Runway Scheduling During Winter Operations

Models, Methods, and Applications

Maximilian Pohl

TECHNISCHE UNIVERSITÄT MÜNCHEN

Lehrstuhl für Operations Management

Runway Scheduling During Winter Operations

—

Models, Methods, and Applications

Maximilian Reinhard Pohl, M.Sc.

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

Doktors der Wirtschaftswissenschaften
(Dr. rer. pol.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr. Maximilian Schiffer
Prüfer der Dissertation:	1. Prof. Dr. Rainer Kolisch
	2. Prof. Christian Artigues, Ph.D. LAAS-CNRS, Toulouse, France

Die Dissertation wurde am 09.12.2019 bei der Technischen Universität München eingereicht und durch die TUM School of Management am 15.04.2020 angenommen.

Acknowledgments

With this thesis, an enjoyable time in school and university comes to an end. I am grateful to all my teachers, professors, mentors, and friends who, over the years, sparked my interest in and passion for knowledge and science.

Amazing kindergarten and school teachers encouraged me to find out how things work. Inspiring professors and colleagues motivated me to work on new things. My advisor Rainer Kolisch and my colleagues at the university provided an excellent environment for my research. Christian Artigues and his team made my research stay in Toulouse fantastic and unforgettable.

With your encouraging support, you made this dissertation possible.

Thank you.

Munich, December 2019

Abstract

This dissertation addresses the runway scheduling problem under consideration of winter operations from an operations research perspective. During snowfall, runways have to be temporarily closed in order to clear them from snow, ice, and slush. This thesis presents two exact optimization approaches, including mathematical model formulations and adequate solution methodologies, to simultaneously plan snow removals for multiple runways and to assign runways and take-off and landing times to aircraft. Both approaches solve the static and deterministic version of this optimization problem.

The first approach uses a time-continuous model formulation and mixed-integer programming techniques. Pruning rules and valid inequalities improve the computational tractability of the model. Additionally, a start heuristic derives initial start solutions for the branch-and-bound procedure.

The second approach uses a time-discrete binary model formulation based on clique inequalities and an equivalent constraint programming model. A start heuristic based on constraint programming generates a feasible initial start solution. A column generation scheme initialized with the heuristic solution identifies all variables of the binary program which are required to solve it optimally. Finally, a branch-and-bound procedure solves the resulting binary program. For this time-discrete variant of the winter runway scheduling problem, an enhanced time discretization method is presented to balance model size and solution quality.

Both optimization approaches are applied to real-world instances from a large international airport. Optimal schedules are compared to solutions of a practice-oriented benchmark heuristic. A computational study shows that both proposed optimization methodologies outperform the practice-oriented heuristic as well as optimization approaches relying solely on constraint programming techniques. They significantly reduce weighted aircraft delay and compute very good, and often optimal, runway schedules within a few seconds. Regarding the time-continuous model, a detailed analysis of computational times shows the efficiency of the proposed pruning rules, the valid inequalities, and the start heuristic. For the time-discrete variant, an analysis of resulting model sizes proves the ability of the presented approach to significantly reduce the number of required variables and constraints of the time-indexed binary program.

Based on the results of this work, this thesis provides several managerial insights for decision makers at airports and air traffic control. It shows that naive scheduling approaches are not suitable for an application in practice. The thesis outlines guiding principles which help human planners to generate good snow removal sequences manually. Additionally, this work provides recommendations for an application of the proposed algorithms in practice. For most instances, the time-continuous mixed-integer programming approach computes optimal solutions very fast and is suitable for airports with up to two runways or if linear cost functions are assumed. The time-discrete binary program and the corresponding column generation scheme, however, outperform the time-continuous formulation for airports with at least three runways and if non-linear cost functions are applied.

This dissertation concludes with an outlook suggesting potential future research directions.

Contents

List of Tables

List of Figures

1 Introduction

Runway availability is a scarce resource at many major international airports. This situation will presumably intensify in the future since the number of operated aircraft and, therefore, corresponding flight movements is expected to almost double until 2040 (Boeing, 2019). This continues to put enormous pressure on airport infrastructure and existing runway systems in particular. During snowfall, runway availability is further limited as runways have to be intermittently closed in order to clear them from snow, ice, and slush. Thus, it is crucial for airport operators and air traffic control to utilize the available runway capacity in the best possible way.

This dissertation thesis proposes models and corresponding solution methods to solve the problem of assigning runways as well as take-off and landing times to aircraft during winter operations while simultaneously planning snow removal activities on runways considering the scarcity of snow removal equipment. This is a complex decision problem since human planners, i.e., the responsible air traffic controllers and runway managers, have to make decisions fast, taking into account a frequently changing environment.

- The set of aircraft to be considered often changes since aircraft are constantly taking-off or landing and new aircraft are entering the near-terminal airspace.
- Attributes of aircraft such as target take-off or landing times, time windows, and cost functions for delay are aircraft-specific and can change over time.
- The planning of snow removal activities is highly dependent on the current weather and the weather forecast as well as on the availability of snow removal equipment.

To account for changing and evolving decision variables and parameters, especially for modifications in the set of considered aircraft, a recalculation of the optimal solution is necessary approximately once per minute. Due to the large number of considered aircraft, runways, and snow removal groups and the crucial role of the time dimension, the possible solution space is extremely large, which further complicates the planning task. Additionally, the multitude of involved stakeholders, such as passengers, airlines, air traffic control, and the airport operator, require a fair, transparent, and traceable decision. In practice, air traffic controllers and runway managers conduct the aircraft scheduling process manually based on a First-Come-First-Served (FCFS) principle. The decision when and in which order runways are closed for snow removal is based on human experience and intuition instead of data-driven mathematical optimization. To facilitate the required planning activities for aircraft and snow removals, this thesis proposes optimization based methods for the underlying scheduling tasks.

1.1 Scope and Scientific Contribution

This dissertation contributes to the field of airport operations by presenting the first integrated models which simultaneously schedule departing and arriving aircraft and snow removals on runways. While the first models investigating runway scheduling date back to the 1970ies (cf. Dear, 1976; Psaraftis, 1980), the integrated consideration of aircraft scheduling and snow removal planning on runways has not been studied in the scientific literature so far.

This thesis presents two exact approaches to model and solve the static and deterministic version of this winter runway scheduling problem (WRSP). The first approach models the problem as a time-continuous mixed-integer program (MIP). Pruning rules are used to presolve and fix binary variables during preprocessing, and problem specific valid inequalities are applied to exploit compulsory precedence relations between aircraft and snow removals. The second approach combines constraint programming (CP) concepts with a column generation scheme. For that, the problem is modeled as a time-discrete binary program (BP) based on clique inequalities and as a CP model. A start heuristic based on the CP formulation generates a feasible initial start solution. A column generation scheme initialized with that start solution identifies all variables of the BP which are required to solve it optimally. Finally, a branch-and-bound procedure computes the optimal solution of the resulting BP. For this time-discrete variant of the WRSP, an enhanced time discretization method is proposed, which balances model size and solution quality.

In this thesis, the applicability, effectiveness, and computational efficiency of both approaches is shown through a detailed computational study. Both approaches are applied to real-world data from Munich International Airport and thoroughly evaluated regarding their ability to compute optimal solutions and with respect to their performance in terms of computational times. Additional analyses of model sizes of the BP formulation and potential model size reductions complete the computational study.

The methodologies proposed in this dissertation enable planners to solve real-world instances of the WRSP to optimality. The presented optimal approaches significantly outperform a practice-oriented heuristic in terms of overall weighted delay cost. To derive managerial insights, this thesis compares proven optimal solutions against schedules of practice-oriented heuristics applied by human planners. It also discusses the practical applicability of the presented algorithms.

This dissertation is based on two working papers. Pohl et al. (2019b) investigate the time-continuous approach and Pohl et al. (2019a) focus on the time-discrete approach for the WRSP.

1.2 Structure and Outline

The remaining part of this dissertation is structured as follows. Chapter 2 details the characteristics of the WRSP. Chapter 3 discusses previous work and related research articles and positions this thesis within the existing literature. In Chapter 4, a time-continuous solution approach for the WRSP is presented. This includes a mathematical MIP model, pruning rules, valid inequalities, and a heuristic to derive initial start solutions. A time-discrete solution approach for the WRSP is proposed in Chapter 5. This time-discrete approach includes two model formulations, a BP and a CP model, and combines CP techniques with a column generation scheme. This chapter also introduces an enhanced time discretization method for the time-discrete BP. Chapter 6 applies both presented approaches, the time-continuous as well as the time-discrete variant, to real-world data from Munich International Airport. A computational study proves that both approaches outperform a human planner and investigates advantages and drawbacks of both variants. For the time-continuous model, this includes an analysis of the efficiencies of the proposed pruning rules, valid inequalities, and the start heuristic. For the time-discrete variant, the computational study analyzes resulting model sizes and proves the ability of the presented method to significantly reduce the number of required variables and constraints of the binary program. It also highlights the value of the proposed enhanced time discretization method. Furthermore, the computational study compares the time-continuous and the time-discrete solution approach regarding their computational times and their applicability in real-world settings. Chapter 7 concludes this thesis with managerial insights and an outlook including potential further research opportunities.

2 The Winter Runway Scheduling Problem

The WRSP solves the problem of scheduling aircraft and snow removals on multiple airport runways while minimizing earliness and tardiness cost for aircraft. This thesis considers an operational planning horizon for the WRSP of up to two hours. Specifically, the WRSP schedules aircraft which depart or arrive between the next 15 to 120 minutes. At airports, air traffic controllers assign runways and take-off and landing times to departing aircraft currently on the apron or at gate positions and to arriving aircraft in the near-terminal airspace currently approaching the airport. In the last phase of the take-off or landing preparation of an aircraft, approximately 15 minutes before departure or arrival, changing an aircraft's runway assignment or its position in the runway sequence is usually not permitted anymore due to safety regulations. In addition, modifications of runway assignments and aircraft sequences are usually impossible if arriving aircraft have entered their final landing trajectory or if departing aircraft are on their way to the assigned runway or lined up in the aircraft queue waiting for departure. Depending on the airport, this final freeze period in which runway sequences remain fixed can be shorter or longer.

2.1 Aircraft Time Windows

Assigned take-off or landing times for aircraft have to respect aircraft-specific time windows and adhere to earliest and latest possible take-off or landing times. For departing aircraft, the earliest possible take-off time is given by the time at which ground operations and taxiing to the runway can be finished and the aircraft can be ready for take-off. A reliable earliest take-off time can usually be predicted one to two hours in advance once the aircraft arrived at the departure gate. The latest possible take-off time is theoretically unrestricted with later departure times leading to ever increasing delay. For arriving aircraft, the earliest possible landing time is given by the shortest flight path to the airport and the maximum flight speed of the aircraft. It can reliably be calculated once the aircraft is airborne. Latest possible landing times are imposed by limited fuel, airport opening times, and regulations regarding working hours of flight personnel. Associated with each aircraft is a preferred target take-off or landing time within the aircraft's time window. This target time reflects the most economical flight path and flight speed for arriving aircraft and standard ground operations for departing aircraft. A deviation from this target time constitutes earliness or tardiness and causes aircraft-specific earliness or tardiness cost. The objective of the WRSP is to minimize the sum of these earliness and tardiness cost over all aircraft and, thus, to minimize the overall weighted delay cost of the schedule.

2.2 Separation Requirements

Aircraft scheduled on the same runway have to follow minimum separation requirements to comply with safety regulations imposed by the Federal Aviation Administration (FAA) and the International Civil Aviation Organization (ICAO). Separation requirements also apply for interdependent runways which are in close proximity to each other or cross each other and, thus, cannot be operated independently. Required separation times between aircraft mainly depend on the wake turbulences caused by the leading aircraft and their impact on following trailing aircraft. Therefore, these separation times are sequence-dependent and based on the operation classes of the involved aircraft, i.e., their weight classes ("Small", "Medium", "Large", "Boeing 757"[1], "Heavy", or "Super"), operation modes ("Take-off" or "Landing") and relative positions ("Leading" or "Trailing"). The models in this thesis consider separation requirements not only for aircraft which directly follow each other, but for all pairs of aircraft on the same runway, which Beasley et al. (2000) defined as complete separation. Table 2.1 shows separation requirements for all combinations of aircraft operation classes usually

[1]Boeing 757 aircraft are handled as heavy aircraft if they are leading and as large aircraft if they are trailing. Therefore, they constitute an own aircraft class with regard to separation requirements (FAA, 2017).

Table 2.1: Separation requirements based on aircraft operation classes according to FAA (2017) (in seconds)

	Trailing							
	Landing				Take-off			
Leading	Large	Boeing 757	Heavy	Super	Large	Boeing 757	Heavy	Super
Landing								
Large	69	69	60	60	75	75	75	75
Boeing 757	157	157	96	96	75	75	75	75
Heavy	157	157	96	96	75	75	75	75
Super	180	180	120	120	180	180	120	120
Take-off								
Large	60	60	60	60	60	60	60	60
Boeing 757	60	60	60	60	120	120	90	90
Heavy	60	60	60	60	120	120	90	90
Super	180	180	120	120	180	180	120	120

operated at large international airports including weight classes from “Large” to “Super”. The data in the table indicate that mixed runway operations, i.e., an alternation of landing and departing aircraft on the same runway, often lead to smaller separation times and, therefore, to runway schedules with higher throughput and less delay. Airports with only a few runways or operating at their capacity limit often use a mixed runway operations mode to increase the capacity of their runway system. Compared to a segregated runway operations mode where specific runways are reserved only for take-offs or landings, mixed runway operations increase the complexity of the scheduling problem and motivate the application of a mathematical optimization approach.

2.3 Snow Removal

During winter operations, runways regularly have to be cleared from snow, ice, and slush to enable safe flight operations. If snow starts piling up on a runway, flight operations continue as long as safe take-offs and landings can be ensured. As soon as safe flight operations cannot be ensured anymore, the runway has to be closed temporarily and can only be reopened after the completion of a snow removal activity. When a runway is closed for snow removal or when safe operations can momentarily not be guaranteed, aircraft have to be delayed or reassigned to another runway. Due to advanced weather predictions and a close monitoring of the runway conditions, the exact point in time at which runways become unsafe can accurately be forecast up to two hours and, thus, is assumed to be known at the beginning of the operational planning horizon. After beginning snowfall, flight operations usually become unsafe at all runways of an airport at approximately the same time. Under continuous winter operations, i.e., if it is snowing for a longer period of several hours, the time at which flight operations on a runway become unsafe usually depends not only on the current snowfall but also on the elapsed time since the last snow removal on that runway.

For snow removal and runway de-icing, airports use dedicated snow removal groups. These snow removal groups consist of multiple snowplows and trucks, which clear a runway collaboratively in a coordinated performance. At most larger airports, the number of runways exceeds the number of snow removal groups. Thus, not all runways can be cleared at the same time and snow removal groups have to clear multiple runways sequentially. Hence, the snow removal planning for multiple runways becomes a scheduling problem itself. Snow removal schedules have to consider snow removal durations per runway and transit times between runways. In practice, clearing a runway from snow and ice usually takes around 20 minutes. The exact snow removal duration, however, depends on the runway length and the used snow removal equipment. Transit times between runways are typically sequence-dependent since distances and driving times between runways depend on the physical layout of the airport and its road network.

After a runway has been cleared by a snow removal group, it usually stays safe for at least two hours. Hence, each runway has to be cleared not more than once within the operational planning horizon of the WRSP.

This thesis considers only snow removal activities on runways. During snowfall, also taxiways, the apron, and airport roads have to be cleared from snow and ice. Snow removal on these parts of the infrastructure, however, is often a subordinated planning task and conducted by separate snow removal groups using different equipment, e.g., smaller snow plows and other de-icing liquids or gravel. Thus, it is not considered in this work.

3 Literature Review

This chapter presents related previous work and research articles and positions this thesis within the existing literature. Since the WRSP extends the classical runway scheduling problem (RSP) and has not been investigated so far, this literature review focuses primarily on existing work on the RSP and related machine scheduling problems. First, various heuristic solution approaches for the RSP are discussed. Then, exact methods including dynamic programming (DP) approaches and MIP models are presented. This chapter also reviews applications of CP for aircraft scheduling and existing methods which combine CP and column generation schemes. Since the WRSP is closely related to machine scheduling problems with sequence-dependent setup times and maintenance activities, corresponding machine scheduling literature is briefly discussed as well.

In general, the literature review focuses on recent contributions and articles most relevant for this thesis. Comprehensive overviews of publications concerning the RSP and related problems in airport arrival management, departure management, and surface management can be found in Bennell et al. (2011), Lieder & Stolletz (2016), and Samà et al. (2019). Bennell et al. (2011) compiled an extensive literature overview of articles published until 2011. Lieder & Stolletz (2016) considered more recent contributions until 2015 focusing on articles with heterogeneous (or interdependent) runways and single (or independent) runways. Samà et al. (2019) organized their literature discussion around arrival scheduling, departure scheduling, and mixed arrival-departure scheduling.

Table 3.1 compares this dissertation with existing RSP literature regarding objective functions, runway configurations and solution approaches. While most papers minimize the weighted earliness and tardiness (weigh. E&T), some articles consider only tardiness (T) or makespan. A few papers combine different cost components, e.g., fuel cost, flight cancellation cost or indirect cost for constraint violation.

Table 3.1: Comparison of this dissertation with existing literature

Reference	Objective functions	Runways	Solution approaches
RSP — heuristic approaches			
Abela et al. (1993)	min $\sum$ weigh. E&T	single	GA
Fahle et al. (2003)	min $\sum$ weigh. E&T	single	LS, CP
Hansen (2004)	min max T	multiple	GA
Díaz & Mena (2005)	min $\sum$ weigh. E&T	single	CP
Bianco et al. (2006)	min makespan, min avg. T	multiple	LS
Pinol & Beasley (2006)	min $\sum$ weigh. E&T	multiple	SS, BA
Salehipour et al. (2009)	min $\sum$ T	multiple	VNS
Liu (2010)	min $\sum$ T (squared)	multiple	GA
Salehipour et al. (2013)	min $\sum$ weigh. E&T	multiple	SA, VNS
Vadlamani & Hosseini (2014)	min $\sum$ weigh. E&T	single	ALNS
Sabar & Kendall (2015)	min $\sum$ weigh. E&T	multiple	LS
Bennell et al. (2017)	min $\sum$ weigh. cost comp. (makespan, avg. T, fuel cost, constr. violation)	single	SA
RSP — exact approaches			
Abela et al. (1993)	min $\sum$ weigh. E&T	single	TC MIP
Beasley et al. (2000)	min $\sum$ weigh. E&T	multiple	TC MIP, TD MIP
Fahle et al. (2003)	min $\sum$ weigh. E&T	single	TC MIP, TD MIP
Balakrishnan & Chandran (2010)	min makespan, min $\sum$ T, min max T	single	DP (CPS)
Furini et al. (2012)	min $\sum$ weigh. T	single	TC MIP (rol. hor.)
Samà et al. (2013)	min max T	multiple	TC MIP (rol. hor.)
Kjenstad et al. (2013)	min $\sum$ weigh. cost comp. (canceled flights, weigh. E&T)	multiple	TD MIP
Heidt et al. (2014)	min $\sum$ weigh. E&T (squared)	single	TD MIP (diff. step sizes)
Furini et al. (2014)	min $\sum$ weigh. E&T	single	MIP (CPS), DP (CPS)
Furini et al. (2015)	min $\sum$ weigh. T	single	TC MIP (rol. hor.)
Faye (2015)	min $\sum$ weigh. E&T	multiple	TD MIP
Bertsimas & Frankovich (2015)	min $\sum$ weigh. cost com. (flight holding cost, travel cost, runway config. change cost)	multiple	TD MIP
Lieder & Stolletz (2016)	min $\sum$ weigh. T	multiple	TC MIP, DP
Avella et al. (2017)	min $\sum$ weigh. cost comp. (canceled flights, weigh. E&T)	single	TD MIP
WRSP — exact approaches			
This dissertation	*min $\sum$ weigh. E&T*	*multiple*	*TC MIP, TD MIP*

3.1 Heuristic Solution Approaches

Since the general RSP is known to be NP-hard (Bianco et al., 1999), many heuristic approaches to solve the problem have been developed. Bianco et al. (2006) and Sabar & Kendall (2015) presented local search (LS) algorithms, while variable neighborhood search (VNS) and adaptive large neighborhood search (ALNS) methods were proposed by Salehipour et al. (2009) and Vadlamani & Hosseini (2014). Salehipour et al. (2013) and Bennell et al. (2017) suggested simulated annealing (SA) approaches to solve the aircraft landing problem. Various population based heuristics have been proposed as well. Abela et al. (1993), Hansen (2004), and Liu (2010) developed genetic search algorithms (GA) and Pinol & Beasley (2006) presented a scatter search (SS) and bionomic algorithms (BA) to solve the aircraft landing problem.

3.2 Exact Methods

Despite the computational complexity of the problem, several exact approaches for solving the RSP have been developed. They often succeed to compute optimal solutions for relatively small instances (up to approximately 50 aircraft on 3 runways) but frequently fail to optimally solve large instances. Existing exact approaches mainly use DP or MIP techniques.

To reduce the computational complexity of exact problem formulations, preprocessing techniques and pruning rules received more attention in recent years. A comprehensive set of preprocessing methods and pruning rules for the RSP was developed by Maere et al. (2017). They presented different pruning rules for the RSP to decrease the solution space of the problem. Their pruning rules are model-independent and generic. Thus, they can be applied to a variety of solution methods including exact DP and MIP approaches.

3.2.1 Dynamic Programming

Bennell et al. (2017) presented a DP for the single runway problem and Lieder & Stolletz (2016) developed a DP approach for multiple interdependent and heterogeneous runways. Balakrishnan & Chandran (2010) proposed DP algorithms for the single runway problem under consideration of constrained position shifting (CPS), which reduces the solution space of the DP by prohibiting large deviations from a FCFS order. Furini et al. (2014) built up on that and presented state space reduction techniques to further improve computational times for DP approaches.

3.2.2 Mixed-Integer Programming

Several MIP formulations have been developed to solve the RSP in multiple settings with different sets of assumptions. Two types of MIP formulations emerged and prevail in literature. The first model type solves the RSP in continuous time and uses big-M formulations. The second model type solves the problem in discrete time using time-indexed model formulations.

Time-Continuous (TC) MIP Formulations Model formulations solving the RSP in continuous time typically represent the take-off or landing time of an aircraft as a single continuous variable. To reflect sequence-dependent separation requirements between aircraft, they rely on big-M constraints. Abela et al. (1993) presented such a MIP formulation for the single runway case. Beasley et al. (2000) developed a MIP model for the multiple runway case, which became the basis for many succeeding model formulations and publications. Time-continuous formulations also build the basis for many rolling horizon frameworks as in Furini et al. (2012), Samà et al. (2013), and Furini et al. (2015).

The time-continuous approach presented in Chapter 4 of this thesis is based upon the model formulation in Beasley et al. (2000) and extends their work in order to reflect winter operations, i.e., snowfall and snow removal activities on runways.

Time-Discrete (TD) MIP Formulations To solve the RSP in discrete time, time-indexed model formulations represent the take-off or landing time of an aircraft by a set of binary variables. Each binary variable is associated with a feasible take-off or landing time within an aircraft's time window. In a feasible solution of the RSP, exactly one binary variable of this set equals one, indicating the aircraft's take-off or landing time. Such time-indexed model formulations typically have a larger number of variables but often return much stronger bounds than time-continuous big-M formulations. For real-world instances of the RSP with a realistic number of aircraft and runways, time-discrete formulations with time-indexed models were long considered to be computationally intractable, especially due to their enormous model sizes (cf. Beasley et al., 2000; Avella et al., 2017). However, due to increasing computational power and algorithmic advances of MIP solvers, they received more attention in recent years. Kjenstad et al. (2013) proposed a decomposition of the RSP and used a time-indexed formulation to optimally assign arrival and departure times to aircraft. Heidt et al. (2014) proposed a dynamic variant of a time-discrete model formulation, in which aircraft-specific slot sizes depend on an aircraft's distance to the runway. Faye (2015) developed a time-discrete model based on a decomposition of separation times. Bertsimas & Frankovich (2015) presented a time-discrete approach to optimize the air traffic flow through airports in a holistic way. Closest to this work is the time-indexed formulation for the RSP by Avella et al. (2017), which computes optimal schedules for departing and arriving aircraft on a single runway. For this, the authors developed a new family of clique constraints, which generalizes a family of clique inequalities presented in Nogueira et al. (2019).

The work of Avella et al. (2017) builds the foundation for the time-discrete approach proposed in Chapter 5 of this dissertation. The time-indexed model in this thesis extends the model formulation in Avella et al. (2017) in order to consider multiple runways as well as winter operations. This thesis also presents a novel solution algorithm for this model formulation combining a column generation scheme with a CP start heuristic.

3.3 Constraint Programming

With regard to CP, Allignol et al. (2012) compiled a survey of CP approaches for air traffic management, including, but not restricted to runway scheduling. Fahle et al. (2003) compared different exact and heuristic methods, including a CP model, for the aircraft sequencing problem on a single runway. For the same single runway problem, Díaz & Mena (2005) presented a CP implementation based on the Oz/Mozart framework[1]. Junker et al. (1999) and Yunes et al. (2000) independently developed a column generation method in which the pricing subproblem is solved using CP techniques. This method was applied to a wide range of applications (cf. Gualandi & Malucelli, 2013), specifically for the tail assignment problem (Grönkvist, 2006; Gabteni & Grönkvist, 2008) and the crew assignment problem (Junker et al., 1999; Fahle et al., 2002).

3.4 Related Machine Scheduling Problems

The RSP is related to machine scheduling problems with sequence-dependent processing or setup times where runways correspond to machines, aircraft correspond to jobs, and separation requirements correspond to sequence-dependent setup times. For the parallel machine scheduling problem with sequence-dependent setup times, Lee & Pinedo (1997) proposed a heuristic based on dispatching rules and simulated annealing to minimize weighted tardiness. For the same problem and the objective of minimizing weighted earliness and tardiness, Radhakrishnan & Ventura (2000) presented a simulated annealing approach. In order to minimize makespan, Vallada & Ruiz (2011) suggested a genetic algorithm and Lopes & de Carvalho (2007) developed a branch-and-price algorithm to minimize an additive cost function. An overview of scheduling problems with setup times and related articles until 2008 was compiled by Allahverdi et al. (2008). From a machine scheduling point of view, snow removal has structural similarities to preventive maintenance, especially when maintenance crews have to maintain machines within a given time window and when travel times between machines apply for crews moving from one machine to the next machine. Scheduling problems with maintenance activities have received growing interest in recent years. For the general problem of scheduling jobs and maintenance activities on parallel machines and the objective of minimizing total weighted completion time, Lee & Chen (2000) presented a branch-and-bound algorithm based on column generation. Variants of this problem with aging effects, multiple maintenance activities during the planning horizon, or maintenance activities with start-time dependent

[1] Oz is a multiparadigm programming language providing functional, object-oriented, and constraint programming concepts. The Mozart Programming System is an implementation of the Oz programming language released by the Mozart Consortium.

durations have been investigated by Cheng et al. (2011) and Yang et al. (2012). Ma et al. (2010) presented a survey of scheduling problems with deterministic machine availability constraints considering articles until 2009. Gao et al. (2006) studied the flexible job shop scheduling problem considering maintenance activities with flexible starting times and proposed a hybrid genetic algorithm to solve it. For a similar problem with scarce maintenance resources where only one machine can be maintained at any given time, Wang & Yu (2010) proposed a heuristic algorithm based on a filtered beam search. Yoo & Lee (2016) investigated different objective functions and job characteristics for the parallel machine scheduling problem with maintenance activities and developed corresponding dynamic programming algorithms. The specific variant of a parallel machine scheduling problem with sequence-dependent setup times for jobs and flexible maintenance activities has not been studied yet.

Concluding, existing literature focuses on a wide range of aspects of runway scheduling. However, the WRSP, i.e., the RSP under consideration of winter operations, has not been investigated so far. This dissertation presents the first model formulations for the WRSP and corresponding solution methodologies.

4 Time-Continuous Approach – Using Pruning Rules, Valid Inequalities and a Mixed-Integer Programming Model

This chapter details and explains the first solution approach for the WRSP based on a time-continuous MIP model. Section 4.1 introduces a deterministic model formulation considering multiple heterogeneous runways and multiple snow removal groups. The model builds upon and extends the model of Beasley et al. (2000). In Section 4.2, pruning rules are derived, which can be applied during preprocessing to improve the performance and tractability of the model. To derive better LP bounds, valid inequalities are presented in Section 4.3. Additionally, in Section 4.4, a start heuristic is developed, which regularly yields good feasible start solutions for the MIP solver and decreases computational times.

4.1 Mixed-Integer Programming Model

Using the sets and parameters defined in Table 4.1 and the decision variables defined in Table 4.2, the proposed model reflects the problem setting described in Chapter 2. In particular, the formulation assigns runways $r \in \mathcal{R}$ and take-off or landing times x_a to departing and arriving aircraft $a \in \mathcal{A}$ of operation class $c(a) \in \mathfrak{C}$. Further, it assigns snow removal groups $g \in \mathcal{G}$ and snow removal times v_r to runways $r \in \mathcal{R}$. The model formulation allows mixed runway operations, i.e., take-offs and landings can be scheduled on the same runway. The objective function of the model minimizes the delay cost of the schedule, i.e., the

Table 4.1: Sets and parameters of the time-continuous model

Notation	Definition
$a \in \mathcal{A}$	Set of aircraft
T_a	Target take-off or landing time of aircraft a
L_a	Latest possible take-off or landing time of aircraft a
C_a	Cost coefficient per time unit for scheduling aircraft a after T_a
$\mathcal{C} \in \mathfrak{C}$	Set of operation classes
$c(a)$	Operation class of aircraft a
$S_{\mathcal{C}\mathcal{C}'}$	Sequence-dependent separation time between a leading aircraft of operation class $\mathcal{C}$ and a trailing aircraft of operation class $\mathcal{C}'$ if both aircraft are scheduled on the same runway
$O_{\mathcal{C}}$	Required separation time between an aircraft of operation class $\mathcal{C}$ and a subsequent snow removal on the same runway
$g \in \mathcal{G}$	Set of snow removal groups
$r \in \mathcal{R}$	Set of runways
U_r	Time at which flight operations on runway r become unsafe
P_r	Required time to clear runway r
Q_{rs}	Sequence-dependent setup time between starts of snow removals on runways r and s conducted by the same snow removal group (including snow removal time P_r and required transit time from runway r to runway s)
$i \in \mathcal{A} \cup \mathcal{R}$	Set of activities which can either be aircraft or snow removals

Table 4.2: Decision variables of the time-continuous model

Notation	Definition
$x_a \geq 0$	Take-off or landing time of aircraft a
$v_r \geq 0$	Start time of snow removal on runway r
δ_{ij}	$= \begin{cases} 1 & \text{if activity } i \text{ starts before activity } j \\ \in \{0,1\} & \text{otherwise} \end{cases}$
y_{ar}	$= \begin{cases} 1 & \text{if aircraft } a \text{ is scheduled on runway } r \\ 0 & \text{otherwise} \end{cases}$
z_{ab}	$= \begin{cases} 1 & \text{if aircraft } a \text{ and } b \text{ are scheduled on the same runway} \\ 0 & \text{otherwise} \end{cases}$
ρ_{rg}	$= \begin{cases} 1 & \text{if snow removal on runway } r \text{ is conducted by snow removal group } g \\ 0 & \text{otherwise} \end{cases}$
ϕ_{rs}	$= \begin{cases} 1 & \text{if snow removals on runways } r \text{ and } s \text{ are conducted by the same snow removal group} \\ 0 & \text{otherwise} \end{cases}$

total weighted aircraft delay $\sum_{a\in\mathcal{A}} C_a(x_a - T_a)$ as the weighted deviation of the scheduled time x_a from the target time T_a.

The model formulation considers time windows $T_a \leq x_a \leq L_a$ for aircraft a limited by target times T_a and latest possible take-off or landing times L_a. Thus, aircraft are not allowed to be scheduled before their target time and earliest possible take-off or landing times are implicitly set as target times. The model considers sequence-dependent separation requirements $S_{c(a)c(b)}$ between pairs of aircraft a and b on the same runway. It also considers separation requirements $O_{c(a)}$ between aircraft a and a following snow removal on the same runway. This time $O_{c(a)}$ is mainly defined by the duration of the take-off or landing procedure of aircraft a. The model ensures separation times P_r between a snow removal and a following aircraft on runway r. It is defined by the duration P_r of a snow removal procedure on a specific runway r. Additionally, the model considers runway closings in case too much snow, ice, or slush has piled up on a runway and safe operations cannot be guaranteed any more. Once a runway has become unsafe at time U_r, it must be cleared before it can be reopened. If a runway r is cleared before it becomes unsafe, it stays safe for the remaining planning horizon. The model formulation schedules exactly one snow removal per runway. If a runway does not become unsafe or if a snow removal on a runway is not necessary within the planning horizon due to low aircraft traffic, the model schedules a dummy snow removal at the end of the planning horizon. For snow removal groups, the model considers sequence-dependent setup times Q_{rs} between runways r and s, which, for the ease of notation, include the time P_r to clear runway r and the transfer time from runway r to runway s. The model formulation uses the notion of activities, which can either be aircraft or snow removals, and their relative order δ_{ij} with $i, j \in \mathcal{A} \cup \mathcal{R} : i \neq j$ indicating whether i or j starts earlier.

The complete MIP formulation is as follows:

$$\textbf{minimize} \quad \sum_{a \in \mathcal{A}} C_a(x_a - T_a) \tag{4.1}$$

subject to

$$T_a \leq x_a \leq L_a \qquad \forall\, a \in \mathcal{A} \tag{4.2}$$

$$\delta_{ij} + \delta_{ji} = 1 \qquad \forall\, i, j \in \mathcal{A} \cup \mathcal{R} : i \neq j \tag{4.3}$$

$$\sum_{r \in \mathcal{R}} y_{ar} = 1 \qquad \forall\, a \in \mathcal{A} \tag{4.4}$$

$$z_{ab} \geq y_{ar} + y_{br} - 1 \qquad \forall\, a, b \in \mathcal{A} : a \neq b; r \in \mathcal{R} \tag{4.5}$$

$$x_b \geq x_a + S_{c(a)c(b)} z_{ab} - M\delta_{ba} \qquad \forall\, a, b \in \mathcal{A} : a \neq b \tag{4.6}$$

$$\sum_{g \in \mathcal{G}} \rho_{rg} = 1 \qquad \forall\, r \in \mathcal{R} \tag{4.7}$$

$$\phi_{rs} \geq \rho_{rg} + \rho_{sg} - 1 \qquad \forall\, r, s \in \mathcal{R} : r \neq s; g \in \mathcal{G} \tag{4.8}$$

$$v_s \geq v_r + Q_{rs}\phi_{rs} - M\delta_{sr} \qquad \forall\, r, s \in \mathcal{R} : r \neq s \tag{4.9}$$

$$v_r \geq x_a + O_{c(a)} y_{ar} - M\delta_{ra} \qquad \forall\, r \in \mathcal{R}; a \in \mathcal{A} \tag{4.10}$$

$$x_a \geq v_r + P_r y_{ar} - M\delta_{ar} \qquad \forall\, r \in \mathcal{R}; a \in \mathcal{A} \tag{4.11}$$

$$x_a \leq U_r + M(1 - y_{ar}) + M\delta_{ra} \qquad \forall\, r \in \mathcal{R}; a \in \mathcal{A} \tag{4.12}$$

$$x_a, v_r \geq 0 \qquad \forall\, a \in A; r \in \mathcal{R}$$

$$\delta_{ij}, y_{ar}, z_{ab}, \rho_{rg}, \phi_{rs} \in \{0, 1\} \qquad \forall\, i, j \in \mathcal{A} \cup \mathcal{R} : i \neq j;\; a, b \in \mathcal{A} : a \neq b;\; r, s \in \mathcal{R} : r \neq s; g \in \mathcal{G}$$

The Objective (4.1) minimizes the delay cost of the schedule as the sum of weighted delay over all aircraft. Constraints (4.2) secure that all flights are scheduled within their allowed time windows. Constraints (4.3) sequence all activities by deciding for each pair of activities i and j if i precedes j or vice versa. The presented model formulation and definition of decision variables δ_{ij} also allow that activities can start at the same time if they are scheduled on different runways. Constraints (4.4) secure that each aircraft is scheduled on exactly one runway. Constraints (4.5) determine whether two aircraft are scheduled on the same runway. Constraints (4.6) guarantee that all separation requirements

$S_{c(a)c(b)}$ between aircraft a and b on the same runway are met, imposing complete separation. Constraints (4.7) assign exactly one snow removal group to each runway. Constraints (4.8) determine whether two runways are cleared by the same snow removal group. Constraints (4.9) secure sufficient setup time Q_{rs} between two consecutive snow removals on runways r and s which are conducted by the same snow removal group. Constraints (4.10) secure sufficient separation time $O_{c(a)}$ between an aircraft a and a following snow removal on the same runway. Similarly, Constraints (4.11) secure sufficient separation time P_r between a snow removal and a following aircraft on the same runway r. Finally, Constraints (4.12) make sure that an aircraft is only scheduled on a runway as long as flight operations are safe, i.e., before a runway becomes unsafe or after snow removal on that runway has been completed. All big-M coefficients are sufficiently large if $M = \max_{a \in \mathcal{A}}\{L_a\}$ holds.

The model allows that runways are heterogeneous in the sense that specific combinations of aircraft operations and runways can be prohibited. This provides the possibility to model situations in which specific runways are not permitted for specific aircraft (e.g., because the runways are too short) or situations in which only take-offs or landings are allowed on certain runways. Fixing variables y_{ar} prohibits or enforces such assignments of aircraft a to runways r and decreases the complexity of the model since it prunes the branch-and-bound search tree. Snow removal groups are assumed to be identical in terms of driving speed and snow removal speed and, therefore, interchangeable. Nevertheless, it is possible to prohibit or enforce the assignments of snow removal groups g to runways r by fixing variables ρ_{rg} accordingly.

4.2 Problem-Specific Pruning Rules

Pruning rules are applied during preprocessing to improve the computational performance of the model. They precalculate and fix binary variables to zero or one in order to decrease the solution space and to accelerate the branch-and-bound process. Maere et al. (2017) give a comprehensive overview of various pruning rules for the RSP.

This section discusses three types of pruning rules. Pruning rules of type I order aircraft according to their non-overlapping time windows and were presented by

Beasley et al. (2000). Pruning rules of type II order aircraft of the same operation class and with overlapping time windows according to their target time and are an extension of a pruning rule developed by Briskorn & Stolletz (2014). Pruning rules of type III precalculate a dominant, i.e., optimal, snow removal pattern independently from actual aircraft traffic for many instances with homogeneous runways and have not been discussed in literature before.

Pruning Rules of Type I: Strict Orders of Aircraft Based on Time Windows

For some pairs of aircraft, an optimal order can be determined based on their time windows. This was proven by Beasley et al. (2000).

Theorem 1. *If the time window $[T_a, L_a]$ of aircraft a and the time window $[T_b, L_b]$ of aircraft b do not overlap with $L_a < T_b$, then aircraft a must be scheduled before aircraft b:*

$$\delta_{ab} = 1 \; \forall \, a, b \in \mathcal{A} : L_a < T_b$$

Pruning Rules of Type II: Strict Orders of Aircraft Within the Same Aircraft Operation Class

Pruning rules of type II are based on the fact that aircraft a and a' of the same operation class $\mathcal{C}$ are separation identical, i.e., they have the same pairwise separation requirements in relation to all other aircraft b: $S_{c(a)c(b)} = S_{c(a')c(b)} \wedge S_{c(b)c(a)} = S_{c(b)c(a')} \; \forall \, b \in \mathcal{A} \setminus \{a, a'\}$ since $c(a) = c(a') = \mathcal{C}$.

For the objectives of minimizing makespan and minimizing weighted delay, Psaraftis (1980) showed that, within an aircraft operation class, a complete order can be inferred under the assumptions that no time window restrictions exist and that all aircraft of this operation class have the same cost function. Briskorn & Stolletz (2014) showed that such complete orders within aircraft operation classes also exist if a time window order (see Definition 1) exists for all pairs of aircraft within a operation class and if class-specific piecewise linear convex cost functions are assumed.

Definition 1. *Time window order.* Two aircraft a and a' of the same operation class have a time window order $a \prec a'$ if $T_a < T_{a'} \Rightarrow L_a \leq L_{a'} \; \wedge \; L_a < L_{a'} \Rightarrow T_a \leq T_{a'}$.

Briskorn & Stolletz (2014) showed that scheduling two aircraft a and a' of the same operation class according to their time window order is optimal for minimizing delay cost by proving that swapping a and a' cannot improve the objective value. Different to Briskorn & Stolletz (2014), this thesis relaxes the assumption of class-specific piecewise linear convex cost functions and assumes aircraft-specific linear cost functions instead, i.e., aircraft a and a' of the same operation class are allowed to have different cost coefficients $C_a \neq C_{a'}$. In practice, these cost coefficients often depend on the number of passengers aboard an aircraft and the price segment of the carrier, e.g., premium or low cost. In order to determine aircraft orders within aircraft operation classes if cost factors are aircraft-specific, the concept of cost compliance is introduced.

Definition 2. *Cost compliance.* Two aircraft a and a' with time window order $a \prec a'$ are cost compliant if aircraft a has the same or a higher cost coefficient than aircraft a': $a \prec a' \Rightarrow C_a \geq C_{a'}$.

A strict order within a pair of aircraft a and a' of the same operation class can be precalculated if a time window order exists and the aircraft are cost compliant.

Theorem 2. *It is always optimal to schedule aircraft a and a' of the same operation class in their corresponding time window order if they are cost compliant:*

$$\begin{aligned} \delta_{aa'} = 1 \; \forall\, a, a' \in \mathcal{A}: \quad & c(a) = c(a'), && \textit{(same operation class)} \\ & T_a \leq T_{a'} \wedge L_a \leq L_{a'}, && \textit{(time window order } a \prec a'\textit{)} \\ & C_a \geq C_{a'} && \textit{(cost compliance)} \end{aligned}$$

Proof. Consider two cases:

Case 1: The time windows of aircraft a and a' do not overlap. In this case, scheduling a' before a is not feasible and a must be scheduled before a'. Note that Theorem 1 applies in this case.

Case 2: The time windows of aircraft a and a' overlap with $L_a \geq T_{a'}$. Consider a feasible aircraft schedule $\mathcal{S}$ with $x_a > x_{a'}$ (a' is scheduled before a) and assume time window order ($T_a \leq T_{a'} \wedge L_a \leq L_{a'}$) and cost compliance ($C_a \geq C_{a'}$). Swapping aircraft a and a' cannot increase the objective value: Generate a new schedule $\bar{\mathcal{S}}$ by swapping aircraft a and a' and scheduling aircraft a at $\bar{x}_a = x_{a'}$ and aircraft a' at $\bar{x}_{a'} = x_a$. This schedule $\bar{\mathcal{S}}$ is feasible with regard to other aircraft since a and a' are of the same operation class and, thus, separation identical. Schedule $\bar{\mathcal{S}}$ is also feasible with regard to possible time windows since $T_a \leq T_{a'} \leq x_{a'} = \bar{x}_a < x_a = \bar{x}_{a'} \leq L_a \leq L_{a'}$. From cost compliance (4.13) and $x_{a'} = \bar{x}_a < x_a = \bar{x}_{a'}$, it can be concluded by (4.14)–(4.19) that the objective value $O(\bar{\mathcal{S}}) = C_a(\bar{x}_a - T_a) + C_{a'}(\bar{x}_{a'} - T_{a'})$ of the new schedule $\bar{\mathcal{S}}$ where a and a' are swapped (and a is scheduled before a' according to their time window order) is better than or equal to the objective value $O(\mathcal{S}) = C_a(x_a - T_a) + C_{a'}(x_{a'} - T_{a'})$ of the original schedule $\mathcal{S}$:

$$C_a \geq C_{a'} \tag{4.13}$$

$$\Leftrightarrow \quad C_a(\bar{x}_a - x_a) \leq C_{a'}(x_{a'} - \bar{x}_{a'}) \tag{4.14}$$

$$\Leftrightarrow \quad C_a\bar{x}_a - C_a x_a \leq C_{a'}x_{a'} - C_{a'}\bar{x}_{a'} \tag{4.15}$$

$$\Leftrightarrow \quad C_a\bar{x}_a + C_{a'}\bar{x}_{a'} \leq C_{a'}x_{a'} + C_a x_a \tag{4.16}$$

$$\Leftrightarrow \quad C_a\bar{x}_a + C_{a'}\bar{x}_{a'} - C_a T_a - C_{a'}T_{a'} \leq C_{a'}x_{a'} + C_a x_a - C_a T_a - C_{a'}T_{a'} \tag{4.17}$$

$$\Leftrightarrow \quad C_a(\bar{x}_a - T_a) + C_{a'}(\bar{x}_{a'} - T_{a'}) \leq C_a(x_a - T_a) + C_{a'}(x_{a'} - T_{a'}) \tag{4.18}$$

$$\Leftrightarrow \quad O(\bar{\mathcal{S}}) \leq O(\mathcal{S}) \tag{4.19}$$

□

Pruning Rules of Type III: Dominant Snow Removal Patterns for Homogeneous Runways

For a set of homogeneous runways (see Definition 3), it is often possible to pre-calculate a snow removal pattern (see Definition 4) which dominates all other possible snow removal patterns.

Definition 3. *Homogeneous runways.* A set of runways $\bar{\mathcal{R}} \subseteq \mathcal{R}$ is considered to be homogeneous if all runways require the same amount of time for a snow removal activity ($P_r = P_{r'}\ \forall\, r, r' \in \bar{\mathcal{R}}$) and if they allow for the same operation modes, i.e., an aircraft that can take-off or land on a runway $r \in \bar{\mathcal{R}}$ can also take-off or land at all other runways $r' \in \bar{\mathcal{R}}$ with $r' \neq r$.

Definition 4. *Snow removal pattern.* A snow removal pattern $\mathfrak{P}$ consists of an $|\mathcal{R}|$-tuple of triples (r_i, g_i, e_i), formally $\mathfrak{P} := \big((r_1, g_1, e_1), (r_2, g_2, e_2), \ldots, (r_{|\mathcal{R}|}, g_{|\mathcal{R}|}, e_{|\mathcal{R}|})\big)$. Herein, triple (r_i, g_i, e_i) defines the i-th snow removal activity in non-decreasing order of snow removal start times with $r_i \in \mathcal{R}$ denoting the runway being cleared, $g_i \in \mathcal{G}$ with $\rho_{r_i g_i} = 1$ denoting the assigned snow removal group and e_i denoting the earliest possible start time of the i-th snow removal activity. Earliest possible start times e_i are computed based on preceding snow removals and sequence-dependent setup times $Q_{r_j r_i}$ with $j \leq i$ and $g_j = g_i$. Due to the order of the triples, $e_i \leq e_{i+1} \, \forall \, i \in 1, 2, \ldots, |\mathcal{R}| - 1$ holds.

Each snow removal pattern $\mathfrak{P}$ contains each runway $r \in \mathcal{R}$ exactly once since each runway is cleared exactly once within the planning horizon. Thus, only a finite number of different snow removal patterns exists. $\mathcal{P}$ denotes the finite set of all possible snow removal patterns and is bounded by $|\mathcal{P}| = |\mathcal{R}|! \cdot |\mathcal{G}|^{|\mathcal{R}|}$.

Similar to the concept of active schedules (cf. Giffler & Thompson, 1960), this thesis introduces pseudo-active snow removal patterns.

Definition 5. *Pseudo-active snow removal pattern.* A snow removal pattern $\mathfrak{P}'$ is called pseudo-active if and only if no other snow removal pattern $\mathfrak{P}$ exists in which the i-th snow removal ($i = 1, 2, \ldots, |\mathcal{R}|$) can be started (and, given homogeneous runways, also finished) earlier. Formally, $\mathfrak{P}' := \big((r'_1, g'_1, e'_1), (r'_2, g'_2, e'_2), \ldots, (r'_{|\mathcal{R}|}, g'_{|\mathcal{R}|}, e'_{|\mathcal{R}|})\big)$ is pseudo-active if $e'_i = \min_{\mathfrak{P} \in \mathcal{P}} \{e_i\} \, \forall \, i \in 1, 2, \ldots, |\mathcal{R}|$ holds.

Extending a snow removal pattern with actual start times for each snow removal activity yields a snow removal schedule.

Definition 6. *Snow removal schedule.* A snow removal schedule $\mathfrak{S}$ extends a snow removal pattern and is an $|\mathcal{R}|$-tuple of quadruples $(r_i, g_i, e_i, \sigma_i)$, formally $\mathfrak{S} := \big((r_1, g_1, e_1, \sigma_1), (r_2, g_2, e_2, \sigma_2), \ldots, (r_{|\mathcal{R}|}, g_{|\mathcal{R}|}, e_{|\mathcal{R}|}, \sigma_{|\mathcal{R}|})\big)$ where σ_i denotes the start time of the i-th snow removal.

Note that, for feasible snow removal schedules, $e_i \leq \sigma_i \, \forall \, i \in 1, 2, \ldots, |\mathcal{R}|$ holds.

A snow removal pattern is dominant if it can be extended to an optimal snow removal schedule and, thus, enables an optimal aircraft schedule. An optimal snow removal schedule must always consider actual aircraft traffic in the planning horizon and requires complete information about occurring aircraft. A dominant snow removal pattern, however, can often be precalculated in advance, independently from the occurring aircraft traffic. Finding a dominant snow removal pattern allows for fixing binary variables $\delta_{rs} \, \forall \, r, s \in \mathcal{R} : r \neq s$, $\rho_{rg} \, \forall \, r \in \mathcal{R}; g \in \mathcal{G}$, and $\phi_{rs} \, \forall \, r, s \in \mathcal{R} : r \neq s$ during preprocessing.

Theorem 3. *For homogeneous runways, a snow removal pattern* $\mathfrak{P}^* := \big((r_1^*, g_1^*, e_1^*), (r_2^*, g_2^*, e_2^*), \dots, (r_{|\mathcal{R}|}^*, g_{|\mathcal{R}|}^*, e_{|\mathcal{R}|}^*)\big)$ *is dominant, i.e., it allows at least one optimal snow removal schedule, if*

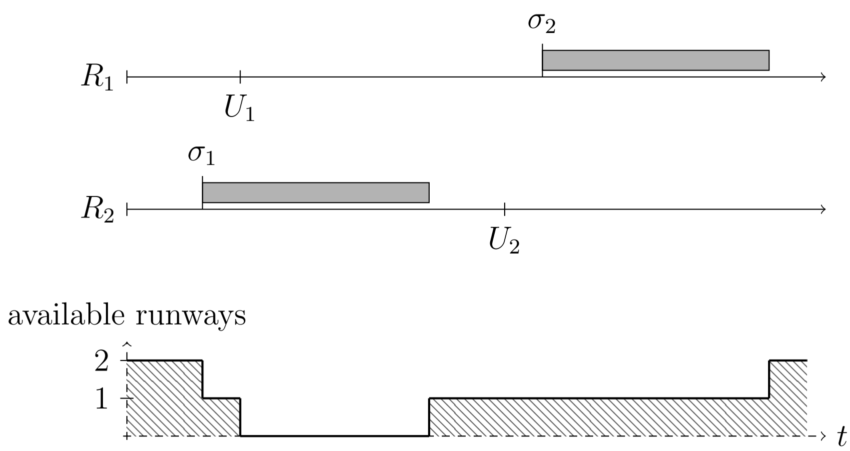
1. *snow removal pattern* $\mathfrak{P}^*$ *is pseudo-active and*

2. *all runways are cleared in non-decreasing order with regard to their parameter* U_r.

Proof. In the following, it is shown that an optimal snow removal schedule $\mathfrak{S}^*$ can always be constructed by extending a pseudo-active snow removal pattern $\mathfrak{P}^* = \big((r_1^*, g_1^*, e_1^*), (r_2^*, g_2^*, e_2^*), \dots, (r_{|\mathcal{R}|}^*, g_{|\mathcal{R}|}^*, e_{|\mathcal{R}|}^*)\big)$ with $U_{r_1^*} \leq U_{r_2^*} \leq \dots \leq U_{r_{|\mathcal{R}|}^*}$ if runways are homogeneous. Consider a feasible and optimal snow removal schedule $\mathfrak{S} = \big((r_1, g_1, e_1, \sigma_1), (r_2, g_2, e_2, \sigma_2), \dots, (r_{|\mathcal{R}|}, g_{|\mathcal{R}|}, e_{|\mathcal{R}|}, \sigma_{|\mathcal{R}|})\big)$. Now, construct a schedule $\mathfrak{S}^* = \big((r_1^*, g_1^*, e_1^*, \sigma_1), (r_2^*, g_2^*, e_2^*, \sigma_2), \dots, (r_{|\mathcal{R}|}^*, g_{|\mathcal{R}|}^*, e_{|\mathcal{R}|}^*, \sigma_{|\mathcal{R}|})\big)$ extending dominant snow removal pattern $\mathfrak{P}^*$ with snow removal start times $\sigma_i \; \forall \; i = 1, 2, \dots, |\mathcal{R}|$ of the optimal snow removal schedule $\mathfrak{S}$. From schedule $\mathfrak{S}$ being feasible ($e_i \leq \sigma_i \; \forall \; i \in 1, 2, \dots, |\mathcal{R}|$) and pattern $\mathfrak{P}^*$ being pseudo-active ($e_i^* = \min_{\mathfrak{P} \in \mathcal{P}}\{e_i\} \forall i \in 1, 2, \dots, |\mathcal{R}|$), it follows that $e_i^* \leq e_i \leq \sigma_i \forall i = 1, 2, \dots, |\mathcal{R}|$ and, hence, that schedule $\mathfrak{S}^*$ is feasible. In schedule $\mathfrak{S}^*$, all homogeneous runways are cleared in non-decreasing order with regard to their parameter U_r. Thus, schedule $\mathfrak{S}^*$ has the same or a better runway availability profile than schedule $\mathfrak{S}$, i.e., at every point in time, schedule $\mathfrak{S}^*$ provides at least as many available runways than schedule $\mathfrak{S}$ (cf. Figure 4.1). Consequently, $\mathfrak{S}^*$ allows the same aircraft schedule and, therefore, the same objective value than $\mathfrak{S}$ and is also optimal. □

Since the number of runways and snow removal groups at airports is rather small, all possible snow removal patterns in $\mathcal{P}$ can be enumerated in order to find a dominant pattern.

(a) Runway availability of $\mathfrak{S}$ with snow removal sequence $R_2 \to R_1$



(b) Runway availability of $\mathfrak{S}^*$ with snow removal sequence $R_1 \to R_2$

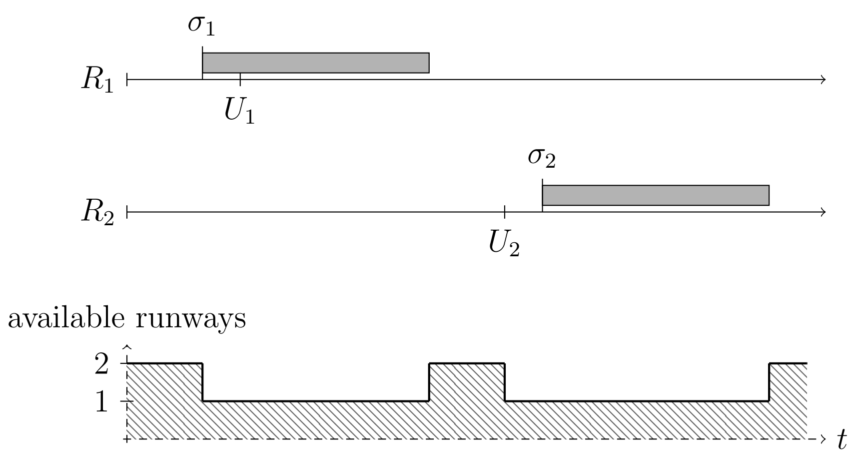


Figure 4.1: Example: $\mathfrak{S}^*$ outperforms $\mathfrak{S}$ in terms of runway availability

4.3 Problem-Specific Valid Inequalities

In the following, two new sets of valid inequalities are presented which are related to snow removal activities. Although these constraints are redundant in the MIP formulation due to a combination of Constraints (4.2) and (4.12), they often tighten the LP relaxation of the MIP.

Valid Inequalities I: Order Between Snow Removals and Aircraft on the Same Runway

If an aircraft a is scheduled on runway r and this aircraft's target time T_a is later than the time U_r at which operations on runway r become unsafe, snow removal on runway r must be completed before aircraft a is scheduled and, thus, $\delta_{ra} = 1$. Consequently, for all pairs of aircraft a and runways r with $U_r < T_a$, the following inequality holds:

$$\delta_{ra} \geq y_{ar} \; \forall \, a \in \mathcal{A}; r \in \mathcal{R} : U_r < T_a \tag{4.20}$$

Valid Inequalities II: Order Between Global Snow Removals and Aircraft

If the target time T_a of an aircraft a is later than all times U_r at which the different runways become unsafe, at least one snow removal activity has to be completed before aircraft a can be scheduled. Consequently, for all aircraft a with $\max_{r \in \mathcal{R}}\{U_r\} < T_a$, the following inequality holds:

$$\sum_{r \in \mathcal{R}} \delta_{ra} \geq 1 \; \forall \, a \in \mathcal{A} : \max_{r \in \mathcal{R}}\{U_r\} < T_a \tag{4.21}$$

4.4 Start Solution Heuristic

The objective of the start solution heuristic is to derive an initial start (incumbent) solution for a problem instance $\mathfrak{I}$. A mapping $\mathfrak{I} \rightarrow \mathfrak{I}'$ is developed to obtain a less complex instance $\mathfrak{I}'$ which can be solved efficiently. $\mathfrak{I}'$ is designed to make maximal use of the proposed pruning rules. To derive $\mathfrak{I}'$ from $\mathfrak{I}$, all separation requirements $S'_{\mathcal{C}\mathcal{C}'}$ between pairs of aircraft operation classes $\mathcal{C}$ and $\mathcal{C}'$ are set to the constant S', where S' equals the minimum of all separation requirements: $S'_{\mathcal{C}\mathcal{C}'} = S' \; \forall \, \mathcal{C}, \mathcal{C}' \in \mathfrak{C}$ where $S' = \min_{\mathcal{C},\mathcal{C}' \in \mathfrak{C}}\{S_{\mathcal{C}\mathcal{C}'}\}$. Hence, in $\mathfrak{I}'$, all aircraft are separation identical, i.e., they can be considered to belong to the same operation class. Thus, pruning rules of type I and II (cf. Section 4.2) yield a complete order of all aircraft. This allows for solving $\mathfrak{I}'$ efficiently even for a large number of aircraft by applying the extended MIP consisting of (4.1)–(4.12), (4.20), and (4.21). From the optimal solution of $\mathfrak{I}'$, the order of activities (aircraft and snow removals) defined by variables $\delta'^{*}_{ij}\ \forall\, i, j \in \mathcal{A} \cup \mathcal{R} : i \neq j$ and the optimal snow removal schedule defined by variables $v'^{*}_{r}\ \forall\, r \in \mathcal{R}$ and $\rho'^{*}_{rg}\ \forall\, r \in \mathcal{R}; g \in \mathcal{G}$ is stored. Based on this information, an initial solution for $\mathfrak{I}$ is constructed by solving an instance $\mathfrak{I}^{\text{ini}}$ which equals instance $\mathfrak{I}$ with fixed values $\delta_{ij} = \delta'^{*}_{ij}\ \forall\, i, j \in \mathcal{A} \cup \mathcal{R} : i \neq j$, $v_r = v'^{*}_{r}\ \forall\, r \in \mathcal{R}$, $\rho_{rg} = \rho'^{*}_{rg}\ \forall\, r \in \mathcal{R}; g \in \mathcal{G}$, and $\phi_{rs} = \phi'^{*}_{rs}\ \forall\, r, s \in \mathcal{R} : r \neq s$. For many instances, $\mathfrak{I}^{\text{ini}}$ yields a good initial solution for $\mathfrak{I}$. For some instances, $\mathfrak{I}^{\text{ini}}$ can be infeasible due to adverse parameter values $S_{\mathcal{C}\mathcal{C}'}$.

The time-continuous solution approach presented in this chapter is tested and evaluated in a computational study in Chapter 6.

5 Time-Discrete Approach – Combining Constraint Programming and Column Generation

This chapter presents the second solution approach for the WRSP using a time-discrete model formulation and a novel combination of CP and column generation. While the time-continuous model discussed in the previous chapter is limited to linear cost functions, a time-indexed model offers greater flexibility with regard to the objective function. From a technical viewpoint, a time-discrete model is favorable since time-indexed model formulations often provide relatively strong bounds compared to big-M formulations.

Section 5.1 introduces a time-discrete BP for the WRSP considering multiple heterogeneous runways, multiple snow removal groups, and sequence-dependent separation requirements for all pairs of aircraft as well as sequence-dependent transit times for snow removal groups. Mandatory precedence relations between aircraft and snow removal activities lead to additional constraints for the BP. Section 5.2 presents an equivalent CP model formulation including additional precedence constraints for the CP model. In Section 5.3, a method to discretize the planning horizon is proposed. This enables improved solutions closer to optimal time-continuous solutions and allows a good balance between solution quality and model size. Finally, Section 5.4 presents the exact solution methodology for time-discrete WRSP. During preprocessing, problem-specific dominance rules reduce variable domains. The CP model is used to further reduce variable domains through constraint propagation. It is also the basis for a CP start heuristic. Afterwards, the proposed BP solves the WRSP to optimality. To keep the size, i.e., the number of variables and constraints, of the time-discrete binary model formulation manageable, a column generation scheme is proposed, which generates all variables required to optimally solve the BP.

5.1 Binary Program Based on Clique Inequalities

This section introduces a time-discrete mathematical problem formulation for the WRSP. The model considers the finite planning horizon $[1, T^{\max}]$ and a set $\mathcal{T}$ of time points $t \in \mathcal{T} : 1 \leq t \leq T^{\max}$ within that planning horizon. If an aircraft or snow removal is scheduled at time t, the take-off, landing or snow removal operation starts at t. If a runway becomes unsafe at time t, it must be closed directly after t and t is the latest time at which an aircraft can still safely take-off or land.

If aircraft $a \in \mathcal{A}$ is scheduled on runway $r \in \mathcal{R}$ at time $t \in \mathcal{T}$, binary variable x_{art} equals one and associated cost C_{art} occur. Respectively, if snow removal group $g \in \mathcal{G}$ is scheduled on runway r at time t, binary variable y_{grt} equals one. If runway r is not cleared during the planning horizon, binary variable z_r equals one.

The time-discrete model in this chapter considers runway-specific aircraft time windows $\mathcal{T}_{ar} = \{E_{ar}, \ldots, L_{ar}\} \subseteq \mathcal{T}$ where E_{ar} denotes the earliest possible take-off or landing time of aircraft a on runway r and L_{ar} denotes the latest possible take-off or landing time respectively. If $\mathcal{T}_{ar} = \emptyset$, aircraft a cannot be assigned to runway r. Similarly, the model considers snow removal time windows $\mathcal{T}_{rg} = \{E_{rg}, \ldots, L_{rg}\} \subseteq \mathcal{T}$ for combinations of runways r and snow removal groups g. If snow removal on runway r is conducted by group g, snow removal cannot start before E_{rg} or after L_{rg}. If $\mathcal{T}_{rg} = \emptyset$, snow removal group g cannot be used to clear runway r.

The operation class of aircraft a is denoted by $c(a)$. The model enforces sequence-dependent separation requirements $S_{c(a)c(b)}$ between pairs of aircraft a and b on the same runway, safety buffers $O_{c(a)}$ between aircraft and following snow removals on the same runway, and runway-specific snow removal durations P_r. Additionally, it considers that each runway r can become unsafe due to snow, ice, or slush at time U_r if it has not been cleared before within the planning horizon. If $U_r \geq T^{\max}$, runway r does not become unsafe during the planning horizon. Once a runway has become unsafe, it must be cleared before it can be reopened and used again. Due to the operational planning horizon of up to two hours, at most one snow removal activity must be scheduled for each runway. For snow removal groups, the model considers sequence-dependent setup times Q_{rs} between runways r and s, which include the time required to clear runway r and the driving time from runway r to runway s. The model minimizes the total delay cost of the schedule. Table 5.1 summarizes the sets and parameters used for the BP and Table 5.2 defines the decision variables of the model.

The BP for the WRSP is based on a family of clique inequalities which have been introduced and referred to as (S,t)-clique inequalities by Avella et al. (2017). The model considers subsets (cliques) of aircraft and snow removal groups $\mathcal{V} \subseteq (\mathcal{A} \cup \mathcal{G})$ with $|\mathcal{V}| \geq 2$. Here, for all aircraft $a \in (\mathcal{V} \cap \mathcal{A})$, $v_a(\mathcal{V})$ denotes a clique-specific minimum separation time after scheduling aircraft a and is defined as

$$v_a(\mathcal{V}) = \begin{cases} \min_{b \in \mathcal{V} \setminus \{a\}} \{S_{c(a)c(b)}\} & \text{if} \quad \mathcal{V} \subseteq \mathcal{A} \\ \min_{b \in (\mathcal{V} \cap \mathcal{A}) \setminus \{a\}} \{O_{c(a)}, S_{c(a)c(b)}\} & \text{else} \end{cases}$$

The set of all cliques required for a specific combination of runway r and time t is denoted as $\mathfrak{V}_{rt}$.

Similarly, the model considers subsets (cliques) of runways $\mathcal{W} \subseteq \mathcal{R}$ with $|\mathcal{W}| \geq 2$. Here, for all runways $r \in \mathcal{W}$, $w_r(\mathcal{W})$ denotes a clique-specific minimum setup time after clearing runway r and is defined as

$$w_r(\mathcal{W}) = \min_{s \in \mathcal{W} \setminus \{r\}} \{Q_{rs}\}$$

The set of all cliques required for a specific combination of snow removal group g and time t is denoted as $\mathfrak{W}_{gt}$.

Table 5.1: Sets and parameters of the time-discrete model

Notation	Definition
$t \in \mathcal{T}$	Set of considered time points within the planning horizon
$r \in \mathcal{R}$	Set of runways
U_r	Time at which flight operations on runway r become unsafe
P_r	Required time to clear runway r
Q_{rs}	Sequence-dependent setup time between starts of snow removals on runways r and s conducted by the same snow removal group (including snow removal time P_r and required transit time from runway r to runway s)
$a \in \mathcal{A}$	Set of aircraft
$\mathcal{C} \in \mathfrak{C}$	Set of operation classes
$c(a)$	Operation class of aircraft a
E_{ar}	Earliest possible take-off or landing time of aircraft a on runway r
T_{ar}	Target take-off or landing time of aircraft a on runway r
L_{ar}	Latest possible take-off or landing time of aircraft a on runway r
$\mathcal{T}_{ar} = \{E_{ar}, \ldots, L_{ar}\} \subseteq \mathcal{T}$	Time window of aircraft a on runway r
$S_{\mathcal{C}\mathcal{C}'}$	Sequence-dependent separation time between a leading aircraft of operation class $\mathcal{C}$ and a trailing aircraft of operation class $\mathcal{C}'$ if both aircraft are scheduled on the same runway
$O_{\mathcal{C}}$	Required separation time between an aircraft of operation class $\mathcal{C}$ and a subsequent snow removal on the same runway
$g \in \mathcal{G}$	Set of snow removal groups
E_{rg}	Earliest possible time for snow removal on runway r conducted by snow removal group g
L_{rg}	Latest possible time for snow removal on runway r conducted by snow removal group g
$\mathcal{T}_{rg} = \{E_{rg}, \ldots, L_{rg}\} \subseteq \mathcal{T}$	Time window for snow removal on runway r conducted by snow removal group g
C_{art}	Cost coefficient for scheduling aircraft a on runway r at time t

Table 5.2: Decision variables of the time-discrete model

Notation	Definition
x_{art}	$= \begin{cases} 1 & \text{if aircraft } a \text{ is scheduled on runway } r \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$
y_{grt}	$= \begin{cases} 1 & \text{if snow removal group } g \text{ clears runway } r \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$
z_r	$= \begin{cases} 1 & \text{if runway } r \text{ is not cleared during the planning horizon} \\ 0 & \text{otherwise} \end{cases}$

With this, the BP formulation is as follows:

$$\textbf{minimize} \quad \sum_{a \in \mathcal{A}} \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}_{ar}} C_{art} x_{art} \tag{5.1}$$

subject to

$$\sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}_{ar}} x_{art} = 1 \qquad \forall\, a \in \mathcal{A} \tag{5.2}$$

$$\sum_{g \in \mathcal{G}} \sum_{t \in \mathcal{T}_{rg}} y_{grt} + z_r = 1 \qquad \forall\, r \in \mathcal{R} \tag{5.3}$$

$$\sum_{a \in \mathcal{V} \cap \mathcal{A}} \sum_{l \in [t - v_a(\mathcal{V}) + 1, t] \cap \mathcal{T}_{ar}} x_{arl} + \sum_{g \in \mathcal{V} \cap \mathcal{G}} \sum_{m \in [t - P_r + 1, t] \cap \mathcal{T}_{rg}} y_{grm} \leq 1$$
$$\forall\, r \in \mathcal{R}; t \in \mathcal{T} : t \leq U_r; \mathcal{V} \in \mathfrak{V}_{rt} \tag{5.4}$$

$$\sum_{a \in \mathcal{V} \cap \mathcal{A}} \sum_{l \in [t - v_a(\mathcal{V}) + 1, t] \cap \mathcal{T}_{ar}} x_{arl} + \sum_{g \in \mathcal{V} \cap \mathcal{G}} \sum_{m \in [t - P_r + 1, T^{\max}] \cap \mathcal{T}_{rg}} y_{grm} + z_r \leq 1$$
$$\forall\, r \in \mathcal{R}; t \in \mathcal{T} : t > U_r; \mathcal{V} \in \mathfrak{V}_{rt} \tag{5.5}$$

$$\sum_{r \in \mathcal{W}} \sum_{l \in [t - w_r(\mathcal{W}) + 1, t] \cap \mathcal{T}_{rg}} y_{grl} \leq 1 \qquad \forall\, g \in \mathcal{G}; t \in \mathcal{T}; \mathcal{W} \in \mathfrak{W}_{gt} \tag{5.6}$$

$$x_{art}, y_{grt}, z_r \in \{0, 1\} \qquad \forall\, a \in \mathcal{A}; r \in \mathcal{R}; t \in \mathcal{T}; g \in \mathcal{G}$$

The objective function (5.1) minimizes the overall delay cost of the schedule. Constraints (5.2) assign exactly one runway and one take-off or landing time to each aircraft. Constraints (5.3) assign a snow removal group and a clearing time to a runway r if this runway is cleared during the planning horizon. Constraints (5.4) and (5.5) make sure that, at any point in time, each runway is occupied by at most one activity (aircraft or snow removal) respecting all separation requirements. Constraints (5.5) also consider the possibility that a runway is unsafe and ensure that runways which do not allow safe operations are not used for aircraft operations. Constraints (5.6) secure that, at any point in time, each snow removal group is busy with at most one snow removal activity respecting setup times between runways.

The number of constraints (5.4)–(5.6) mainly depends on the cardinalities of sets $\mathfrak{V}_{rt}$, $\mathfrak{W}_{gt}$ and $\mathcal{T}$ since the cardinalities of sets $\mathcal{A}$, $\mathcal{R}$ and $\mathcal{G}$ are comparatively small.

Constraints (5.4)–(5.6) yield a sufficient and exhaustive set of clique inequalities if they constitute a complete cover of the problem's underlying incompatibility structure, i.e., if all incompatibilities between aircraft operations, snow removal operations and unsafe runways are reflected in at least one constraint. Formally, this is the case if

$$\forall\, a,b \in \mathcal{A} : a \neq b; r \in \mathcal{R}; t \in \{E_{ar}, \ldots, L_{ar} + S_{c(a)c(b)} - 1\} \cap \mathcal{T}_{br}$$
$$\exists\, \mathcal{V} \in \mathfrak{V}_{rt} : a,b \in \mathcal{V} \wedge S_{c(a)c(b)} = v_a(\mathcal{V}) \tag{5.7}$$
$$\forall\, a \in \mathcal{A}; g \in \mathcal{G}; r \in \mathcal{R}; t \in \{E_{ar}, \ldots, L_{ar} + O_{c(a)} - 1\} \cap \mathcal{T}_{rg}$$
$$\exists\, \mathcal{V} \in \mathfrak{V}_{rt} : a,g \in \mathcal{V} \wedge O_{c(a)} = v_a(\mathcal{V}) \tag{5.8}$$
$$\forall\, a \in \mathcal{A}; g \in \mathcal{G}; r \in \mathcal{R}; t \in \{E_{rg}, \ldots, L_{rg} + P_r - 1\} \cap \mathcal{T}_{ar}$$
$$\exists\, \mathcal{V} \in \mathfrak{V}_{rt} : a,g \in \mathcal{V} \tag{5.9}$$
$$\forall\, r,s \in \mathcal{R} : r \neq s; g \in G; t \in \{E_{rg}, \ldots, L_{rg} + Q_{rs} - 1\} \cap \mathcal{T}_{sg}$$
$$\exists\, \mathcal{W} \in \mathfrak{W}_{gt} : r,s \in \mathcal{W} \wedge Q_{rs} = w_r(\mathcal{W}) \tag{5.10}$$
$$\forall\, a \in \mathcal{A}; r \in \mathcal{R}; t \in \mathcal{T}_{ar} : t > U_r$$
$$\exists\, \mathcal{V} \in \mathfrak{V}_{rt} : a \in \mathcal{V} \tag{5.11}$$

(5.7) covers incompatibilities between two aircraft on the same runway due to separation requirements. (5.8) covers incompatibilities between aircraft and following snow removals, while (5.9) covers incompatibilities between snow removals and aircraft ensuring that no aircraft is scheduled during a snow removal. (5.10) covers incompatibilities between two snow removals conducted by the same snow removal group ensuring sufficient setup time between snow removals. (5.11) covers incompatibilities between aircraft and unsafe runways making sure that no aircraft is scheduled on a runway which is unsafe.

5.1.1 Construction Algorithms to Create Sets of Required Cliques

Two construction algorithms are used to create sets $\mathfrak{V}_{rt}$ and $\mathfrak{W}_{gt}$ satisfying conditions (5.7)–(5.11). They generate sufficient sets of cliques while keeping the cardinalities of sets $\mathfrak{V}_{rt}$ and $\mathfrak{W}_{gt}$ small. Algorithm 5.1 *Construct-*$\mathfrak{V}_{rt}$ is based on the matrix of sequence-dependent separation times $S_{\mathcal{C}\mathcal{C}'}$ between aircraft operation classes (as in Table 2.1) and on required separation times $O_{\mathcal{C}}$ between aircraft and following snow removals. Algorithm 5.2 *Construct-*$\mathfrak{W}_{gt}$ is based on the matrix of sequence-dependent setup times Q_{rs} between runways. The common underlying idea for both algorithms is to add subsets (cliques) of aircraft and snow removal groups to $\mathfrak{V}_{rt}$ and subsets (cliques) of runways to $\mathfrak{W}_{gt}$ in a systematic and structured way until all separation requirements between aircraft and snow removals (c.f. *Construct-*$\mathfrak{V}_{rt}$) and all setup times between pairs of snow removals (c.f. *Construct-*$\mathfrak{W}_{gt}$) are reflected and, therefore, conditions (5.7)–(5.11) are fulfilled. Algorithms *Construct-*$\mathfrak{V}_{rt}$ and *Construct-*$\mathfrak{W}_{gt}$ can generate dominated cliques, which can lead to redundant or non-binding constraints in the BP. Hence, in the last steps of both algorithms, dominated cliques are identified and removed from sets $\mathfrak{V}_{rt}$ and $\mathfrak{W}_{gt}$. Here, clique $\mathcal{V}$ dominates clique $\mathcal{V}'$ if $\mathcal{V} \supseteq \mathcal{V}'$ and $v_a(\mathcal{V}) \geq v_a(\mathcal{V}') \, \forall \, a \in \mathcal{V}' \cap \mathcal{A}$. Similarly, clique $\mathcal{W}$ dominates clique $\mathcal{W}'$ if $\mathcal{W} \supseteq \mathcal{W}'$ and $w_r(\mathcal{W}) \geq w_r(\mathcal{W}') \, \forall \, r \in \mathcal{W}'$ (cf. Proposition 5 in Avella et al., 2017).

5.1.2 Extensions Compared to the Time-Continuous Model

While the time-continuous model only considers aircraft-specific time windows, the time-discrete model allows aircraft time windows also to be runway-specific. For arriving and departing aircraft, different time windows can be specified for

Algorithm 5.1 *Construct-*$\mathfrak{V}_{rt}$

initialize set of cliques $\mathfrak{V}_{rt} = \emptyset$
for all leading operation classes $\mathcal{L} \in \mathfrak{C}$ **do**
  create set of separation times $\mathcal{S} = \{S_{\mathcal{L}\mathcal{F}} : \mathcal{F} \in \mathfrak{C}, O_{\mathcal{L}}\}$
  for all separation times $s \in \mathcal{S}$ **do**
    if $s \leq S_{\mathcal{L}\mathcal{L}}$ **then**
      create new clique $\mathcal{V}$ of leading aircraft a, trailing aircraft b and snow removal groups g
      with $\mathcal{V} = \big\{a : a \in \mathcal{L} \wedge t \in \{E_{ar}, \ldots, L_{ar} + s - 1\},$
        $b : S_{\mathcal{L}c(b)} \geq s \wedge t \in \mathcal{T}_{br},$
        $g : g \in \mathcal{G} \wedge O_{\mathcal{L}} \geq s \wedge t \in \mathcal{T}_{rg}\big\}$
      add $\mathcal{V}$ to $\mathfrak{V}_{rt}$
    if $s > S_{\mathcal{L}\mathcal{L}}$ **then**
      for all leading aircraft $a \in \mathcal{L} : t \in \{E_{ar}, \ldots, L_{ar} + s - 1\}$ **do**
        create new clique $\mathcal{V}$ of leading aircraft a, trailing aircraft b and snow removal groups g
        with $\mathcal{V} = \big\{a,$
          $b : S_{\mathcal{L}c(b)} \geq s \wedge t \in \mathcal{T}_{br},$
          $g : g \in \mathcal{G} \wedge O_{\mathcal{L}} \geq s \wedge t \in \mathcal{T}_{rg}\big\}$
        add $\mathcal{V}$ to $\mathfrak{V}_{rt}$
remove dominated cliques from $\mathfrak{V}_{rt}$

Algorithm 5.2 *Construct-*$\mathfrak{W}_{gt}$

initialize set of cliques $\mathfrak{W}_{gt} = \emptyset$
for all preceding runways $r \in \mathcal{R}$ **do**
  create set of setup times $\mathcal{Q} = \{Q_{rs} : s \in \mathcal{R}\}$
  for all setup times $q \in \mathcal{Q}$ **do**
    create new clique $\mathcal{W}$ of preceding runway r, succeeding runways s
    with $\mathcal{W} = \big\{r : t \in \{E_{rg}, \ldots, L_{rg} + q - 1\},$
      $s : s \in \mathcal{R} \wedge Q_{rs} \geq q \wedge t \in \mathcal{T}_{rg}\big\}$
    add $\mathcal{W}$ to $\mathfrak{W}_{gt}$
remove dominated cliques from $\mathfrak{W}_{gt}$

different runways. This allows for modeling situations in which aircraft have different distances and flight paths to different runways and, thus, can reach the runways at different times. Similar to runway-specific aircraft time windows, the time-discrete model also supports runway-specific time windows for snow removal groups. Adjusting earliest possible snow removal times E_{rg} allows to reflect current locations of snow removal groups g and their respective distances to runways r. Additionally, the time-discrete model allows earliness, i.e., aircraft can be accelerated and scheduled before their target time T_{ar} and $E_{ar} \leq T_{ar} \leq L_{ar}$ holds. In the time-discrete model formulation, for every combination of aircraft a and runway r, each possible take-off or landing time t is associated with a specific cost factor C_{art}. Hence, cost functions for earliness and tardiness are not restricted to linear functions as in the time-continuous model formulation, but can assume any arbitrary shape including non-linear, non-convex, or discontinuous functions.

5.1.3 Additional Constraints for the Time-Discrete Binary Program

Similar to pruning rules for the time-continuous model, additional constraints which decrease the solution space can be derived from compulsory precedence relations. Such compulsory precedence relations can exist between aircraft of the same class and between snow removal activities (cf. Section 4.2). This section presents aggregated-time and disaggregated-time versions for both types of precedence constraints for the time-discrete BP.

Precedence Constraints for Aircraft of the Same Operation Class

For aircraft of the same operation class, it is often possible to determine an optimal order during preprocessing and, hence, to derive precedence constraints between all pairs of aircraft within the respective operation class. For the time-continuous WRSP with aircraft-specific cost coefficients and without earliness ($E_{ar} = T_{ar} \leq L_{ar}$), Section 4.2 showed that it is always optimal to schedule aircraft of the same operation class in their corresponding time window order if they are cost compliant. A similar observation can be made for the time-discrete WRSP if aircraft acceleration is permitted ($E_{ar} \leq T_{ar} \leq L_{ar}$).

Theorem 4. *For two aircraft a and b of the same operation class, it is always optimal to schedule a not after b if $E_{ar} \leq E_{br} \wedge L_{ar} \leq L_{br} \; \forall\, r \in \mathcal{R}$ (time window order) and $C_{art} + C_{br't'} \leq C_{ar't'} + C_{brt} \forall r, r' \in \mathcal{R}; t \in \mathcal{T}_{ar} \cap \mathcal{T}_{br}; t' \in \mathcal{T}_{ar'} \cap \mathcal{T}_{br'} : t \leq t'$ (cost compliance).*

To prove Theorem 4, a swap argument can be applied analogously to the proof of Theorem 2. Either, it is not possible to swap aircraft a and b due to a violation of respective time windows or, if a swap is possible, the objective function value cannot be improved.

For pairs of aircraft a and b, for which it is known that a must not be scheduled after b, the compulsory precedence relation is denoted as $a \preceq b$.

Inequalities (5.12) express the corresponding precedence constraints between aircraft for the BP in an aggregated-time version.

$$\sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}_{br}} (t \cdot x_{brt}) - \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}_{ar}} (t \cdot x_{art}) \geq 0 \qquad \forall\, a, b \in \mathcal{A} : a \preceq b \tag{5.12}$$

Disaggregated-time versions of the same aircraft precedence constraints can be formulated as Inequalities (5.13).

$$\sum_{r \in \mathcal{R}} \sum_{l \in [0,t] \cap \mathcal{T}_{ar}} x_{arl} - \sum_{r \in \mathcal{R}} \sum_{m \in [0,t] \cap \mathcal{T}_{br}} x_{brm} \geq 0 \qquad \forall\, a, b \in \mathcal{A} : a \preceq b; t \in \bigcup_{r \in \mathcal{R}} \mathcal{T}_{br} \tag{5.13}$$

By adjusting the set $\mathcal{R}$ in (5.12) and (5.13), both versions of aircraft precedence constraints can be adapted to situations where precedence constraints only hold for a certain runway or for a subset of runways.

Computational experiments showed that precedence constraints between snow removals improve computational times by a larger degree than precedence constraints between aircraft. The experiments also showed that the aggregated-time versions are more beneficial since disaggregated-time versions regularly yield too many constraints.

Precedence Constraints for Snow Removals Based on Dominant Snow Removal Patterns

Section 4.2 showed that, for a set of homogeneous runways, i.e., runways of the same length allowing for the same flight operations, it is often possible to precompute a dominant snow removal pattern $\mathfrak{P}^* = \big((r_1^*, g_1^*, e_1^*), (r_2^*, g_2^*, e_2^*), \ldots, (r_{|\mathcal{R}|}^*, g_{|\mathcal{R}|}^*, e_{|\mathcal{R}|}^*)\big)$ and, thereby, an optimal sequence of snow removals. If a dominant, i.e., optimal, snow removal pattern is known, Inequalities (5.14) express all corresponding precedence constraints between snow removals for the time-discrete BP in an aggregated-time version.

$$\sum_{t \in \mathcal{T}_{r_j g_j}} (t \cdot y_{g_j r_j t}) + \max_{t \in \mathcal{T}_{r_i g_i}} (t) \cdot z_{r_j} - \sum_{t \in \mathcal{T}_{r_i g_i}} (t \cdot y_{g_i r_i t}) \geq 0$$
$$\forall\, (r_i, g_i, e_i), (r_j, g_j, e_j) \in \mathfrak{P}^* : i < j \quad (5.14)$$

Disaggregated-time versions of the same precedence constraints between snow removals can be formulated as Inequalities (5.15).

$$\sum_{l \in [0,t] \cap \mathcal{T}_{r_i g_i}} y_{g_i r_i l} - \sum_{m \in [0,t] \cap \mathcal{T}_{r_j g_j}} y_{g_j r_j m} \geq 0$$
$$\forall\, (r_i, g_i, e_i), (r_j, g_j, e_j) \in \mathfrak{P}^* : i < j; t \in \mathcal{T}_{r_j g_j} \quad (5.15)$$

5.2 Constraint Programming Model

This section presents a CP model formulation for the WRSP. The first part of this section introduces the used CP constructs. Specifically, used variable types, expressions, and cumulative functions are defined. Then, the CP model formulation is presented and additional constraints based on compulsory precedence relations are developed.

5.2.1 Variable Types, Expressions, and Cumulative Functions

The notation of the CP model presented in this section is based on common scheduling concepts from CP and a terminology which is similar to the one used in IBM

ILOG CPLEX CP Optimizer and in Goel et al. (2015) and Novara et al. (2016). This section briefly defines and describes the used concepts and constructs.

Interval variables represent aircraft and snow removals. In a solution of a constraint program, an interval variable $\tilde{x}$ can be present with a discrete start time s and end time e or can be absent (denoted as $\tilde{x} = \perp$). Hence, an interval variable $\tilde{x}$ is formally defined as $\tilde{x} \in \{[s, e[\colon s, e \in \mathbb{Z}, s \leq e\} \cup \{\perp\}$. The set of possible values for an interval variable is called its domain. The model uses the following expressions and constraints for interval variables $\tilde{x}$ and $\tilde{y}$ and sets $\tilde{\mathcal{A}}$ of interval variables.

presenceOf($\tilde{x}$) returns `TRUE` if interval $\tilde{x}$ is present and `FALSE` if interval $\tilde{x}$ is absent.

startOf($\tilde{x}$) returns the start time s of a present interval $\tilde{x} = [s, e[$.

lengthOf($\tilde{x}$) returns the size $e - s$ of a present interval $\tilde{x} = [s, e[$.

startBeforeStart($\tilde{x}$, $\tilde{y}$) enforces interval $\tilde{x}$ to start not after interval $\tilde{y}$ if both intervals are present.

alternative($\tilde{x}$, $\tilde{\mathcal{A}}$) defines an exclusive alternative between intervals in set $\tilde{\mathcal{A}}$. If interval $\tilde{x}$ is present, then exactly one interval of set $\tilde{\mathcal{A}}$ is present and this specific interval starts and ends together with interval $\tilde{x}$. If interval $\tilde{x}$ is absent, all intervals in set $\tilde{\mathcal{A}}$ are absent.

S*equence variables* represent sequences for specific runways and snow removal groups. A sequence variable $\hat{\xi}$ is defined over a set $\tilde{\mathcal{A}}$ of intervals and represents an order of all present intervals in set $\tilde{\mathcal{A}}$. Each interval $\tilde{a}$ in set $\tilde{\mathcal{A}}$ is also associated with a type $\theta_{\tilde{a}} \in \Theta$ through a mapping function $\Omega : \tilde{\mathcal{A}} \rightarrow \Theta$. A transition matrix can be specified as a function $M : \Theta \times \Theta' \rightarrow \mathbb{Z}$ to express the minimal distance between the end of an interval of type Θ and the start of an interval of type Θ'. In combination with a noOverlap-constraint, this allows to model sequence-dependent transition (setup) times. The model uses the following expressions and constraints for sequence variables $\hat{\xi}$, sets $\tilde{\mathcal{A}}$ of interval variables, mappings Ω to associated interval types Θ and transition matrices M.

sequenceVar($\tilde{\mathcal{A}}$, Ω) defines a sequence variable $\hat{\xi}$ over interval set $\tilde{\mathcal{A}}$ and type mapping Ω.

noOverlap($\hat{\xi}$, M) enforces sequence-dependent transition times for all intervals in $\hat{\xi}$ according to transition matrix M.

Cumulative functions represent runways and snow removal groups as constrained resources. Hereby, a cumulative function's value over time models a resource availability over time. Intervals contribute individually to a cumulative function by changing the function's value through elementary step functions and, thus, represent the use and release of a resource. The model uses the following elementary step functions.

pulse($\tilde{x}$, k) changes the value of a cumulative function by k during the interval $\tilde{x} = [s, e[$. It increases the function's value by k at time s and decreases the value by k at time e.

stepAtEnd($\tilde{x}$, k) increases the value of a cumulative function by k at the end e of interval $\tilde{x} = [s, e[$.

step(t, k) increases the value of a cumulative function by k at time t.

5.2.2 Model Formulation

For scheduling aircraft, mandatory interval variables $\tilde{x}_a$ are defined for all aircraft $a \in \mathcal{A}$. To represent the specific scheduling of an aircraft a on runway r, optional interval variables $\tilde{x}_{ar} \in \tilde{\mathcal{A}}$ are used for all aircraft $a \in \mathcal{A}$ and runways $r \in \mathcal{R}$. Similarly, for scheduling snow removals on runways r, optional interval variables $\tilde{y}_r$ are defined for all runways $r \in \mathcal{R}$. To represent the specific scheduling of a snow removal on runway r by snow removal group g, optional interval variables $\tilde{y}_{gr} \in \tilde{\mathcal{Y}}$ are used for all snow removal groups $g \in \mathcal{G}$ and runways $r \in \mathcal{R}$.

To represent a sequence of aircraft and snow removals on a specific runway r, the model uses sequence variables $\hat{\sigma}_r$ for all runways $r \in \mathcal{R}$. To represent a sequence of snow removals for a specific snow removal group, the model uses sequence variables $\hat{\tau}_g$ for all snow removal groups $g \in \mathcal{G}$.

To facilitate constraint propagation and domain reduction in the CP model, required separation times $S_{\mathcal{C}\mathcal{C}'}$ between aircraft of operation class $\mathcal{C}$ and aircraft of operation class $\mathcal{C}'$ are split into a constant processing time $p = \min_{\mathcal{C},\mathcal{C}' \in \mathfrak{C}}\{S_{\mathcal{C}\mathcal{C}'}\}$ and a setup time $S^{\text{CP}}_{\mathcal{C}\mathcal{C}'} = S_{\mathcal{C}\mathcal{C}'} - p$. Subsequently, to preserve congruency with time-discrete BP and to allow U_r as the last possible take-off or landing time for aircraft on runways which become unsafe at U_r, parameter U_r^{CP} is introduced as $U_r^{\text{CP}} = U_r + p$.

This thesis defines two mappings of interval variables to types and two transition matrices accordingly. With regard to specific runway sequences, mapping $\Omega^R : \tilde{\mathcal{A}} \cup \tilde{\mathcal{Y}} \rightarrow \mathfrak{C} \cup \{srg\}$ associates each aircraft interval variable $\tilde{x}_{ar}$ with a type corresponding to its operation class $c(a)$ and all snow removal interval variables $\tilde{y}_{gr}$ with a type srg:

$$\Omega^R(\tilde{a}) = \begin{cases} c(a) & \forall\, \tilde{a} \in \{\tilde{x}_{ar} : a \in \mathcal{A} \wedge r \in \mathcal{R}\} \\ srg & \forall\, \tilde{a} \in \{\tilde{y}_{gr} : g \in \mathcal{G} \wedge r \in \mathcal{R}\} \end{cases}$$

Transition matrix $M^R : \mathfrak{C} \cup \{srg\} \times \mathfrak{C} \cup \{srg\} \rightarrow \mathbb{Z}$ represents separation requirements between aircraft and snow removals accordingly:

$$M^R(\mathcal{C}, \mathcal{C}') = \begin{cases} S^{\text{CP}}_{\mathcal{C}\mathcal{C}'} & \forall\, \mathcal{C}, \mathcal{C}' \in \mathfrak{C} \\ 0 & \text{otherwise} \end{cases}$$

With regard to snow removal sequences of specific snow removal groups, mapping $\Omega^G : \tilde{\mathcal{Y}} \rightarrow \mathcal{R}$ associates all snow removal interval variables $\tilde{y}_{gr}$ with a type corresponding to its runway r:

$$\Omega^G(\tilde{y}_{gr}) = r$$

Transition matrix $M^G : \mathcal{R} \times \mathcal{R} \rightarrow \mathbb{Z}$ represents required transfer times between runways accordingly:

$$M^G(r, s) = Q_{rs} - P_r$$

The function $Cost_{ar} : \mathcal{T} \rightarrow \mathbb{R}$ denotes an aircraft- and runway-specific cost function and determines the resulting earliness or tardiness cost if aircraft a is scheduled on runway r at time t.

Using this notation, the CP model formulation is as follows:

$$\textbf{minimize} \qquad \sum_{a\in\mathcal{A}} \sum_{r\in\mathcal{R}} Cost_{ar}(\text{startOf}(\tilde{x}_{ar})) \tag{5.16}$$

subject to

$$\tilde{x}_{ar} \in [E_{ar}, L_{ar} + p) \qquad \forall\, a \in \mathcal{A}; r \in \mathcal{R} \tag{5.17}$$

$$\text{lengthOf}(\tilde{x}_{ar}) = p \qquad \forall\, a \in \mathcal{A}; r \in \mathcal{R} \tag{5.18}$$

$$\text{presenceOf}(\tilde{x}_a) = \texttt{TRUE} \qquad \forall\, a \in \mathcal{A} \tag{5.19}$$

$$\text{alternative}(\tilde{x}_a,\, \{\tilde{x}_{ar} : r \in \mathcal{R}\}) \qquad \forall\, a \in \mathcal{A} \tag{5.20}$$

$$\text{lengthOf}(\tilde{y}_{gr}) = P_r \qquad \forall\, g \in \mathcal{G}; r \in \mathcal{R} \tag{5.21}$$

$$\text{alternative}(\tilde{y}_r,\, \{\tilde{y}_{gr} : g \in \mathcal{G}\}) \qquad \forall\, r \in \mathcal{R} \tag{5.22}$$

$$\hat{\sigma}_r = \text{sequenceVar}(\{\tilde{x}_{ar} : a \in \mathcal{A}, \tilde{y}_{gr} : g \in \mathcal{G}\},\, \Omega^R) \qquad \forall\, r \in \mathcal{R} \tag{5.23}$$

$$\text{noOverlap}(\hat{\sigma}_r,\, M^R) \qquad \forall\, r \in \mathcal{R} \tag{5.24}$$

$$\hat{\tau}_g = \text{sequenceVar}(\{\tilde{y}_{gr} : r \in \mathcal{R}\},\, \Omega^G) \qquad \forall\, g \in \mathcal{G} \tag{5.25}$$

$$\text{noOverlap}(\hat{\tau}_g,\, M^G) \qquad \forall\, g \in \mathcal{G} \tag{5.26}$$

$$\sum_{a\in\mathcal{A}} \text{pulse}(\tilde{x}_{ar},\, 1) + \text{step}(U_r^{\text{CP}},\, 1) + \sum_{g\in\mathcal{G}} \text{stepAtEnd}(\tilde{y}_{gr},\, -1) \leq 1 \qquad \forall\, r \in \mathcal{R} \tag{5.27}$$

Objective function (5.16) minimizes the overall cost of the schedule depending on the take-off and landing times of aircraft. Constraints (5.17) and (5.18) define the time windows and processing times for all aircraft. Constraints (5.19) and (5.20) ensure that each aircraft is scheduled on exactly one runway. Constraints (5.21) define the duration of snow removals on runways and Constraints (5.22) assign exactly one snow removal group to each runway which is cleared. Constraints (5.23) declare runway-specific sequences of aircraft and snow removals and Constraints (5.24) ensure that all separation times between aircraft and snow removals on the same runway are met. Similarly, Constraints (5.25) declare a snow removal sequence for each snow removal group and Constraints (5.26) ensure sufficiently large setup times between snow removals for each snow removal group. Finally, Constraints (5.27) represent the availability of each runway and make sure that aircraft are only scheduled on a runway if the runway is safe.

Although time points in IBM ILOG CPLEX CP Optimizer are internally represented as integers, the optimal solution of the CP model is equivalent to the optimal solution of a time-continuous model as long as all time parameters are integer. Since all time parameters can be converted to integers by multiplying with a large constant factor, the CP model can be considered to solve the time-continuous problem.

5.2.3 Additional Constraints for the Constraint Programming Model

Similar to the additional constraints for the BP presented in Section 5.1, a set of valid constraints can be used to decrease the solution space of the CP model. These constraints can be derived from compulsory precedence relations between aircraft of the same operation class and between snow removals on runways.

Precedence Constraints for Aircraft of the Same Operation Class

For pairs of aircraft a and b with $a \preceq b$, Constraints (5.28) express the corresponding precedence relation.

$$\text{startBeforeStart}(\tilde{x}_a,\ \tilde{x}_b) \qquad \forall\, a, b \in \mathcal{A} : a \preceq b \tag{5.28}$$

Precedence Constraints for Snow Removals Based on Dominant Snow Removal Patterns

Given an optimal pattern of snow removals $\mathfrak{P}^* = \big((r_1^*, g_1^*, e_1^*), (r_2^*, g_2^*, e_2^*), \dots, (r_{|\mathcal{R}|}^*, g_{|\mathcal{R}|}^*, e_{|\mathcal{R}|}^*)\big)$, Constraints (5.29) assign to each runway the corresponding snow removal group and Constraints (5.30) express all obtainable precedence constraints between snow removals.

$$\text{presenceOf}(\tilde{y}_{g_i r_i}) = \texttt{TRUE} \qquad \forall\, (r_i, g_i, e_i) \in \mathfrak{P}^* \tag{5.29}$$

$$\text{startBeforeStart}(\tilde{y}_{g_i r_i},\ \tilde{y}_{g_j r_j}) \qquad \forall\, (r_i, g_i, e_i), (r_j, g_j, e_j) \in \mathfrak{P}^* : i < j \tag{5.30}$$

5.3 Enhanced Time Discretization

To apply the time-discrete BP, the continuous planning horizon $[1, T^{\max}]$ is discretized by defining a set $\mathcal{T}$ of considered time points $t \in \mathcal{T} : 1 \leq t \leq T^{\max}$. From an aircraft- and runway-specific cost function $Cost_{ar} : \mathcal{T} \to \mathbb{R}$ which calculates earliness and tardiness cost for aircraft a being scheduled on runway r at time t, cost factors C_{art} are computed as $C_{art} = Cost_{ar}(t)$.

The size of the BP, the required computational effort to solve it, and the objective function value of the optimal solution significantly depend on the chosen time discretization and the resulting cardinality of $\mathcal{T}$. If the number $|\mathcal{T}|$ of considered time points increases, then the number of variables x_{art} and y_{grt} as well as the number of constraints (5.4)–(5.6) grows. Increasing $|\mathcal{T}|$, in general, increases the required computational effort to solve the model but can lead to better optimal objective function values, which are closer to an optimal time-continuous solution. The opposite is true for decreasing the number $|\mathcal{T}|$ of considered time points and, with that, the number of variables and constraints of the model. If $|\mathcal{T}|$ is decreased, the model becomes easier and faster to solve, but, in general, optimal objective function values become worse due to a loss of time granularity. If all time parameters of the model are integer, the time-discrete model with a step size of one between considered time points is equivalent to the time-continuous model.

This thesis proposes the following time discretization approach to keep $|\mathcal{T}|$ small while, at the same time, enabling good optimal objective function values close to time-continuous solutions (cf. Figure 5.1):

1. Start with a general time discretization by considering equidistant time points $t \in \mathcal{T}$ which span the planning horizon with a constant step size.

2. For each aircraft a and runway r, add the aircraft's target time T_{ar} to the set $\mathcal{T}_{ar}$ of considered time points for aircraft a on runway r and to the set $\mathcal{T}$. This opens the possibility to schedule aircraft a on runway r exactly at its target time T_{ar} and, thus, to avoid any earliness or tardiness cost. It eliminates the need to deviate from T_{ar} if an optimal solution with $x_{art} = 1$ exists.

3. Similarly, for each aircraft a and runway r, add the time $x^{\mathrm{CP}}_{ar} = \mathrm{startOf}(\tilde{x}_{ar})$

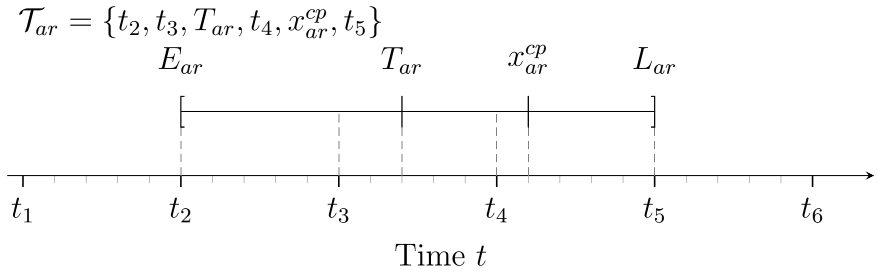


Figure 5.1: Enhanced time discretization with resulting time window T_{ar} of aircraft a on runway r

as computed by the CP start heuristic to the sets $\mathcal{T}_{ar}$ and $\mathcal{T}$. This adds the solution of the CP start heuristic to the BP solution space. Notably, this adds the optimal time-continuous solution to the BP solution space if it is found by the CP start heuristic.

The proposed enhanced time discretization approach often enables significantly better solutions or even the optimal time-continuous solution while adding only marginal complexity, i.e., only a few variables and constraints, to the model.

5.4 Exact Solution Approach

This thesis proposes an exact algorithm to solve the time-discrete WRSP to proven optimality. This section details all steps of the proposed algorithm. In a preprocessing step, dominant snow removal patterns are computed to reduce the number of variables and the solution space. Constraint propagation based on the CP model reduces variable domains. The CP formulation is also used to generate an initial (incumbent) solution heuristically using a standard CP optimization engine. This incumbent solution from the CP start heuristic provides the basis for the enhanced time discretization. It is also used to initialize a column generation scheme which solves the LP relaxation of the BP optimally. The presented column generation approach identifies all variables which are potentially required to solve the BP to integer optimality, resulting in a column-reduced BP. In the last step of the algorithm, a branch-and-bound procedure solves the column-reduced BP optimally. Figure 5.2 gives an overview of the proposed approach.

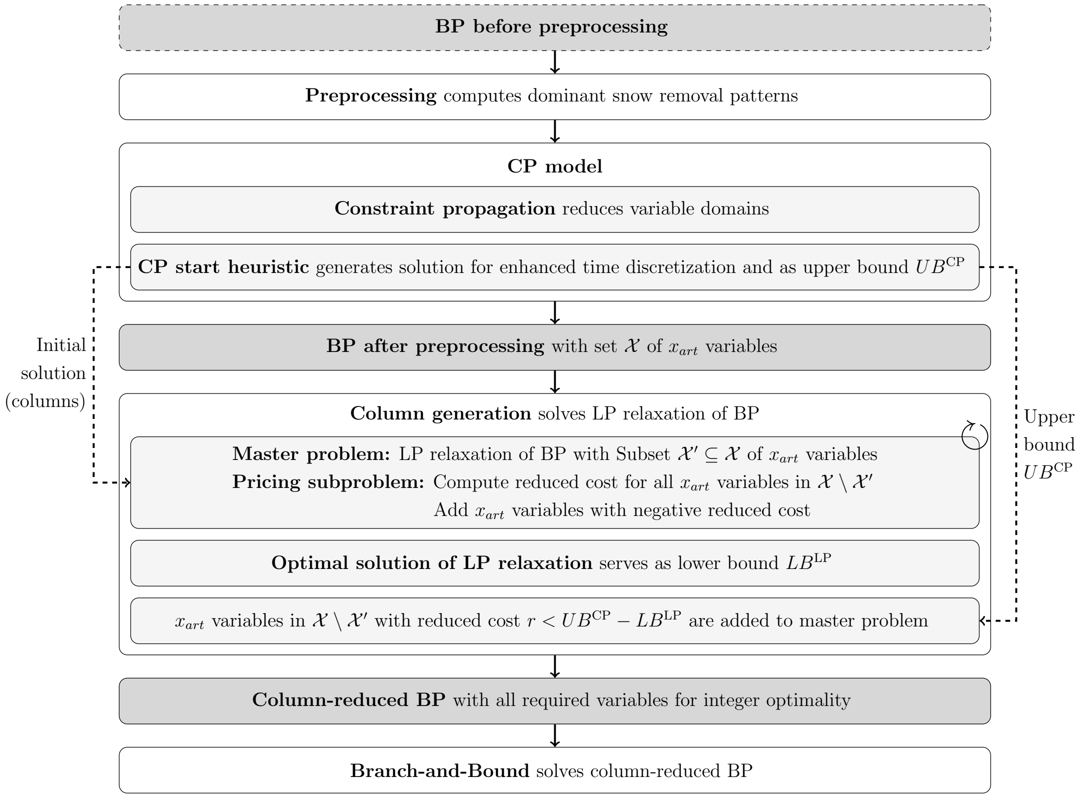


Figure 5.2: Overview of the exact approach using constraint programming and column generation

A key feature of the algorithm is that it decreases the model size, i.e., the number of variables and constraints, of the time-discrete BP in order to accelerate the final branch-and-bound procedure. The proposed preprocessing, the constraint propagation, and the column generation scheme primarily aim at reducing the number of variables in the model. This is particularly efficient, since, in the proposed model formulation, a reduction of x_{art} and y_{grt} variables usually decreases the sizes of sets $\mathcal{T}_{ar}$, $\mathcal{T}_{rg}$, $\mathfrak{V}_{rt}$, and $\mathfrak{W}_{gt}$. This, subsequently, reduces the number of constraints (5.4)–(5.6).

5.4.1 Preprocessing

In a preprocessing step, the concept of dominant snow removal patterns is used to reduce the number of y_{grt} variables in order to decrease the solution space. Given a dominant snow removal pattern $\mathfrak{P}^* = \big((r_1^*, g_1^*, e_1^*), (r_2^*, g_2^*, e_2^*), \ldots, (r_{|\mathcal{R}|}^*, g_{|\mathcal{R}|}^*, e_{|\mathcal{R}|}^*)\big)$, only corresponding precomputed combinations of runways, snow removal groups and earliest possible snow removal times have to be considered in order to solve the WRSP optimally. Therefore, y_{grt} variables can be restricted to all tuples (g, r, t) with $(r, g, e) \in \mathfrak{P}^*$ and $e \leq t$.

5.4.2 Constraint Propagation

Based on the CP model, constraint propagation computes possible assignments of aircraft to runways and tightens time windows for aircraft and for snow removals. To achieve this, constraint propagation makes logical deductions about the presence and possible domains of $\tilde{x}_{ar}$ and $\tilde{y}_{gr}$ interval variables. The proposed algorithm includes only binary variables x_{art} and y_{grt} in the time-discrete BP if the corresponding interval variables in the CP model are not reduced by the constraint propagation procedure. This significantly reduces the number of y_{grt} and x_{art} variables in the BP model. Furthermore, all time points t, which cannot be the start time of any interval variable, are also be excluded from $\mathcal{T}$. This reduces the number of constraints (5.4)–(5.6) in the BP and, thus, its model size and complexity.

5.4.3 Start Heuristic Based on Constraint Programming

A CP optimization engine is used to solve the CP model of the WRSP heuristically in order to derive an initial solution and upper bound UB^{CP} for the problem. The CP solution procedure is terminated after a given limit of failed tries to construct a solution or after a given time limit. To adjust for the complexity of the instances to be solved, these limits are based on the number of considered aircraft and runways. The resulting, best found (time-continuous) solution of the CP model is used for the enhanced time discretization approach. It is included in the BP solution space by adding the corresponding time points $x^{\text{CP}}_{ar} = \text{startOf}(\tilde{x}_{ar})$ to $\mathcal{T}_{ar}$ and $\mathcal{T}$ as described in Section 5.3. The solution from the CP start heuristic is also used to construct initial columns for the column generation scheme.

5.4.4 Column Generation Scheme

After reducing the number of variables during preprocessing, the resulting BP still has a large number of x_{art} variables. Most of them are non-basic in the optimal solution. In the following, $\mathcal{X}$ denotes the full set of x_{art} variables after preprocessing. A column generation procedure identifies a subset $\mathcal{X}' \subseteq \mathcal{X}$ which includes all x_{art} variables required to solve the BP to optimality.

The master problem of the column generation scheme is the LP relaxation of the BP containing all y_{grt} and z_r variables, but only x_{art} variables with $x_{art} \in \mathcal{X}'$. Since the number of x_{art} variables exceeds the number of y_{grt} and z_r variables by far, the column generation scheme focuses on the generation of x_{art} variables. The master problem is initialized with the subset $\mathcal{X}'$ of x_{art} variables corresponding to the solution of the CP start heuristic. In each iteration of the column generation scheme, the pricing step computes reduced cost for all x_{art} variables in $\mathcal{X} \backslash \mathcal{X}'$. This pricing procedure can be implemented very efficiently since the number of x_{art} variables in $\mathcal{X} \backslash \mathcal{X}'$ is finite and the pricing of each variable is solely a linear combination of respective dual variables. In each iteration of the column generation procedure, x_{art} variables with negative reduced cost enter the variable set $\mathcal{X}'$ for the next iteration. The column generation terminates if no more variables with negative reduced cost are found. The final solution of the master problem after the last iteration of the column generation procedure is the optimal solution for the LP relaxation of the BP and, thus, constitutes a lower bound LB^{LP} for the BP.

5.4.5 Deriving an Optimal Solution

After the last iteration of the column generation procedure, when no more variables with negative reduced cost are found in the pricing step, all variables required to optimally solve the master problem, i.e., the LP relaxation of the BP, have been generated. Solving the BP to integer optimality, however, could require additional variables from set $\mathcal{X} \backslash \mathcal{X}'$. For all variables $x_{art} \in \mathcal{X} \backslash \mathcal{X}'$, their reduced cost $\bar{c}_{x_{art}}$ describe their contribution to the objective function value of the master problem if these variables become basic. Given the lower bound LB^{LP} from the LP relaxation and the upper bound UB^{CP} from the CP start heuristic, it is obvious that variables $x_{art} \in \mathcal{X} \backslash \mathcal{X}'$ with $LB^{\text{LP}} + \bar{c}_{x_{art}} \geq UB^{\text{CP}}$ cannot be required for the optimal integer solution. Note that all variables which are required to construct the solution from the CP start heuristic resulting in UB^{CP} are present in the master problem since the column generation scheme was initialized with these variables. To secure that also all variables potentially required for integer optimality of the BP are present in the master problem, all variables $x_{art} \in \mathcal{X} \setminus \mathcal{X}'$ with $LB^{\mathrm{LP}} + \bar{c}_{x_{art}} < UB^{\mathrm{CP}}$ are added to the master problem, resulting in the column-reduced BP. Finally, a standard branch-and-bound procedure of a MIP solver computes the optimal solution for the integer variant of the column-reduced BP.

The time-discrete solution approach presented in this chapter is tested and evaluated in a computational study in the next Chapter.

6 Computational Results

To show the applicability and efficiency of the presented approaches, both methods were tested on real-world data from Munich International Airport. Section 6.1 describes the setup and design of the computational study. In Section 6.2, a practice-oriented benchmark heuristic is introduced which mimicks the scheduling procedure applied manually by runway managers and air traffic controllers. Section 6.3 details the computational results for the time-continuous approach based on the MIP model. Optimal schedules are compared against benchmark solutions of the practice-oriented heuristic with regard to objective function values and improvements. This section also evaluates the time-continuous approach and its corresponding pruning rules, valid inequalities, and start solution heuristic with regard to computational efficiency. Section 6.4 presents computational results for the time-discrete approach using constraint programming and a column generation method. It shows that the enhanced time discretization approach enables good solutions even for larger time steps and that it provides a good balance between model size and solution quality. An analysis of resulting model sizes demonstrates that the proposed preprocessing, constraint propagation, and column generation scheme significantly reduce the number of variables and, subsequently, also the number of constraints in the time-indexed BP. Additionally, computational times for the time-discrete approach are presented including detailed runtime analyses of all components of the solution method. Section 6.5 compares the time-continuous and the time-discrete approach with regard to optimal solutions and computational times. Finally, this chapter concludes with a discussion regarding the practical applicability of both proposed solution methodologies.

All instances of the computational study were computed on an Intel i7-8700K processor with 3.7 GHz and 32 GB RAM using Python 3.6 with Gurobi 8.1 as MIP solver and IBM ILOG CPLEX CP Optimizer for constraint propagation and as CP optimization engine.

6.1 Study Design

The computational study considers multiple real-world instances with various combinations of setups and parameters. This section describes the underlying data set. It also explains the different parameters and assumptions used in the computational study.

Data Sets

All instances were generated from a publicly available flight database obtaining real arrival and departure data of a winter day at Munich International Airport. As a typical hub airport, Munich International Airport has time windows in which mainly domestic flights arrive and depart and time windows in which additional long-distance flights are processed. The computational study considers three data sets with real flight operations recorded in November 2017, each with a representative mix of arriving and departing, domestic and long-distance flights. The data sets reflect different times of the day: 9 a.m. - 10.30 a.m. (data set “morning”), 12 p.m. - 1.30 p.m. (data set “noon”), and 9 p.m. - 10.30 p.m. (data set “evening”).

Aircraft Time Windows

The computational study considers two types of time windows for aircraft. Both are used in existing literature on runway scheduling. First, it considers instances without earliness, where an aircraft’s target time constitutes its earliest possible take-off or landing time. Second, it considers instances allowing earliness, where aircraft can be scheduled up to 10 minutes ahead of their target time causing respective earliness cost. In both cases, it allows aircraft delays (tardiness) of up to 20 minutes after the respective target time.

Cost Functions

The computational study considers two established types of cost functions for aircraft delay (cf. Figure 6.1). The cost function type “linear” assumes that earliness cost (if earliness is permitted) and tardiness cost are linearly related to the

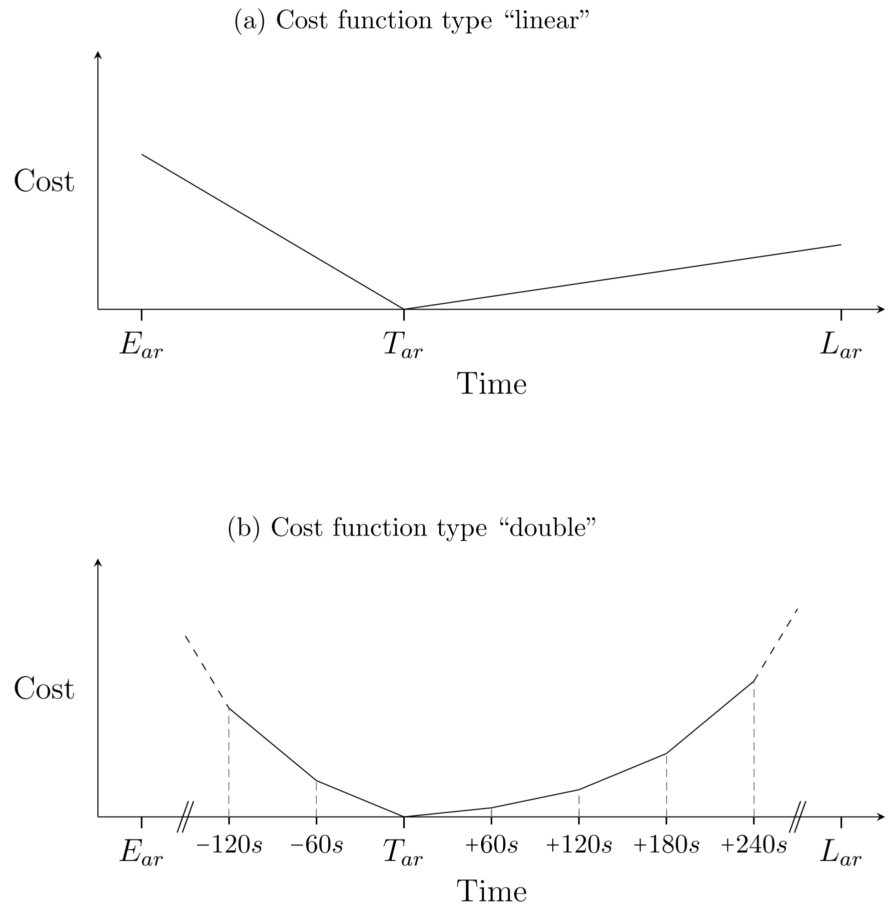


Figure 6.1: Cost functions for aircraft delay

deviation from an aircraft’s target time. In practice, earliness often causes higher cost than tardiness due to higher fuel consumption at higher aircraft speeds. The cost increase rate, i.e., the slope of the cost curve, is assumed to be 50% higher for earliness than for tardiness. To take into account that delays of larger aircraft with a higher number of passengers are more critical, the study assumes cost increase rates for tardiness of one, two, three, and four monetary units per second for aircraft of the classes “Large”, “Boeing 757”, “Heavy”, and “Super” respectively. For the cost function type “double”, the cost increase rate of the linear cost function doubles every minute for earliness cost as well as for tardiness cost.

Table 6.1: Transit times between runways (in seconds)

	Succeeding runway		
Preceding runway	Runway 1	Runway 2	Runway 3
Runway 1	-	600	1,200
Runway 2	900	-	1,200
Runway 3	1,500	1,500	-

Runways and Snow Removal Groups

The computational study considers instances with either two runways and one snow removal group or with three runways and two snow removal groups. It considers sets of homogeneous and independent runways, i.e., that runways have the same length and allow for the same flight operations and that flight operations on one runway do not affect flight operations on other runways. Transit times between runways are based on the physical layout and road network of Munich International Airport and are shown in Table 6.1. For instances with three runways, this thesis considers an extension of the existing runway system as approved and published by the Bavarian state authorities (cf. Regierung von Oberbayern, 2011).

Aircraft and Flight Density

Instances of the computational study cover 30, 45, 60, or 75 aircraft to be scheduled. For instances with two runways and one snow removal group, a flight density of 45 flight operations per hour is assumed. For instances with three runways and two snow removal groups, 60 flight operations per hour are assumed. This corresponds to planning horizons between 49 minutes and 128 minutes.

Snowfall Scenarios

The computational study considers three different scenarios regarding snowfall.

In the scenario "beginning snowfall", all runways have the same initial conditions at the start of the planning horizon, and, due to beginning snowfall, operations become unsafe at the same point in time on all runways ($U_r = U_{r'} \; \forall \, r, r' \in$

$\mathcal{R} : r \neq r'$). In the computational study, runways become unsafe 25 minutes after the start of the planning horizon.

In the scenario “continuous winter operations”, runways have previously been cleared from snow, ice, and slush at different times. Thus, the times at which runways become unsafe mainly depend on the times elapsed since the previous snow removals and, consequently, runways become unsafe at different times ($U_r \neq U_{r'}\ \forall\, r, r' \in \mathcal{R} : r \neq r'$). In the computational experiments, the first runway becomes unsafe 10 minutes after the start of the planning horizon and the second runway becomes unsafe after 25 minutes. In case of three runways, the third runway becomes unsafe after 40 minutes.

In the scenario “ending snowfall”, runways have previously been cleared and not all runways become unsafe during the planning horizon as snowfall is about to end. In case of three runways, two runways become unsafe after 10 and 25 minutes respectively. In case of two runways, one runway becomes unsafe after 25 minutes. To reflect the lower demand for snow removals in this scenario, one snow removal group is used regardless of the number of runways.

6.2 Heuristic Scheduling Approaches

To show the efficiency of the proposed optimization approaches for the WRSP, heuristic scheduling approaches have been developed as benchmark methods. A naive scheduling approach schedules aircraft on a FCFS basis according to target times and closes runways for snow removals as soon as they become unsafe. This leads, in general, to high aircraft delays and is, therefore, not applicable in practice. The practice-oriented benchmark heuristic is based on discussions with decision makers of runway management and air traffic control at Munich International Airport. It simulates and mimicks the actual decision making process of human managers and serves as an approximation to decisions observed in practice. Therefore, schedules computed by the practice-oriented heuristic are used as benchmark solutions for the proposed optimization approaches.

6.2.1 Naive Scheduling Approach

A simple and naive heuristic method schedules aircraft and snow removals on runways according to a FCFS order. Aircraft are scheduled in the order of their target time on the first runway which is available. As soon as a runway becomes unsafe, it is closed for snow removal and the first available snow removal group is used to clear the runway. This often leads to situations where runways have to be closed without having a snow removal group available for clearing it. As a consequence, runway capacity significantly deteriorates and large aircraft delays occur causing high operational cost. Therefore, most airports follow more advanced heuristic approaches yielding significantly better schedules.

6.2.2 Practice-Oriented Benchmark Heuristic

To derive meaningful insights regarding the advantages and efficiency gains of the exact methods proposed in this thesis, optimal schedules of the time-continuous and the time-discrete optimization approach are compared against schedules generated by a practice-oriented heuristic. This heuristic mimicks the decisions of human planners at airports and air traffic control resulting in runway schedules which are very similar to schedules observed in practice. The following practice-oriented benchmark heuristic is based on observations and interviews with practitioners at airports and follows a stepwise sequential approach. First, it schedules snow removals. Then, it schedules aircraft based on the resulting runway availability:

Step 1 Schedule snow removals by sequentially applying the following two rules:

1. *Maximize runway availability* in order to increase the capacity for scheduling aircraft. This is equivalent to minimizing the overall time at which runways are unavailable because they are unsafe or are being cleared.
2. *Start snow removals as late as possible* in order to postpone the need for the next snow removals in the next planning period.

Step 2 Assign runways and take-off or landing times to aircraft by scheduling them in parallel on all runways on a FCFS basis according to their target time.

The computational study additionally considers a variant of this heuristic in which the optimization model of Beasley et al. (2000) is used to schedule aircraft optimally in Step 2.

For more complex instances with at least three runways and two snow removal groups, the solution space for scheduling snow removals rapidly grows. In these cases, the benchmark heuristic computes snow removal schedules which are typically not found by human planners. Consequently, the heuristic often constructs better schedules than human planners and solutions of the benchmark heuristic constitute lower bounds for manually created solutions.

6.3 Results for the Time-Continuous Approach Using Mixed-Integer Programming

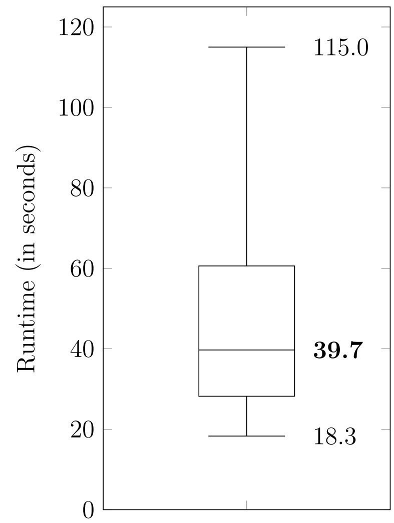
This section presents computational results for the time-continuous WRSP presented in Chapter 4.

For the time-continuous approach, 24 instances are considered: The scenario “beginning snowfall” is computed on the data set “morning”, scenarios “continuous winter operations” and “ending snowfall” are based on data sets “noon” and “evening” respectively. All scenarios are solved with two and three runways and for 30, 45, 60, and 75 aircraft. All instances consider linear cost functions and no aircraft acceleration to keep them computationally tractable. Table 6.2 provides a complete list of configurations and parameters for all instances considered for the time-continuous approach.

When solving the same problem instance multiple times, a high variance in computational times became apparent. This variance resulted from the degeneracy in the LP relaxation of the MIP. If a problem’s LP relaxation has multiple optimal solutions, the MIP solver chooses one of these LP solutions randomly. Resulting branching decisions change the search tree and, thus, lead to varying computational times. In order to derive meaningful conclusions about the performance of the proposed approach, each instance of the computational study was solved 60 times with varying random seeds for the MIP solver. Figure 6.2 exemplary shows the distribution of solver runtimes for instance *T/l/begin/m/2/75* using all pruning rules and valid inequalities. In general, computational times

Table 6.2: Configurations and parameters of instances for the time-continuous approach

Instance	Aircraft time windows (in seconds)	Cost function	Snowfall scenario	Data set	Runways (snow removal groups) / aircraft	Unsafe times U_1 / U_2 / U_3 (in seconds)	Required snow removal times P_1 / P_2 / P_3 (in seconds)	Departing aircraft of class "Large" / "Boeing 757" / "Heavy" / "Super" | landing aircraft of class "Large" / "Boeing 757" / "Heavy" / "Super"	Planning horizon $\max_{a \in \mathcal{A}}\{L_a\}$ (in seconds)
T/l/begin/m/2/30	$[T, T + 1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	2 (1) / 30	1,500 / 1,500 / -	1,200 / 1,200 / -	14 / 0 / 2 / 0 | 12 / 1 / 1 / 0	3,664
T/l/begin/m/2/45	$[T, T + 1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	2 (1) / 45	1,500 / 1,500 / -	1,200 / 1,200 / -	19 / 0 / 2 / 0 | 21 / 1 / 2 / 0	4,789
T/l/begin/m/2/60	$[T, T + 1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	2 (1) / 60	1,500 / 1,500 / -	1,200 / 1,200 / -	25 / 1 / 2 / 0 | 28 / 1 / 3 / 0	5,593
T/l/begin/m/2/75	$[T, T + 1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	2 (1) / 75	1,500 / 1,500 / -	1,200 / 1,200 / -	31 / 2 / 2 / 0 | 35 / 1 / 4 / 0	6,608
T/l/begin/m/3/30	$[T, T + 1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	3 (2) / 30	1,500 / 1,500 / 1,500	1,200 / 1,200 / 1,200	16 / 0 / 1 / 0 | 8 / 1 / 4 / 0	3,097
T/l/begin/m/3/45	$[T, T + 1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	3 (2) / 45	1,500 / 1,500 / 1,500	1,200 / 1,200 / 1,200	21 / 0 / 2 / 0 | 16 / 1 / 5 / 0	3,939
T/l/begin/m/3/60	$[T, T + 1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	3 (2) / 60	1,500 / 1,500 / 1,500	1,200 / 1,200 / 1,200	24 / 0 / 2 / 0 | 27 / 1 / 6 / 0	4,819
T/l/begin/m/3/75	$[T, T + 1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	3 (2) / 75	1,500 / 1,500 / 1,500	1,200 / 1,200 / 1,200	31 / 1 / 2 / 0 | 33 / 1 / 7 / 0	5,346
T/l/cont/n/2/30	$[T, T + 1200]$	linear	Continuous winter operations	"noon" / 12 p.m.	2 (1) / 30	600 / 1,500 / -	1,200 / 1,200 / -	12 / 0 / 6 / 0 | 10 / 0 / 1 / 1	3,600
T/l/cont/n/2/45	$[T, T + 1200]$	linear	Continuous winter operations	"noon" / 12 p.m.	2 (1) / 45	600 / 1,500 / -	1,200 / 1,200 / -	21 / 0 / 9 / 0 | 13 / 0 / 1 / 1	4,622
T/l/cont/n/2/60	$[T, T + 1200]$	linear	Continuous winter operations	"noon" / 12 p.m.	2 (1) / 60	600 / 1,500 / -	1,200 / 1,200 / -	28 / 0 / 9 / 0 | 20 / 0 / 2 / 1	5,726
T/l/cont/n/2/75	$[T, T + 1200]$	linear	Continuous winter operations	"noon" / 12 p.m.	2 (1) / 75	600 / 1,500 / -	1,200 / 1,200 / -	32 / 0 / 10 / 0 | 30 / 0 / 2 / 1	7,689
T/l/cont/n/3/30	$[T, T + 1200]$	linear	Continuous winter operations	"noon" / 12 p.m.	3 (2) / 30	600 / 1,500 / 2,400	1,200 / 1,200 / 1,200	12 / 0 / 5 / 0 | 11 / 1 / 0 / 1	2,970
T/l/cont/n/3/45	$[T, T + 1200]$	linear	Continuous winter operations	"noon" / 12 p.m.	3 (2) / 45	600 / 1,500 / 2,400	1,200 / 1,200 / 1,200	21 / 0 / 7 / 0 | 14 / 1 / 1 / 1	3,872
T/l/cont/n/3/60	$[T, T + 1200]$	linear	Continuous winter operations	"noon" / 12 p.m.	3 (2) / 60	600 / 1,500 / 2,400	1,200 / 1,200 / 1,200	28 / 0 / 12 / 0 | 17 / 1 / 1 / 1	4,647
T/l/cont/n/3/75	$[T, T + 1200]$	linear	Continuous winter operations	"noon" / 12 p.m.	3 (2) / 75	600 / 1,500 / 2,400	1,200 / 1,200 / 1,200	35 / 1 / 12 / 0 | 22 / 1 / 3 / 1	5,459
T/l/end/e/2/30	$[T, T + 1200]$	linear	Ending snowfall	"evening" / 9 p.m.	2 (1) / 30	1,500 / - / -	1,200 / - / -	9 / 0 / 2 / 0 | 18 / 1 / 0 / 0	3,489
T/l/end/e/2/45	$[T, T + 1200]$	linear	Ending snowfall	"evening" / 9 p.m.	2 (1) / 45	1,500 / - / -	1,200 / - / -	16 / 0 / 3 / 1 | 22 / 2 / 0 / 1	4,645
T/l/end/e/2/60	$[T, T + 1200]$	linear	Ending snowfall	"evening" / 9 p.m.	2 (1) / 60	1,500 / - / -	1,200 / - / -	23 / 0 / 5 / 1 | 28 / 2 / 0 / 1	5,962
T/l/end/e/2/75	$[T, T + 1200]$	linear	Ending snowfall	"evening" / 9 p.m.	2 (1) / 75	1,500 / - / -	1,200 / - / -	34 / 0 / 5 / 1 | 32 / 2 / 0 / 1	7,291
T/l/end/e/3/30	$[T, T + 1200]$	linear	Ending snowfall	"evening" / 9 p.m.	3 (2) / 30	1,500 / - / 600	1,200 / - / 1,200	10 / 0 / 2 / 0 | 18 / 0 / 0 / 0	3,058
T/l/end/e/3/45	$[T, T + 1200]$	linear	Ending snowfall	"evening" / 9 p.m.	3 (2) / 45	1,500 / - / 600	1,200 / - / 1,200	15 / 0 / 2 / 0 | 27 / 1 / 0 / 0	3,841
T/l/end/e/3/60	$[T, T + 1200]$	linear	Ending snowfall	"evening" / 9 p.m.	3 (2) / 60	1,500 / - / 600	1,200 / - / 1,200	21 / 0 / 3 / 1 | 32 / 2 / 0 / 1	4,683
T/l/end/e/3/75	$[T, T + 1200]$	linear	Ending snowfall	"evening" / 9 p.m.	3 (2) / 75	1,500 / - / 600	1,200 / - / 1,200	29 / 0 / 5 / 1 | 37 / 2 / 0 / 1	5,635



	Quartile				
	min	.25	**.50**	.75	max
Runtime (in seconds)	18.3	28.2	**39.7**	60.6	115.0

Figure 6.2: Examplary distribution of runtimes for instance *T/l/begin/m/2/75*

between the .25-quartile and .75-quartile are closely centered around the median. For some random seeds, considerably lower or higher computational times were measured with upper outliers deviating considerably stronger.

Throughout the following sections, computational times are reported as the median of runtimes over all random seeds.

Table 6.3 presents results for all computed instances. For each instance, the table shows the objective function value z^{naive} of the naive scheduling approach and the objective function value $z^{\text{heu/FCFS}}$ of the practice-oriented heuristic as a benchmark solution. It also shows the objective function value $z^{\text{heu/opt}}$ of a variant of the practice-oriented heuristic which uses the optimization model of Beasley et al. (2000) to schedule aircraft optimally. $\Delta^{\text{heu/opt}} = (z^{\text{heu/FCFS}} - z^{\text{heu/opt}})/z^{\text{heu/FCFS}}$ measures the improvement through state-of-the-art aircraft scheduling over the FCFS-based aircraft scheduling approach used in practice. It assumes that snow removals are scheduled with the practice-oriented heuristic. The table also reports the objective function value z^{MIP} of the integrated time-continuous MIP and the improvement $\Delta^{\text{MIP}} = (z^{\text{heu/FCFS}} - z^{\text{MIP}})/z^{\text{heu/FCFS}}$ over the practice-oriented

Table 6.3: Computational results for the time-continuous approach

Instance	z^{naive}	$z^{heu/FCFS}$	$z^{heu/opt}$ / $\Delta^{heu/opt}$	z^{MIP} / Δ^{MIP}	Computational times of different configurations (in seconds)				
					C1 (MIP)	C2 (MIP + PR I and II)	C3 (MIP + PR I–III)	C4 (MIP + PR I–III + VI I and II)	Solution with less than 1% deviation from optimal value
T/l/begin/m/2/30	15,757	2,415	1,726 / 29%	1,710 / 29%	41	1	0	0	0 (+0)*
T/l/begin/m/2/45	31,466	3,592	2,933 / 18%	2,933 / 18%	>3,600	29	6	6	1 (+0)
T/l/begin/m/2/60	49,761	3,732	3,070 / 18%	3,070 / 18%	>3,600	112	21	20	1 (+0)
T/l/begin/m/2/75	62,083	3,804	3,142 / 17%	3,142 / 17%	>3,600	206	44	40	2 (+1)
T/l/begin/m/3/30	11,057	1,027	1,006 / 2%	1,006 / 2%	>3,600	48	4	3	1 (+1)*
T/l/begin/m/3/45	25,492	3,655	2,825 / 23%	1,683 / 54%	>3,600	> 3,600	627	601	2 (+4)
T/l/begin/m/3/60	32,837	4,179	3,185 / 24%	1,747 / 58%	>3,600	> 3,600	1,306	1,015	8 (+7)*
T/l/begin/m/3/75	38,476	4,198	3,201 / 24%	1,763 / 58%	>3,600	> 3,600	1,728	1,680	14 (+28)*
T/l/cont/n/2/30	7,423	4,034	3,157 / 22%	3,157 / 22%	961	2	4	4	2 (+0)
T/l/cont/n/2/45	12,192	6,150	4,352 / 29%	4,352 / 29%	>3,600	19	15	14	5 (+0)*
T/l/cont/n/2/60	15,690	6,451	4,522 / 30%	4,522 / 30%	>3,600	42	43	45	8 (+1)
T/l/cont/n/2/75	15,752	6,513	4,584 / 30%	4,584 / 30%	>3,600	53	47	48	9 (+1)
T/l/cont/n/3/30	1,225	499	499 / -	499 / -	31	2	2**	3**	1 (+0)**
T/l/cont/n/3/45	2,653	1,268	1,266 / -	634 / 50%	1757	29	29**	24**	1 (+4)**
T/l/cont/n/3/60	2,901	1,529	1,491 / 2%	707 / 54%	>3,600	95	95**	76**	2 (+7)**
T/l/cont/n/3/75	2,982	1,610	1,548 / 4%	739 / 54%	>3,600	216	216**	191**	3 (+7)**
T/l/end/e/2/30	721	721	700 / 3%	700 / 3%	31	3	2	3	1 (+0)
T/l/end/e/2/45	1,400	1,400	1,176 / 16%	891 / 36%	203	7	7	7	1 (+1)
T/l/end/e/2/60	1,474	1,474	1,221 / 17%	936 / 36%	272	9	8	9	1 (+1)
T/l/end/e/2/75	1,819	1,819	1,566 / 14%	1,239 / 32%	>3,600	242	214	215	7 (+2)
T/l/end/e/3/30	735	387	261 / 33%	261 / 33%	21	2	1	2	1 (+0)
T/l/end/e/3/45	838	490	364 / 26%	364 / 26%	219	25	6	7	1 (+1)
T/l/end/e/3/60	931	531	405 / 24%	405 / 24%	1,240	43	14	13	2 (+2)
T/l/end/e/3/75	960	560	418 / 25%	418 / 25%	1,879	56	20	19	2 (+2)

* start heuristic does not yield a feasible start solutions   ** pruning rules of type III do not yield an optimal snow removal schedule

PR I, II, III = pruning rules of type I, II, III   VI I, II= valid inequalities I, II heuristic. For the time-continuous MIP formulation, computational times for four model configurations are reported:

- Configuration C1: MIP as defined by (4.1)–(4.12)
- Configuration C2: MIP and pruning rules of type I and II
- Configuration C3: MIP and pruning rules of type I–III
- Configuration C4: MIP, pruning rules of type I–III, and valid inequalities I and II

In the computational experiments, it became evident that good solutions are computed early during the branch-and-bound process. Therefore, the last column of Table 6.3 shows required times to compute solutions which deviate less than 1% from the optimal objective function value z^{MIP} using configuration C4 and the start solution heuristic. Computational times of the start heuristic are reported in brackets.

6.3.1 Reduction of Weighted Delay Through Integrated Planning

An analysis of z^{naive} shows that, for most instances, the naive scheduling approach constructs solutions significantly worse than all other considered variants and is not suitable for an application in practice.

The comparison of the objective function values of the practice-oriented heuristic $z^{\mathrm{heu/FCFS}}$ and the optimization model z^{MIP} shows that the integrated approach significantly reduces weighted delay. In 23 out of 24 instances, improvements Δ^{MIP} of up to 58% are achieved. These improvements reflect the benefit of an optimal aircraft schedule with integrated snow removal decisions. Improvements $\Delta^{\mathrm{heu/opt}}$ show that using an optimal aircraft schedule instead of a FCFS-based aircraft schedule within the practice-oriented sequential heuristic reduces weighted delay only up to 33% (on avg. 18%). Comparing Δ^{MIP} and $\Delta^{\mathrm{heu/opt}}$ indicates that, for many instances, substantial reductions of delay cost originate from an integration of the scheduling decisions for snow removals and aircraft. Since solutions of the benchmark heuristic represent lower bounds for solutions created by human planners, actual improvements by the integrated approach are most likely even higher.

6.3.2 Improvements of Computational Times Through Pruning Rules, Valid Inequalities, and the Start Solution Heuristic

Computational results show that for 13 out of 24 instances, the original MIP formulation (C1) does not yield a proven optimal solution within a time limit of one hour.

For all instances, significant improvements of computational times can be observed if pruning rules are applied. The use of pruning rules of type I and II (C2) allows for solving 21 of 24 instances to optimality within one hour.

Pruning rules of type III (C3), which compute optimal snow removal patterns, allow for solving all instances to optimality. For 18 instances, runtimes are less than one minute. Additionally, these pruning rules reduce the computational times by up to 92% in case of beginning snowfall and by up to 76% in case of ending snowfall. Under continuous winter operations, they yield optimal snow removal patterns only for two runways and one snow removal group. In these cases, they reduce computational times by up to 21%.

For six instances, valid inequalities (C4) yield a significant speed-up of at least 10%. Especially for instances *T/l/cont/n/3/45*, *T/l/cont/n/3/60*, and *T/l/cont/n/3/75*, where pruning rules III do not yield optimal snow removal patterns, the proposed valid inequalities improve computational times by up to 20%. For all six instances which cannot be solved optimally within one minute, schedules with less than 1% deviation from the optimal objective function value are computed in less than one minute. With the start heuristic, initial start solutions can be computed within a few seconds. These are particularly helpful to find good solutions early in the branch-and-bound procedure. As a result, it is possible to use the presented approach heuristically by terminating the computation before a proven optimum with an optimality gap of zero is obtained.

6.4 Results for the Time-Discrete Approach Using Constraint Programming and Column Generation

This section presents computational results for the time-discrete WRSP presented in Chapter 5.

The time-discrete approach is tested on 24 instances considering three different setups. The first setup assumes linear cost functions and no aircraft acceleration, which is congruent to the assumptions for the time-continuous approach in Section 6.3. The second setup also assumes linear cost functions but allows aircraft acceleration and, thus, earliness. The third setup assumes cost functions of type "double" and allows aircraft acceleration. Each setup is solved for two and three runways and for 45 and 75 aircraft assuming beginning snowfall and continuous winter operations. All instances are based on data set "morning". Table 6.4 shows the complete list of configurations and parameters for all instances considered for the time-discrete approach.

For the CP start heuristic, a limit of $2000 \times |\mathcal{R}| \times |\mathcal{A}|$ fails was set for instances without earliness and a limit of $8000 \times |\mathcal{R}| \times |\mathcal{A}|$ fails was set for instances with earliness to account for their larger solution space.

6.4.1 Balancing Model Size and Solution Quality Through Enhanced Time Discretization

Table 6.5 proves that the proposed enhanced time discretization approach enables high quality solutions while keeping the model size comparatively small. This is shown by analyzing the number of variables and constraints and the optimal objective function values z^{TD1}, z^{TD3}, z^{TD5}, and z^{TD5e} of four different time discretization variants with various step sizes. Specifically, step sizes of one second (TD1), three seconds (TD3), and five seconds (TD5) using a standard time discretization (without special consideration of target times T_{ar} and heuristic solutions x_{ar}^{CP}) are compared to the enhanced time discretization approach (TD5e) which uses a step size of five seconds and includes target times T_{ar} and heuristic solutions x_{ar}^{CP} in the solution space. With regard to the optimal solution, TD1 is

Table 6.4: Configurations and parameters of instances for the time-discrete approach

Instance	Aircraft time windows (in seconds)	Cost function	Snowfall scenario	Data set	Runways (snow removal groups) / aircraft	Unsafe times U_1 / U_2 / U_3 (in seconds)	Required snow removal times P_1 / P_2 / P_3 (in seconds)	Departing aircraft of class "Large" / "Boeing 757" / "Heavy" $\mid$ landing aircraft of class "Large" / "Boeing 757" / "Heavy"	Planning horizon $\max_{a \in \mathcal{A}}\{L_a\}$ (in seconds)
T/l/begin/m/2/45	$[T, T+1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	2 (1) / 45	1,500 / 1,500 / -	1,200 / 1,200 / -	19 / 0 / 2 $\mid$ 21 / 1 / 2	4,789
T/l/begin/m/2/75	$[T, T+1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	2 (1) / 75	1,500 / 1,500 / -	1,200 / 1,200 / -	31 / 2 / 2 $\mid$ 35 / 1 / 4	6,608
T/l/begin/m/3/45	$[T, T+1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	3 (2) / 45	1,500 / 1,500 / 1,500	1,200 / 1,200 / 1,200	21 / 0 / 2 $\mid$ 16 / 1 / 5	3,939
T/l/begin/m/3/75	$[T, T+1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	3 (2) / 75	1,500 / 1,500 / 1,500	1,200 / 1,200 / 1,200	31 / 1 / 2 $\mid$ 33 / 1 / 7	5,346
T/l/cont/m/2/45	$[T, T+1200]$	linear	Continuous winter operations	"morning" / 9 a.m.	2 (1) / 45	2,400 / 1,500 / -	1,200 / 1,200 / -	19 / 0 / 2 $\mid$ 21 / 1 / 2	4,789
T/l/cont/m/2/75	$[T, T+1200]$	linear	Continuous winter operations	"morning" / 9 a.m.	2 (1) / 75	2,400 / 1,500 / -	1,200 / 1,200 / -	31 / 2 / 2 $\mid$ 35 / 1 / 4	6,608
T/l/cont/m/3/45	$[T, T+1200]$	linear	Continuous winter operations	"morning" / 9 a.m.	3 (2) / 45	2,400 / 1,500 / 600	1,200 / 1,200 / 1,200	21 / 0 / 2 $\mid$ 16 / 1 / 5	3,939
T/l/cont/m/3/75	$[T, T+1200]$	linear	Continuous winter operations	"morning" / 9 a.m.	3 (2) / 75	2,400 / 1,500 / 600	1,200 / 1,200 / 1,200	31 / 1 / 2 $\mid$ 33 / 1 / 7	5,346
E+T/l/begin/m/2/45	$[T-600, T+1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	2 (1) / 45	1,500 / 1,500 / -	1,200 / 1,200 / -	19 / 0 / 2 $\mid$ 21 / 1 / 2	4,789
E+T/l/begin/m/2/75	$[T-600, T+1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	2 (1) / 75	1,500 / 1,500 / -	1,200 / 1,200 / -	31 / 2 / 2 $\mid$ 35 / 1 / 4	6,608
E+T/l/begin/m/3/45	$[T-600, T+1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	3 (2) / 45	1,500 / 1,500 / 1,500	1,200 / 1,200 / 1,200	21 / 0 / 2 $\mid$ 16 / 1 / 5	3,939
E+T/l/begin/m/3/75	$[T-600, T+1200]$	linear	Beginning snowfall	"morning" / 9 a.m.	3 (2) / 75	1,500 / 1,500 / 1,500	1,200 / 1,200 / 1,200	31 / 1 / 2 $\mid$ 33 / 1 / 7	5,346
E+T/l/cont/m/2/45	$[T-600, T+1200]$	linear	Continuous winter operations	"morning" / 9 a.m.	2 (1) / 45	2,400 / 1,500 / -	1,200 / 1,200 / -	19 / 0 / 2 $\mid$ 21 / 1 / 2	4,789
E+T/l/cont/m/2/75	$[T-600, T+1200]$	linear	Continuous winter operations	"morning" / 9 a.m.	2 (1) / 75	2,400 / 1,500 / -	1,200 / 1,200 / -	31 / 2 / 2 $\mid$ 35 / 1 / 4	6,608
E+T/l/cont/m/3/45	$[T-600, T+1200]$	linear	Continuous winter operations	"morning" / 9 a.m.	3 (2) / 45	2,400 / 1,500 / 600	1,200 / 1,200 / 1,200	21 / 0 / 2 $\mid$ 16 / 1 / 5	3,939
E+T/l/cont/m/3/75	$[T-600, T+1200]$	linear	Continuous winter operations	"morning" / 9 a.m.	3 (2) / 75	2,400 / 1,500 / 600	1,200 / 1,200 / 1,200	31 / 1 / 2 $\mid$ 33 / 1 / 7	5,346
E+T/d/begin/m/2/45	$[T-600, T+1200]$	double	Beginning snowfall	"morning" / 9 a.m.	2 (1) / 45	1,500 / 1,500 / -	1,200 / 1,200 / -	19 / 0 / 2 $\mid$ 21 / 1 / 2	4,789
E+T/d/begin/m/2/75	$[T-600, T+1200]$	double	Beginning snowfall	"morning" / 9 a.m.	2 (1) / 75	1,500 / 1,500 / -	1,200 / 1,200 / -	31 / 2 / 2 $\mid$ 35 / 1 / 4	6,608
E+T/d/begin/m/3/45	$[T-600, T+1200]$	double	Beginning snowfall	"morning" / 9 a.m.	3 (2) / 45	1,500 / 1,500 / 1,500	1,200 / 1,200 / 1,200	21 / 0 / 2 $\mid$ 16 / 1 / 5	3,939
E+T/d/begin/m/3/75	$[T-600, T+1200]$	double	Beginning snowfall	"morning" / 9 a.m.	3 (2) / 75	1,500 / 1,500 / 1,500	1,200 / 1,200 / 1,200	31 / 1 / 2 $\mid$ 33 / 1 / 7	5,346
E+T/d/cont/m/2/45	$[T-600, T+1200]$	double	Continuous winter operations	"morning" / 9 a.m.	2 (1) / 45	2,400 / 1,500 / -	1,200 / 1,200 / -	19 / 0 / 2 $\mid$ 21 / 1 / 2	4,789
E+T/d/cont/m/2/75	$[T-600, T+1200]$	double	Continuous winter operations	"morning" / 9 a.m.	2 (1) / 75	2,400 / 1,500 / -	1,200 / 1,200 / -	31 / 2 / 2 $\mid$ 35 / 1 / 4	6,608
E+T/d/cont/m/3/45	$[T-600, T+1200]$	double	Continuous winter operations	"morning" / 9 a.m.	3 (2) / 45	2,400 / 1,500 / 600	1,200 / 1,200 / 1,200	21 / 0 / 2 $\mid$ 16 / 1 / 5	3,939
E+T/d/cont/m/3/75	$[T-600, T+1200]$	double	Continuous winter operations	"morning" / 9 a.m.	3 (2) / 75	2,400 / 1,500 / 600	1,200 / 1,200 / 1,200	31 / 1 / 2 $\mid$ 33 / 1 / 7	5,346

Table 6.5: Improving objective function values through enhanced time discretization

	TD1*			TD3			TD5			TD5e		
Instance	Variables	Constraints	z^{TD1}	Variables	Constraints	z^{TD3}	Variables	Constraints	z^{TD5}	Variables	Constraints	z^{TD5e}
T/l/begin/m/2/45	115,272	90,948	*2,933*	38,420	30,345	*3,013 (+ 2.7%)*	23,048	18,226	*3,147 (+ 7.3%)*	23,172	19,467	*2,933 (+ 0.0%)*
T/l/begin/m/2/75	190,970	151,981	*3,142*	63,646	50,697	*3,262 (+ 3.8%)*	38,184	30,456	*3,448 (+ 9.7%)*	38,384	33,075	*3,142 (+ 0.0%)*
T/l/begin/m/3/45	178,578	143,640	*1,683*	59,529	47,928	*1,757 (+ 4.4%)*	35,709	28,779	*1,812 (+ 7.7%)*	35,880	31,086	*1,683 (+ 0.0%)*
T/l/begin/m/3/75	295,110	245,841	*1,763*	98,367	82,002	*1,884 (+ 6.9%)*	59,013	49,245	*1,965 (+ 11.5%)*	59,274	53,637	*1,763 (+ 0.0%)*
T/l/cont/m/2/45	115,272	90,948	*2,933*	38,420	30,345	*3,013 (+ 2.7%)*	23,048	18,226	*3,147 (+ 7.3%)*	23,172	19,467	*2,933 (+ 0.0%)*
T/l/cont/m/2/75	190,970	151,981	*3,142*	63,646	50,697	*3,262 (+ 3.8%)*	38,184	30,456	*3,448 (+ 9.7%)*	38,384	33,075	*3,142 (+ 0.0%)*
T/l/cont/m/3/45	178,578	143,640	*664*	59,529	47,928	*725 (+ 9.2%)*	35,709	28,779	*777 (+ 17.0%)*	35,862	30,792	*664 (+ 0.0%)*
T/l/cont/m/3/75	295,110	245,841	*744*	98,367	82,002	*852 (+ 14.5%)*	59,013	49,245	*930 (+ 25.0%)*	59,256	53,343	*744 (+ 0.0%)*
E+T/l/begin/m/2/45	165,702	124,941	*1,966*	55,226	41,938	*1,991 (+ 1.3%)*	33,132	25,345	*2,032 (+ 3.4%)*	33,254	27,063	*2,004 (+ 1.9%)*
E+T/l/begin/m/2/75	277,400	208,259	*2,133*	92,452	70,175	*2,181 (+ 2.3%)*	55,468	42,580	*2,236 (+ 4.8%)*	55,654	45,844	*2,170 (+ 1.7%)*
E+T/l/begin/m/3/45	253,413	190,796	*1,086*	84,468	63,884	*1,119 (+ 3.0%)*	50,673	38,504	*1,147 (+ 5.6%)*	50,835	41,426	*1,093 (+ 0.6%)*
E+T/l/begin/m/3/75	423,945	330,397	*1,166*	141,306	110,875	*1,230 (+ 5.5%)*	84,777	66,988	*1,270 (+ 8.9%)*	85,031	72,775	*1,175 (+ 0.8%)*
E+T/l/cont/m/2/45	165,702	124,941	*1,966*	55,226	41,938	*1,991 (+ 1.3%)*	33,132	25,345	*2,032 (+ 3.4%)*	33,251	26,979	*1,999 (+ 1.7%)*
E+T/l/cont/m/2/75	277,400	208,259	*2,133*	92,452	70,175	*2,181 (+ 2.3%)*	55,468	42,580	*2,236 (+ 4.8%)*	55,660	45,944	*2,171 (+ 1.8%)*
E+T/l/cont/m/3/45	253,413	190,796	*428*	84,468	63,884	*466 (+ 8.9%)*	50,673	38,504	*502 (+ 17.3%)*	50,823	41,140	*430 (+ 0.5%)*
E+T/l/cont/m/3/75	423,945	330,397	*586*	141,306	110,875	*653 (+ 11.4%)*	84,777	66,988	*705 (+ 20.3%)*	85,023	72,523	*592 (+ 1.0%)*
E+T/d/begin/m/2/45	165,702	124,941	*2,373*	55,226	41,938	*2,405 (+ 1.3%)*	33,132	25,345	*2,464 (+ 3.8%)*	33,254	27,024	*2,413 (+ 1.7%)*
E+T/d/begin/m/2/75	277,400	208,259	*2,540*	92,452	70,175	*2,595 (+ 2.2%)*	55,468	42,580	*2,667 (+ 5.0%)*	55,659	45,900	*2,585 (+ 1.8%)*
E+T/d/begin/m/3/45	253,413	190,796	*1,247*	84,468	63,884	*1,279 (+ 2.6%)*	50,673	38,504	*1,313 (+ 5.3%)*	50,840	41,410	*1,261 (+ 1.1%)*
E+T/d/begin/m/3/75	423,945	330,397	*n/a*	141,306	110,875	*1,390 (n/a)*	84,777	66,988	*1,436 (n/a)*	85,032	72,775	*1,336 (n/a)*
E+T/d/cont/m/2/45	165,702	124,941	*2,373*	55,226	41,938	*2,405 (+ 1.3%)*	33,132	25,345	*2,464 (+ 3.8%)*	33,259	27,075	*2,429 (+ 2.4%)*
E+T/d/cont/m/2/75	277,400	208,259	*2,540*	92,452	70,175	*2,595 (+ 2.2%)*	55,468	42,580	*2,667 (+ 5.0%)*	55,664	45,957	*2,602 (+ 2.4%)*
E+T/d/cont/m/3/45	253,413	190,796	*428*	84,468	63,884	*466 (+ 8.9%)*	50,673	38,504	*502 (+ 17.3%)*	50,826	41,207	*432 (+ 0.9%)*
E+T/d/cont/m/3/75	423,945	330,397	*n/a*	141,306	110,875	*656 (n/a)*	84,777	66,988	*708 (n/a)*	85,026	72,601	*602 (n/a)*

* TD1 is equivalent to a time-continuous model z^{TD1}, z^{TD3}, z^{TD5}, and z^{TD5e} rounded to integer

equivalent to a time-continuous model formulation since all time parameters in the computed instances are integer. All four time discretization variants were generated for all instances to determine respective model sizes. While solving some instances with TD1, the solver ran out of memory and the solution procedure could not be finished (indicated by "n/a"). For optimal objective function values z^{TD3}, z^{TD5}, and z^{TD5e}, the difference to z^{TD1} is reported in brackets.

Regarding the model size, the number of variables as well as the number of constraints is almost directly proportional to the chosen step size of the time discretization. A step size of three seconds reduces the number of variables and constraints by approximately a factor of three and, thus, the matrix size of the BP by a factor of nine. Similarly, a step size of five seconds reduces the number of variables and constraints by approximately a factor of five and, thus, the matrix size of the BP by a factor of 25. This comes at the cost of loosing granularity and solution quality compared to a time-continuous solution using a step size of one. TD3 yields optimal objective function values which are up to 14.5% (on avg. 4.3%) higher than objective function values of a time-continuous formulation. With TD5, respective optimal objective function values increase up to 25.0% (on avg. 8.7%). The enhanced time discretization approach TD5 achieves optimal solutions which are only up to 2.4% (on avg. 0.9%) higher than time-continuous solutions while realizing almost the full model size reduction of factor 25.

6.4.2 Reducing the Number of Variables Through Preprocessing, Constraint Propagation, and Column Generation

Table 6.6 shows the impact of preprocessing, constraint propagation, and the column generation approach (cf. Figure 5.2) on the size of the resulting model. This analysis is based on time discretization variant TD5e due to its favorable trade-off between model size and solution quality. For all considered instances, the table reports the number of variables and constraints of the original BP before preprocessing, which corresponds to columns "TD5e" of Table 6.5. It also shows the number of variables and constraints after preprocessing and constraint propagation. The last three columns report the number of variables and constraints which remain after the column generation scheme in the column-reduced

Table 6.6: Reducing the model size through preprocessing, constraint propagation, and column generation

	BP before preprocessing		BP after preprocessing and constraint propagation		Column-reduced BP after column generation (including variables x_{art} with $LB^{\mathrm{LP}} + \bar{c}_{x_{art}} < UB^{\mathrm{CP}}$)		
Instance	Variables	Constraints	Variables	Constraints	Variables	Constraints	*Matrix Size (in % of BP before preprocessing)*
T/l/begin/m/2/45	23,172	19,467	16,295	14,579	8,746 (37.7%)	7,969 (34.4%)	*13.0%*
T/l/begin/m/2/75	38,384	33,075	31,507	28,187	13,574 (35.4%)	13,715 (35.7%)	*12.6%*
T/l/begin/m/3/45	35,880	31,086	33,876	28,823	20,098 (56.0%)	16,770 (46.7%)	*26.2%*
T/l/begin/m/3/75	59,274	53,637	56,424	49,814	32,767 (55.3%)	29,342 (49.5%)	*27.4%*
T/l/cont/m/2/45	23,172	19,467	15,659	13,984	4,519 (19.5%)	4,171 (18.0%)	*3.5%*
T/l/cont/m/2/75	38,384	33,075	30,871	27,592	6,219 (16.2%)	6,872 (17.9%)	*2.9%*
T/l/cont/m/3/45	35,862	30,792	35,015	30,052	8,802 (24.5%)	8,560 (23.9%)	*5.9%*
T/l/cont/m/3/75	59,256	53,343	58,409	52,603	13,003 (21.9%)	14,599 (24.6%)	*5.4%*
E+T/l/begin/m/2/45	33,254	27,063	23,508	19,799	14,352 (43.2%)	12,021 (36.1%)	*15.6%*
E+T/l/begin/m/2/75	55,654	45,844	45,192	38,218	27,464 (49.3%)	24,076 (43.3%)	*21.3%*
E+T/l/begin/m/3/45	50,835	41,426	48,828	39,072	20,874 (41.1%)	17,731 (34.9%)	*14.3%*
E+T/l/begin/m/3/75	85,031	72,775	82,172	68,773	34,221 (40.2%)	30,561 (35.9%)	*14.5%*
E+T/l/cont/m/2/45	33,251	26,979	23,221	19,647	12,140 (36.5%)	10,602 (31.9%)	*11.6%*
E+T/l/cont/m/2/75	55,660	45,944	44,902	38,057	21,411 (38.5%)	20,006 (35.9%)	*13.8%*
E+T/l/cont/m/3/45	50,823	41,140	50,375	40,800	8,245 (16.2%)	8,538 (16.8%)	*2.7%*
E+T/l/cont/m/3/75	85,023	72,523	84,575	72,242	17,177 (20.2%)	19,805 (23.3%)	*4.7%*
E+T/d/begin/m/2/45	33,254	27,024	23,512	19,824	6,299 (18.9%)	6,626 (19.9%)	*3.8%*
E+T/d/begin/m/2/75	55,659	45,900	45,193	38,245	12,502 (22.5%)	13,940 (25.0%)	*5.6%*
E+T/d/begin/m/3/45	50,840	41,410	48,830	39,072	12,081 (23.8%)	12,609 (24.8%)	*5.9%*
E+T/d/begin/m/3/75	85,032	72,775	82,182	68,934	19,930 (23.4%)	21,767 (25.6%)	*6.0%*
E+T/d/cont/m/2/45	33,259	27,075	23,225	19,700	5,784 (17.4%)	6,240 (18.8%)	*3.3%*
E+T/d/cont/m/2/75	55,664	45,957	44,901	38,040	12,055 (21.7%)	13,535 (24.3%)	*5.3%*
E+T/d/cont/m/3/45	50,826	41,207	50,378	40,818	7,478 (14.7%)	7,849 (15.4%)	*2.3%*
E+T/d/cont/m/3/75	85,026	72,601	84,575	72,243	14,295 (16.8%)	17,198 (20.2%)	*3.4%*

BP and the resulting matrix size in relation to the BP before preprocessing. This includes all variables x_{art} with $LB^{\text{LP}} + \bar{c}_{x_{art}} < UB^{\text{CP}}$ and, thus, all variables which are required to solve the BP to integer optimality. The column generation scheme generates only 14.7–56.0% (on avg. 29.6%) of all variables of the original BP before preprocessing. This corresponds to 15.4–49.5% (on avg. 28.5%) of all constraints. As a result, the size of the matrix of the column-reduced BP is considerably smaller and only 2.3–27.4% (on avg. 9.6%) of the matrix size of the original BP.

In summary, preprocessing, constraint propagation, and column generation reduce the model size by more than 90% on average, which not only reduces memory demand but also computational times for solving the model.

6.4.3 Analysis of Computational Times

Table 6.7 evaluates the computational performance of the proposed time-discrete algorithm using enhanced time discretization variant TD5e. It presents details on the three main components of the algorithm, namely the CP start heuristic, the column generation phase, and the final branch-and-bound procedure for the column-reduced BP. With regard to the CP start heuristic, the table shows the best found solution as upper bound UB^{CP}, its optimality gap, and the computational time after the start heuristic has terminated due to reaching the given fail limit. For the column generation phase, it reports the number of required iterations before no more variables with negative reduced cost are found and the corresponding final solution as LB^{LP}. It also reports computational times for the column generation phase, which denote the combined time over all iterations to compute the LP solutions. For the final branch-and-bound procedure, the table shows computational times to solve the column-reduced BP to integer optimality and objective values z^{BP} of the final optimal solution. The last column shows the sum of computational times over all three components as overall time.

The proposed time-discrete approach solves all 24 considered instances to optimality within three minutes. In 15 out of 24 cases, it computes the optimal solution in less than one minute. Within the chosen fail limits, the CP start heuristics calculates good solutions in 2–49 seconds. These upper bounds deviate at most 26.4% (on avg. 8.4%) from the optimal solution. For instances without

Table 6.7: Computational results for the time-discrete approach per component

	CP start heuristic			Column generation to solve LP relaxation			Branch-and-bound on column-reduced BP		
Instance	UB^{CP}	Gap (in %)	*Time*	LB^{LP}	Iterations	*Time*	z^{BP}	*Time*	*Overall Time*
T/l/begin/m/2/45	2,933	87	*2*	2,469	9	*1*	2,933	*4*	*7*
T/l/begin/m/2/75	3,142	90	*5*	2,852	16	*4*	3,142	*11*	*20*
T/l/begin/m/3/45	1,683	100	*2*	977	9	*1*	1,683	*45*	*48*
T/l/begin/m/3/75	1,763	100	*6*	1,070	16	*4*	1,763	*63*	*73*
T/l/cont/m/2/45	2,933	87	*2*	2,857	12	*1*	2,933	*1*	*4*
T/l/cont/m/2/75	3,142	88	*5*	3,082	17	*4*	3142	*3*	*12*
T/l/cont/m/3/45	664	100	*3*	533	16	*5*	664	*7*	*15*
T/l/cont/m/3/75	744	100	*6*	604	15	*7*	744	*23*	*36*
E+T/l/begin/m/2/45	2,206	92	*14*	1,637	8	*2*	2,004	*20*	*36*
E+T/l/begin/m/2/75	2,492	94	*40*	1,919	13	*6*	2,170	*50*	*96*
E+T/l/begin/m/3/45	1,167	100	*17*	748	11	*3*	1,093	*42*	*62*
E+T/l/begin/m/3/75	1,238	100	*48*	831	16	*7*	1,175	*101*	*156*
E+T/l/cont/m/2/45	2,349	93	*11*	1,875	8	*1*	1,999	*8*	*20*
E+T/l/cont/m/2/75	2,530	94	*33*	2,078	11	*4*	2,171	*19*	*56*
E+T/l/cont/m/3/45	448	100	*17*	351	13	*4*	430	*8*	*29*
E+T/l/cont/m/3/75	618	100	*49*	481	18	*15*	592	*45*	*109*
E+T/d/begin/m/2/45	2,686	90	*12*	1,934	8	*1*	2,413	*5*	*18*
E+T/d/begin/m/2/75	3,050	93	*34*	2,206	10	*4*	2,585	*36*	*74*
E+T/d/begin/m/3/45	1,507	100	*17*	756	10	*2*	1,261	*55*	*74*
E+T/d/begin/m/3/75	1,544	100	*49*	843	14	*6*	1,336	*104*	*159*
E+T/d/cont/m/2/45	2,839	91	*10*	2,231	9	*1*	2,429	*3*	*14*
E+T/d/cont/m/2/75	3,288	93	*28*	2,461	11	*4*	2,602	*12*	*44*
E+T/d/cont/m/3/45	458	100	*15*	351	15	*5*	432	*7*	*27*
E+T/d/cont/m/3/75	647	100	*46*	484	15	*12*	602	*55*	*113*

UB^{CP}, LB^{LP}, and z^{BP} rounded to integer   Times in seconds earliness, the CP start heuristic finds the optimal solution but cannot prove its optimality within the fail limit and terminates with significant optimality gaps above 87%. For all instances, it requires less than 20 iterations to generate all required columns and, in each iteration, the solution of the LP is computed in less than a second on average.

6.5 Comparison of Solution Approaches

This section compares both solution approaches presented in this dissertation regarding solution quality and computational times. It also demonstrates that both proposed exact methods outperform a pure CP approach. For this comparison, all 24 instances considered for the time-discrete BP approach have also been solved with a time-continuous MIP model and a CP optimization engine. The time-continuous MIP model is based on the mathematical model presented in Chapter 4. To incorporate earliness, i.e., to allow the acceleration of aircraft, Constraint (4.2) was replaced by $E_a \leq x_a \leq L_a$. Correspondingly, the Objective (4.1) was modified to support piecewise linear cost functions. The CP optimization engine was initialized with a time limit of five minutes.

Table 6.8 reports results for the comparative computational study. For the time-continuous approach based on the MIP, it shows optimal objective function values z^{MIP} and computational times. As in Table 6.3, the comparison presents computational times required to compute schedules which deviate less than 1% from the optimal solution z^{MIP}. For the time-discrete approach using column generation for the BP, the table presents computational times and the optimal solution z^{BP}. The difference to the optimal solution z^{MIP} found by the time-continuous approach is shown brackets. The table also presents the best found solution z^{CP}, the remaining optimality gap and computational times of the pure CP approach. Again, the difference of the heuristic solution z^{CP} to the time-continuous solution z^{MIP} is reported in brackets.

Objective function values z^{MIP} always reflect the best possible solution for the WRSP in continuous time. Objective function values z^{BP}, however, represent optimal solutions for the time-discrete WRSP and can deviate from z^{MIP} since the model inherent time discretization implies a loss of granularity and potentially excludes optimal (time-continuous) solutions from the solution space. Although

Table 6.8: Comparison of solution approaches

	Time-continuous approach			Time-discrete approach		CP model		
Instance	z^{MIP}	*Time*	*Time (1% dev.)*	z^{BP}	*Time*	z^{CP}	Gap (in %)	*Time*
T/l/begin/m/2/45	2,933	*6*	*1*	2,933 (+0.0%)	*7*	2,933 (+0.0%)	87	>*300*
T/l/begin/m/2/75	3,142	*40*	*3*	3,142 (+0.0%)	*20*	3,142 (+0.0%)	90	>*300*
T/l/begin/m/3/45	1,683	*601*	*6*	1,683 (+0.0%)	*48*	1,683 (+0.0%)	100	>*300*
T/l/begin/m/3/75	1,763	*1,680*	*42*	1,763 (+0.0%)	*73*	1,763 (+0.0%)	100	>*300*
T/l/cont/m/2/45	2,933	*5*	*1*	2,933 (+0.0%)	*4*	2,933 (+0.0%)	87	>*300*
T/l/cont/m/2/75	3,142	*26*	*1*	3,142 (+0.0%)	*12*	3,142 (+0.0%)	88	>*300*
T/l/cont/m/3/45	664	*32*	*7*	664 (+0.0%)	*15*	664 (+0.0%)	100	>*300*
T/l/cont/m/3/75	744	*90*	*44*	744 (+0.0%)	*36*	744 (+0.0%)	100	>*300*
E+T/l/begin/m/2/45	1,966	*7*	*4*	2,004 (+1.9%)	*36*	2,070 (+5.3%)	92	>*300*
E+T/l/begin/m/2/75	2,133	*46*	*17*	2,170 (+1.7%)	*96*	2,508 (+17.6%)	94	>*300*
E+T/l/begin/m/3/45	1,086	*1,454*	*30*	1,093 (+0.6%)	*62*	1,125 (+3.6%)	100	>*300*
E+T/l/begin/m/3/75	1,166	*2,486*	*49*	1,175 (+0.8%)	*156*	1,916 (+64.3%)	100	>*300*
E+T/l/cont/m/2/45	1,966	*6*	*3*	1,999 (+1.7%)	*20*	2,136 (+8.6%)	92	>*300*
E+T/l/cont/m/2/75	2,133	*40*	*17*	2,171 (+1.8%)	*56*	2,511 (+17.7%)	94	>*300*
E+T/l/cont/m/3/45	428	*123*	*10*	430 (+0.5%)	*29*	435 (+1.6%)	100	>*300*
E+T/l/cont/m/3/75	586	*1,503*	*42*	592 (+1.0%)	*109*	590 (+0.7%)	100	>*300*
E+T/d/begin/m/2/45	2,373	*3*	*2*	2,413 (+1.7%)	*18*	2,917 (+22.9%)	91	>*300*
E+T/d/begin/m/2/75	2,540	*24*	*11*	2,585 (+1.8%)	*74*	3,254 (+28.1%)	93	>*300*
E+T/d/begin/m/3/45	1,247	*2,779*	*32*	1,261 (+1.1%)	*74*	1,292 (+3.6%)	100	>*300*
E+T/d/begin/m/3/75	1,327	>*3,600*	*63*	1,336 (+0.7%)	*159*	1,379 (+3.9%)	100	>*300*
E+T/d/cont/m/2/45	2,373	*3*	*2*	2,429 (+2.4%)	*14*	3,113 (+31.2%)	92	>*300*
E+T/d/cont/m/2/75	2,540	*29*	*11*	2,602 (+2.4%)	*44*	3,391 (+33.5%)	93	>*300*
E+T/d/cont/m/3/45	428	*140*	*18*	432 (+0.9%)	*27*	436 (+1.9%)	100	>*300*
E+T/d/cont/m/3/75	594	>*3,600*	*564*	602 (+1.3%)	*113*	625 (+5.2%)	100	>*300*

All objective values rounded to integer   Times in seconds the proposed enhanced time discretization approach reduces these losses, a moderate increase of the objective function values of less than 2.5% can be observed for instances allowing aircraft earliness.

For instances without earliness of aircraft and with linear delay cost functions, the time-discrete approach achieves the same optimal solutions as the time-continuous approach and, in seven out of eight instances, has significantly lower computational times. Especially instances with three runways can be solved much faster and runtimes can be reduced by more than 50% in these cases. For these instances, the time-discrete approach computes optimal solutions in up to 73 seconds and outperforms the time-continuous model.

If aircraft are allowed to be scheduled early, i.e., before their target time, or if piecewise linear cost functions are used, the time-continuous approach regularly computes better schedules. In these cases, delay cost are up to 2.4% higher with the time-discrete approach. For instances with three runways, the time-discrete approach computes solutions much faster and reduces computational times by at least 76% resulting in runtimes of at most 159 seconds. For instances with two runways, the time-continuous solution computes not only superior schedules but also has lower computational times of at most 46 seconds.

The CP approach relying solely on the CP optimization engine cannot solve any of the 24 instances to proven optimality within the given time limit of five minutes. For instances without earliness and with linear cost functions, it finds the optimal solution but terminates with an optimality gap of at least 87%. For all other instances, the best found solution at termination has up to 64.3% (on avg. 15.6%) higher delay cost than the optimal time-continuous schedule.

For all instances, the time-discrete approach outperforms the pure CP approach regarding resulting schedules and computational times.

The time-continuous approach also is superior to the CP approach. For 16 instances, the time-continuous approach computes optimal solutions within five minutes while the CP model terminates with high optimality gaps. Neither the time-continuous approach nor the CP model can solve eight instances to optimality within the given time limit. However, for these instances, the branch-and-bound process of the time-continuous approach can be terminated early which results in near-optimal schedules.

6.6 Applicability in Practice

An important aspect of optimization models is their applicability in real-world settings. Therefore, this section analyzes the proposed solution approaches for the WRSP with regard to a potential implementation in decision support systems for runway management and air traffic control. During winter operations, the situation in the near-terminal area and on the airport ground often changes. New arriving aircraft are constantly entering the near-terminal area or landing. Departing aircraft are taking-off and leaving the airspace around the airport control tower. Snowfall and other weather influences alter the conditions on the runways. Snow removal groups are clearing runways requiring runway closings or enabling the reopening of runways. To reflect this changing environment, optimal runway schedules must be recalculated once the situation around the airport has significantly changed. Given the actual flight density and frequency of aircraft movements at large international airports, a recalculation of optimal runway schedules every minute is reasonable.

This thesis proposes to embed the presented solution algorithms in an optimization framework which recomputes optimal runway schedules once per minute. After each optimization procedure, the decisions for the first minute of the schedule are implemented and optimal schedules are recomputed with a planning horizon shifted by one minute. This requires computational times of less than one minute per optimization procedure.

The time-continuous solution approach for the WRSP presented in this thesis computes optimal runway schedules in less than one minute for all instances with up to 75 aircraft and at most two runways. Thus, airports with up to two runways can integrate and optimize their aircraft and snow removal schedules by using the time-continuous solution methodology.

For larger airports with at least three runways, the time-continuous approach often cannot solve the WRSP to optimality within one minute. In these cases, the time-discrete approach performs significantly faster at the cost of marginally increased delay due to loss of time granularity. It computes solutions with a moderate delay cost increase of less than 1% on average within one minute. If the time-continuous and the time-discrete algorithm cannot compute optimal solutions in the required short amount of time, it is possible to use the time-continuous approach heuristically by terminating the solution procedure early. This regularly results in optimal or near-optimal schedules.

7 Conclusion, Managerial Insights, and Outlook

This dissertation presented two exact optimization approaches for the WRSP. Both methodologies simultaneously schedule aircraft and snow removals on runways during winter operations at airports. They minimize the operational cost of the resulting schedule by minimizing cost for weighted aircraft delay.

The first approach modeled the problem as a time-continuous MIP. In order to accelerate the branch-and-bound procedure, problem specific pruning rules based on compulsory precedence relations as well as valid inequalities were developed. Additionally, a method was presented to derive initial start solutions for the MIP solver heuristically.

The second approach presented a time-discrete variant of the WRSP and an exact solution algorithm to solve it. This thesis proposed a novel combination of CP and column generation techniques. It proposed a start heuristic based on a CP model and presented a time-indexed BP which was solved using a column generation scheme and a branch-and-bound procedure. An enhanced time discretization approach was developed and applied to balance solution quality and model size of the BP. The proposed time discretization method enables high quality runway schedules which are close to optimal time-continuous solutions while maintaining small model sizes.

To substantiate the value of this dissertation, both presented approaches were applied to real-world instances from a large international airport. The numerical experiments showed significant reductions of weighted delay compared to a manual scheduling process by a human planner. The computational study also proved the efficiency of the proposed methods and showed that optimal or near-optimal runway schedules can be computed in a short amount of time.

The results of this work yield several key insights which are of high practical relevance for managers and decision makers overseeing runway scheduling at airports during winter operations.

- *A naive scheduling approach is not suitable for an application in practice.* Simple scheduling rules, which start snow removals as soon as runways are unsafe and snow removal groups are available, yield only poor schedules with high delay. In terms of weighted delay cost, these naive solutions are often by a factor of 10 to 20 worse than optimal schedules.

- *When creating runway schedules manually, human planners should follow guiding principles to generate good snow removal schedules.* An analysis of optimal schedules shows that it is crucial to avoid situations in which a runway has to be closed due to snow or ice without having a snow removal group available for clearing it. This often leads to optimal schedules where a (preceding) snow removal on a runway already starts before operations on that runway would become unsafe so that a succeeding snow removal on the next runway can start on time. Furthermore, human planners can use the concept of dominant snow removal patterns. If dominant snow removal patterns are precalculated, planners need to consider only schedules that use such dominant patterns.

- *An integrated scheduling of snow removals and aircraft using the proposed solution methodologies yields optimal schedules and significantly reduces weighted delay cost.* A substantial part of delay cost reduction originates from the integration of these scheduling decisions. Potential improvements increase with growing complexity of the underlying optimization problem. For larger airports with at least three runways, an integrated optimization of runway schedules can often reduce delay cost by more than 50% compared to the practice-oriented heuristic.

This thesis also discussed the real-world applicability of the proposed algorithms. For airports with at most two runways, the presented time-continuous approach computes optimal runway schedules in less than one minute and is applicable in practice. For larger airports, the time-discrete method computes very good runway schedules in a reasonable amount of time. It offers the possibility to concede some time granularity, and, thereby, solution quality, in exchange for better computational performance. In all cases where both approaches cannot compute optimal runway schedules in the required amount of time, the time-continuous approach can be applied heuristically. Terminating its branch-and-bound process early regularly results in near-optimal runway schedules.

The results of this work motivate some potential future research directions. This thesis considered the static and deterministic version of the WRSP. To better represent real-world settings, it is reasonable to investigate the dynamic version of the problem or to incorporate it into a sound rolling horizon approach. Since many aspects of the problem, e.g., aircraft parameters and weather conditions, are subject to uncertainty and change, it might be beneficial to account for the stochasticity of the problem. Related to that, concepts of robust optimization can help to create more stable schedules which are less likely to change with minor modifications in the parameters. From a methodological point of view, this thesis presented a novel combination of constraint programming and column generation for a time-discrete BP. Such time-indexed model formulations typically have a lot of variables and constraints. This thesis used preprocessing techniques, constraint propagation, and column generation to reduce the size of the time-discrete model resulting in a reduced BP whose matrix size is, on average, less than 10% of the matrix size of the initial BP. The proposed algorithm combining CP techniques and column generation might be transferable or adaptable also to time-discrete optimization models with high numbers of variables and constraints in other domains. Further research in that direction is highly encouraged.

Reinforced by future advancements, the results of this dissertation enable airport operators and air traffic control to integrate their planning and to achieve better overall runway schedules for winter days with considerable snowfall.

Bibliography

Abela, J., Abramson, D., Krishnamoorthy, M., De Silva, A., & Mills, G. (1993). Computing optimal schedules for landing aircraft. In *Proceedings of the 12th national conference of the Australian Society for Operations Research* (pp. 71–90).

Allahverdi, A., Ng, C., Cheng, T., & Kovalyov, M. Y. (2008). A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, *187*, 985–1032.

Allignol, C., Barnier, N., Flener, P., & Pearson, J. (2012). Constraint programming for air traffic management: A survey. *The Knowledge Engineering Review*, *27*, 361–392.

Avella, P., Boccia, M., Mannino, C., & Vasilyev, I. (2017). Time-indexed formulations for the runway scheduling problem. *Transportation Science*, *51*, 1196–1209.

Balakrishnan, H., & Chandran, B. G. (2010). Algorithms for scheduling runway operations under constrained position shifting. *Operations Research*, *58*, 1650–1665.

Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., & Abramson, D. (2000). Scheduling aircraft landings - The static case. *Transportation Science*, *34*, 180–197.

Bennell, J. A., Mesgarpour, M., & Potts, C. N. (2011). Airport runway scheduling. *4OR*, *9*, 115–138.

Bennell, J. A., Mesgarpour, M., & Potts, C. N. (2017). Dynamic scheduling of aircraft landings. *European Journal of Operational Research*, *258*, 315–327.

Bertsimas, D., & Frankovich, M. (2015). Unified optimization of traffic flows through airports. *Transportation Science*, *50*, 77–93.

Bianco, L., Dell'Olmo, P., & Giordani, S. (1999). Minimizing total completion time subject to release dates and sequence-dependent processing times. *Annals of Operations Research*, *86*, 393–415.

Bianco, L., Dell'Olmo, P., & Giordani, S. (2006). Scheduling models for air traffic control in terminal areas. *Journal of Scheduling*, *9*, 223–253.

Boeing (2019). Commercial market outlook 2019–2038. https://www.boeing.com/commercial/market/commercial-market-outlook/.

Briskorn, D., & Stolletz, R. (2014). Aircraft landing problems with aircraft classes. *Journal of Scheduling*, *17*, 31–45.

Cheng, T., Hsu, C.-J., & Yang, D.-L. (2011). Unrelated parallel-machine scheduling with deteriorating maintenance activities. *Computers & Industrial Engineering*, *60*, 602–605.

Dear, R. G. (1976). The dynamic scheduling of aircraft in the near terminal area. Flight Transportation Laboratory, Massachusetts Institute of Technology, Cambridge.

Díaz, J. F., & Mena, J. A. (2005). Solving the aircraft sequencing problem using concurrent constraint programming. In *Multiparadigm Programming in Mozart/Oz (Lecture Notes in Computer Science, vol. 3389)* (pp. 292–304).

FAA (2017). Federal Aviation Administration Order JO 7110.65X. Air Traffic Control. https://www.faa.gov/regulations_policies/orders_notices/index.cfm/go/document.current/documentNumber/7110.65.

Fahle, T., Feldmann, R., Götz, S., Grothklags, S., & Monien, B. (2003). The aircraft sequencing problem. In *Computer Science in Perspective (Lecture Notes in Computer Science, vol. 2598)* (pp. 152–166).

Fahle, T., Junker, U., Karisch, S. E., Kohl, N., Sellmann, M., & Vaaben, B. (2002). Constraint programming based column generation for crew assignment. *Journal of Heuristics*, *8*, 59–81.

Faye, A. (2015). Solving the aircraft landing problem with time discretization approach. *European Journal of Operational Research*, *242*, 1028–1038.

Furini, F., Kidd, M. P., Persiani, C. A., & Toth, P. (2014). State space reduced dynamic programming for the aircraft sequencing problem with constrained position shifting. In *Combinatorial Optimization (Lecture Notes in Computer Science, vol. 8596)* (pp. 267–279).

Furini, F., Kidd, M. P., Persiani, C. A., & Toth, P. (2015). Improved rolling horizon approaches to the aircraft sequencing problem. *Journal of Scheduling*, *18*, 435–447.

Furini, F., Persiani, C. A., & Toth, P. (2012). Aircraft sequencing problems via a rolling horizon algorithm. In *Combinatorial Optimization (Lecture Notes in Computer Science, vol. 7422)* (pp. 273–284).

Gabteni, S., & Grönkvist, M. (2008). Combining column generation and constraint programming to solve the tail assignment problem. *Annals of Operations Research*, *171*, 61–76.

Gao, J., Gen, M., & Sun, L. (2006). Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm. *Journal of Intelligent Manufacturing*, *17*, 493–507.

Giffler, B., & Thompson, G. L. (1960). Algorithms for solving production-scheduling problems. *Operations Research*, *8*, 487–503.

Goel, V., Slusky, M., van Hoeve, W.-J., Furman, K., & Shao, Y. (2015). Constraint programming for LNG ship scheduling and inventory management. *European Journal of Operational Research*, *241*, 662–673.

Grönkvist, M. (2006). Accelerating column generation for aircraft scheduling using constraint propagation. *Computers & Operations Research*, *33*, 2918–2934.

Gualandi, S., & Malucelli, F. (2013). Constraint programming-based column generation. *Annals of Operations Research*, *204*, 11–32.

Hansen, J. V. (2004). Genetic search methods in air traffic control. *Computers & Operations Research*, *31*, 445–459.

Heidt, A., Helmke, H., Liers, F., & Martin, A. (2014). Robust runway scheduling using a time-indexed model. In *Proceedings of the 4th SESAR Innovation Days* (pp. 1–8).

Junker, U., Karisch, S. E., Kohl, N., Vaaben, B., Fahle, T., & Sellmann, M. (1999). A framework for constraint programming based column generation. In *Principles and Practice of Constraint Programming – CP'99 (Lecture Notes in Computer Science, vol. 1713)* (pp. 261–274).

Kjenstad, D., Mannino, C., Nordlander, T., Schittekat, P., & Smedsrud, M. (2013). Optimizing aman-sman-dman at hamburg and arlanda airport. In *Proceedings of the 3rd SESAR Innovation Days*.

Lee, C.-Y., & Chen, Z.-L. (2000). Scheduling jobs and maintenance activities on parallel machines. *Naval Research Logistics*, *47*, 145–165.

Lee, Y. H., & Pinedo, M. (1997). Scheduling jobs on parallel machines with sequence-dependent setup times. *European Journal of Operational Research*, *100*, 464–474.

Lieder, A., & Stolletz, R. (2016). Scheduling aircraft take-offs and landings on interdependent and heterogeneous runways. *Transportation Research Part E: Logistics and Transportation Review*, *88*, 167–188.

Liu, Y.-H. (2010). A genetic local search algorithm with a threshold accepting mechanism for solving the runway dependent aircraft landing problem. *Optimization Letters*, *5*, 229–245.

Lopes, M. J. P., & de Carvalho, J. V. (2007). A branch-and-price algorithm for scheduling parallel machines with sequence dependent setup times. *European Journal of Operational Research*, *176*, 1508–1527.

Ma, Y., Chu, C., & Zuo, C. (2010). A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering*, *58*, 199–211.

Maere, G. D., Atkin, J. A. D., & Burke, E. K. (2017). Pruning rules for optimal runway sequencing. *Transportation Science*, *52*, 898–916.

Nogueira, T. H., Carvalho, C. R. V. d., Ravetti, M. G., & Souza, M. C. d. (2019). Analysis of mixed integer programming formulations for single machine scheduling problems with sequence dependent setup times and release dates. *Pesquisa Operacional*, *39*, 109–154.

Novara, F. M., Novas, J. M., & Henning, G. P. (2016). A novel constraint programming model for large-scale scheduling problems in multiproduct multistage batch plants: Limited resources and campaign-based operation. *Computers & Chemical Engineering*, *93*, 101–117.

Pinol, H., & Beasley, J. (2006). Scatter search and bionomic algorithms for the aircraft landing problem. *European Journal of Operational Research*, *171*, 439–462.

Pohl, M., Artigues, C., & Kolisch, R. (2019a). Solving the time-discrete winter runway scheduling problem with column generation and constraint programming. Working Paper. TUM School of Management. Technical University of Munich.

Pohl, M., Kolisch, R., & Schiffer, M. (2019b). Runway scheduling during winter operations. Working Paper. TUM School of Management. Technical University of Munich.

Psaraftis, H. N. (1980). A dynamic programming approach for sequencing groups of identical jobs. *Operations Research*, *28*, 1347–1359.

Radhakrishnan, S., & Ventura, J. A. (2000). Simulated annealing for parallel machine scheduling with earliness-tardiness penalties and sequence-dependent setup times. *International Journal of Production Research*, *38*, 2233–2252.

Regierung von Oberbayern (2011). Planfeststellungsverfahren Flughafen München — 3. Start- und Landebahn (98. Änderungsplanfeststellungsbeschluss). https://www.regierung.oberbayern.bayern.de/aufgaben/wirtschaft/luftamt/planfeststellung/07727/index.php.

Sabar, N. R., & Kendall, G. (2015). An iterated local search with multiple perturbation operators and time varying perturbation strength for the aircraft landing problem. *Omega*, *56*, 88–98.

Salehipour, A., Modarres, M., & Naeni, L. M. (2013). An efficient hybrid meta-heuristic for aircraft landing problem. *Computers & Operations Research*, *40*, 207–213.

Salehipour, A., Moslemi Naeni, L., & Kazemipoor, H. (2009). Scheduling aircraft landings by applying a variable neighborhood descent algorithm: Runway-dependent landing time case. *Journal of Applied Operational Research*, *1*, 39–49.

Samà, M., D'Ariano, A., & Pacciarelli, D. (2013). Rolling horizon approach for aircraft scheduling in the terminal control area of busy airports. *Procedia - Social and Behavioral Sciences*, *80*, 531–552.

Samà, M., D'Ariano, A., Palagachev, K., & Gerdts, M. (2019). Integration methods for aircraft scheduling and trajectory optimization at a busy terminal manoeuvring area. *OR Spectrum*, *41*, 641–681.

Vadlamani, S., & Hosseini, S. (2014). A novel heuristic approach for solving aircraft landing problem with single runway. *Journal of Air Transport Management*, *40*, 144–148.

Vallada, E., & Ruiz, R. (2011). A genetic algorithm for the unrelated parallel machine scheduling problem with sequence dependent setup times. *European Journal of Operational Research*, *211*, 612–622.

Wang, S., & Yu, J. (2010). An effective heuristic for flexible job-shop scheduling problem with maintenance activities. *Computers & Industrial Engineering*, *59*, 436–447.

Yang, D.-L., Cheng, T., Yang, S.-J., & Hsu, C.-J. (2012). Unrelated parallel-machine scheduling with aging effects and multi-maintenance activities. *Computers & Operations Research*, *39*, 1458–1464.

Yoo, J., & Lee, I. S. (2016). Parallel machine scheduling with maintenance activities. *Computers & Industrial Engineering*, *101*, 361–371.

Yunes, T. H., Moura, A. V., & De Souza, C. C. (2000). Solving very large crew scheduling problems to optimality. In *Proceedings of the 15th Symposium on Applied Computing (SAC 2000)* (pp. 446–451).