are not considered, and the number of gates used to park aircraft across all demand patterns.



(a) Greenfield (b) Soft brownfield (c) True brownfield

Figure 3.6: Optimal solutions for different planning scenarios, demand pattern 5

Compared to the true brownfield case, sets $\mathcal{G}$ and $\mathcal{L}$ contain significantly more gates and lead-in lines in the greenfield and soft brownfield scenarios, which increases planning flexibility. Consequently, the number of aircraft that cannot be parked at contact gates weighted by aircraft class and demand pattern (z_1^*) is substantially lower in the greenfield and soft brownfield scenarios than in the true brownfield scenario. However, there is no difference in that value between the greenfield and soft brownfield scenarios, suggesting that the existing gates and lead-in lines that distinguish the two scenarios from each other do not lead to an increase in parking capacity in our case.

Our results indicate that the substantial decrease in the number of aircraft that cannot be parked at contact gates weighted by aircraft class and demand pattern in the soft brownfield scenario compared to the true brownfield scenario does not lead to an equally

Table 3.6: Performance data of optimal layouts for different planning scenarios

Planning scenario	z_1^*	z_2^*	z_2^* without already existing gates	Number of gates used
Greenfield	3.2	36	36	27
Soft brownfield	3.2	37	23	28
True brownfield	6.2	35	21	25

significant increase in construction effort for gates. Instead, the increase in construction effort is comparatively small, also when taking the already existing gates into account. Thus, by allowing changes to the existing layout in the southern part of the terminal, a significant improvement in parking capacity can be achieved, while the actual changes to the existing gate layout remain relatively small.

Figure 3.6 shows that mainly orthogonal lead-in lines are used in the optimal layouts regardless of the scenario, without explicitly incentivizing the use of orthogonal lead-in lines in our model. Non-orthogonal lead-in lines are only used in three cases: (i) At corners of the terminal building where no orthogonal lead-in lines can be used, (ii) when specific geometric situations favor the use of non-orthogonal lead-in lines (for example, consider the situation at the southern facade of the new extension building in the true brownfield scenario), and (iii) when there is slack space that does not suffice to park another aircraft, even if all lead-in lines were orthogonal. However, in situations (i) and (ii), the use of non-orthogonal lead-in lines enables parking more aircraft simultaneously, so non-orthogonal lead-in lines should always be considered in the planning.

Finally, we find that the optimal layouts in Figure 3.6 and Appendix B.8 are quite distinct from each other. Nevertheless, the solutions for the greenfield and soft brownfield scenarios are very similar in terms of our objective functions. Therefore, we suspect that several solutions exist per scenario that are equivalent with respect to our objective functions. In that case, it might be useful to identify all these equivalent solutions in order to be able to provide the planner with several alternatives.

3.7 Conclusion

We introduced the Airport Gate Layout Problem (AGLP) and presented a mixed-integer model formulation for the problem that can be applied to both greenfield and brownfield instances. To solve the problem efficiently, we presented a decomposition framework, which features a custom bounding algorithm as well as problem-specific acceleration techniques. In our computational experiments, we demonstrated the superior performance of our approach compared to CPLEX and investigated the impact of each acceleration technique on runtimes. In addition, we explored how our approach performs when the number of independent areas, demand patterns, or aircraft classes is increased. Finally, we analyzed and compared particular layouts for Munich Airport Terminal 1 in different planning scenarios.

Solving the AGLP provides valuable decision support to planning experts. First, the optimal layout shows how to make the best use of the available space, especially for complex terminal geometries. Second, the optimal solution can be used as a benchmark for layout alternatives created by planning experts. Finally, when a new terminal is to be built and the layout of the building has not yet been determined, solving the AGLP can be used to evaluate different terminal layouts in terms of their potential to handle as many of the expected aircraft as possible at contact gates. Our decomposition approach considerably reduces the time needed to solve the AGLP for a given instance, which allows us to solve instances that are intractable for CPLEX. Thus, we can solve the AGLP with increased planning granularity for a particular terminal building, resulting in a better solution.

To conclude, we outline some possibilities for future research on the AGLP. First, we have not considered the affiliation of aircraft to airlines so far. Airlines that operate a base at an airport tend to have their own terminal or terminal areas where all flights of these airlines are handled. Of course, these terminal areas do not necessarily correspond to the terminal areas we define in our solution approach. However, different airlines are associated with different fleet compositions. Therefore, if the affiliation of aircraft to airlines is to be taken into account, the parking positions in each terminal area must reflect the fleet mix of the particular airline. Second, to reduce operational complexity on the apron, the AGLP may be extended by another objective function to minimize the number of lead-in lines, especially those that are non-orthogonal to the terminal facade, used across all demand

patterns. Third, it might be useful for decision makers to receive not only one solution, but several alternative layouts that are equivalent with respect to the given objectives and constraints, provided there is more than one optimal solution. While different layouts may be equally optimal from the perspective of our model, one alternative layout might be easier to realize than others, especially in brownfield situations.

Finally, future work should be concerned with further shortening the runtimes of the AGLP. In particular, pre-generation of parking patterns and demand decompositions is computationally expensive for instances with many independent areas, demand patterns, and/or aircraft classes. To address this problem, our approach could be supplemented with a column generation mechanism for stepwise generation of parking patterns and demand decompositions. Furthermore, approaches that reduce the runtime of the relaxed subproblems could be promising. Ideally, in the long run planners can be equipped with an interactive optimization tool in which the optimal layout can be manually adjusted and then re-optimized.

4 Dynamic Configuration of Aircraft Stands

We consider the configuration of airport gates with passenger boarding bridges. The set of aircraft types that can be serviced at a gate depends on the installed boarding bridge(s). For instance, the Airbus A380 can only be serviced at gates equipped with a passenger boarding bridge that is able to access its upper level. Given the dynamic development of both the number of aircraft movements and the fleet mix at airports, the recurring decision problem is to determine for each gate whether and when the passenger boarding bridge configuration should be changed. The objective is to minimize investment and operating costs associated with the bridges as well as penalty costs for aircraft which cannot be processed because gates that are equipped with adequate gate configurations are not available. We propose a mixed-integer model formulation for the Dynamic Gate Configuration Problem and present its underlying network structure. To solve the problem, we employ a column generation based heuristic approach. We demonstrate the good performance of the heuristic in a computational study and present a detailed discussion of the decisions taken as part of a case study.

4.1 Introduction

Airports operate in a highly competitive environment, and the efficient use of infrastructure is a key driver of success (e.g., Caves 1994, Dorndorf et al. 2007). This chapter focuses on the infrastructure provided at the gates, which represent the interface between the terminal complex and the apron and where the aircraft are accommodated during their service time on the ground. Analogous to Chapter 3, we will use the terms gate and

parking position synonymously. Passengers need to be conveyed safely, comfortably and efficiently from the terminal building to the parked aircraft and vice versa. To facilitate these deplaning and boarding operations, passenger boarding bridges (PBBs) are employed at the contact gates of most airports, especially large ones. For each gate, airports have to decide whether and, if so, how many PBBs of each type should be installed. We denote the resulting options - including the "no PBB" option - as PBB configurations that a contact gate can be equipped with. The decision of which gate should be equipped with which PBB configuration depends on the aircraft types to be processed at the gate in question, as the compatibility between aircraft types and PBB configurations is restricted by the various aircraft geometries (e.g., Kazda and Caves 2008). Thus, the decision of which PBB configuration to install at which gate clearly affects the gate capacity of an airport with respect to each aircraft type.

In addition to the compatibilities between aircraft types and PBB configurations, restrictions caused by the adjacent parking of aircraft have to be taken into account. As we already discussed in Section 3.2, parking an aircraft at a particular position might be temporarily prohibited due to potential wingtip collisions with other aircraft parked at adjacent parking positions at the same time (cf. Dorndorf et al. 2017). Also, Multi Aircraft Ramping Stands (MARS), where up to two aircraft can be accommodated simultaneously, must be considered. In the following, we will refer to a parking position that is configured as a MARS as a MARS gate. Due to their greater operational flexibility, MARS gates require special PBB configurations.

Decisions regarding the PBB configurations have to be considered in a dynamic context, as the total number of aircraft present at an airport evolves over time, as does the fleet mix. Changes in traffic volume and/or fleet mix can occur gradually or abruptly and may have either temporary or enduring effects. For instance, the average annual growth in global air traffic volume yields a gradual and enduring increase in traffic at many airports. One example of a gradual and permanent change in the fleet mix is the current trend towards employing smaller but more fuel efficient aircraft on long haul flights rather than large Boeing 747s or Airbus A380s (Boston Consulting Group 2020). It was in the course of this trend that both Airbus and Boeing recently announced that they would cease production of their largest aircraft models in the near future (Airbus 2019, Boeing 2020). An example of a sudden and permanent reduction in traffic volume at an airport is the opening of a new major airport in the vicinity. Finally, the recent global crisis

caused by the SARS-CoV-2 virus can be classified as an abrupt but temporary decline in air traffic volume. Adapting the gate capacity too early or too late to a fluctuating traffic volume and fleet structure inevitably leads to either unused PBBs or aircraft which cannot be processed at a contact gate due to capacity shortages. Consequently, airports not only have to decide which PBB configuration should be installed at which parking position but also when. These decisions can be supported by various long-term forecasts that are regularly undertaken by important actors in the aviation industry (refer to Airbus 2018 or Eurocontrol 2018, for instance).

Facing a highly uncertain future, airport managers would prefer the most flexible PBB configurations over others if no further aspects were taken into account. However, higher flexibility is associated with higher cost, and decisions need to be considered carefully, as the cost of a single PBB is approximately $450,000$€ (Airport Improvement Magazine 2010, Travel PR News 2019). For instance, as discussed in Section 3.6, Terminal 1 at Munich Airport is currently undergoing expansion. Unaware of the aforementioned development concerning the Airbus A380, decision makers at Munich Airport opted for two additional parking positions equipped with PBBs with sufficient flexibility for processing the A380 at the new terminal building (Munich Airport 2016). As the airport already possesses gates at which the A380 can be handled in sufficient number to serve pre-SARS-CoV-2 traffic, it is at least questionable from today's point of view whether the investment in additional PBBs will eventually pay off.

The overall question is when to equip which gate with which PBB configuration, balancing the operational flexibility and total cost with respect to the PBB configurations. We propose a model to formally describe this problem, which we call the Dynamic Gate Configuration Problem (DGCP). To increase the tractability of the problem, we apply Dantzig-Wolfe decomposition, which exploits the inherent network structure. Based on a column generation methodology, we propose a heuristic solution approach to the problem and show that it performs well for realistic data instances.

The remainder of this chapter is structured as follows: In Section 4.2, we discuss the related literature. Section 4.3 introduces the DGCP in detail. A mixed-integer model is presented in Section 4.4, while Section 4.5 describes our proposed solution approach. The computational study presented in Section 4.6 investigates the performance of the proposed procedure. In Section 4.7, we present a case study for Munich Airport Terminal

2, based on real data. Finally, conclusions and recommendations for future research are presented in Section 4.8.

4.2 Related literature

The work presented in this chapter is positioned at the interface between the strategic capacity planning of aprons and the gate assignment problem (i.e., the optimal assignment of aircraft to gates). We will briefly discuss the existing literature from both fields of research and state the similarities and differences of both streams compared to our approach.

Strategic capacity planning of aprons The strategic management of apron capacity has been addressed from different perspectives. An elaborate review of the relevant literature is provided by Mirković and Tošić (2014). Simple formulas for calculating the capacity of a given apron can be found in Ashford et al. (2011), Horonjeff et al. (2010), and de Neufville and Odoni (2013). These formulae either assume that each gate can accommodate all aircraft types or they take existing compatibility restrictions into account and calculate apron capacities for different types of aircraft. More elaborate methods provide advice on the number of gates an apron should have to be able to fulfill certain requirements, mostly under stochastic conditions (see Bandara and Wirasinghe 1989, Wirasinghe and Bandara 1990, Steuart 1974, Hassounah and Steuart 1993, Narciso and Piera 2015). Furthermore, comparing hub airports to non-hub airports, Mirković and Tošić (2017) demonstrate how gate capacity depends on the traffic characteristics of the airport.

This chapter relates to the strategic planning of apron capacity, as the choice of PBB configurations at gates determines how many aircraft of each type can be served simultaneously. However, while the above literature provides formulae with which to perform a one-time calculation of apron capacity for the number of gates available, the objective of this work is to dynamically reconfigure the gates so as to enable the most cost-efficient use of the existing infrastructure in the long run[1]. We therefore investgate not only how much

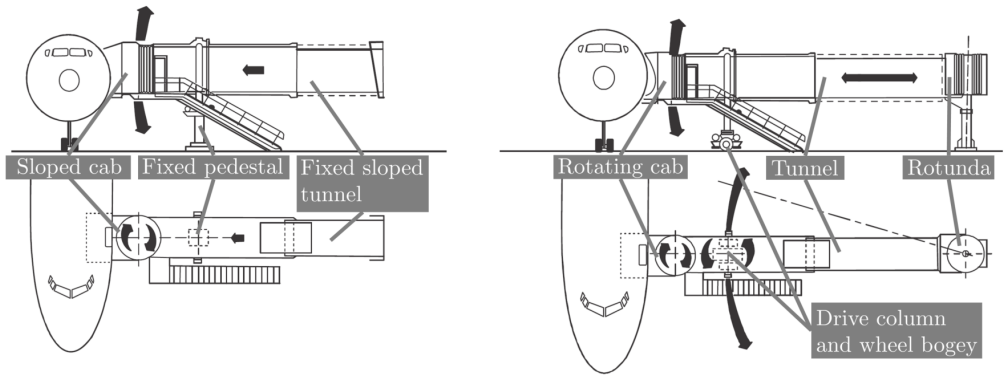
[1]This follows the recommendations by Caves 1994 and Narciso and Piera 2015 to improve the efficiency of existing apron space instead of investing in new infrastructure.

capacity of which type needs to be provided, but also when it is needed. Furthermore, existing work determines the number of aircraft that a given apron can process per time unit considering a continuous time horizon of operational length. In contrast, we aim to dynamically adjust the number of aircraft which can be processed simultaneously (i.e., at a given moment in time) over a strategic time horizon. Finally, contrasting the existing literature, we consider such operational aspects as MARS gates and adjacency restrictions in our model.

Gate assignment problem The second stream of research that our problem relates to is the gate assignment problem, where arriving aircraft need to be assigned to available, compatible gates. Extensive reviews of the literature considering this problem are provided by Dorndorf et al. (2007), Cheng et al. (2012), Guépet et al. (2015), and recently by Daş et al. (2020). The main similarity of our approach to existing literature that discusses the gate assignment problem is that we assign aircraft to gates subject to capacity and compatibility constraints, while additionally considering aircraft parked at both adjacent gates and MARS gates. However, rather than creating a schedule for the operative level of decision making, where individual aircraft are assigned to gates, we determine on a strategic level which type of aircraft each gate accommodates to enable long-term adaption of PBB configurations and in turn efficiently service the aggregate traffic. Our objective is thus to minimize the number of aircraft that cannot be accommodated at a contact gate due to gate unavailability or incompatibility. In contrast, existing literature on the gate assignment problem considers such objectives as the minimization of the total walking distance for passengers and the minimization of the number of required towing procedures or else attempts to find robust schedules that remain stable in the face of uncertainty. Daş et al. (2020) find that more recent approaches tend to incorporate multiple objectives. While the gate assignment problem itself has been proven to be NP-hard (Guépet et al. 2015), we employ a basic version of it embedded into our strategic decision problem.

4.3 Problem description

We consider the PBBs at the contact gates of an airport over a strategic time horizon of up to 10 years. Time is discretized into periods with a length of half a year each, as this interval matches the seasons over which flight plans generally remain unchanged and thus the capacity requirements posed to airports remain stable. The objective is to strike a balance between PBB costs and the probability of not being able to service aircraft at contact gates. For each gate and time period it needs to be decided whether the PBB configuration from the previous time period should be retained or exchanged. Selecting a PBB configuration for a gate implies deciding on (a) the type(s) and (b) the number of bridges of the selected type(s) that are installed at the gate. Regarding (a), PBBs can be grouped into stationary and apron-drive bridges, see Figure 4.1. In contrast to stationary PBBs, apron-drive bridges are equipped with a rotatable connection to the terminal building and a drive column with wheel bogey. Consequently, they exhibit a higher operational flexibility but with the downside of higher investment and operational costs due to their greater technical complexity.



(a) Stationary loading bridge (b) Apron-drive loading bridge

Figure 4.1: Stationary and apron drive loading bridges (source: International Civil Aviation Organization 2005, edited according to National Academies of Sciences, Engineering, and Medicine 2010).

With respect to (b), either a single PBB or two PBBs of the same type are usually installed per gate. At gates that accommodate very large aircraft as the Airbus A380, a third PBB is employed in order to expedite the boarding and deplaning operations. The

aggregate of all PBB configurations represents all possible states of a gate with respect to PBBs, including the no-PBB case. The number and the positions of contact gates as well as the initial PBB configuration of each gate are predefined, enabling both greenfield and brownfield scenarios to be modeled.

In addition to the diverse PBB configurations, a multitude of commercial aircraft types exists, each with different specifications. As indicated by Kazda and Caves (2008), the variation in doorsill height across aircraft of different types causes incompatibilities between PBB configurations and aircraft types (also see National Academies of Sciences, Engineering, and Medicine 2010). In addition to these inevitable incompatibilities owing to aircraft and PBB geometries, it is not practical to serve certain types of aircraft with certain PBB configurations. For instance, there might be a PBB configuration that is defined as a single but flexible apron-drive PBB. Although this PBB might reach the doorsill of all aircraft on the market, boarding and deplaning operations would require an unacceptable duration for large types of aircraft. Consequently, the compatibility between PBB configurations and aircraft types is de facto restricted beyond the physical limitations, such that accommodation is physically possible *and* operations can be conducted with adequate efficiency. Due to the large variety of aircraft types and to account for aircraft that will be introduced to the market within the time horizon, we classify aircraft by wingspan similar to Section 3.2. Consequently, we define the compatibilities between PBB configurations and aircraft classes rather than individual aircraft types. If a PBB configuration is compatible with a particular aircraft class, all aircraft of that class can be accommodated at a parking position equipped with that PBB configuration.

As we consider the problem in a dynamic context, we rely on forecasts that estimate the demand for accommodation for each aircraft class and time period in the time horizon. The decisions are taken at the beginning of each time period and configuration changes are realized instantly, such that any new configurations can be utilized in the respective time period. The following three cost types are relevant to the DGCP: First, a change in PBB configuration causes investment cost for dismantling and disposing the previous PBB configuration, gate unavailability during reconfiguration and the purchase and setup of the new PBB configuration. The actual amount depends on the PBB configurations before and after the change. For each period, we consider a constrained budget available to change PBB configurations. Second, installed PBB configurations cause operating costs over the course of the respective time period. These include the maintenance costs

of the installed equipment (including vertical and horizontal drive motors, control systems and air-conditioning units) and thus depend on the PBB configuration. Third, in each time period, penalty costs are incurred for each aircraft that cannot be accommodated at one of the contact gates. Aircraft that cannot be processed at a contact gate need to be parked at a remote parking position and passengers then have to be conveyed from and to the aircraft by bus. On the one hand, this has a direct financial impact on the airport, as airlines pay fees for the use of PBBs at contact gates (for instance, see Aena 2020), which are not incurred for aircraft processed at remote parking positions. On the other hand, this diminishes the operational efficiency at the airport, as it leads to longer connecting times for transfer passengers. Longer connecting times reduce the hub potential of the affected airport as fewer connections can be offered within a given amount of time. The penalty costs include the resulting costs accruing due to lost income, deteriorated efficiency and lost connections, and the amount depends on the class of the relocated aircraft. The objective is to minimize the total cost, i.e., the sum of the investment, operating and penalty costs.

Demand patterns Airlines and airports generally issue two flight plans per year, one for the summer and one for the winter season. As we have chosen a time period length of half a year, the length of a season matches the length of one time period, and each season is represented by one time period. Within each season, the flight schedules generally repeat in a weekly rhythm. Therefore, the flight traffic for each season can be represented by the flights observed in one representative week of that season. In the course of a week, the number of aircraft processed simultaneously is not constant but varies over time. Figure 4.7 in Section 4.7 shows an illustrative example of the number of aircraft processed simultaneously at Munich Airport Terminal 2 for a typical week.

Analogous to Section 3.3, we only consider those situations where traffic reaches its peaks, and we represent each traffic peak by one demand pattern in our model. Each demand pattern defines the number of aircraft per class that must be processed simultaneously. For example, a demand pattern $(3, 4, 5)$ states that 3, 4, and 5 aircraft of classes 1, 2, and 3, respectively, need to be processed in parallel. Furthermore, each demand pattern is assigned a weighting factor for each time period based on the relative frequency of the pattern in that time period. For example, if there are three different demand patterns that occur in the representative week of a period with absolute frequencies of two, three,

and one, respectively, the weights of the demand patterns in that period are 0.33, 0.50, and 0.17, respectively.

The use of demand patterns has three major benefits. First, the amount of demand data is drastically reduced, while information relevant for demand fulfillment is retained. If the demand can be covered at all traffic peaks, a capacity shortage is unlikely to occur at other moments in time due to unavailable or incompatible PBBs. Second, estimating the number of aircraft and the fleet mix to be expected at traffic peaks is easier for airports to implement than forecasting the complete flight plan over the entire time horizon. Third, to account for forecasting uncertainty, several demand patterns reflecting different possible future scenarios of one traffic peak can be created without any further changes being made to the concept.

Neighborhood restrictions The accommodation of an aircraft at a gate might temporarily be prohibited to prevent collisions with other aircraft simultaneously parked at adjacent parking positions (Dorndorf et al. 2017). This is relevant to our decision problem, as it induces restrictions to reasonable PBB configurations at the parking positions depending on the PBB configurations at the adjacent gates. For instance, there is no point in installing PBB configurations for large aircraft types at two adjacent gates if only one of the gates can serve one large aircraft at a time due to spatial restrictions. We consider a nose-in parking orientation of aircraft accommodated at contact gates as common at most airports (for further reference, see Horonjeff et al. 2010 and Kazda and Caves 2008). Gates comprise lead-in lines, which guide aircraft from the taxiway on the apron to their final parking positions. The fuselages of parked aircraft are aligned with the lead-in lines. To accommodate aircraft of different types and increase operational flexibility, a single gate may have several lead-in lines. However, in contrast to Chapter 3, we now assume that all lead-in lines are perpendicular to the terminal facade and that adjacent lead-in lines are parallel to each other.

In order to prevent collisions of aircraft processed simultaneously at adjacent gates, the sum of the half wingspans of aircraft being processed on any combination of lead-in lines belonging to the gates must not exceed the lateral distance between the lead-in lines the aircraft are parked on. Consider the situation at the adjacent gates depicted in Figure 4.2, for example, where both gates are equipped with two lead-in lines. As we have classified

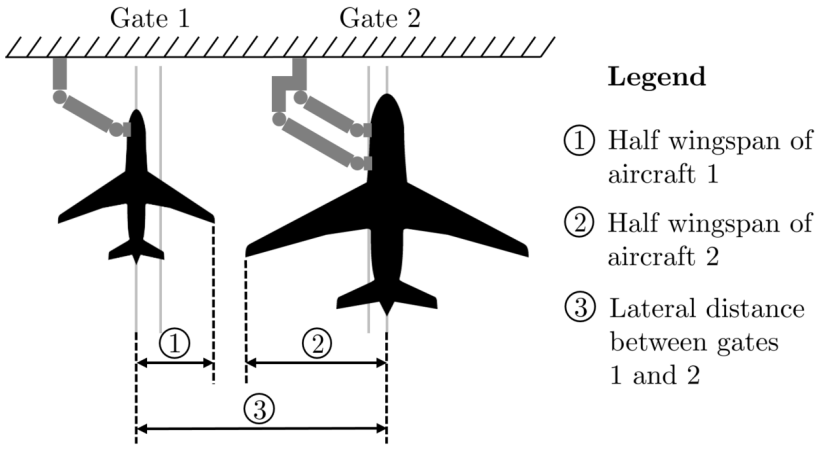


Figure 4.2: Lateral distance between adjacent gates and half wingspans

aircraft by wingspan, the wingspan of an aircraft is equal to the maximum wingspan of its aircraft class.

Multi Aircraft Ramping Stands In order to enhance spatial efficiency, MARS gates have been introduced at many airports (see National Academies of Sciences, Engineering, and Medicine 2010). Up to two aircraft can be accommodated simultaneously at a MARS gate, depending on the size of the aircraft. To be able to process two smaller aircraft simultaneously, MARS gates must be equipped with two PBBs at least. Furthermore, a parking position must be equipped with at least three lead-in lines in order to be used as a MARS gate. These include a central lead-in line for processing a single aircraft and two outer lead-in lines to enable two aircraft to be accommodated simultaneously. This is different compared to Chapter 3, where we made no assumptions in this regard and MARS gates could also be equipped with two lead-in lines only. Figure 4.3 presents the possible operating modes of a MARS gate and illustrates the three parallel lead-in lines described.

For each PBB configuration, we define whether it can be employed at a MARS gate. While normally only one aircraft can be processed at a time at a gate, up to two aircraft can be handled simultaneously at gates where a PBB configuration is installed that can be used at MARS gates. Furthermore, the previous considerations of neighborhood restrictions need to be adjusted to prevent not only collisions between two aircraft parked at adjacent

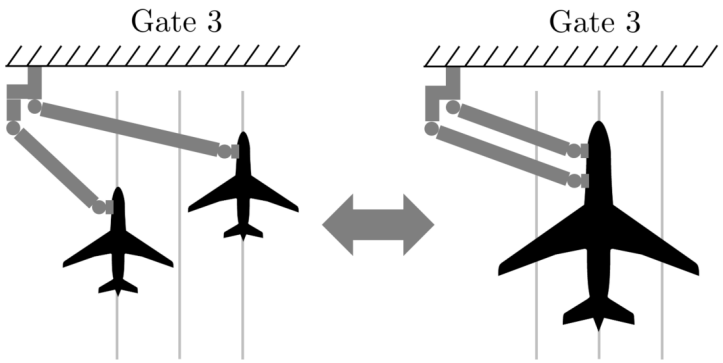


Figure 4.3: Modes of operation and lead-in lines of MARS gates

gates simultaneously, but also to prevent collisions between two aircraft parked at the same MARS gate simultaneously.

4.4 Model

In this section, we present a mathematical model for the DGCP. A summary of the notation is given in Table C.1 of Appendix C.1. As previously stated, the planning goal is to determine which PBB configuration should be installed at each gate, and when, in order to satisfy dynamic demands with minimum total cost. Planning is based on a discrete horizon $\mathcal{T} = \{0, \ldots, T\}$ with T time periods of equal length. The sets of gates and lead-in lines are denoted as $\mathcal{G} = \{0, \ldots, G\}$ and $\mathcal{J} = \{0, \ldots, J\}$, respectively. For each gate, disjoint sets $\mathcal{J}_g \subseteq \mathcal{J}$ contain the lead-in lines associated with gates $g \in \mathcal{G}$. At the beginning of each time period $t = \{0, \ldots, T-1\}$ it needs to be decided for each gate which PBB configuration of set $\mathcal{L} = \{0, \ldots, L\}$ should be installed, where $l = 0$ represents a dummy PBB configuration in which no PBB is installed at the respective gate. The set $\mathcal{L}^{MARS} \subseteq \mathcal{L}$ contains all PBB configurations which indicate that the respective gate is equipped as a MARS gate.

Time-space network We model the DGCP as a time space network, in which each gate is represented by one subnetwork. The example in Figure 4.4 shows two gates, T periods and three PBB configurations.

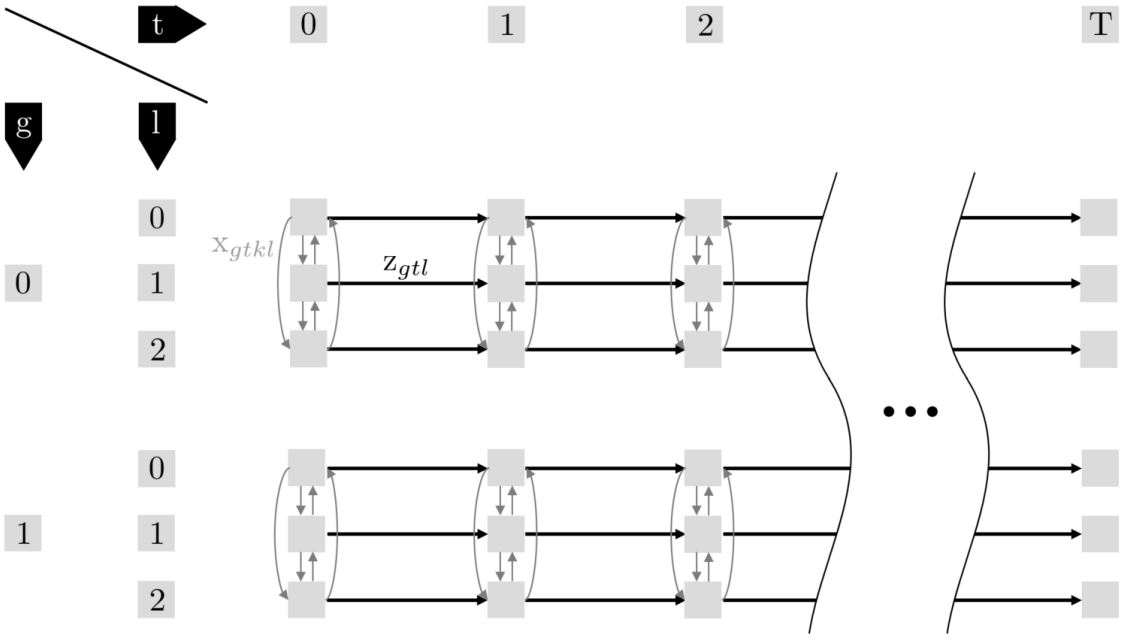


Figure 4.4: Time-space network

The horizontal axis of each subnetwork constitutes the planning horizon, and the vertical axis the PBB configurations. Each node depicts a possible state of gate $g \in \mathcal{G}$ with respect to PBB configuration $l \in \mathcal{L}$ in time period $t \in \mathcal{T}$. The arcs represent feasible transitions from one node to another. The flow through arcs which connect nodes of different PBB configurations $k, l \in \mathcal{L}$ at the beginning of each period $t \in \mathcal{T}$ is determined by binary decision variables $x_{gtkl} \in \{0, 1\}$. Binary decision variables $z_{gtl} \in \{0, 1\}$ determine the flow through arcs connecting nodes that depict PBB configuration $l \in \mathcal{L}$ during period $t \in \mathcal{T}$. Variable x_{gtkl} is 1 if the construction of PBB configuration $l \in \mathcal{L}$ at gate $g \in \mathcal{G}$ is performed at the beginning of time period $t = 0, \ldots, T-1$ to substitute PBB configuration $k \in \mathcal{L} \setminus \{l\}$, and 0 otherwise. For the PBB configuration $l \in \mathcal{L}$ which is employed at gate $g \in \mathcal{G}$ during time period $t \in \mathcal{T}$, the respective variable z_{gtl} equals 1, and 0 otherwise. At each node, a parameter e_{gtl} gives the difference between the sum of all inflows and the sum of all outflows. At the source node, i.e., the node at $t = 0$ representing the initial PBB configuration of the respective gate, inflows are 0 and outflows are 1 ($e_{gtl} = 1$). Furthermore, at each node in $t = 0, \ldots, T-1$ which is neither a source nor a sink of the network, the sum of all inflows must be equal to the sum of all outflows ($e_{gtl} = 0$) in order

to ensure that the balance of flow is maintained.

Aircraft types and demand patterns Aircraft are classified by wingspan, as stated in Section 4.3, and the resulting set of aircraft classes is denoted as $\mathcal{A} = \{1, \ldots, A\}$. For each aircraft class $a \in \mathcal{A}$, the set of PBB configurations that can be employed to accommodate aircraft of that class is introduced as $\mathcal{L}_a \subseteq \mathcal{L}$.

Future traffic volume is estimated by means of demand patterns as stated in Section 4.3. The demand patterns are enumerated in the set $\mathcal{B} = \{1, \ldots, B\}$. Parameter D_{ba} gives the number of aircraft of class $a \in \mathcal{A}$ in demand pattern $b \in \mathcal{B}$. Furthermore, for each time period $t \in \mathcal{T}$, the set $\mathcal{B}_t \subseteq \mathcal{B}$ contains the demand patterns of time period t. Weight f_{tb} gives the relative frequency of demand pattern $b \in \mathcal{B}_t$ in time period $t \in \mathcal{T}$. As we assume that demand patterns and weights for the time horizon are given at the beginning of the decision process, our model is deterministic.

The degree to which demand (as represented by the demand patterns) can be fulfilled depends on the PBB configurations of the gates. Binary decision variable y_{gjtba} states if for demand pattern $b \in \mathcal{B}_t$ an aircraft of class $a \in \mathcal{A}$ is assigned to gate $g \in \mathcal{G}$ in time period $t \in \mathcal{T}$ using lead-in line $j \in \mathcal{J}_g$. Decision variables y^-_{tba} count the number of ungated aircraft. If for any demand pattern $b \in \mathcal{B}_t$ in time period $t \in \mathcal{T}$ an aircraft of class $a \in \mathcal{A}$ cannot be accommodated at any of the contact gates, y^-_{tba} is increased by one.

Aircraft at adjacent gates For each parking position $g \in \mathcal{G}$, set $\mathcal{N}_g \subseteq \mathcal{G}$ is introduced, containing gate g and its right side neighbor gate $g+1$[2]. Furthermore, the lateral distance between the lead-in lines $i \in \mathcal{J}$ and $j \in \mathcal{J}$ is denoted by the parameter d_{ij}. We define the parameter s_a as the lateral distance from the center of the fuselage to one of the wingtips (i.e. half the wingspan) of the aircraft with the largest wingspan in aircraft class $a \in \mathcal{A}$ plus an additional minimum separation distance, which must be adhered to. In the model, constraints will assure for each gate $g \in \mathcal{G}$ that for no combination of lead-in lines $i \in \mathcal{J}_h$ and $j \in \mathcal{J}_h$ ($i \neq j$ and $h \in \mathcal{N}_g$) does the sum of the adjusted aircraft wingspans s_a of the aircraft parked simultaneously at lead-in lines i and j exceed the value of d_{ij}. As lead-in lines i and j might be associated with the same gate, these constraints also

[2]The left side neighbor gate $g-1$ does not need to be considered as the combination of gates $g-1$ and g is already considered in the set $\mathcal{N}_{g-1}$.

cover the necessary neighborhood relations between two aircraft parked simultaneously at a MARS gate.

Costs and objective The objective of minimizing the total cost implies minimizing the sum of three cost types. First, investment costs c^{i}_{kl} represent the costs of changing the PBB configuration at any gate for each combination $(k, l \in \mathcal{L})$ of non-identical PBB configurations. The total investment cost incurred in period $t \in \mathcal{T}$ is limited by the available budget r_t. Second, operating costs c^{o}_l measure the costs per period for maintenance and of accommodating aircraft at an aircraft stand where PBB configuration $l \in \mathcal{L}$ is installed. Third, penalty costs c^- are incurred for each passenger who leaves or boards an aircraft parked at a remote position. In the objective function, c^- is multiplied by the parameter p_a, which gives the average number of passengers on board an aircraft of class $a \in \mathcal{A}$. Thus, the penalty cost incurred depends on the class of the respective aircraft which cannot be assigned to a contact gate.

Extensive formulation We now present the extensive formulation of the DGCP, which we denote as Original Problem OP.

$$\begin{aligned}\min\ z_{OP} \coloneqq & \sum_{g\in\mathcal{G}}\sum_{l\in\mathcal{L}}\sum_{k\in\mathcal{L}\setminus\{l\}}\sum_{t=0}^{T-1} c^{\mathrm{i}}_{kl} x_{gtkl} \\ & + \sum_{g\in\mathcal{G}}\sum_{t=0}^{T-1}\sum_{l\in\mathcal{L}} c^{\mathrm{o}}_l z_{gtl} \\ & + c^- \sum_{a\in\mathcal{A}} p_a \sum_{t=0}^{T-1}\sum_{b\in\mathcal{B}_t} f_{tb} y^-_{tba}\end{aligned} \tag{4.1a}$$

subject to

$$\sum_{k\in\mathcal{L}\setminus\{l\}} x_{gtlk} + z_{gtl} \qquad \forall g \in \mathcal{G}, t = 0, \ldots, T-1, \tag{4.1b}$$

$$- \sum_{k\in\mathcal{L}\setminus\{l\}} x_{gtkl} - \sum_{\substack{t'=t:\\ t'>0}} z_{g,t'-1,l} = e_{gtl} \qquad l \in \mathcal{L}$$

$$\sum_{g\in\mathcal{G}}\sum_{j\in\mathcal{J}_g} y_{gjtba} + y^-_{tba} \geq D_{ba} \qquad \forall t = 0,\dots,T-1, \quad b \in \mathcal{B}_t, a \in \mathcal{A} \tag{4.1c}$$

$$\sum_{g\in\mathcal{G}}\sum_{l\in\mathcal{L}}\sum_{k\in\mathcal{L}\setminus\{l\}} c^{\mathrm{i}}_{kl} x_{gtkl} \leq r_t \qquad \forall t = 0,\dots,T-1 \tag{4.1d}$$

$$\sum_{a\in\mathcal{A}} y_{gjtba} \leq \sum_{l\in\mathcal{L}\setminus\{0\}} z_{gtl} \qquad \forall g \in \mathcal{G}, j \in \mathcal{J}_g, \quad t = 0,\dots,T-1, b \in \mathcal{B}_t \tag{4.1e}$$

$$\sum_{a\in\mathcal{A}}\sum_{j\in\mathcal{J}_g} y_{gjtba} \leq 2 - \sum_{l\in\mathcal{L}\setminus\mathcal{L}^{MARS}} z_{gtl} \qquad \forall g \in \mathcal{G}, t = 0,\dots,T-1, \quad b \in \mathcal{B}_t \tag{4.1f}$$

$$y_{gjtba} \leq \sum_{l\in\mathcal{L}_a} z_{gtl} \qquad \forall g \in \mathcal{G}, j \in \mathcal{J}_g, t = 0,\dots,T-1, \quad b \in \mathcal{B}_t, a \in \mathcal{A} \tag{4.1g}$$

$$\sum_{a\in\mathcal{A}} y_{hitba} s_a + \sum_{a\in\mathcal{A}} y_{hjtba} s_a \leq d_{ij} + M \cdot \left(2 - \sum_{a\in\mathcal{A}} y_{hitba} - \sum_{a\in\mathcal{A}} y_{hjtba}\right) \qquad \forall g \in \mathcal{G}, i,j \in \mathcal{J}_h : h \in \mathcal{N}_g, j > i, \quad t = 0,\dots,T-1, b \in \mathcal{B}_t \tag{4.1h}$$

$$x_{gtkl} \in \{0,1\} \qquad \forall g \in \mathcal{G}, k \in \mathcal{L}, l \in \mathcal{L}\setminus\{k\}, \quad t = 0,\dots,T-1 \tag{4.1i}$$

$$z_{gtl} \in \{0,1\} \qquad \forall g \in \mathcal{G}, t = 0,\dots,T-1, \quad l \in \mathcal{L} \tag{4.1j}$$

$$y^-_{tba} \geq 0 \qquad \forall t = 0,\dots,T-1, \quad b \in \mathcal{B}_t, a \in \mathcal{A} \tag{4.1k}$$

$$y_{gjtba} \in \{0,1\} \qquad \forall g \in \mathcal{G}, j \in \mathcal{J}_g, t = 0,\dots,T-1, \quad b \in \mathcal{B}_t, a \in \mathcal{A} \tag{4.1l}$$

Objective Function (4.1a) minimizes the sum of the investment, operating and penalty costs. Constraints (4.1b) are the flow balance equations and Constraints (4.1c) require that demands are satisfied, otherwise a penalty is incurred. Budget Constraints (4.1d) make sure that the budget available for PBB configuration changes is not exceeded in any

time period. Capacity Constraints (4.1e) ensure that no aircraft can be assigned to a lead-in line which belongs to a gate that does not have any PBB installed. Additional Capacity Constraints (4.1f) ensure that at gates which are not equipped with a PBB configuration $l \in \mathcal{L}^{MARS}$, no more than one aircraft can be serviced simultaneously, while at gates where a PBB configuration $l \in \mathcal{L}^{MARS}$ is installed, at most two aircraft can be accommodated at a time. Constraints (4.1g) ensure that only aircraft stands with compatible PBB configurations are used to serve the demands of each aircraft class. Neighborhood Constraints (4.1h) ensure that restrictions on aircraft parking simultaneously at adjacent gates and MARS gates are adhered to. We set the value of the Big-M parameter used in these constraints to $\max_{a \in \mathcal{A}} s_a$. Finally, Constraints (4.1i) – (4.1l) define the variable domains.

4.5 Solution methodology

In the computational study that follows in Section 4.6, we will show that OP (4.1a)-(4.1l) cannot be solved within a reasonable amount of time for data instances of realistic size. When ignoring the gate assignment variables y^{-}_{tba} and y_{gjtba}, the penalty costs in Objective Function (4.1a) and related Constraints (4.1c)-(4.1h), OP can be reduced to a multi-commodity flow problem, where for each gate $g \in \mathcal{G}$ a shortest path problem has to be solved (see Figure 4.4). Due to this natural decomposition of the problem, we suggest to reformulate OP by applying Dantzig-Wolfe decomposition and to solve the reformulated model using a column generation (CG) heuristic. Desrosiers and Lübbecke (2005) show that the extreme points of the polyhedron defined by a shortest path problem correspond to paths through the flow network. In our case, the flow network is the one presented in Figure 4.4 and a path is defined as a sequence of nodes leading from the source to the sink of the graph associated with one gate. A path thus defines the PBB configuration of one gate for each time period within the time horizon. In addition to the notation presented in Table C.1, the terminology provided in Table C.2 will be used in the following. All paths (of all gates) are contained in the set $\mathcal{P} = \{1, \ldots, P\}$, and for each gate $g \in \mathcal{G}$, a set $\mathcal{P}_g \subseteq \mathcal{P}$ contains all paths associated with the respective gate. Furthermore, each path is associated with path cost c_p, which equals the sum of investment and operating costs accumulated through the respective path. Instead of deciding for each gate and time period whether the current PBB configuration should be

retained or replaced by a different one by setting binary decision variables x_{gtkl} and z_{gtl}, the decision now is whether or not to select path $p \in \mathcal{P}_g$ for gate $g \in \mathcal{G}$, represented by binary decision variables λ_p. Binary parameters z^p_{gtl} indicate whether or not gate $g \in \mathcal{G}$ is equipped with PBB configuration $l \in \mathcal{L}$ in time period $t \in \mathcal{T}$ and path $p \in \mathcal{P}_g$. Similarly, binary parameters x^p_{gtkl} indicate whether or not the PBB configuration $l \in \mathcal{L}$ is replaced by PBB configuration $k \in \mathcal{L}$ at gate $g \in \mathcal{G}$ in time period $t \in \mathcal{T}$.

Master Problem Given our definition of a path and the presented notation, reformulation of OP yields the following Master Problem MP:

$$\min \; z_{MP} \coloneqq \sum_{p \in \mathcal{P}} c_p \lambda_p + c^- \sum_{a \in \mathcal{A}} p_a \sum_{t=0}^{T-1} \sum_{b \in \mathcal{B}_t} f_{tb} y^-_{tba} \tag{4.2a}$$

subject to

$$\sum_{g \in \mathcal{G}} \sum_{j \in \mathcal{J}_g} y_{gjtba} + y^-_{tba} \geq D_{ba} \qquad \forall t = 0, \ldots, T-1, \; b \in \mathcal{B}_t, a \in \mathcal{A} \tag{4.2b}$$

$$\sum_{g \in \mathcal{G}} \sum_{p \in \mathcal{P}_g} \lambda_p \left(\sum_{l \in \mathcal{L}} \sum_{k \in \mathcal{L} \setminus \{l\}} c^{\mathrm{i}}_{kl} x_{gtkl} \right) \leq r_t \qquad \forall t = 0, \ldots, T-1 \tag{4.2c}$$

$$\sum_{a \in \mathcal{A}} y_{gjtba} \leq \sum_{p \in \mathcal{P}_g} \lambda_p \left(\sum_{l \in \mathcal{L} \setminus \{0\}} z^p_{gtl} \right) \qquad \forall g \in \mathcal{G}, j \in \mathcal{J}_g, \; t = 0, \ldots, T-1, b \in \mathcal{B}_t \tag{4.2d}$$

$$\sum_{a \in \mathcal{A}} \sum_{j \in \mathcal{J}_g} y_{gjtba} \leq 2 - \sum_{p \in \mathcal{P}_g} \lambda_p \left(\sum_{l \in \mathcal{L} \setminus \mathcal{L}^{MARS}} z^p_{gtl} \right) \qquad \forall g \in \mathcal{G}, t = 0, \ldots, T-1, \; b \in \mathcal{B}_t \tag{4.2e}$$

$$y_{gjtba} \leq \sum_{p \in \mathcal{P}_g} \lambda_p \left(\sum_{l \in \mathcal{L}_a} z^p_{gtl} \right) \qquad \forall g \in \mathcal{G}, j \in \mathcal{J}_g, t = 0, \ldots, T-1, \quad (4.2f)$$
$$b \in \mathcal{B}_t, a \in \mathcal{A}$$

$$\sum_{a \in \mathcal{A}} y_{hitba} s_a + \sum_{a \in \mathcal{A}} y_{hjtba} s_a \leq \qquad \forall g \in \mathcal{G}, i, j \in \mathcal{J}_h : h \in \mathcal{N}_g, j > i, \quad (4.2g)$$
$$d_{ij} + M \cdot \left(2 - \sum_{a \in \mathcal{A}} y_{hitba} - \sum_{a \in \mathcal{A}} y_{hjtba} \right) \qquad t = 0, \ldots, T-1, b \in \mathcal{B}_t$$

$$\sum_{p \in \mathcal{P}_g} \lambda_p = 1 \qquad \forall g \in \mathcal{G} \quad (4.2h)$$

$$y^-_{tba} \geq 0 \qquad \forall t = 0, \ldots, T-1, b \in \mathcal{B}_t, a \in \mathcal{A} \quad (4.2i)$$

$$y_{gjtba} \in \{0, 1\} \qquad \forall g \in \mathcal{G}, j \in \mathcal{J}_g, t = 0, \ldots, T-1, \quad (4.2j)$$
$$b \in \mathcal{B}_t, a \in \mathcal{A}$$

$$\lambda_p \in \{0, 1\} \qquad \forall p \in \mathcal{P} \quad (4.2k)$$

Note that the flow balance constraint is incorporated in the definition of a path and is therefore not represented in MP. The same holds for Constraints (4.1i) and (4.1j), because decision variables x_{gtkl} and z_{gtl} have been replaced and x^p_{gtkl} and z^p_{gtl} are now parameters. Constraints (4.1d), (4.1e), (4.1f) and (4.1g) are modified to (4.2c), (4.2d), (4.2e) and (4.2f) according to the altered notation, and Convexity Constraints (4.2h) need to be added as one path per gate needs to be selected.

Column generation For practical data instances, $\mathcal{P}$ will be prohibitively large. Explicitly enumerating and evaluating all possible paths in MP therefore becomes intractable. Hence, we derive the Restricted Master Problem RMP from MP by substituting $\mathcal{P}$ with the subset of paths $\mathcal{P}' \subseteq \mathcal{P}$. In order to apply the CG approach, we relax the integrality requirements on decision variables λ_p and y_{gjtba} and denote the resulting, relaxed version of RMP as $\widetilde{\text{RMP}}$. Similarly, the version of MP where integrality requirements on decision variables λ_p and y_{gjtba} are relaxed is referred to as $\widetilde{\text{MP}}$. After the initialization of $\widetilde{\text{RMP}}$ with a dummy path for each aircraft stand $g \in \mathcal{G}$, the CG procedure is commenced by solving $\widetilde{\text{RMP}}$ to optimality. Then, the dual variable values of Constraints (4.2c), (4.2d), (4.2e), (4.2f) and (4.2h) are used to determine whether for any aircraft stand a new path can be found with negative reduced cost. This is done by solving G subproblems - one

for each gate $g \in \mathcal{G}$. If for any gate(s) a path with negative reduced cost can be found, the latter is/are added to $\widetilde{\text{RMP}}$ and the process is repeated until no path with negative reduced cost can be found. The resulting solution is optimal to $\widetilde{\text{MP}}$ and the CG process is terminated.

A CG approach has to define how many subproblems should be solved after each re-optimization of $\widetilde{\text{RMP}}$ and how many of the found columns should be added to $\widetilde{\text{RMP}}$ in each iteration (see, e.g., Morabit et al. 2020). While adding only one column per iteration may lead to a large number of (potentially expensive) $\widetilde{\text{RMP}}$ re-optimizations, columns which are added to $\widetilde{\text{RMP}}$ in the same iteration are derived from the same dual variable values and thus are likely to be very similar, possibly inducing undesired symmetries to the problem. We opt for a balanced approach, which considers both aspects, in which, before launching the CG procedure, the gates are assigned to groups of size ≥ 1. In each iteration, the subproblems associated with one group are solved to optimality and the columns that are found (at most one per gate) are added to $\widetilde{\text{RMP}}$. The gates are allocated to the groups, such that the lateral distances between the gates of one group are maximized. The aim is to mitigate the effect of adding columns responding to the same dual variable values.

Subproblems Let π_t^{bud}, $\pi_{jtb}^{\text{cap,line}}$, $\pi_{tb}^{\text{cap,gate}}$, π_{jtba}^{com} and π^{con} denote the dual variable values of Constraints (4.2c), (4.2d), (4.2e), (4.2f) and (4.2h) associated with aircraft stand $g \in \mathcal{G}$, respectively. Then, for this gate the reduced cost of the best path which could be added to $\widetilde{\text{RMP}}$ is calculated by solving the following Subproblem SP to optimality:

$$\begin{aligned}\min\ \bar{c}^g \coloneqq & \sum_{l\in\mathcal{L}} \sum_{k\in\mathcal{L}\setminus\{l\}} \sum_{t=0}^{T-1} c_{kl}^{\text{i}} \left(1 - \pi_t^{\text{bud}}\right) x_{tkl} \\ & + \sum_{t=0}^{T-1} \sum_{l\in\mathcal{L}} \left(c_l^{\text{o}} + \sum_{j\in\mathcal{J}_g} \sum_{b\in\mathcal{B}_t} \sum_{\substack{l'\in\mathcal{L}\setminus\{0\}:\\ l'=l}} \pi_{jtb}^{\text{cap,line}} \right. \\ & \left. - \sum_{b\in\mathcal{B}_t} \sum_{\substack{l'\in\mathcal{L}\setminus\mathcal{L}^{MARS}:\\ l'=l}} \pi_{tb}^{\text{cap,gate}} \right.\end{aligned} \tag{4.3a}$$

$$+\sum_{j\in\mathcal{J}_g}\sum_{b\in\mathcal{B}_t}\sum_{a\in A}\sum_{\substack{l'\in\mathcal{L}_a:\\ l'=l}}\pi^{\text{com}}_{jtba}\Bigg)\, z_{tl} - \pi^{\text{con}}$$

subject to

$$\sum_{k\in\mathcal{L}\setminus\{l\}} x_{tlk} + z_{tl} - \sum_{k\in\mathcal{L}\setminus\{l\}} x_{t'kl} - \sum_{\substack{t'=t:\\ t'>0}} z_{t'-1,l} = e_{tl} \qquad \forall t = 0,\dots,T-1, l\in\mathcal{L} \tag{4.3b}$$

$$x_{tkl}\in\{0,1\} \qquad \forall k\in\mathcal{L}, l\in\mathcal{L}\setminus\{k\}, \; t=0,\dots,T-1 \tag{4.3c}$$

$$z_{tl}\in\{0,1\} \qquad \forall t=0,\dots,T-1, l\in\mathcal{L} \tag{4.3d}$$

The index g is removed from all parameters and variables, as SP is solved for each aircraft stand $g \in \mathcal{G}$ individually. We solve the subproblems efficiently by means of dynamic programming. That is, we find the shortest path through the (sub-)network corresponding to the respective gate as shown in Figure 4.4. The initial PBB configuration of the gate in $t = 0$ constitutes the source of the network, and in $t = T+1$, we add a supersink which is connected with each node of $t = T$. The weight w_{klt} of the arc connecting the node representing PBB configuration $k \in \mathcal{L}$ in time period $t \in \mathcal{T}$ with the node depicting PBB configuration $l \in \mathcal{L}$ in time period $(t+1) \in \mathcal{T}$ is calculated as

$$\begin{aligned} w_{klt} =& c^{\text{o}}_l + c^{\text{i}}_{kl}\left(1-\pi^{\text{bud}}_t\right) + \sum_{j\in\mathcal{J}_g}\sum_{b\in\mathcal{B}_t}\sum_{\substack{l'\in\mathcal{L}\setminus\{0\}:\\ l'=l}}\pi^{\text{cap,line}}_{jtb} \\ & - \sum_{b\in\mathcal{B}_t}\sum_{\substack{l'\in\mathcal{L}\setminus\mathcal{L}^{MARS}_:\\ l'=l}}\pi^{\text{cap,gate}}_{tb} + \sum_{j\in\mathcal{J}_g}\sum_{b\in\mathcal{B}_t}\sum_{a\in A}\sum_{\substack{l'\in\mathcal{L}_a:\\ l'=l}}\pi^{\text{com}}_{jtba}. \end{aligned} \tag{4.4}$$

If $k = l$ the term $c^{\text{i}}_{kl}\left(1-\pi^{\text{bud}}_t\right)$ is omitted. The weight of the arcs connecting the nodes of $t = T$ with the supersink is set to 0.

Obtaining integral solutions Solving $\widetilde{\text{RMP}}$ gives us an optimal solution to $\widetilde{\text{MP}}$, but this may not necessarily be an integral solution. We therefore employ the following heuristic approach to obtain an integral solution to OP. As soon as the CG process is terminated, the integrality constraints on decision variables λ_p and y_{gjtba} are restored in $\widetilde{\text{RMP}}$ and the resulting RMP′ is solved by a commercial solver. This approach is a heuristic, as only the paths added to $\widetilde{\text{RMP}}$ during the CG procedure can be used in the optimal solution to RMP′. Furthermore, to counteract the slow convergence behavior of RMP′ when close to the optimal solution, we abort the optimization process of RMP′ when the duality gap reaches a sufficiently small value. The solution approach is illustrated as pseudo-code in Algorithm 4.1. The computational study will show that this allows us to find integral solutions of a given quality more quickly than the commercial solver when solving OP (4.1a)-(4.1l).

Algorithm 4.1 Proposed CG heursitic

```
1: Initialize $\widetilde{\text{RMP}}$ and one SP per gate, create $N$ gate groups
2: columnFound ← true, $n \leftarrow 0$
3: while columnFound = true do
4:    Solve $\widetilde{\text{RMP}}$ and get dual variable values
5:    columnFound ← false
6:    if $n = N + 1$ then
7:       $n \leftarrow 0$
8:    end if
9:    for all gates belonging to group $n$ do
10:      Solve SP with the dual variable values from $\widetilde{\text{RMP}}$
11:      $c \leftarrow$ objective function value of SP
12:      if $c \leq 0$ then
13:         Add the respective column to $\widetilde{\text{RMP}}$
14:         columnFound ← true
15:      end if
16:   end for
17:   $n \leftarrow n + 1$
18:   if columnFound = false then
19:      for all gates do
20:         Solve SP with the dual variable values from $\widetilde{\text{RMP}}$
21:         $c \leftarrow$ objective function value of SP
22:         if $c \leq 0$ then
23:            Add the respective column to $\widetilde{\text{RMP}}$
24:            columnFound ← true
25:         end if
26:      end for
27:   end if
28: end while
29: Add integrality constraints to the path variables of $\widetilde{\text{RMP}}$ to
    obtain RMP
30: Solve RMP until the desired duality gap is reached
```

4.6 Computational study

The numerical analysis presented in this section investigates the computational performance of the proposed CG heuristic and consists of four parts. First, we introduce how the data instances used in the following experiments were generated. Second, we investigate the performance obtained when solving OP (4.1a)-(4.1l) to optimality using a commercial solver. Third, we vary the parameters of the proposed CG heuristic to find a good setting with respect to both solution quality and runtime. Finally, we demonstrate the effectiveness of the CG heuristic by comparing its performance to solving OP. All experiments were conducted on a computer equipped with an octa-core CPU running at 3.8GHz and 32GB of RAM, while Gurobi 9.1 was employed as a commercial solver. The algorithms were implemented in Java.

4.6.1 Instances

To investigate the performance of the solution methods, we generated problem instances of different sizes. We model an airport terminal with $G \in \{10, 15, 20\}$ contact gates, each gate being equipped with four lead-in lines, thus $J = 4 \cdot G$. The length of the time horizon is $T \in \{10, 15, 20\}$. The number of aircraft classes is set to $A = 3$ (small, medium, large aircraft with maximum half wingspans s_a of 20, 34, and 44 meters, respectively). There are $L = 5$ PBB configurations available, where $l = 0$ denotes the "no PBB" case and $l = 4$ represents a PBB configuration for MARS gates. The compatibilies between aircraft classes and PBB configurations are defined as $\mathcal{L}_1 = \{1, 2, 3, 4\}$, $\mathcal{L}_2 = \{2, 3, 4\}$, and $\mathcal{L}_3 = \{3, 4\}$. Figure 4.5 shows the five configurations employed in the computational study, including the "no PBB" case (configuration 0).

At $t = 0$, all gates have PBB configuration $l = 1$ installed and the distance between each pair of adjacent aircraft stands is set to 60 meters. Each gate is equipped with two inner lead-in lines located five meters to the left and right of the center of the gate, respectively. To allow for MARS operations, each aircraft stand is additionally equipped with two outer lead-in lines, which are placed 25 meters left and right from the center line of the gate, respectively. Since we cannot disclose real cost data, we use the following cost factors

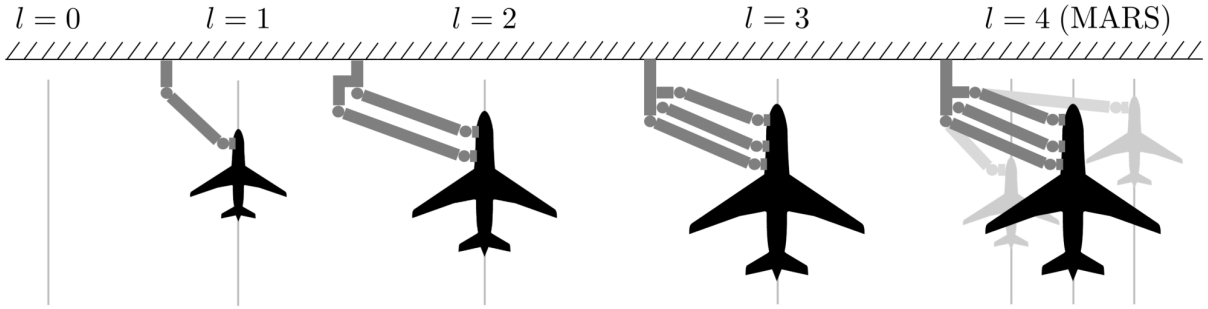


Figure 4.5: PBB configurations in the computational study

reflecting the relationship between the different costs: $c^{\mathrm{i}}_{kl} = \begin{pmatrix} 0 & 8 & 16 & 24 & 32 \\ 8 & 0 & 8 & 16 & 24 \\ 8 & 8 & 0 & 8 & 16 \\ 8 & 8 & 8 & 0 & 16 \\ 8 & 8 & 8 & 8 & 0 \end{pmatrix}$, $c^{\mathrm{o}}_{l} = (\ 0.5 \quad 1.5 \quad 2.5 \quad 3.5 \quad 4.5\ )$ and $c^{-} = 0.02$. The more flexible a PBB configuration l is, the higher are its investment costs c^{i}_{kl} and operating cost c^{o}_{l}. As for the value of c^{-}, the case study presented in Section 4.7 will demonstrate that 0.02 is a reasonable value for a quality oriented airport with respect to the given values of c^{i}_{kl} and c^{o}_{l}. The aircraft classes are weighted with $p_a = (\ 150 \quad 300 \quad 400\ )$, approximately corresponding with the number of passengers carried by the respective types of aircraft. While we investigate the impact of different values of r_t on the solution in the case study in Section 4.7, we relax Budget Constraints (4.1d) and (4.2c) within the scope of the computational study in this section. Regarding the development of air traffic over the time horizon, we model a scenario where a high traffic volume (i.e., approximately G aircraft per demand pattern) throughout the time horizon is combined with a shift in the fleet mix towards larger aircraft. In recent decades, this scenario was frequently observed at airports where the construction of additional capacities (runways, terminals, etc.) was not possible in the medium term although passenger volume was increasing (London Heathrow, for instance).

For each combination of G and T we generated 20 instances, which gives $3 \cdot 3 \cdot 20 = 180$ instances in total. In the following, we will refer to the instances of one G and T as instance class. The 20 instances of one instance class differ with respect to the demand patterns as well as the relative weighting of the patterns:

- The demand patterns in $\mathcal{B}_t$ are selected randomly from a predefined set of demand patterns $\mathcal{B}$ for each time period $t \in \mathcal{T}$, such that three demand patterns are allocated to each time period and the described development of air traffic over the time horizon is replicated[3].

- The weighting factors f_{tb} take random values between $]0;1[$, where for each time period $\sum_{b \in \mathcal{B}_t} f_{tb} = 1$ must hold.

We divided the 180 instances into two batches of 90 instances each, where each batch contains 10 instances of each instance class.

4.6.2 Performance of commercial solver

We applied the commercial solver to the first batch of instances. Table 4.1 presents the average computation times of OP for each instance class.

Table 4.1: Average computation times of OP (4.1a)-(4.1l) [min]

	$T = 10$	$T = 15$	$T = 20$
$G = 10$	1.78	5.00	23.81
$G = 15$	13.59	61.07	129.29
$G = 20$	24.38	94.29	674.54[4]

Computation times increase considerably when the values of T and G are increased. It is apparent that the commercial solver cannot efficiently solve OP to optimality for larger instances. With $T = G = 20$, the solution process is not completed within 24 hours for three instances. OP is thus not applicable to data instances of realistic size where G and T may both take values larger than 20 (e.g., Munich airport has $G = 58$ contact gates[5]) and the number of demand patterns for each time period might be larger than three.

[3] Further details are provided in Appendix C.2.
[4] The optimization process of three instances was aborted after 24 hours.
[5] as of January 2020

4.6.3 Performance of column generation heuristic

When applying the CG heuristic, the following parameters have to be set:

- The number of gates per group as defined for the iterative optimization process of $\widetilde{\text{RMP}}$, denoted as *group_size*.
- The duality gap that needs to be reached to abort the optimization process of RMP′, denoted as *rmp_gap*.

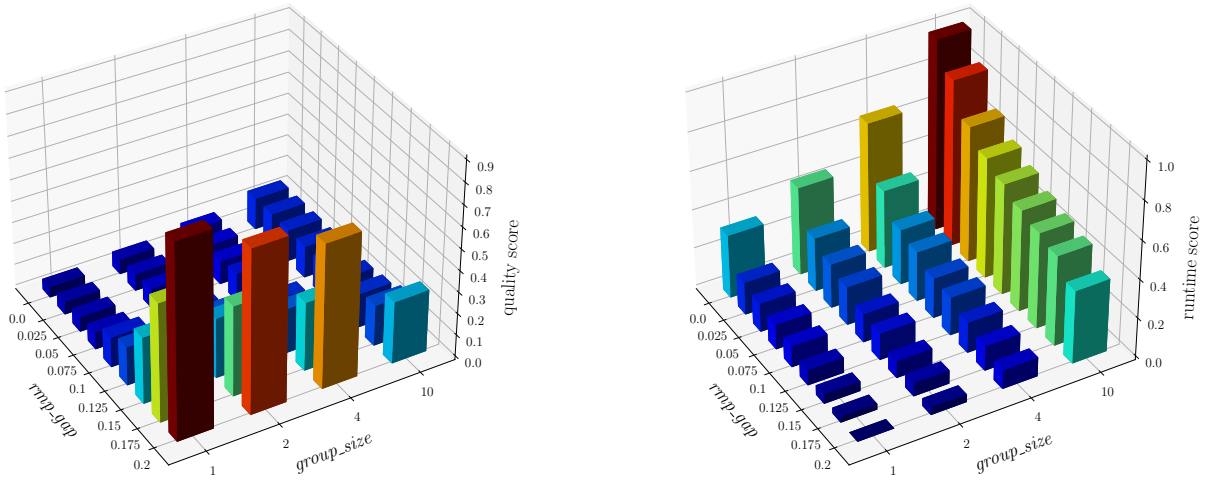
We tested the computational performance of the CG heuristic for all combinations of $group\_size \in \{1, 2, 4, 10\}$ and $rmp\_gap \in \{0, 2.5\%, 5\%, 10\%, 12.5\%, 15\%, 17.5\%, 20\%\}$. Thus, we applied the heuristic $4 \cdot 8 = 32$ times to each of the 90 data instances of the first batch.

To find the best values for *group_size* and *rmp_gap*, we investigated the performance of each value combination of these parameters for every instance class. More precisely, we calculated for each instance class and value combination of *group_size* and *rmp_gap* (a) the average percentage deviation from the optimum (referred to as *optimality gap*) and (b) the average total runtime[6]. We normalized the values of both (a) and (b) to be in $[0; 1]$ by $v^n = \frac{v - v^{min}}{v^{max} - v^{min}}$, where normalized values close to zero indicate small optimality gaps for (a) and short runtimes for (b)[7]. In the following, we will refer to the normalized values as *quality* and *runtime scores*, respectively. Figures 4.6a and 4.6b provide the average quality and runtime scores of each parmeter combination over all instance classes.

As expected, the setting of both *group_size* and *rmp_gap* parameters has a clear impact on the performance of the CG heuristic. Increasing *rmp_gap* leads to a worsening of the quality score (i.e., an increase in the optimality gaps) and an improvement of the runtime score (i.e., shorter computation times). We find that the value of *group_size* only has a notable impact on the quality score for large values of *rmp_gap*. However, the best quality scores are observed when $group\_size = 1$ and *rmp_gap* is small. With respect to the runtime score, *group_size* has a larger impact when the value of *rmp_gap* is small.

[6]The optimum objective function values were obtained by solving OP with unlimited runtime.

[7]For $group\_size = 10$ and $rmp\_gap = 0$, runtimes were significantly higher for all instance classes compared to other parameter combinations. In order to obtain reasonable values by the normalization, we excluded this parameter combination from the normalization procedure and set its value to 1 for all instance classes.



(a) Quality scores

(b) Runtime scores

Figure 4.6: Average quality and runtime scores for different parameter combinations

Again, the best runtime scores were observed for $group\_size = 1$. We conclude that the disadvantage of solving fewer subproblems per iteration (potentially resulting in many necessary $\widetilde{\text{RMP}}$ re-optimization cycles, see Section 4.5) is smaller than the negative effect of adding multiple similar columns to $\widetilde{\text{RMP}}$ per iteration (which results from larger values of $group\_size$) in our case.

As small optimality gaps and short runtimes are conflicting goals, there is no dominating parameter setting. For the following experiments, we selected the parameter combination with the smallest sum of quality and runtime scores, as shown in Figures 4.6a and 4.6b. This is the case for $group\_size = 1$ and $rmp\_gap = 0.075$. With these parameter values, the final heuristic solution is found after 2.15 minutes and the optimality gap is 1.99%, averaged over all instance classes.

Validation To validate the performance of the CG heuristic with the given parameter values, we applied it to the second batch of instances created in Section 4.6.1.

Tables 4.2a-4.2c present the results when applying the CG heuristic with the parameters as selected in Section 4.6.3 to these instances. As a valid benchmark for the computation times of the CG heuristic, we solved each instance with OP (4.1a)-(4.1l) and aborted the

solution procedure once the objective function value obtained by the CG heuristic was reached. Furthermore, we solved OP to optimality as presented in Section 4.6.2 in order to calculate the optimality gaps.

	$T = 10$	$T = 15$	$T = 20$
$G = 10$	0.24	0.92	2.10
$G = 15$	0.35	1.52	3.61
$G = 20$	0.60	2.35	6.29

(a) Average computation times of CG heuristic [min]

	$T = 10$	$T = 15$	$T = 20$
$G = 10$	0.87	1.18	0.87
$G = 15$	0.46	0.51	0.57
$G = 20$	0.62	0.86	0.62

(b) Average computation times of CG heuristic divided by average computation times of OP with abortion criterion

	$T = 10$	$T = 15$	$T = 20$
$G = 10$	1.96	2.68	2.35
$G = 15$	1.54	1.18	2.29
$G = 20$	2.20	1.93	1.95

(c) Average optimality gaps of CG heuristic [%]

Table 4.2: Performance measures of CG heuristic

The results demonstrate the efficiency of the proposed CG heuristic. With optimality gaps between 1.18% and 2.68%, the CG heuristic reaches its final objective function value on average in less than 73% of the runtime needed by the commercial solver to find a solution of equal quality. With the exception of one relatively small instance class ($G = 10$ and $T = 15$), where the commercial solver finds the final objective function value provided by the CG heuristic faster than the CG heuristic and where the optimality gap reaches its maximum, the CG heuristic thoroughly outperforms the commercial solver.

4.7 Case study

In this case study, we consider Munich Airport Terminal 2[8]. For this purpose, we first describe how we modeled the contact gates of the terminal, generated demand patterns

[8]For an introduction to Munich Airport, please refer to Sections 2.5 and 3.6.

based on real traffic data, and considered the recent crisis caused by the SARS-CoV-2 virus in Section 4.7.1[9]. We provide the data set containing all information that is needed to reproduce the case study in Hagspihl et al. (2021). In the following two Sections 4.7.2 and 4.7.3, we investigate the impact of different values of penalty cost c^- and the available budget r_t on the solution proposed by the CG heuristic.

4.7.1 Data

At Munich Airport, passengers are processed via two terminals, with Terminal 2, including its satellite building, serving as a hub terminal for Star Alliance airlines and offering about twice the capacity of Terminal 1. Terminal 2 possesses $G = 47$ contact gates, each of which is equipped with one to three lead-in lines. The contact gates are equipped with apron-drive PBBs, where up to three bridges are installed per gate. Currently four gates are equipped as MARS gates.

Along the lines of the computational study, we considered $L = 5$ PBB configurations (0: No PBB; 1: Single apron-drive PBB; 2: Two apron-drive PBBs; 3: Three apron-drive PBBs; 4: MARS) and $A = 3$ aircraft classes with maximum half wingspans s_a of 20, 34 and 44 meters, respectively. The compatibilities between PBB configurations and aircraft classes were defined as in Section 4.6.1, and we set the length of the time horizon to $T = 20$ time periods, representing the time span from summer 2019 to summer 2029.

As for the development of demand over time, we modeled the recent situation, which was heavily impacted by the SARS-CoV-2 virus. That is, we modeled an abrupt but temporary drop in air traffic volume in 2020, followed by a continuous recovery from the low point reached. To generate the demand patterns, we analyzed a representative week of the flight plan of Munich Airport for the summer period of 2019. Of all flights listed for that week, we only considered those that were processed at a gate (contact or remote) belonging to Terminal 2. Using the list of the relevant arrivals and departures, we calculated the number of aircraft which were simultaneously accommodated on the apron of Terminal 2, with a time granularity of five minutes. Figure 4.7 presents the resulting

[9]We do not need to consider the entire airport, as Terminal 2 is used exclusively by member airlines of the Star Alliance and thus the traffic can be forecasted individually for this part of the airport.

number of aircraft at a gate over time, which demonstrates the aforementioned oscillatory behavior.

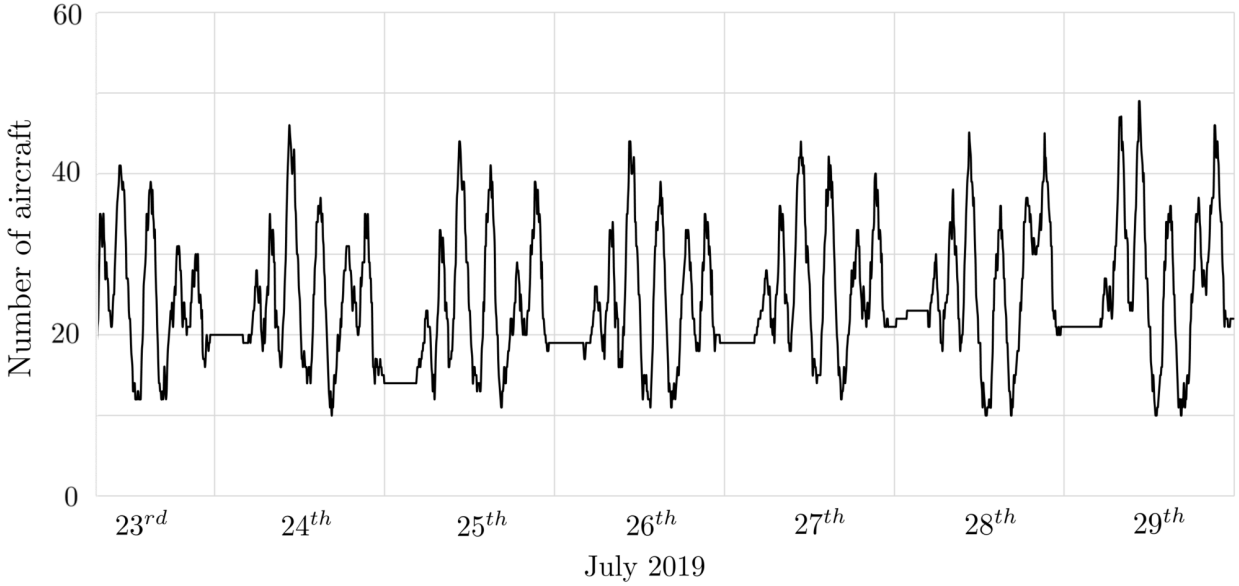


Figure 4.7: Number of aircraft accommodated simultaneously at Munich Airport Terminal 2 over one typical week in summer 2019

To derive the relevant demand patterns, we first determined the number of aircraft of each class to be accommodated simultaneously for each peak. We then identified those demand patterns that are not dominated by others, using the same dominance criterion as introduced in Section 3.3. As a result, we identified four demand patterns to be included in the model for the period of summer 2019. The demand patterns for the following time periods were generated based on those we identified for the period of summer 2019, i.e., $\mid \mathcal{B}_t \mid = 4$ for each $t \in \mathcal{T}$. More specifically, the demand patterns for $t \in \{1, 2, 10, 19\}$ were generated as follows. In periods 1 and 2, the patterns follow the steep decrease in traffic caused by the SARS-CoV-2 crisis. The demand patterns of $t = 10$ equal those of period 0, based on the assumption that the demand will have recovered from the crisis by then. The demand patterns of the last period $t = 19$ are based on those of $t = 10$, ensuring that the demand is stationary with low variance after $t = 10$; however, based on recent developments around the Airbus A380 and the Boeing 747, the largest class of aircraft becomes less frequent. For $t \in [3, 9]$, the demand patterns were interpolated between the

patterns of $t = 2$ and $t = 10$, and random but seasonally biased deviations were added[10]. For $t \in [11, 18]$, the same method was applied using the patterns of $t = 10$ and $t = 19$ for the interpolation. Figure 4.8 shows the average number of aircraft per class across all demand patterns for each time period.

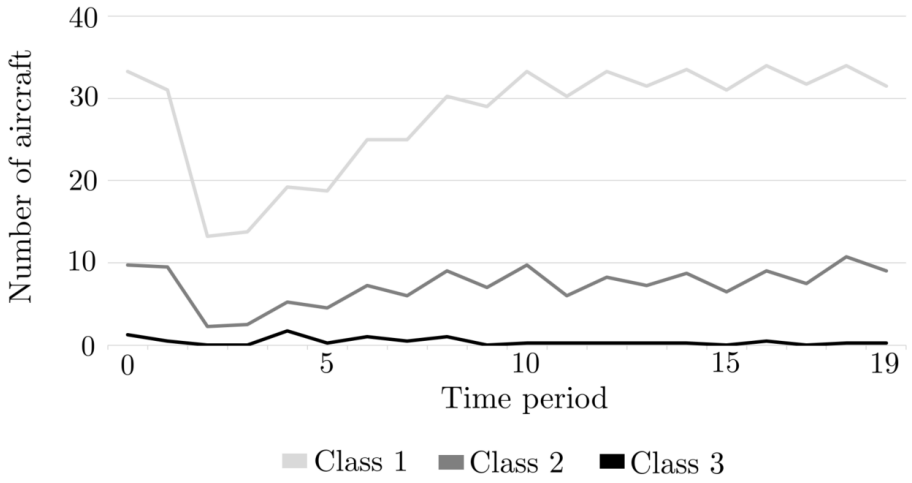


Figure 4.8: Development of the average traffic volume and fleet mix over the time horizon

4.7.2 Variation of penalty costs

In practice, both the costs c^{i}_{kl}, c^{o}_{l}, and the aircraft class weights p_a are given exogeneously by PBB and aircraft manufacturers. In contrast, the penalty cost c^{-} depends on the desired service quality the airport aims to offer to its customers: Accommodating an aircraft at a remote parking position leads to a higher penalty cost at a quality oriented hub airport than at a low cost oriented small airport. For the following analysis, we set c^{i}_{kl}, c^{o}_{l}, and p_a to the values given in Section 4.6.1, whereas we varied the value of c^{-} within $[0; 0.3]$ with a step size of 0.005, which allows us to analyze the effect of service quality on the decisions taken and the different cost components. While we investigate the impact of different budgets in Section 4.7.3, the investigation of different penalty cost values is based on relaxed Budget Constraints (4.2c), which is also in line with Section 4.6.1. For simplicity reasons, all weights f_{tb} were set to 0.25.

[10] A random value was added to each value of D_{ba}. If pattern $b \in \mathcal{B}$ is associated with a winter period, the value was chosen from $[-3, -1]$, otherwise from $[1, 3]$.

Averaged over all values of c^-, the CG heuristic with $group\_size = 1$ and $rmp\_gap = 0.075$ solves the instance in 2.83 minutes. The objective function value (the total cost) obtained by the CG heuristic is 1612.35 on average. When solving the CG heuristic with the additional constraint that the PBB configurations installed in summer 2019 must not be changed, the objective function value is 2006.88 on average. Consequently, averaged over all values of c^- and under the assumption that the future traffic development represented by the demand patterns indeed materializes in the future, changing the PBB configurations over time can yield savings of 19.66%.

The solution found by the CG heuristic depends on the value of c^-. For small values of c^-, a large share of aircraft is processed at remote parking positions. With increasing c^-, more PBB capacities are installed to reduce the share of aircraft processed at remote positions. Figure 4.9 shows the relation between the value of $c^-$and the share of aircraft processed at remote parking positions. Furthermore, the figure presents the total investment, operating and penalty costs obtained depending on the value of c^-. Finally, Table 4.3 presents the number of gates equipped with each PBB configuration at the beginning of the time horizon and - for each value of c^- - the number of gates equipped with each PBB configuration after the first five time periods.

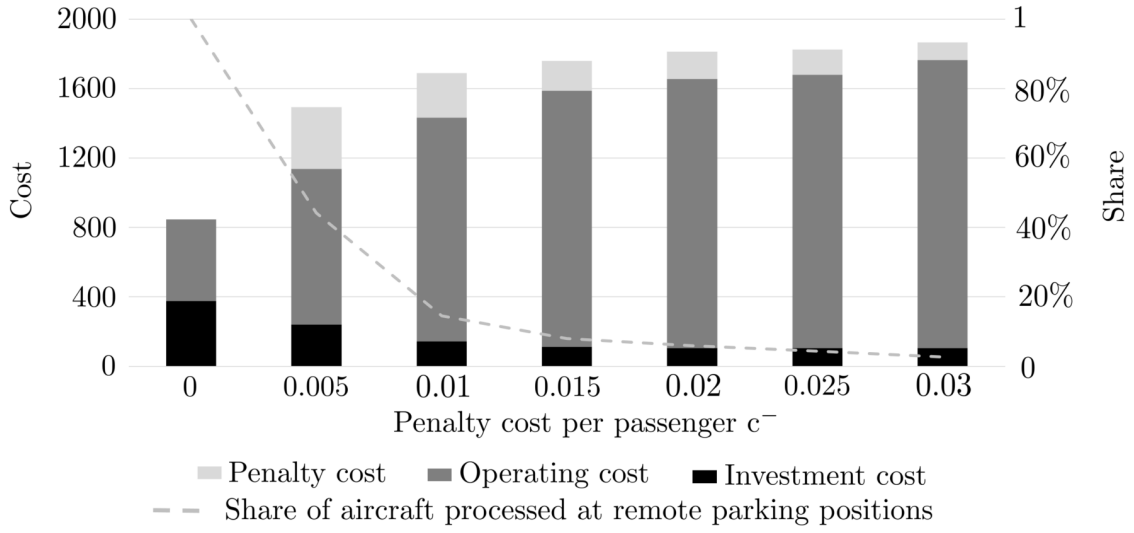


Figure 4.9: Total investment, operating and penalty costs depending on the value of c^-

In their initial state, as presented in Table 4.3, the PBB configurations installed at Munich

Table 4.3: Distribution of PBB configurations at the beginning of the time horizon and after five time periods for each value of c^-

l	$t = 0$	$t = 5$						
		c^-						
		0	0.005	0.010	0.015	0.020	0.025	0.030
0	0	47	26	13	9	7	6	5
1	27	0	21	28	27	27	28	28
2	15	0	0	6	10	12	12	11
3	1	0	0	0	1	1	1	0
4	4	0	0	0	0	0	0	3

Airport Terminal 2 would allow 98.99% of the traffic modelled for summer 2019 to be accommodated at contact gates. In the light of the given demand scenario, with a strong decrease in demand at the beginning of the time horizon followed by a slow recovery in the following years, PBB configurations are more likely to be downgraded or maintained than upgraded, depending on the ratio of the cost components. As shown in Figure 4.9, the operating costs are the largest cost item across all values of c^-. For $c^- \in [0; 0.02]$, the operating costs are increasing in c^- and for $c^- \geq 0.02$ they remain comparably constant. Consequently, the higher the value of c^-, the more operating costs are accepted for larger PBB configurations in order to avoid increasing penalty costs. In contrast, the investment costs for changing the configuration are decreasing in c^- for $c^- \in [0; 0.015]$ and remain constant for $c^- \geq 0.015$. That is, the larger the value of c^-, the fewer gates are downgraded in order to save on operating costs.

In the SARS-CoV-2 scenario considered, PBB configurations are downgraded for all values of c^- (see Table 4.3). That is, PBB configurations which have the flexibility to accommodate larger types of aircraft are substituted by PBB configurations that can only be used to process smaller types of aircraft. Furthermore, for all values of c^-, the PBB configuration is changed to $l = 0$ at some gates, i.e., the gate is considered a remote parking position after the change. When $c^- = 0$, there is no incentive to accommodate aircraft at contact gates, and in order to reduce the operating costs to the minimum, the PBB configuration is changed to $l = 0$ at all gates in $t = 0$. When c^- is increased to 0.005, the number of gates without any PBBs installed in $t = 5$ decreases to 26. However, no gate remains with $l \geq 2$. The more the value of c^- is increased, the smaller the number

of gates without any PBBs after five time periods and the higher the number of gates equipped with $l \geq 2$. Only when $c^- = 0.030$, we observe gates which are equipped with $l = 4$ in $t = 5$. For smaller values of c^-, the high degree of flexibility which justifies the high operating costs of MARS gates cannot be sufficiently exploited in the given traffic scenario.

4.7.3 Variation of budget

In the remainder of the case study, we investigate the impact of different investment budgets available per period on the solution. Assuming that the available budget per period is constant over the time horizon, we omit the time index of r_t in the following. We added Budget Constraints (4.2c) and run the CG heuristic for budget values $r \in \{0, 8, 16, 32, 64, 1504\}$, where $r = 0$ and $r = 1,504$ represent two extreme scenarios: When $r = 0$, there is no budget available and hence, no PBB configuration changes can be undertaken. By contrast, when $r = 1,504$, the available budget equals the upper bound of the budget demand $G \cdot \max_{k,l \in \mathcal{L}} c^{\mathrm{i}}_{kl}$, and thus, all PBB configuration changes can be undertaken. Based on the prior analysis of the penalty cost, we set c^- to 0.025. Figure 4.10 shows the resulting total investment, operating, and penalty costs as well as the share of aircraft processed at remote parking positions for each value of r. Furthermore, Table 4.4 provides a detailed overview of the investment costs per time period for each value of r. As an example, the detailed results of one instance ($c^- = 0.025$ and $r = 32$) are provided in Appendix C.3.

Table 4.4: Investment cost per time period for different values of r

	t																				
r	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	$\sum$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	8	8	8	8	0	8	8	0	0	0	0	0	0	0	0	0	0	0	48
16	16	16	8	16	0	16	0	0	0	8	0	8	0	0	0	0	0	0	0	0	88
32	32	24	16	8	8	0	0	8	0	8	0	0	8	0	0	0	0	0	0	0	112
64	56	24	24	0	0	8	0	0	0	0	8	0	0	0	0	0	0	0	0	0	120
1504	88	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	104

Figure 4.10 demonstrates that the objective function value improves for higher values of r. The total cost difference between the two extreme scenarios ($r = 0$ and $r = 1,504$)

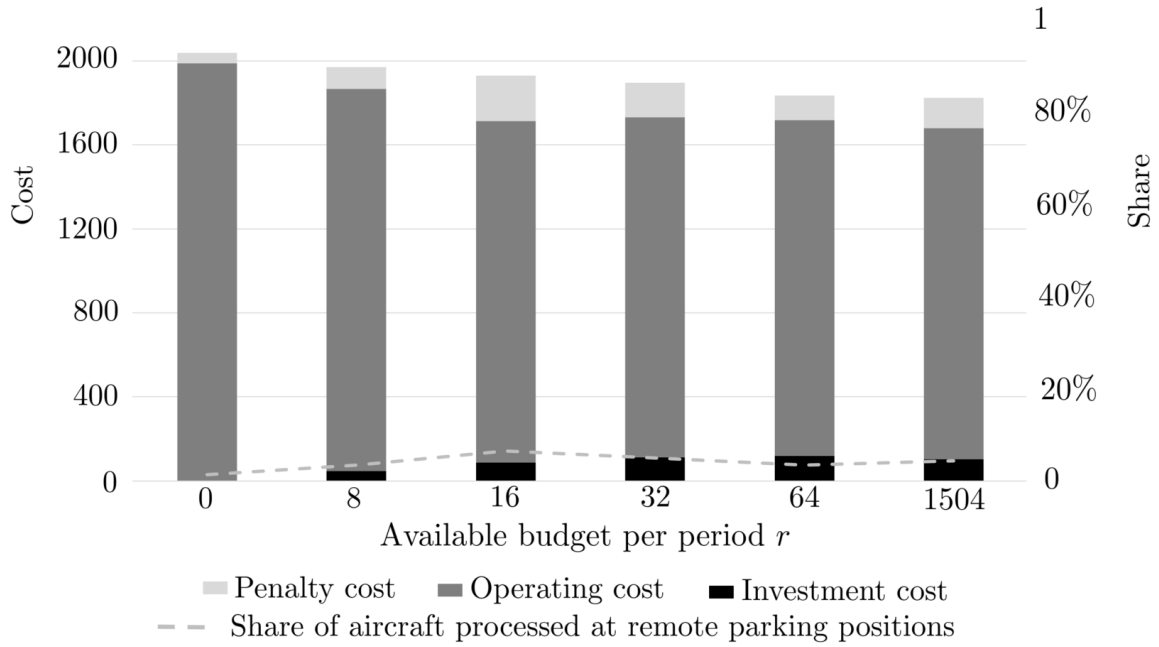


Figure 4.10: Total investment, operating and penalty costs depending on the value of r

is 10.5%. As discussed earlier, in the SARS-CoV-2 scenario, PBB configurations should be downgraded in order to reduce the operating costs. The higher the available budget for PBB configuration changes is, (1) the more of these changes can be realized, (2) the earlier the changes can be made, and (3) the higher the resulting reduction of operating costs is. The last column of Table 4.4 shows that higher values of r are associated with higher total investment cost, except for $r = 1,504$. Furthermore, Table 4.4 shows that the higher the value of r, the more PBB configuration changes tend to be undertaken in early periods of the time horizon where the effect of the change is the largest. Regarding the penalty costs, Figure 4.10 shows a slight variation in the percentage of aircraft processed at remote parking positions for different values of r. However, this percentage is smaller than 6.5% in all cases. The maximum is reached for $r = 16$, where the penalty costs also reach their maximum.

Whether a higher budget leads to lower total costs depends on the characteristics of the demand scenario. If the growth in air traffic volume and the change of the fleet mix are moderate, higher budgets will, in general, not lead to considerable savings. By contrast, the SARS-CoV-2 scenario is characterized by a drastic drop in air traffic volume in 2020

and a slow recovery in the following years. Thus, a higher degree of flexibility - especially at the beginning of the time horizon - leads to substantial reductions of total costs.

Overall, the case clarifies the tradeoffs between investment, operating, and penalty costs in a dynamic environment, as well as the relevance of the available investment budget, and demonstrates that our approach provides airports with highly valuable decision support.

4.8 Conclusion

In this chapter, we introduced the Dynamic Gate Configuration Problem (DGCP) and formulated it as a deterministic mixed-binary problem. By considering neighborhood constraints, MARS gates and the concept of demand patterns, the mathematical formulation incorporates operational constraints in the strategic context of the problem. Our computational results demonstrated the intractability of the extensive formulation for data instances of realistic size. We presented a CG heuristic to circumvent this problem and illustrated its good performance with respect to solution quality as well as computation time for data instances of mutable complexity. Finally, we presented a case study to demonstrate the implications of the results of the CG heuristic in reality. The results provide airport operators with important insights into the extent to which the capacities provided at the gates at present are capable of handling demand in the future. Furthermore, the results indicate at *which* gate(s) the PBB configuration should be changed in the event that the solution recommends such change(s). Upgrades are performed such that they maximize the gain in operational flexibility and downgrades are performed such that they minimize the loss of operational flexibility.

In future research, the model could be extended to allow for the addition of new gates within the time horizon, representing terminal extensions or the construction of entirely new terminals. Furthermore, our model does not take into account that managerial decisions besides the ones considered here, not to mention legal constraints, may pose further restrictions on the accommodation of aircraft at certain aircraft stands, besides spatial neighborhood constraints. For example, airports may have mutual agreements with airlines for the exclusive use of gates (for instance, see Young and Wells 2011) and airports may have to provide gates equipped with security equipment of different levels. On the

one hand, both aspects require a more detailed subdivision of demand as presented in this work, while on the other, they might allow a decomposition of the problem. For example, our approach may be applied to a subset of all gates if these gates are used by one airline only by contract and the traffic development can be estimated separately for that airline. The concept of the model can also be transferred to other problems, where limited capacites need to be (re-)configured to meet changing demand over time. For instance, hospitals have to determine which rooms to allocate to which ward, while the number of patients treated by the wards fluctuates over time and both investment and personnel costs need to be considered.

5 Conclusion and Outlook

We addressed three planning problems, all of which involve the question of how particular elements of airport infrastructure should be designed in order to provide additional capacities or to adapt existing capacities to changes in traffic volume and/or fleet mix. In Chapter 2, we considered the introduction of air taxis as a new means of transport that may provide a more convenient way to reach airports. More specifically, we presented a framework for determining the optimal locations for vertiports in the area surrounding a given airport in order to establish a shuttle network for the airport. In Chapters 3 and 4, we turned our attention towards the management of contact gates. We introduced an approach to find the optimal locations and orientations of aircraft parking positions when a given terminal is planned to be newly built or renovated in Chapter 3. We then assumed this layout to be given in Chapter 4, where we investigated which gate should be equipped with which PBB configuration at which point in time in order to react to changing traffic characteristics over a strategic time horizon.

From a modeling perspective, the problems presented in Chapters 3 and 4 share several commonalities. First, we classify aircraft by wingspan in both problems in order to reduce planning complexity and to take future aircraft types into account. Second, we consider that gates must be equipped appropriately to handle aircraft of a particular class. While we use the term equipment in a broader sense in Chapter 3, we focus specifically on PBB configurations in Chapter 4. Third, we use the precise positions of parked aircraft, which are determined by the lead-in lines they are assigned to, to prevent collisions of aircraft parked adjacently in both problems. Fourth, we focus on hub airports where air traffic arrives and departs in waves, allowing us to consider only the points in time where traffic volume and/or fleet mix reach their peaks, and enabling us to apply our concept of demand patterns to both problems. Finally, we explicitly consider MARS gates in both problems.

Furthermore, neither model presented in Chapters 3 and 4 can be solved to optimality by commercial solvers for realistic instances in reasonable amounts of time. We presented decomposition approaches to solve both problems efficiently. In Chapter 3, we developed a problem-specific decomposition approach, which exploits our observation that individual areas of the apron can be planned independently of each other, as soon as a decision is made for each demand pattern on how many aircraft of each class should be parked in which area. Furthermore, we used several acceleration techniques, which significantly reduce computation times. In Chapter 4, we applied Dantzig-Wolfe decomposition in order to use a column generation based heuristic to solve the original problem. In our approach, a column specifies for one gate which PBB configuration(s) are installed at which point in time. We investigated how many columns should ideally be added to the restricted master problem per iteration and at which duality gap the optimization process of the restricted master problem with restored integrality constraints should be aborted in the last step of our heuristic. In contrast to Chapters 3 and 4, the problem presented in Chapter 2 can be solved to optimality by commercial solvers as long as the set of potential vertiport locations is small. However, embedding the multinomial logit model in the hub location problem initially results in a nonlinear model, which we linearized based on the approach by Haase (2009).

We presented extensive case studies for Munich Airport for all three problems. Our experiments in Chapter 2 indicate that the high ticket prices for air taxi flights predicted for market launch will initially result in a significant number of passengers choosing to travel by air taxi only if the distance between the vertiports and Munich Airport is relatively short. However, since the area around Munich Airport offers a well developed road and rail network by global standards, the air taxi can only make up for the transfer time at the vertiport on longer flight routes, and thus shorten the overall journey to Munich Airport. In consequence, in the short to medium term, the air taxi will not reduce travel times to Munich Airport for most passengers. However, this could be different for airports in other regions of the world, as our findings are primarily based on the price sensitivity of passengers in Bavaria as well as the attractiveness of alternative means of transport in the catchment area of the airport. Our computational results in Chapter 3 highlight the benefits of planning flexibility (see Table 3.6) and, with few exceptions, the spatial efficiency of lead-in lines that are placed perpendicular to the terminal facade. Finally, our experiments in Chapters 3 and 4 demonstrate how airport planners can maximize

capacity and best adapt existing infrastructure to meet changing traffic characteristics without using additional space.

This dissertation paves the way for numerous future research activities. First, in the vertiport location problem in Chapter 2, we currently assume that the number of passengers traveling to Munich Airport from each origin is given and independent of our location decisions. However, neighboring airports compete for passengers, and improving the accessibility of one airport could, ceteris paribus, lead to previously indifferent passengers choosing that airport. For example, passengers from northern Bavaria currently travel about the same distance to Munich Airport as to Frankfurt Airport. The introduction of an air taxi shuttle service to Munich Airport could increase the proportion of passengers from this region who choose Munich Airport, as long as a similar network to Frankfurt Airport is not set up in parallel. This in turn could have a significant impact on the optimal locations for vertiports. Second, our approach to determine the optimal locations and orientations of aircraft parking positions around a given terminal in Chapter 3 could be used as a starting point to investigate whether specific terminal shapes are particularly well suited in general to maximize the number of aircraft that can be handled at contact gates simultaneously. This information could prove especially valuable for the planning of future terminals at space-constrained airports. In addition, our solution methodology can be further improved to achieve even shorter runtimes. For example, machine learning models could be trained to identify particularly promising demand decompositions, whose subproblems could then be solved first. Furthermore, with minor adjustments, our approach could also be applied to seaports to define the positions and orientations of berths for ships of different sizes in a given harbor basin. This problem shares some features of our problem, as collisions between ships must be prevented and infrastructure on land, such as structures for loading and unloading ship cargo or connections for fuel supply, must be included in the planning of berths. Third, regarding the problem to determine optimal PBB configurations at gates of a terminal over time as presented in Chapter 4, we are confident that ideas from the solution methodology in Chapter 3 can be used to significantly reduce the runtime. For example, analogous to Chapter 3, the terminal could be divided into independently plannable areas as soon as it is decided for each demand pattern how many aircraft of which class are to be parked in which area. The existing solution methodology could then be applied to each area individually. Finally, in this dissertation only two elements of airport infrastructure were considered: The gates as

the interface of airside and landside airport infrastructure, and the infrastructure of a new airport feeder system. As described in Chapter 1, airport infrastructure comprises numerous other elements, which could be addressed in future work. For instance, the taxiway layout of an airport could be optimized to minimize expected waiting times of aircraft due to ground traffic. In addition, future technological developments will lead to new elements of infrastructure being added to airports, creating new challenges for airport planners and thus opening up new opportunities for optimization. For example, the use of hydrogen as aircraft fuel will require the installation of appropriate refueling infrastructure at airports. In this context, the question will arise as to where on the apron and to what extent infrastructure needs to be installed in order to meet demand for hydrogen refueling at minimum overall cost.

The results of this dissertation show how mathematical optimization can significantly contribute to the effective planning and use of airport infrastructure. In light of the expected further growth in civil aviation, the importance of this contribution will continue to increase, especially for space-constrained hub airports. We therefore strongly encourage further research in this direction.

Bibliography

Advanced Air Mobility Institute (2023). Our mission. https://aaminstitute.org/mission. Last accessed on Mar 14, 2023.

Aena (2020). Guía de tarifas 2020: Edición octubre. http://www.aena.es/es/comercial/guia-tarifas.html. Last accessed on Dec 14, 2020.

Airbus (2018). Global market forecast: Global networks, global citizens: 2018-2037. https://www.airbus.com/content/dam/corporate-topics/publications/media-day/GMF-2018-2037.pdf. Last accessed on Aug 02, 2019.

Airbus (2019). Airbus and Emirates reach agreement on A380 fleet, sign new widebody orders. https://www.airbus.com/newsroom/press-releases/en/2019/02/airbus-and-emirates-reach-agreement-on-a380-fleet--sign-new-widebody-orders.html. Last accessed on Sep 01, 2020.

Airbus (2022). Airport operations - Autocad 3D view aircraft drawings. https://web.archive.org/web/20211027010324/https:/www.airbus.com/aircraft/support-services/airport-operations-and-technical-data/autocad-3-view-aircraft-drawings.html. Last accessed on Dec 22, 2022.

Airport Improvement Magazine (2010). "like new" boarding bridges return to service at daytona beach. https://airportimprovement.com/article/new-boarding-bridges-return-service-daytona-beach. Last accessed on Jun 16, 2019.

Allgemeiner Deutscher Automobil-Club e. V. (2021). ADAC Autokosten Herbst/Winter 2021/22. https://assets.adac.de/image/upload/v1618309743/ADAC-eV/KOR/Text/PDF/autokostenuebersicht_whgbcz.pdf. Last accessed on Jun 30, 2023.

Alumur, S. and Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1–21.

Alvarez-Valdes, R., Martinez, A., and Tamarit, J. M. (2013). A branch & bound algorithm for cutting and packing irregularly shaped pieces. *International Journal of Production Economics*, 145(2):463–477.

Anjos, M. F. and Vieira, M. V. (2017). Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions. *European Journal of Operational Research*, 261(1):1–16.

Aros-Vera, F., Marianov, V., and Mitchell, J. E. (2013). p-hub approach for the optimal park-and-ride facility location problem. *European Journal of Operational Research*, 226(2):277–285.

Ashford, N. J., Mumayiz, S. A., and Wright, P. H., editors (2011). *Airport Engineering: Planning, Design, and Development of 21st Century Airports*. John Wiley & Sons, Hoboken, New Jersey, 4th edition.

Bandara, S. J. and Wirasinghe, S. C. (1989). Airport gate position estimation under uncertainty. *Transportation Research Record Journal of the Transportation Research Board*, 1199:41–48.

Bennell, J. A. and Oliveira, J. F. (2008). The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 184(2):397–415.

Bennell, J. A. and Oliveira, J. F. (2009). A tutorial in irregular shape packing problems. *Journal of the Operational Research Society*, 60(1):93–105.

Boeing (2020). Boeing CEO updates employees on quarterly results and market realities. `https://boeing.mediaroom.com/news-releases-statements?item=130713`. Last accessed on Sep 01, 2020.

Boston Consulting Group (2020). Seven trends that will reshape the airline industry. `https://www.bcg.com/publications/2020/seven-trends-reshape-airline-industry`. Last accessed on Aug 03, 2020.

Briskorn, D. and Dienstknecht, M. (2019). Mixed-integer programming models for tower crane selection and positioning with respect to mutual interference. *European Journal of Operational Research*, 273(1):160–174.

Brueckner, J. K. (2004). Network structure and airline scheduling. *The Journal of Industrial Economics*, 52(2):291–312.

Burghouwt, G. and de Wit, J. (2005). Temporal configurations of european airline networks. *Journal of Air Transport Management*, 11(3):185–198.

Caves, R. (1994). A search for more airport apron capacity. *Journal of Air Transport Management*, 1(2):109–120.

Chehrazad, S., Roose, D., and Wauters, T. (2022). A fast and scalable bottom-left-fill algorithm to solve nesting problems using a semi-discrete representation. *European Journal of Operational Research*, 300(3):809–826.

Chen, L., Wandelt, S., Dai, W., and Sun, X. (2022). Scalable vertiport hub location selec-

tion for air taxi operations in a metropolitan region. *INFORMS Journal on Computing*, 34(2):834–856.

Cheng, C.-H., Ho, S. C., and Kwan, C.-L. (2012). The use of meta-heuristics for airport gate assignment. *Expert Systems with Applications*, 39(16):12430–12437.

Daş, G. S., Gzara, F., and Stützle, T. (2020). A review on airport gate assignment problems: Single versus multi objective approaches. *Omega*, 92:102–146.

Daskin, M. S. (2011). *Service Science: Service Operations for Managers and Engineers*. John Wiley & Sons Incorporated, New York, 1st ed. edition.

de Neufville, R. and Odoni, A. R. (2013). *Airport Systems: Planning, Design, and Management*. McGraw-Hill, New York.

de Souza Queiroz, L. R. and Andretta, M. (2022). A branch–and–cut algorithm for the irregular strip packing problem with uncertain demands. *International Transactions in Operational Research*, 29(6):3486–3513.

Desrosiers, J. and Lübbecke, M. E. (2005). A primer in column generation. In Desaulniers, G., Desrosiers, J., and Solomon, M. M., editors, *Column Generation*, pages 1–32. Springer US, New York.

Dorndorf, U., Drexl, A., Nikulin, Y., and Pesch, E. (2007). Flight gate scheduling: State-of-the-art and recent developments. *Omega*, 35(3):326–334.

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

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

Dorndorf, U., Jaehn, F., and Pesch, E. (2017). Flight gate assignment and recovery strategies with stochastic arrival and departure times. *OR Spectrum*, 39(1):65–93.

Drira, A., Pierreval, H., and Hajri-Gabouj, S. (2007). Facility layout problems: A survey. *Annual Reviews in Control*, 31(2):255–267.

Eiselt, H. A. and Marianov, V. (2009). A conditional p-hub location problem with attraction functions. *Computers & Operations Research*, 36(12):3128–3135.

Eurocontrol (2018). European aviation in 2040: Challenges of growth. `https://www.eurocontrol.int/sites/default/files/content/documents/official-documents/reports/challenges-of-growth-2018.pdf`. Last accessed on Aug 01, 2019.

European Aviation Safety Agency (2017). Certification specifications and guidance material for aerodromes design CS-ADR-DSN. `https://www.faa.gov/documentLibrary/media/`

Advisory_Circular/150-5300-13A-chg1-interactive-201907.pdf. Last accessed on Oct 22, 2021.

Farahani, R. Z. and Hekmatfar, M., editors (2009). *Facility Location: Concepts, Models, Algorithms and Case Studies*. Springer eBook Collection Business and Economics. Physica-Verlag HD, Heidelberg.

Farahani, R. Z., Hekmatfar, M., Arabani, A. B., and Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096–1109.

Federal Aviation Administration (2012). AC 150/5300-13A - airport design. https://www.faa.gov/airports/resources/advisory_circulars/index.cfm/go/document.information/documentNumber/150_5300-13A. Last accessed on Oct 22, 2021.

Federal Aviation Administration (2022). Urban air mobility and advanced air mobility. https://www.faa.gov/uas/advanced_operations/urban_air_mobility. Last accessed on Mar 14, 2023.

Frej Vitalle, R., Zhang, Y., Normann, B., and Shen, N. (2020). A model for the integration of UAM operations in and near terminal areas. In *AIAA AVIATION 2020 FORUM*, Reston, Virginia. American Institute of Aeronautics and Astronautics.

Fu, M., Rothfeld, R., and Antoniou, C. (2019). Exploring preferences for transportation modes in an urban air mobility environment: Munich case study. *Transportation Research Record: Journal of the Transportation Research Board*, 2673(10):427–442.

Fu, M., Straubinger, A., and Schaumeier, J. (2022). Scenario-based demand assessment of urban air mobility in the greater munich area. *Journal of Air Transportation*, 30(4):125–136.

Garrow, L. A., German, B. J., and Leonard, C. E. (2021). Urban air mobility: A comprehensive review and comparative analysis with autonomous and electric ground transportation for informing future research. *Transportation Research Part C: Emerging Technologies*, 132.

Gelhausen, M. C., Berster, P., and Wilken, D. (2020). *Airport Capacity Constraints and Strategies for Mitigation: A Global Perspective*. Elsevier Science & Technology, Saint Louis.

Goyal, R., Reiche, C., Fernando, C., and Cohen, A. (2021). Advanced air mobility: Demand analysis and market potential of the airport shuttle and air taxi markets. *Sustainability*, 13(13):7421.

Guépet, J., Acuna-Agost, R., Briant, O., and Gayon, J. P. (2015). Exact and heuristic approaches to the airport stand allocation problem. *European Journal of Operational Research*, 246(2):597–608.

Guépet, J., Briant, O., Gayon, J. P., and Acuna-Agost, R. (2016). The aircraft ground routing

problem: Analysis of industry punctuality indicators in a sustainable perspective. *European Journal of Operational Research*, 248(3):827–839.

Haase, K. (2009). Discrete location planning. Technical Report WP-09-07. Institute of Transport and Logistics Studies, University of Sydney.

Haase, K. and Müller, S. (2014). A comparison of linear reformulations for multinomial logit choice probabilities in facility location models. *European Journal of Operational Research*, 232(3):689–691.

Hagspihl, T., Kolisch, R., Fontaine, P., and Schiffels, S. (2023a). Apron layout planning - optimal positioning of aircraft stands. Working Paper. TUM School of Management. Technical University of Munich.

Hagspihl, T., Kolisch, R., Fontaine, P., and Schiffels, S. (2023b). Instances for: Apron layout planning - optimal positioning of aircraft stands. *Mendeley Data*, (V1).

Hagspihl, T., Kolisch, R., Ruf, C., and Schiffels, S. (2021). Case study data for: "dynamic gate configurations at airports: A network optimization approach". *Mendeley Data*, (V1).

Hagspihl, T., Kolisch, R., Ruf, C., and Schiffels, S. (2022). Dynamic gate configurations at airports: A network optimization approach. *European Journal of Operational Research*, 301(3):1133–1148.

Hagspihl, T., Kolisch, R., and Schiffels, S. (2023c). Planning an airport shuttle network with air taxis using choice-based optimization. Working Paper. TUM School of Management. Technical University of Munich.

Hassounah, M. I. and Steuart, G. N. (1993). Demand for aircraft gates. *Transportation Research Record*, 1423(1423):26–33.

Hess, S. and Palma, D. (2019). Apollo: A flexible, powerful and customisable freeware package for choice model estimation and application. *Journal of Choice Modelling*, 32.

Holden, J. and Goel, N. (2016). Fast-forwarding to a future of on-demand urban air transportation. `https://evtol.news/__media/PDFs/UberElevateWhitePaperOct2016.pdf`. Last accessed on Mar 14, 2023.

Horonjeff, R., McKelvey, F. X., Sproule, W. J., and Young, S. B. (2010). *Planning and Design of Airports*. McGraw-Hill, New York, 5th edition.

Huang, C., Wong, C. K., and Tam, C. M. (2011). Optimization of tower crane and material supply locations in a high-rise building site by mixed-integer linear programming. *Automation in Construction*, 20(5):571–580.

Ikli, S., Mancel, C., Mongeau, M., Olive, X., and Rachelson, E. (2021). The aircraft runway scheduling problem: A survey. *Computers & Operations Research*, 132:105336.

Ilahi, A., Belgiawan, P. F., Balać, M., and Axhausen, K. W. (2020). Exploring willingness to pay of urban air mobility and on-demand transport. In *hEART 2020: 9th Symposium of the European Association for Research in Transportation*. European Association for Research in Transportation.

International Air Transport Association (2004). Airport development reference manual: 9th edition effective january 2004. `https://www.iata.org/en/publications/store/airport-development-reference-manual/`. Last accessed on Aug 03, 2020.

International Air Transport Association (2023a). Annual review 2023. `https://www.iata.org/contentassets/c81222d96c9a4e0bb4ff6ced0126f0bb/annual-review-2023.pdf`. Last accessed on Oct 18, 2023.

International Air Transport Association (2023b). Number of flights performed by the global airline industry from 2004 to 2021, with forecasts until 2023. In Statista. `https://www.statista.com/statistics/564769/airline-industry-number-of-flights/`. Last accessed on Oct 18, 2023.

International Air Transport Association (2023c). Number of scheduled passengers boarded by the global airline industry from 2004 to 2022. In Statista. `https://www.statista.com/statistics/564717/airline-industry-passenger-traffic-globally/`. Last accessed on Oct 18, 2023.

International Civil Aviation Organization (1987). Airport planning manual: Part 1 master planning: Doc 9184-AN/902.

International Civil Aviation Organization (2005). Doc 9157 aerodrome design manual: Part 2: Taxiways, aprons and holding bays. `https://www.bazl.admin.ch/dam/bazl/de/dokumente/Fachleute/Flugplaetze/ICAO/icao_doc_9157_aerodromedesignmanual-part2.pdf.download.pdf/icao_doc_9157_aerodromedesignmanual-part2.pdf`. Last accessed on Jul 26, 2019.

International Civil Aviation Organization (2018). Annex 14 to the convention on international civil aviation: Aerodromes: Volume 1 aerodrome design and operations. `https://www.bazl.admin.ch/dam/bazl/de/dokumente/Fachleute/Regulationen_und_Grundlagen/icao-annex/icao_annex_14_aerodromesvolumei-aerodromedesignandoperations.pdf.download.pdf/an14_v1_cons.pdf`. Last accessed on Jul 28, 2019.

Jeong, J., So, M., and Hwang, H.-Y. (2021). Selection of vertiports using k-means algorithm and noise analyses for urban air mobility (uam) in the seoul metropolitan area. *Applied Sciences*, 11(12):5729.

Katsigiannis, F. A. and Zografos, K. G. (2021). Optimising airport slot allocation considering

flight-scheduling flexibility and total airport capacity constraints. *Transportation Research Part B: Methodological*, 146:50–87.

Kazda, A. and Caves, R. E. (2008). *Airport Design and Operation*. Emerald, Bingley, 2nd edition.

Kim, S. H. and Shim, S. (2021). Park-and-ride facility location selection under nested logit demand function. `https://arxiv.org/pdf/2111.09522`. Working paper. Last accessed on Nov 11, 2021.

Laporte, G., Desrochers, M., and Nobert, Y. (1984). Two exact algorithms for the distance-constrained vehicle routing problem. *Networks*, 14(1):161–172.

Laporte, G., Nickel, S., and Saldanha da Gama, F. (2019). *Location science*. Springer, Cham, Switzerland, second edition edition.

Leao, A. A., Toledo, F. M., Oliveira, J. F., Carravilla, M. A., and Alvarez-Valdés, R. (2020). Irregular packing problems: A review of mathematical models. *European Journal of Operational Research*, 282(3):803–822.

Lilium (2020). Designing a scalable vertiport. `https://lilium.com/newsroom-detail/designing-a-scalable-vertiport`. Last accessed on Mar 15, 2023.

Lilium (2022). Revolutionizing sustainable high-speed regional air mobility. `https://investors.lilium.com/static-files/ddddaf2f-e33e-4115-8fa4-7a0779b28301`. Last accessed on Mar 10, 2023.

Lim, E. and Hwang, H. (2019). The selection of vertiport location for on-demand mobility and its application to seoul metro area. *International Journal of Aeronautical and Space Sciences*, 20(1):260–272.

Lüer-Villagra, A. and Marianov, V. (2013). A competitive hub location and pricing problem. *European Journal of Operational Research*, 231(3):734–744.

Luo, Q. and Rao, Y. (2023). Heuristic algorithms for the special knapsack packing problem with defects arising in aircraft arrangement. *Expert Systems with Applications*, 215.

Martinez-Sykora, A., Alvarez-Valdes, R., Bennell, J. A., Ruiz, R., and Tamarit, J. M. (2017). Matheuristics for the irregular bin packing problem with free rotations. *European Journal of Operational Research*, 258(2):440–455.

Martins, E. Q. V. (1984). On a multicriteria shortest path problem. *European Journal of Operational Research*, 16(2):236–245.

McKinsey & Company (2020). To take off, flying vehicles first need places to land. `https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/`

to-take-off-flying-vehicles-first-need-places-to-land. Last accessed on Mar 15, 2023.

Mirković, B. and Tošić, V. (2014). Airport apron capacity: estimation, representation, and flexibility. *Journal of Advanced Transportation*, 48(2):97–118.

Mirković, B. and Tošić, V. (2016). Apron capacity at hub airports – the impact of wave-system structure. *Journal of Advanced Transportation*, 50(7):1489–1505.

Mirković, B. and Tošić, V. (2017). The difference between hub and non-hub airports – an airside capacity perspective. *Journal of Air Transport Management*, 62:121–128.

Morabit, M., Desaulniers, G., and Lodi, A. (2020). Machine-learning-based column selection for column generation. *Technical report, Les Cahiers du GERAD G–2020–29*.

Munich Airport (2016). Münchner Flughafen plant die Erweiterung von Terminal 1. https://www.munich-airport.de/mehr-kapazitat-mehr-qualitat-mehr-komfort-338570. Last accessed on Sep 06, 2019.

Munich Airport (2020). Annual traffic report 2019. https://www.munich-airport.com/_b/00000000000000008934656bb5e9eb677/annual-traffic-report-2019.pdf. Last accessed on Jan 15, 2021.

Munich Airport (2023a). Awards. https://www.munich-airport.com/awards-263282. Last accessed on Apr 14, 2023.

Munich Airport (2023b). Location & expansion. https://www.munich-airport.com/location-expansion-263252. Last accessed on Oct 25, 2023.

Narciso, M. E. and Piera, M. A. (2015). Robust gate assignment procedures from an airport management perspective. *Omega*, (50):82–95.

National Academies of Sciences, Engineering, and Medicine (2010). *Airport Passenger Terminal Planning and Design: Volume 1: Guidebook*. The National Academies Press, Washington, DC.

National Aeronautics and Space Administration (2021). Regional air mobility: Leveraging our national investments to energize the american travel experience. https://ntrs.nasa.gov/citations/20210014033. Last accessed on Sep 26, 2023.

Owen, S. H. and Daskin, M. S. (1998). Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447.

Ploetner, K. O., Al Haddad, C., Antoniou, C., Frank, F., Fu, M., Kabel, S., Llorca, C., Moeckel, R., Moreno, A. T., Pukhova, A., Rothfeld, R., Shamiyeh, M., Straubinger, A., Wagner, H., and Zhang, Q. (2020). Long-term application potential of urban air mobility complementing public transport: an upper bavaria example. *CEAS Aeronautical Journal*, 11(4):991–1007.

Pukhova, A., Llorca, C., Moreno, A., Staves, C., Zhang, Q., and Moeckel, R. (2021). Flying taxis revived: Can urban air mobility reduce road congestion? *Journal of Urban Mobility*, 1.

Qin, Y., Chan, F. T., Chung, S. H., Qu, T., and Niu, B. (2018). Aircraft parking stand allocation problem with safety consideration for independent hangar maintenance service providers. *Computers & Operations Research*, 91:225–236.

Rajendran, S. and Shulman, J. (2020). Study of emerging air taxi network operation using discrete-event systems simulation approach. *Journal of Air Transport Management*, 87.

Rajendran, S. and Zack, J. (2019). Insights on strategic air taxi network infrastructure locations using an iterative constrained clustering approach. *Transportation Research Part E: Logistics and Transportation Review*, 128:470–505.

Rath, S. and Chow, J. Y. (2022). Air taxi skyport location problem with single-allocation choice-constrained elastic demand for airport access. *Journal of Air Transport Management*, 105.

ReVelle, C. S. and Eiselt, H. A. (2005). Location analysis: A synthesis and survey. *European Journal of Operational Research*, 165(1):1–19.

Rezaei, S., Khojandi, A., Mohsena Haque, A., Brakewood, C., Jin, M., and Cherry, C. R. (2022). Park-and-ride facility location optimization: A case study for nashville, tennessee. *Transportation Research Interdisciplinary Perspectives*, 13.

Rimjha, M., Hotle, S., Trani, A., and Hinze, N. (2021a). Commuter demand estimation and feasibility assessment for urban air mobility in northern california. *Transportation Research Part A: Policy and Practice*, 148:506–524.

Rimjha, M., Hotle, S., Trani, A., Hinze, N., and Smith, J. C. (2021b). Urban air mobility demand estimation for airport access: A los angeles international airport case study. In *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, Piscataway, NJ. IEEE.

Schubert, A. (2023). S-Bahn München: Unpünktlich wie noch nie - einige Linien besonders schlecht. *Süddeutsche Zeitung*, 2023. Last accessed on Apr 14, 2023.

Schweiger, K. and Preis, L. (2022). Urban air mobility: Systematic review of scientific publications and regulations for vertiport design and operations. *Drones*, 6(7):179.

Sinha, A. A. and Rajendran, S. (2023). Study on facility location of air taxi skyports using a prescriptive analytics approach. *Transportation Research Interdisciplinary Perspectives*, 18:100761.

SkyDrive (2023). Skydrive makes u.s. market entrance with plans to develop practical use cases in south carolina. `https://en.skydrive2020.com/archives/8635`. Last accessed on Mar 10, 2023.

Statistisches Bundesamt (2022). Löhne und Gehälter nach Bundesländern und Geschlecht 2021. In Statista. https://de.statista.com/statistik/daten/studie/209211/umfrage/loehne-und-gehaelter-in-deutschland-nach-bundeslaendern-und-geschlecht-2010/. Last accessed on Jul 08, 2023.

Statistisches Bundesamt (2023a). Bevölkerung: Deutschland, Stichtag, Altersjahre. https://www-genesis.destatis.de/genesis//online?operation=table&code=12411-0005&bypass=true&levelindex=1&levelid=1689686133123#abreadcrumb. Last accessed on Jul 18, 2023.

Statistisches Bundesamt (2023b). Privathaushalte in Deutschland nach monatlichem Haushaltsnettoeinkommen 2022. In Statista. https://de.statista.com/statistik/daten/studie/3048/umfrage/privathaushalte-nach-monatlichem-haushaltsnettoeinkommen/. Last accessed on Jul 08, 2023.

Stephan, K., Weidinger, F., and Boysen, N. (2021). Layout design of parking lots with mathematical programming. *Transportation Science*, 55(4):930–945.

Steuart, G. N. (1974). Gate position requirements at metropolitan airports. *Transportation Science*, 8(2):169–189.

Straubinger, A., Kluge, U., Fu, M., Al Haddad, C., Ploetner, K. O., and Antoniou, C. (2020a). Identifying demand and acceptance drivers for user friendly urban air mobility introduction. In Müller, B. and Meyer, G., editors, *Towards User-Centric Transport in Europe 2: Enablers of Inclusive, Seamless and Sustainable Mobility*, pages 117–134. Springer, Cham.

Straubinger, A., Rothfeld, R., Shamiyeh, M., Büchter, K.-D., Kaiser, J., and Plötner, K. O. (2020b). An overview of current research and developments in urban air mobility – setting the scene for uam introduction. *Journal of Air Transport Management*, 87.

Sun, Y. and Schonfeld, P. (2015). Stochastic capacity expansion models for airport facilities. *Transportation Research Part B: Methodological*, 80:1–18.

Tomita, E., Tanaka, A., and Takahashi, H. (2006). The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28–42.

Train, K. (2009). *Discrete choice methods with simulation*. Cambridge University Press, Cambridge and New York, 2. ed. edition.

Travel PR News (2019). Budapest airport installs brand new passenger boarding bridges. https://travelprnews.com/budapest-airport-installs-brand-new-passenger-boarding-bridges-649599/travel-press-release/2019/05/27/. Last accessed on Mar 04, 2021.

Tu, N., Li, Z.-C., Fu, X., and Lei, Z. (2020). Airline network competition in inter-continental market. *Transportation Research Part E: Logistics and Transportation Review*, 143.

Tung Tung, C. and Lin Chew, K. (1992). A multicriteria pareto-optimal path algorithm. *European Journal of Operational Research*, 62(2):203–209.

Umetani, S. and Murakami, S. (2022). Coordinate descent heuristics for the irregular strip packing problem of rasterized shapes. *European Journal of Operational Research*, 303(3):1009–1026.

Vertical Aerospace (2023). About us: Revolutionising urban air mobility and electrifying air travel. `https://vertical-aerospace.com/about-us/`. Last accessed on Mar 10, 2023.

Wang, K., Jacquillat, A., and Vaze, V. (2022). Vertiport planning for urban aerial mobility: An adaptive discretization approach. *Manufacturing & Service Operations Management*, 24(6):2797–3306.

Weiszer, M., Burke, E. K., and Chen, J. (2020). Multi-objective routing and scheduling for airport ground movement. *Transportation Research Part C: Emerging Technologies*, 119:102734.

Willey, L. C. and Salmon, J. L. (2021). A method for urban air mobility network design using hub location and subgraph isomorphism. *Transportation Research Part C: Emerging Technologies*, 125.

Wirasinghe, S. C. and Bandara, S. J. (1990). Airport gate position estimation for minimum total costs—approximate closed form solution. *Transportation Research Part B: Methodological*, 24(4):287–297.

Wu, Z. and Zhang, Y. (2021). Integrated network design and demand forecast for on-demand urban air mobility. *Engineering*, 7(4):473–487.

Xiao, Y., Fu, X., and Zhang, A. (2013). Demand uncertainty and airport capacity choice. *Transportation Research Part B: Methodological*, 57:91–104.

Young, S. B. and Wells, A. T. (2011). *Airport Planning & Management*. McGraw-Hill Professional, New York, NY, 6th edition.

Zhang, H., Liu, Q., Wei, L., Zeng, J., Leng, J., and Yan, D. (2022). An iteratively doubling local search for the two-dimensional irregular bin packing problem with limited rotations. *Computers & Operations Research*, 137.

A Appendices Vertiport Location

A.1 Notation

Table A.1: Notation for hub location problem

Sets, parameters	
$\mathcal{O} = \{1, \ldots, O\}$	Set of origins
$\mathcal{K} = \{1, \ldots, K\}$	Set of potential vertiport locations
$\mathcal{I} = \{1, \ldots, I\}$	Set of itineraries
$\mathcal{I}^{\text{gnd}} \subset \mathcal{I}$	Subset of itineraries where passengers only use ground-based means of transport
$\mathcal{I}^{\text{air}} \subset \mathcal{I}$	Subset of itineraries where passengers also use air taxis
$\mathcal{I}_o \subset \mathcal{I}$	Subset of itineraries that is available to passengers at origin $o \in \mathcal{O}$
$\mathcal{I}_o^{\text{gnd}} \subset \mathcal{I}$	Subset of itineraries available to passengers at origin $o \in \mathcal{O}$ where only ground-based means of transport are used
$\mathcal{I}_o^{\text{air}} \subset \mathcal{I}$	Subset of itineraries available to passengers at origin $o \in \mathcal{O}$ where also air taxis are used
V_{oi}	Representative utitlity for passengers from origin $o \in \mathcal{O}$ when using itinerary $i \in \mathcal{I}_o$ to get to the airport
n_o	Number of passengers (demand) from origin $o \in \mathcal{O}$
q	Number of vertiports to be opened
r_{oi}	Probability that passengers from origin $o \in \mathcal{O}$ use itinerary $i \in \mathcal{I}_o$ to get to the airport if all vertiports $k \in \mathcal{K}$ were built, i.e. $r_{oi} = \frac{e^{V_{oi}}}{\sum_{i \in \mathcal{I}_o} e^{V_{oi}}}$
$\hat{i}_o$	Itinerary from set $\mathcal{I}_o^{\text{gnd}}$ with the largest choice probability for passengers from origin $o \in \mathcal{O}$, i.e. $r_{o\hat{i}_o} \geq r_{oi}$ for all $i \in \mathcal{I}_o^{\text{gnd}}$

γ	Small value (e.g., 0.0001)
Decision variables	
$y_k \in \{0, 1\}$	1, if vertiport $k \in \mathcal{K}$ is opened; 0, otherwise
$p_{oi} \in \mathbb{R}_0^+$	Probability that passengers from origin $o \in \mathcal{O}$ use itinerary $i \in \mathcal{I}_o$ to get to the airport

Table A.2: Notation for discrete choice model

Sets, parameters	
$\mathcal{Q} = \{1, \ldots, Q\}$	Set of criteria considered by the passengers when evaluating alternative itineraries
k_i	Vertiport $k \in \mathcal{K}$ where passengers transfer from any ground-based means of transport to air taxi on itinerary $i \in \mathcal{I}$
U_{oi}	Utility that passengers from origin $o \in \mathcal{O}$ obtain from reaching the airport using itinerary $i \in \mathcal{I}$
ϵ_{oi}	Difference between true utility U_{oi} and representative utility V_{oi} for passengers from origin $o \in \mathcal{O}$ on itinerary $i \in \mathcal{I}$
β_i^{asc}	Alternative-specific constant of itinerary $i \in \mathcal{I}$
β_i^q	Weighting parameter for itinerary $i \in \mathcal{I}$ with respect to criterion $q \in \mathcal{Q}$
η_{oi}^q	Performance of itinerary $i \in \mathcal{I}_o$ for passengers from origin $o \in \mathcal{O}$ with respect to criterion $q \in \mathcal{Q}$

Table A.3: Notation for model application and case study

Sets, parameters	
$\mathcal{I}^{\text{car}}$	Set of itineraries where passengers reach the airport directly by car
$\mathcal{I}^{\text{PT}}$	Set of itineraries where passengers reach the airport directly by public transport

$\mathcal{I}^{\text{car\_AT}}$	Set of itineraries where passengers reach the airport by air taxi, and the vertiport is reached by car
$\mathcal{I}^{\text{PT\_AT}}$	Set of itineraries where passengers reach the airport by air taxi, and the vertiport is reached by public transport
$\mathcal{I}_o^{\text{car}}$	Set of itineraries where passengers reach the airport directly by car available to passengers at origin $o \in \mathcal{O}$
$\mathcal{I}_o^{\text{PT}}$	Set of itineraries where passengers reach the airport directly by public transport available to passengers at origin $o \in \mathcal{O}$
$\mathcal{I}_o^{\text{car\_AT}}$	Set of itineraries where passengers reach the airport by air taxi, and the vertiport is reached by car available to passengers at origin $o \in \mathcal{O}$
$\mathcal{I}_o^{\text{PT\_AT}}$	Set of itineraries where passengers reach the airport by air taxi, and the vertiport is reached by public transport available to passengers at origin $o \in \mathcal{O}$
$\beta^{\text{car,asc}}$	Alternative-specific constant of itinerary $i \in \mathcal{I}^{\text{car}}$ when determining the representative utility
$\beta^{\text{PT,asc}}$	Alternative-specific constant of itinerary $i \in \mathcal{I}^{\text{PT}}$ when determining the representative utility
$\beta^{\text{car\_AT,asc}}$	Alternative-specific constant of itinerary $i \in \mathcal{I}^{\text{car\_AT}}$ when determining the representative utility
$\beta^{\text{PT\_AT,asc}}$	Alternative-specific constant of itinerary $i \in \mathcal{I}^{\text{PT\_AT}}$ when determining the representative utility
$\beta^{\text{car},tt}$	Weighting factor for travel time of itinerary $i \in \mathcal{I}^{\text{car}}$ when determining the representative utility
$\beta^{\text{PT},tt}$	Weighting factor for travel time of itinerary $i \in \mathcal{I}^{\text{PT}}$ when determining the representative utility
$\beta^{\text{car\_AT},tt}$	Weighting factor for travel time of itinerary $i \in \mathcal{I}^{\text{car\_AT}}$ when determining the representative utility
$\beta^{\text{PT\_AT},tt}$	Weighting factor for travel time of itinerary $i \in \mathcal{I}^{\text{PT\_AT}}$ when determining the representative utility
$\beta^{\text{car},tc}$	Weighting factor for travel costs of itinerary $i \in \mathcal{I}^{\text{car}}$ when determining the representative utility
$\beta^{\text{PT},tc}$	Weighting factor for travel costs of itinerary $i \in \mathcal{I}^{\text{PT}}$ when determining the representative utility

$\beta^{\text{car\_AT},tc}$	Weighting factor for travel costs of itinerary $i \in \mathcal{I}^{\text{car\_AT}}$ when determining the representative utility
$\beta^{\text{PT\_AT},tc}$	Weighting factor for travel costs of itinerary $i \in \mathcal{I}^{\text{PT\_AT}}$ when determining the representative utility
η_{oi}^{tt}	Travel time of itinerary $i \in \mathcal{I}$ for passengers at origin $o \in \mathcal{O}$
η_{oi}^{tc}	Travel costs of itinerary $i \in \mathcal{I}$ for passengers at origin $o \in \mathcal{O}$
$t_{o,k}^{\text{car}}$	Travel time from origin $o \in \mathcal{O}$ to vertiport $k \in \mathcal{K}$ by car
$t_{o,\text{apt}}^{\text{car}}$	Travel time from origin $o \in \mathcal{O}$ to the airport by car
$t_{o,k}^{\text{PT}}$	Travel time from origin $o \in \mathcal{O}$ to vertiport $k \in \mathcal{K}$ by public transport
$t_{o,\text{apt}}^{\text{PT}}$	Travel time from origin $o \in \mathcal{O}$ to the airport by public transport
$t_{k,\text{apt}}^{\text{AT}}$	Travel time from vertiport $k \in \mathcal{K}$ to the airport by air taxi
$t^{\text{car+}}$	Time needed to park the car at a vertiport or the airport
$t^{\text{PT+}}$	Waiting time at the departure station of the public transport service
$d_{o,k}$	Ground (i.e., street) distance between origin $o \in \mathcal{O}$ and vertiport $k \in \mathcal{K}$
$d_{o,\text{apt}}$	Ground (i.e., street) distance between origin $o \in \mathcal{O}$ and the airport
$d_{k,\text{apt}}$	Direct (i.e., Euclidean) distance between vertiport $k \in \mathcal{K}$ and the airport
v^{AT}	Air taxi cruising speed
$c^{\text{car,fix}}$	Fixed cost component of all itineraries where a car is used, accounting for the parking cost at the vertiport/airport
$c^{\text{car,var}}$	Per-kilometer cost component of all itineraries where a car is used
$c^{\text{PT,low}}, c^{\text{PT,mid}}, c^{\text{PT,high}}$	Cost levels of public transport
$c^{\text{AT,fix}}$	Fixed cost component of all itineraries where an air taxi is used
$c^{\text{AT,var}}$	Per-kilometer cost component of all itineraries where an air taxi is used

B Appendices Apron Layout

B.1 Notation

Table B.1: Notation for AGLP

Sets, parameters	
$\mathcal{A} = \{1, \ldots, A\}$	Set of aircraft classes
$\mathcal{L} = \{1, \ldots, L\}$	Set of lead-in lines
$\mathcal{G} = \{1, \ldots, G\}$	Set of gates
$\mathcal{K} = \{1, \ldots, K\}$	Set of demand patterns
$\mathcal{A}^{\text{small}}$	Subset of $\mathcal{A}$, which includes the aircraft classes from which up to two aircraft can be processed at one gate simultaneously
$\mathcal{A}^{\text{large}}$	Subset of $\mathcal{A}$, which includes the aircraft classes from which only one aircraft can be processed at one gate at one point in time
C_{al}	1, if an aircraft of class $a \in \mathcal{A}$ can be parked at lead-in-line $l \in \mathcal{L}$; 0, otherwise
E_{lmab}	1, if aircraft of classes $a \in \mathcal{A}$ and $b \in \mathcal{A}$ cannot be parked simultaneously at lead-in lines $l \in \mathcal{L}$ and $m \in \mathcal{L} \setminus \{l\}$; 0, otherwise
F_{lg}	1, if lead-in line $l \in \mathcal{L}$ can be assigned to gate $g \in \mathcal{G}$; 0, otherwise
H_{ga}	1, if aircraft of class $a \in \mathcal{A}$ could be parked at gate $g \in \mathcal{G}$ in the past; 0, otherwise
D_{ak}	Number of aircraft of class $a \in \mathcal{A}$ that need to be parked simultaneously for demand pattern $k \in \mathcal{K}$

W_{ak}	Weighting factor for aircraft class $a \in \mathcal{A}$ and demand pattern $k \in \mathcal{K}$
Δ	Distance between adjacent lead-in line starting points
κ	Rotation angle between the lead-in lines that share the same starting point
Decision variables	
$y_l \in \{0, 1\}$	1, if lead-in line $l \in \mathcal{L}$ is used to park an aircraft for at least one demand pattern; 0, otherwise
$x_{glak} \in \{0, 1\}$	1, if lead-in line $l \in \mathcal{L}$ is assigned to gate $g \in \mathcal{G}$ and is used to park an aircraft of class $a \in \mathcal{A}$ for demand pattern $k \in \mathcal{K}$; 0, otherwise
$u_{gl} \in \{0, 1\}$	1, if lead-in line $l \in \mathcal{L}$ is assigned to gate $g \in \mathcal{G}$; 0, otherwise
$q_{ak} \geq 0$	Number of aircraft belonging to class $a \in \mathcal{A}$ that cannot be parked at any contact gate for demand pattern $k \in \mathcal{K}$
$v_{ga} \in \{0, 1\}$	1, if aircraft of class $a \in \mathcal{A}$ can be parked at gate $g \in \mathcal{G}$; 0, otherwise

Table B.2: Notation for solution approach

Sets, parameters	
$\mathcal{S} = \{1, \ldots, S\}$	Set of airport areas that are independent with respect to Constraints (3.1j)
$\mathcal{C} = \{1, \ldots, C\}$	Set of demand decompositions
$\mathcal{C}_k$	Set of demand decompositions pretending that demand pattern $k \in \mathcal{K}$ is the only demand pattern that exists
$\mathcal{L}_s$	Subset of $\mathcal{L}$ containing all lead-in lines belonging to area $s \in \mathcal{S}$
$\mathcal{G}_s$	Subset of $\mathcal{G}$ containing all gates belonging to area $s \in \mathcal{S}$
$\mathcal{P}_s$	Set of efficient parking patterns associated with area $s \in \mathcal{S}$
$\mathcal{P}_{cs}$	Set of parking patterns $p \in \mathcal{P}_s$ contained in demand decomposition $c \in \mathcal{C}$ for area $s \in \mathcal{S}$
r_{ap}	Number of aircraft of class $a \in \mathcal{A}$ contained in parking pattern $p \in \mathcal{P}_s$
$p\,(c, s, k)$	Parking pattern from $\mathcal{P}_s$ that is contained in demand decomposition $c \in \mathcal{C}$ for area $s \in \mathcal{S}$ and demand pattern $k \in \mathcal{K}$
π	A path through the network created to find efficient parking patterns and to check whether a solution to the relaxed subproblem is feasible for a parking pattern
z_1^*, z_2^*	Optimal values of Objective Functions (3.1a) and (3.1b), respectively
z_{1k}^*	Optimal value of Objective Function (3.1a) that can be reached for demand pattern $k \in \mathcal{K}$
z_{1c}^*, z_{2c}^*	Optimal values for Objective Functions (3.1a) and (3.1b) when aircraft are assigned to areas according to demand decomposition $c \in \mathcal{C}$
z_{2cs}^*	Optimal value for Objective Function (3.1b) in area $s \in \mathcal{S}$ when aircraft are assigned to areas according to demand decomposition $c \in \mathcal{C}$
Variables	

LB_c	Lower bound for the value of Objective Function (3.1a) when aircraft are assigned to areas according to demand decomposition $c \in \mathcal{C}$
LB_{cs}	Lower bound for the value of Objective Function (3.1a) in area $s \in \mathcal{S}$ when aircraft are assigned to areas according to demand decomposition $c \in \mathcal{C}$
UB	Upper bound for the value of Objective Function (3.1a)
$\sigma_{a\pi}$	Number of class $a \in \mathcal{A}$ aircraft parked in path π
ω_{csa}	Minimum number of gates equipped for class $a \in \mathcal{A}$ aircraft required in area $s \in \mathcal{S}$ to park all aircraft contained in $\mathcal{P}_{cs}$
ψ_{csak}	Number of class $a \in \mathcal{A}$ aircraft that can be parked at gates which are equipped for aircraft of a larger class $a' > a$ in area $s \in \mathcal{S}$ for demand pattern $k \in \mathcal{K}$ when aircraft are assigned to areas according to demand decomposition $c \in \mathcal{C}$
ζ_{cak}	Number of aircraft belonging to classes $\{a, \ldots, A\}$ that are parked at all areas for demand pattern $k \in \mathcal{K}$ when aircraft are assigned to areas according to demand decomposition $c \in \mathcal{C}$

B.2 Process to compute aircraft safety envelopes

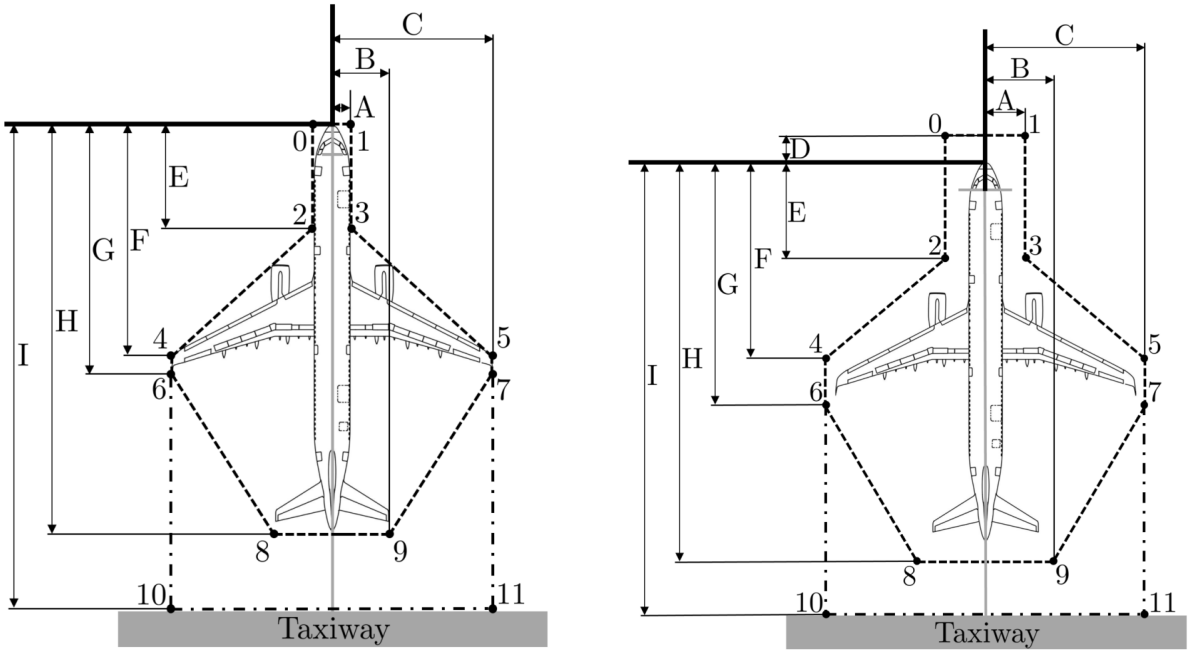
To compute the safety envelope for each aircraft class, we first determine the dimensions of a so-called critical design aircraft. We then compute the aircraft safety envelope based on the critical design aircraft.

Critical design aircraft The critical design aircraft represents a fictitious aircraft type, which in all respects has the maximum dimensions of all real aircraft types included in the respective ADG. Thus, all real aircraft belonging to a class can be handled at a parking position if the critical design aircraft corresponding to the class can be parked there.

To determine the dimensions of the critical design aircraft of a class, the following key figures need to be known for each aircraft type belonging to that class: The width of the

fuselage, the wingspan, the width of the horizontal stabilizer, the linear positions of the wingbox, leading and trailing edges of the wingtip, and the overall length of the aircraft. The critical design aircraft of a class is initialized using the dimensions of the longest aircraft belonging to the class. Next, the maximum wingspan, the maximum fuselage width, and the maximum width of the horizontal stabilizer are identified from all aircraft belonging to the class and transferred to the critical design aircraft. Finally, the critical design aircraft is compared to each aircraft type belonging to the class and it is checked whether the outer shape of the respective aircraft is covered entirely by the outer shape of the critical design aircraft. In case the outer shape of the critical design aircraft does not entirely cover another aircraft of the class, its geometry is changed accordingly.

Aircraft safety envelopes We can now compute the safety envelope for the critical design aircraft of each class. Figure B.1a shows the aircraft safety envelope without minimum safety clearance for one aircraft.



(a) Without safety clearance

(b) With safety clearance

Figure B.1: Aircraft safety envelopes (Source of aircraft models: Airbus 2022)

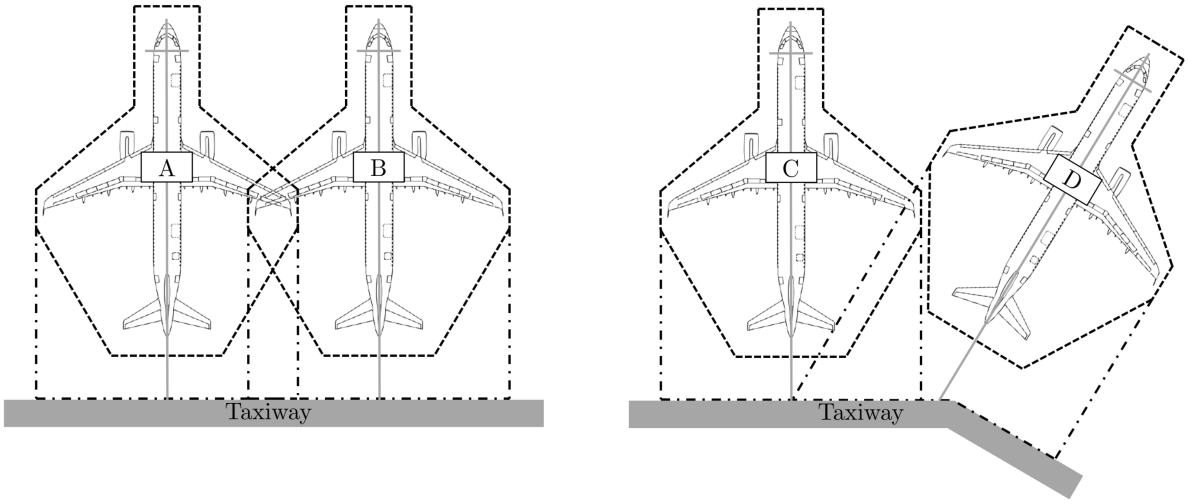
The aircraft safety envelope as shown in Figure B.1a is defined by 6 pairs of points,

where each pair is located symmetrically to the aircraft center line. As the aircraft is accommodated at a parking position, the aircraft center line equals the lead-in line of the parking position. Points (0, 1) are located at the nose of the aircraft, with a lateral distance of half the fuselage width (A) to the lead-in line. Points (4, 5) and (6, 7) are located in front of (F) and behind (G) the wingtips, where the lateral distance of each point from the lead-in line equals half the wingspan (C). The position of points (2, 3) is chosen such that the distance to the nose of the aircraft (E) is maximized while the edges (2, 4) and (3, 5) do not interfere with the engines or the wingbox of the aircraft. Points (8, 9) are located at the tail of the aircraft (H), where the lateral distance to the lead-in line equals the width of the horizontal stabilizer (B). Finally, the path to the taxiway is approximated using two additional points (10, 11). These points are positioned at the intersection of the lead-in line and the taxiway (I), and the lateral distance between the points and the lead-in line equals half the wingspan of the aircraft (C).

For our purpose, the aircraft safety envelope must be extended to include the minimum safety clearances, see Figure B.1b, where the safety clearance is added to both coordinates of each point, so that the points are shifted away from the aircraft.

B.3 Collisions of aircraft safety envelopes

We consider two types of collisions, illustrated in Figure B.2. Figure B.2a shows the case where the safety envelopes of two parked aircraft overlap. Figure B.2b demonstrates the case where the safety envelopes of two aircraft parked adjacently do not overlap when both aircraft are in their parking positions, but where the pushback of aircraft D would lead to an infringement of the safety envelope of aircraft C.



(a) Collision when both aircraft are parked

(b) Collision when one aircraft is moving

Figure B.2: Collisions of aircraft safety envelopes (Source of aircraft models: Airbus 2022)

B.4 Subproblems

The subproblem for demand decomposition $c \in \mathcal{C}$ and area $s \in \mathcal{S}$ is defined as follows:

$$\min \; z_{2cs} = \sum_{g \in \mathcal{G}_s} \sum_{a \in \mathcal{A}} v_{ga} \tag{B.1a}$$

subject to

$$\sum_{g \in \mathcal{G}_s} \sum_{l \in \mathcal{L}_s : C_{al}=1 \cap F_{lg}=1} x_{glak} = r_{ap(c,s,k)} \qquad \forall a \in \mathcal{A}; k \in \mathcal{K} \tag{B.1b}$$

$$\sum_{g \in \mathcal{G}_s : F_{lg}=1} \sum_{a \in \mathcal{A} : C_{al}=1} x_{glak} \leq y_l \qquad \forall l \in \mathcal{L}_s; k \in \mathcal{K} \tag{B.1c}$$

$$\sum_{g \in \mathcal{G}_s : F_{lg}=1} u_{gl} = y_l \qquad \forall l \in \mathcal{L}_s \tag{B.1d}$$

$$x_{glak} \leq z_{gl} \qquad \forall g \in \mathcal{G}_s; l \in \mathcal{L}_s; a \in \mathcal{A}; k \in \mathcal{K} : F_{lg} = 1; C_{al} = 1 \tag{B.1e}$$

$$\sum_{l \in \mathcal{L}_s : F_{lg}=1} \sum_{a \in \mathcal{A} : C_{al}=1} a \cdot x_{glak} \leq \sum_{a \in \mathcal{A}} v_{ga} \qquad \forall g \in \mathcal{G}_s; k \in \mathcal{K} \tag{B.1f}$$

$$v_{ga} \leq v_{gb} \qquad \forall g \in \mathcal{G}_s; a \in \{2, \ldots, A\}; b = a - 1 \tag{B.1g}$$

$$\sum_{l \in \mathcal{L}_s : F_{lg}=1} \left(\sum_{a \in \mathcal{A}^{\text{small}} : C_{al}=1} x_{glak} + \sum_{a \in \mathcal{A}^{\text{large}} : C_{al}=1} 2 \cdot x_{glak} \right) \leq 2 \quad \forall g \in \mathcal{G}_s; k \in \mathcal{K} \tag{B.1h}$$

$$\sum_{g \in \mathcal{G}_s : F_{lg}=1} x_{glak} + \sum_{h \in \mathcal{G}_s : F_{mh}=1} x_{hmbk} \leq 1 \quad \forall l, m \in \mathcal{L}_s; a, b \in \mathcal{A}; k \in \mathcal{K} : E_{lmab} = C_{al} = C_{bm} = 1; l < m \tag{B.1i}$$

$$v_{ga} = H_{ga} \quad \forall g \in \mathcal{G}_s; a \in \mathcal{A} : \sum_{\hat{a} \in \mathcal{A}} H_{g\hat{a}} > 0 \tag{B.1j}$$

$$v_{ga} \in \{0, 1\} \quad \forall g \in \mathcal{G}_s; a \in \mathcal{A} \tag{B.1k}$$

$$x_{glak} \in \{0, 1\} \quad \forall g \in \mathcal{G}_s; l \in \mathcal{L}_s; a \in \mathcal{A}; k \in \mathcal{K} : F_{lg} = 1; C_{al} = 1 \tag{B.1l}$$

$$y_l \in \{0, 1\} \quad \forall l \in \mathcal{L}_s \tag{B.1m}$$

$$u_{gl} \in \{0, 1\} \quad \forall g \in \mathcal{G}_s; l \in \mathcal{L}_s : F_{lg} = 1 \tag{B.1n}$$

In contrast to Model (3.1a)-(3.1o), D_{ak} is substituted by parking patterns $\mathcal{P}_{cs}$. The q_{ak} variables are eliminated, because the optimal value of z_1 has been determined already and all aircraft contained in parking patterns $\mathcal{P}_{cs}$ have to be parked at contact gates. Demand Constraints (3.1c) are changed to Constraints (B.1b) accordingly. Sets $\mathcal{G}$ and $\mathcal{L}$ are replaced throughout the model by the area-specific subsets $\mathcal{G}_s$ and $\mathcal{L}_s$.

B.5 Algorithm to compute lower bounds for demand decompositions in brownfield scenarios

Let parameter $\Omega_{sa} \in \mathbb{N}$ denote the number of existing gates in subarea $s \in \mathcal{S}$ that are equipped to handle aircraft of class $a \in \mathcal{A}$. Algorithm B.1 shows how we match the results from Algorithm 3.2 with Ω_{sa} and adapt the values of ω_{csa} accordingly if necessary.

According to Constraints (3.1p), already existing gates must not be changed in a brownfield scenario. Hence, ω_{csa} cannot be smaller than Ω_{sa} in any area $s \in \mathcal{S}$ and any aircraft class $a \in \mathcal{A}$. If ω_{csa} has to be increased for that reason and $a > 1$, it follows from the downward compatibility of gates that $\omega_{cs\hat{a}}$ can be reduced for a smaller aircraft class $\hat{a} < a$. Again, we explicitly consider the MARS mode. That is, if ω_{csa} is increased by 1 for an aircraft class $a \in \mathcal{A}^{\text{large}}$, $\omega_{cs\hat{a}}$ for an aircraft class $\hat{a} \in \mathcal{A}^{\text{small}}$ can be reduced by 2.

Algorithm B.1 Algorithm to determine the value of LB_{cs} in brownfield scenarios

```
1: Initialize Ω_sa and integer counter variable f
2: Perform Algorithm 3.2, lines 1 to 19
3: for all aircraft classes a ∈ A do
4:     Ω_sa ← Σ_{g∈G_s} H_ga
5: end for
6: for all aircraft classes a ∈ A (in descending order) do
7:     while Ω_sa > ω_csa do
8:         ω_csa ← (ω_csa + 1)
9:         if a ∈ A^small : a > 1 then
10:            for all a' ∈ A^small : a' < a (in descending order) do
11:                if ω_csa' > 0 then
12:                    ω_csa' ← (ω_csa' − 1)
13:                    break
14:                end if
15:            end for
16:        else if a ∈ A^large then
17:            f ← 2
18:            for all a' ∈ A^large : a' < a (in descending order) do
19:                if ω_csa' > 0 then
20:                    ω_csa' ← (ω_csa' − 1)
21:                    f ← 0
22:                    break
23:                end if
24:            end for
25:            for all a' ∈ A^small : a' < a (in descending order) do
26:                if f > 0 and ω_csa' > 1 then
27:                    ω_csa' ← (ω_csa' − 2)
28:                    f ← 0
29:                    break
30:                else if f > 0 and ω_csa' > 0 then
31:                    ω_csa' ← (ω_csa' − 1)
32:                    f ← (f − 1)
33:                end if
34:            end for
35:        end if
36:    end while
37: end for
38: LB_cs ← Σ_{a∈A} a · ω_csa
39: return LB_cs
```

B.6 Algorithm to compute consistency of a demand decomposition

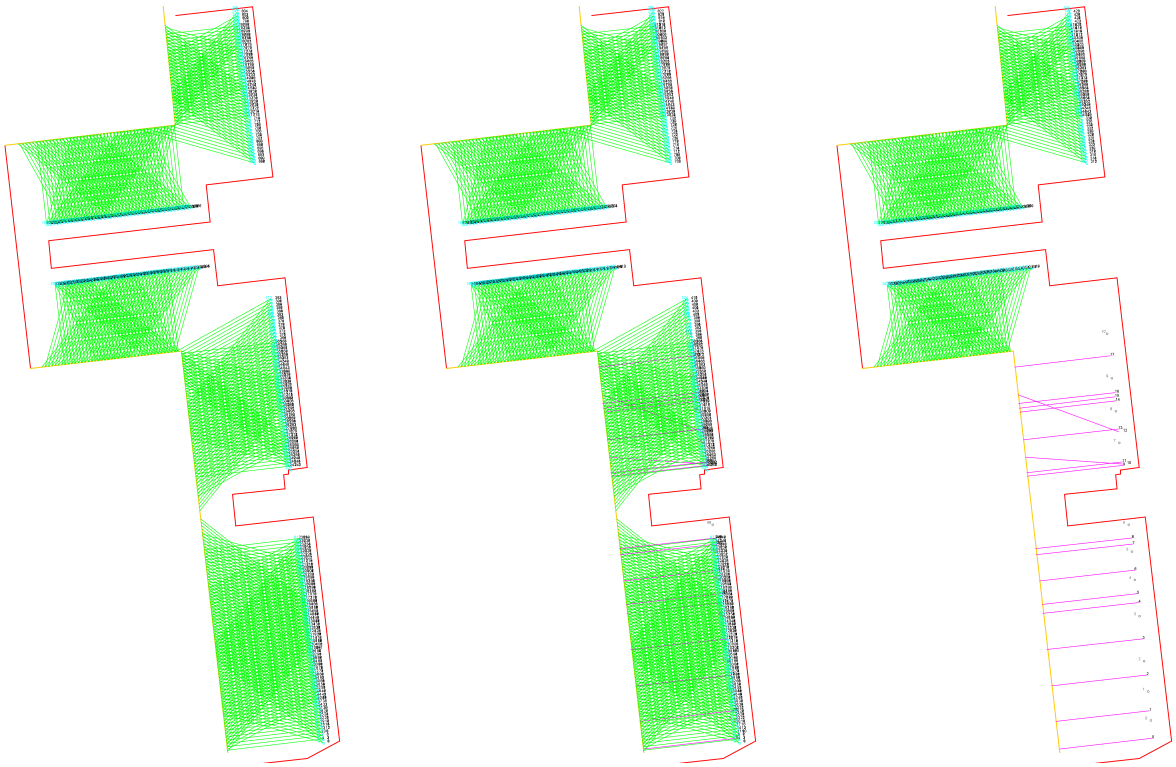
Let $n_{sk}^{a'} \in \mathbb{N}$ be defined as the aggregated number of aircraft belonging to classes $a', \ldots, A \in \mathcal{A}$ that must be parked at area $s \in \mathcal{S}$ for demand pattern $k \in \mathcal{K}$ when aircraft are assigned to areas according to demand decomposition $c \in \mathcal{C}$. The procedure we employ to determine whether a demand decomposition is consistent or not is provided in Algorithm B.2.

Algorithm B.2 Algorithm to determine whether a demand decomposition is consistent or not

```
Initialize n_sk^a
for all aircraft classes a_min ∈ A^large do
    n_sk^{a_min} ← Σ_{a∈A^large : a≥a_min} r_{ap(c,s,k)}
    for all demand patterns k_1 ∈ K do
        for all demand patterns k_2 ∈ K : k_2 > k_1 do
            if Σ_{a∈A^large : a≥a_min} D_{ak_1} ≤ Σ_{a∈A^large : a≥a_min} D_{ak_2} then
                if n_{sk_1}^{a_min} > n_{sk_2}^{a_min} then
                    return false        ▷ demand decomposition is not consistent
                end if
            end if
        end for
    end for
end for
return true                 ▷ demand decomposition is consistent
```

B.7 Planning scenarios for Munich Airport Terminal 1

The following figures illustrate the sets of gates $\mathcal{G}$ and lead-in lines $\mathcal{L}$ for Munich Airport Terminal 1 in the different planning scenarios, with $\Delta = 5\text{m}$ and $\kappa = 22.5°$. Figure B.3a depicts the greenfield scenario, Figure B.3b shows the soft brownfield case, and Figure B.3c displays the true brownfield instance. In all figures, physical obstacles that must not be touched by aircraft at any time are visualized by red lines, taxiways are represented by yellow lines. Furthermore, existing gates are marked by gray circles, all other gates are represented by cyan circles. Finally, existing lead-in lines are represented by purple lines, all other lead-in lines by green lines.

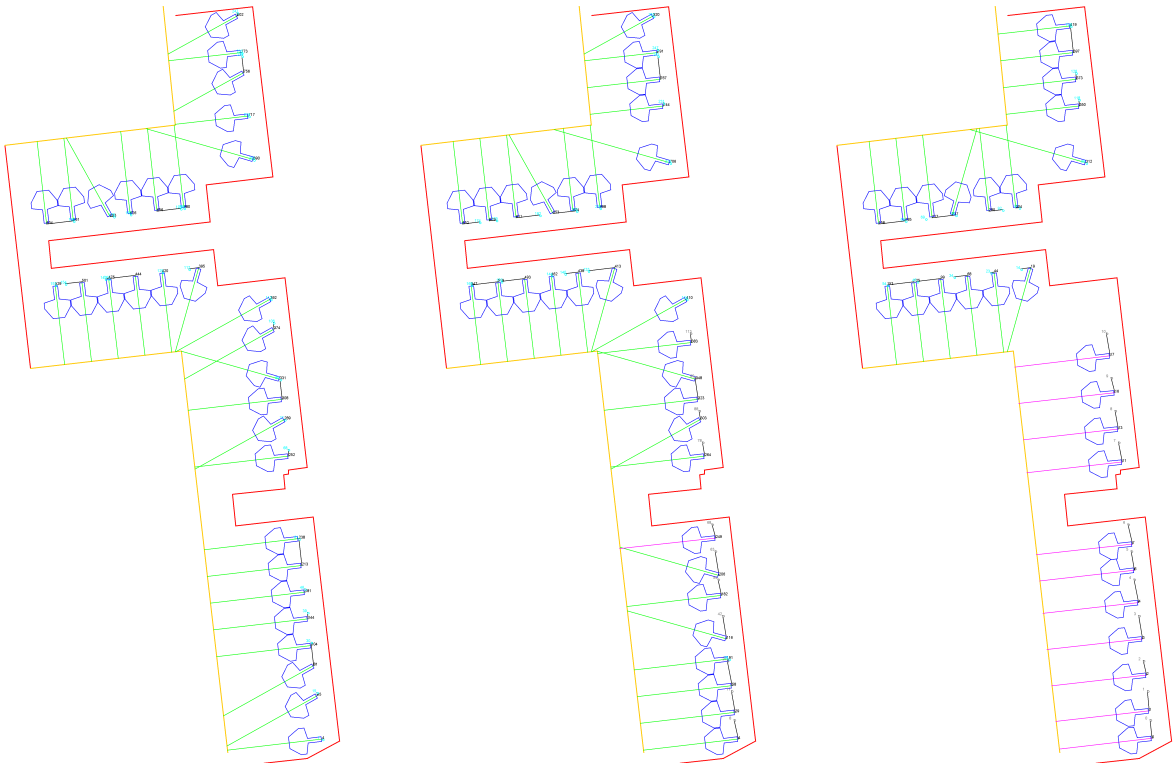


(a) Greenfield (b) Soft brownfield (c) True brownfield

Figure B.3: Lead-in lines and gates at Munich Airport Terminal 1 in different planning scenarios

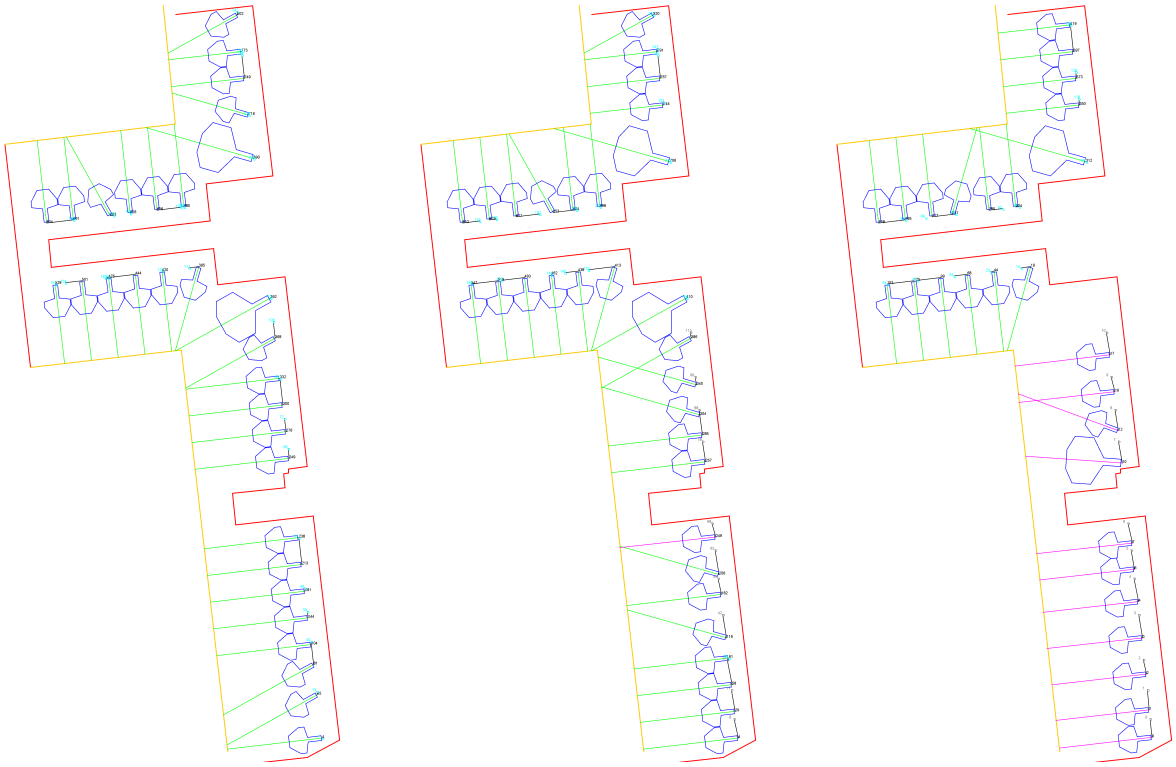
B.8 Optimal layouts for Munich Airport Terminal 1, demand patterns 1-4

The following figures are analogous to Figure 3.6 and provide the optimal layouts of all planning scenarios for demand patterns 1 to 4.



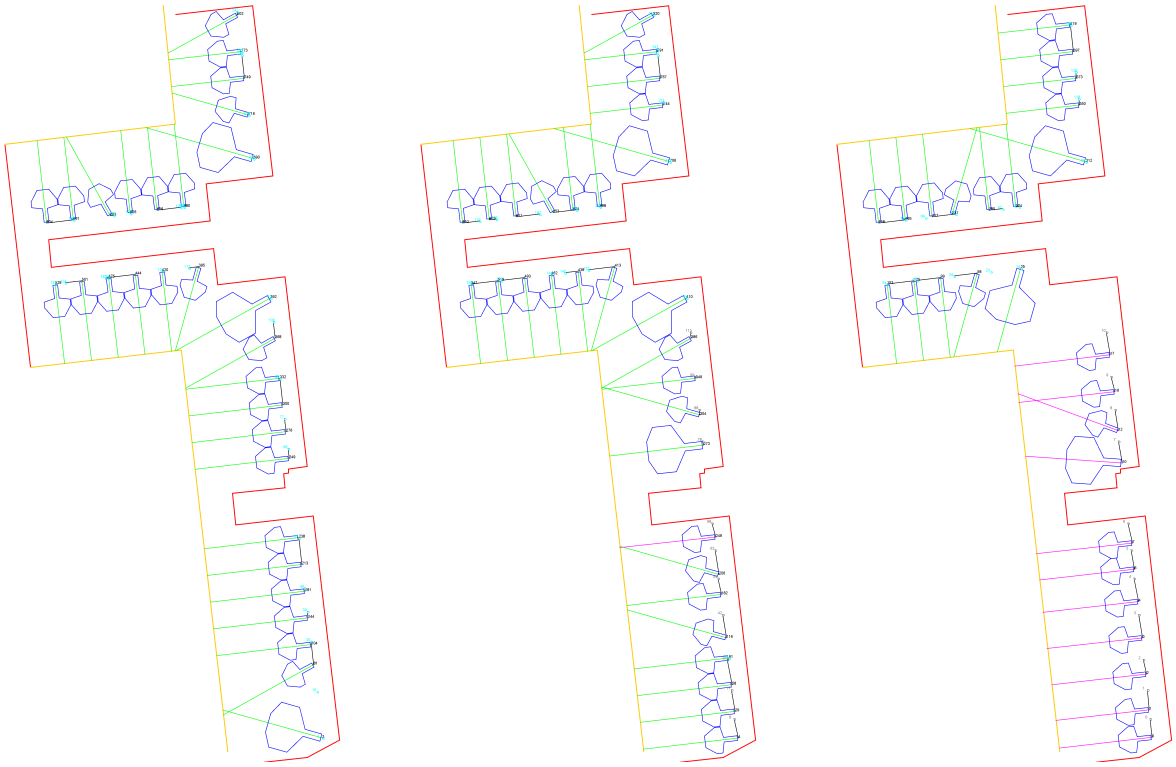
(a) Greenfield (b) Soft brownfield (c) True brownfield

Figure B.4: Optimal solutions for different planning scenarios, demand pattern 1



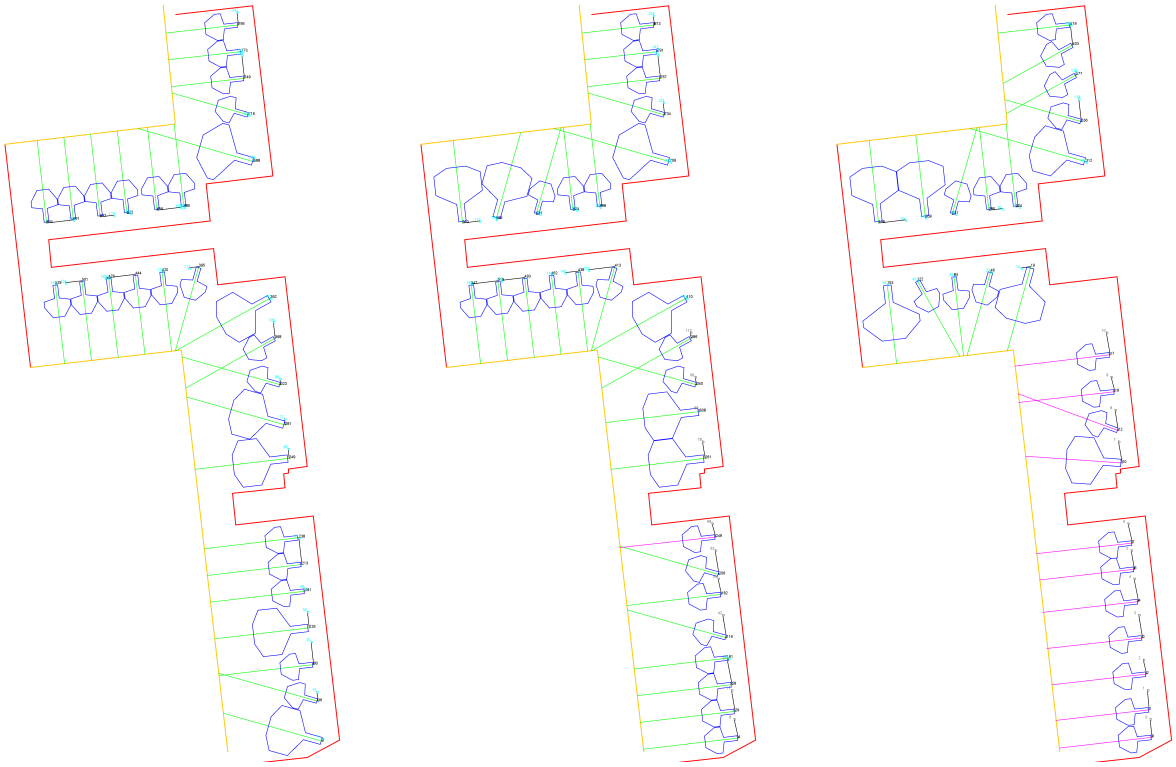
(a) Greenfield (b) Soft brownfield (c) True brownfield

Figure B.5: Optimal solutions for different planning scenarios, demand pattern 2



(a) Greenfield (b) Soft brownfield (c) True brownfield

Figure B.6: Optimal solutions for different planning scenarios, demand pattern 3



(a) Greenfield (b) Soft brownfield (c) True brownfield

Figure B.7: Optimal solutions for different planning scenarios, demand pattern 4

C Appendices Gate Configurations

C.1 Notation

Table C.1: Notation for DGCP

Sets, parameter	
$\mathcal{T} = \{0, \ldots, T\}$	Set of time periods
$\mathcal{L} = \{0, \ldots, L\}$	Set of PBB configurations
$\mathcal{A} = \{1, \ldots, A\}$	Set of aircraft classes
$\mathcal{G} = \{1, \ldots, G\}$	Set of aircraft stands
$\mathcal{J} = \{1, \ldots, J\}$	Set of lead-in lines
$\mathcal{B} = \{1, \ldots, B\}$	Set of demand patterns
$\mathcal{L}^{MARS}$	Subset of $\mathcal{L}$ containing all PBB configurations that indicate a MARS configuration
$\mathcal{L}_a$	Subset of $\mathcal{L}$ containing all PBB configurations that can be used to accommodate aircraft class $a \in \mathcal{A}$
$\mathcal{J}_g$	Subset of $\mathcal{J}$ containing all lead-in lines belonging to gate $g \in \mathcal{G}$
$\mathcal{N}_g$	Subset of $\mathcal{G}$ containing gate $g \in \mathcal{G}$ and its dexter neighbor gate
$\mathcal{B}_t$	Subset of $\mathcal{B}$containing all demand patterns of period $t \in \mathcal{T}$
c^{i}_{kl}	Investment cost of PBB configuration $l \in \mathcal{L}$, if PBB configuration $k \in \mathcal{L}$ is currently installed
c^{o}_{l}	Operating cost of PBB configuration $l \in \mathcal{L}$
c^{-}	Penalty cost per passenger
r_t	Available investment budget in period $t \in \mathcal{T}$

e_{gtl}	1, if aircraft stand $g \in \mathcal{G}$ has PBB configuration $l \in \mathcal{L}$ installed in period $t = 0$; 0, otherwise
d_{ij}	Lateral distance between lead-in lines $i \in \mathcal{J}$ and $j \in \mathcal{J}$
D_{ba}	Demand of aircraft class $a \in \mathcal{A}$ according to demand pattern $b \in \mathcal{B}$
f_{tb}	Relative weight of demand pattern $b \in \mathcal{B}$ in period $t \in \mathcal{T}$
p_a	Average number of passengers on board an aircraft of class $a \in \mathcal{A}$
s_a	Maximum half wingspan of aircraft belonging to class $a \in \mathcal{A}$ including additional space for minimum separation requirements
M	Sufficiently large number
Decision variables	
$z_{gtl} \in \{0, 1\}$	1, if PBB configuration $l \in \mathcal{L}$ is installed at aircraft stand $g \in \mathcal{G}$ in period $(t, t+1)$ for $t = 0, \ldots, T-1$; 0, otherwise
$x_{gtkl} \in \{0, 1\}$	1, if PBB configuration $l \in \mathcal{L}$ is built at aircraft stand $g \in \mathcal{G}$ starting with PBB configuration $k \in \mathcal{L} \setminus \{l\}$ at time $t = 0, \ldots, T-1$; 0, otherwise
$y_{gjtba} \in \{0, 1\}$	1, if aircraft stand $g \in \mathcal{G}$ (lead-in line $j \in \mathcal{J}$) is used for aircraft of class $a \in \mathcal{A}$ in period $(t, t+1)$ for demand pattern $b \in \mathcal{B}_t$; 0, otherwise
$y^-_{tba} \geq 0$	Uncovered demand in period $(t, t+1)$ for $t = 0, \ldots, T-1$ for aircraft of class $a \in \mathcal{A}$ for demand pattern $b \in \mathcal{B}_t$

Table C.2: Notation for solution approach

Sets, parameter	
$\mathcal{P} = \{1, \ldots, P\}$	Set of paths
$\mathcal{P}_g$	Subset of $\mathcal{P}$ containing all paths associated with gate $g \in \mathcal{G}$
c_p	Total (investment and operating) cost of path $p \in \mathcal{P}$
z^p_{gtl}	1, if PBB configuration $l \in \mathcal{L}$ is installed at aircraft stand $g \in \mathcal{G}$ in period $(t, t+1)$ for $t = 0, \ldots, T-1$ in path $p \in \mathcal{P}$; 0 otherwise
$\pi^{\text{cap,line}}_{jtb}$	Dual variable of the respective (lead-in line) capacity constraint in the master problem
$\pi^{\text{cap,gate}}_{tb}$	Dual variable of the respective (gate) capacity constraint in the master problem
π^{com}_{jtba}	Dual variable of the respective compatibility constraint in the master problem
π^{con}	Dual variable of the respective convexity constraint in the master problem
Decision variables	
$\lambda_p \in \{0, 1\}$	1, if path $p \in \mathcal{P}$ is used; 0, otherwise

C.2 Allocation of demand patterns to time periods

Depending on the number of gates G, we pre-define a set of $B = 10$ demand patterns as

$$D_{ba} = \begin{pmatrix} 1.2 \cdot G & 0 & 0 \\ G & 0.1 \cdot G & 0 \\ 0.9 \cdot G & 0.05 \cdot G & 0.05 \cdot G \\ 0.9 \cdot G & 0 & 0.1 \cdot G \\ 0.5 \cdot G & 0.4 \cdot G & 0 \\ 0.5 \cdot G & 0.25 \cdot G & 0.05 \cdot G \\ 0.5 \cdot G & 0.2 \cdot G & 0.1 \cdot G \\ 0.5 \cdot G & 0 & 0.2 \cdot G \\ 0.25 \cdot G & 0.25 \cdot G & 0.25 \cdot G \\ 0.25 \cdot G & 0.15 \cdot G & 0.35 \cdot G \end{pmatrix}.$$

The matrix is defined such that a higher value of b corresponds with a higher share of large and very large aircraft. We allocate the demand patterns to time periods according to the following scheme:

- If $t \leq \frac{T}{3}$: Randomly allocate three demand patterns from the subset of demand patterns in $\mathcal{B}$ with index $[0; 4]$ to time period t
- Else if $t > \frac{T}{3}$ and $t \leq \frac{2 \cdot T}{3}$: Randomly allocate three demand patterns from the subset of demand patterns in $\mathcal{B}$ with index $[3; 7]$ to time period t
- Otherwise: Randomly allocate three demand patterns from the subset of demand patterns in $\mathcal{B}$ with index $[6; 9]$ to time period t

C.3 Detailed results for case study

In the following we present detailed results for $c^- = 0.025$ and $r = 32$. Table C.3 provides the PBB configurations for each gate $g \in \mathcal{G}$ and time period $t \in \mathcal{T}$. PBB configuration changes are highlighted for visibility. When the PBB configuration of a gate is downgraded, the respective cell is dyed gray. When it is upgraded, the cell is enframed.

Table C.3: Resulting PBB configurations at Munich Airport for $c^- = 0.025$ and $r = 32$

	t																				
g	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
29	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
30	3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
31	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
33	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
34	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
35	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
36	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
37	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
38	4	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
39	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
40	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
41	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
42	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
43	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
44	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
45	4	4	4	4	4	4	4	4	4	4	2	2	2	2	2	2	2	2	2	2	2
46	4	4	4	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3