European Journal of Operational Research 299 (2022) 674–689

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Innovative Applications of O.R.

Solving the time-discrete winter runway scheduling problem: A column generation and constraint programming approach

Maximilian Pohl[a,*], Christian Artigues[b], Rainer Kolisch[a]

[a] TUM School of Management, Technical University of Munich, Arcisstrasse 21, Munich 80333, Germany
[b] Laboratory for Analysis and Architecture of Systems, Centre National de la Recherche Scientifique, 7 Avenue du Colonel Roche, Toulouse 31031, France

ARTICLE INFO

Article history:
Received 27 January 2020
Accepted 12 August 2021
Available online 25 August 2021

Keywords:
Scheduling
Airport operations
Column generation
Constraint programming

ABSTRACT

We address the runway scheduling problem under consideration of winter operations. During snowfall, runways have to be temporarily closed in order to clear them from snow, ice and slush. We propose an integrated optimization model to simultaneously plan snow removal for multiple runways and to assign runways and take-off and landing times to aircraft. For this winter runway scheduling problem, we present a time-discrete binary model formulation using clique inequalities and an equivalent constraint programming model. To solve the winter runway scheduling problem optimally, we propose an exact solution methodology. Our start heuristic based on constraint programming generates a feasible initial start solution. We use a column generation scheme, which we initialize with a heuristic solution, to identify all variables of the binary program which are required to solve it optimally. Finally, we apply a branch-and-bound procedure to our resulting binary program. Additionally, we present an enhanced time discretization method to balance model size and solution quality. We apply our algorithm to realistic instances from a large international airport. An analysis of resulting model sizes proves the ability of our approach to significantly reduce the number of required variables and constraints of the time-discrete binary program. We also show that our method computes optimal schedules in a short amount of time and often outperforms a time-continuous formulation as well as a pure constraint programming approach.

1. Introduction

At most major airports, runway capacity is a scarce resource especially during winter operations, when runways have to be intermittently closed in order to clear them from snow, ice, and slush. This situation will presumably intensify in the future since the number of operated aircraft and, therefore, corresponding flight movements are expected to almost double until 2040 (Boeing, 2019). Hence, it is crucial to utilize the runway system in the best possible way. The winter runway scheduling problem (WRSP) assigns runways and take-off or landing times to departing and arriving aircraft and, simultaneously, schedules snow removals on runways. While the first models looking at runway scheduling date back to the 1970s (Dear, 1976; Psaraftis, 1980), the integrated consideration of aircraft scheduling and snow removal planning on runways has only been addressed recently. Pohl, Kolisch, & Schiffer (2021) proposed a time-continuous model for the WRSP, which computes very good schedules in a short amount of time. Larger instances, however, can not be solved to proven optimality within reasonable time limits. In this paper, we propose an exact solution procedure based on a time-discrete model formulation to overcome this drawback. Our solution approach combines constraint programming (CP) techniques with a column generation approach. Furthermore, the model of Pohl et al. (2021) is limited to piecewise linear convex cost functions. Our time-indexed model, however, offers greater flexibility with regard to the objective function. From a technical viewpoint, a time-discrete model is favorable since such model formulations often provide relatively strong bounds.

This paper is organized as follows. In Section 2, we detail the structure and characteristics of the problem. Section 3 presents on overview of related research articles and places our work into existing literature. In Section 4, we present two model formulations. We propose a time-discrete binary program (BP) based on clique inequalities. Our BP is based on problem-inherent incompatibilities and builds up on a model for the runway scheduling problem (RSP) of Avella, Boccia, Mannino, & Vasilyev (2017). We also present an equivalent CP model formulation. In Section 5, we propose a method to discretize the planning horizon, which enables improved solutions and allows an efficient balance between solution quality and model size. We present our exact solution methodology in Section 6. During preprocessing, we

* Corresponding author.
E-mail addresses: maximilian.pohl@tum.de (M. Pohl), artigues@laas.fr (C. Artigues), rainer.kolisch@tum.de (R. Kolisch).

https://doi.org/10.1016/j.ejor.2021.08.028
0377-2217/

apply problem-specific dominance rules to reduce variable domains. We use the CP model to further reduce variable domains through constraint propagation and as a start heuristic. We use our BP to solve the WRSP optimally. To keep the size, i.e., the number of variables and constraints, of our time-discrete binary model formulation manageable, we present a column generation scheme, which generates all variables required to solve the BP to optimality. In Section 7, we apply our method in a computational study to realistic data from Munich International Airport. We show the applicability and effectiveness of our approach by analyzing resulting model sizes and computational times. Most importantly, we show that, for large instances, our method outperforms a time-continuous formulation and a pure CP approach. Section 8 concludes our work with a brief summary.

2. Problem statement

We define the WRSP as the problem of scheduling aircraft and snow removals on multiple airport runways while minimizing earliness and tardiness cost, i.e., lateness cost, for aircraft. We consider an operational planning horizon of up to two hours. A feasible solution to the WRSP assigns a runway as well as a take-off or landing time to each departing or arriving aircraft. Assigned take-off and landing times have to lie within aircraft- and runway-specific time windows. 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. The latest possible take-off time is theoretically unrestricted until, ultimately, the flight is canceled. For arriving aircraft, the earliest possible landing time is given by the shortest flight path and the maximum flight speed of the aircraft. 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. In this paper, we examine convex cost functions which are piecewise linear with integer break points.

Aircraft scheduled on the same runway (or, in general, on interdependent runways) 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). These separation times between aircraft are sequence-dependent and depend on the operation classes of the aircraft, i.e., their weight classes (e.g., "Small", "Medium", "Large", "Boeing 757", "Heavy", "Super"), operation modes ("Take-off" or "Landing") and relative positions ("Leading" or "Trailing"). Table 1 lists separation requirements for a typical set of aircraft operation classes operated at large international airports.

During winter operations, runways regularly have to be cleared from snow, ice, and slush to enable safe flight operations. If a runway is covered by a critical amount of snow, ice, or slush, the runway becomes unsafe. At this point, the previous flight operation on that runway must have been completed and new flight operations can only start once the runway has been cleared. Due to advanced weather predictions, the exact point in time within the planning horizon at which runways become unsafe can be accurately forecast two hours in advance and, thus, is known at the beginning of the operational planning horizon. Airports use snow removal groups to clear runways from snow, ice, and slush. 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. Hence, not all runways can be cleared at the same time and snow removal groups have to clear multiple runways sequentially. Consequently, the snow removal planning for multiple runways becomes a scheduling problem itself. Snow removal schedules have to consider snow removal durations per runway (as processing times) and transit times between runways (as setup times). These transit times are sequence-dependent since distances and driving durations between runways depend on the physical layout of the airport and its road network. After a runway has been cleared, 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 up to two hours.

3. Related literature

This section briefly presents related literature. We focus on recent contributions most relevant for our work. For a general overview of publications concerning the RSP and related problems in airport arrival management, departure management, and surface management, we refer to Bennell, Mesgarpour, & Potts (2011), Lieder & Stolletz (2016), and Samà, D'Ariano, Palagachev, & Gerdts (2019). Bennell et al. (2011) gave a comprehensive 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 comprehensive literature discussion around aircraft arrival scheduling, aircraft departure scheduling, and mixed arrival-departure scheduling.

Since the general RSP is known to be NP-hard (Bianco, Dell'Olmo, & Giordani, 1999), many heuristic approaches to solve the problem have been developed. Bianco, Dell'Olmo, & Giordani (2006) and Sabar & Kendall (2015) presented local search algorithms, while variable neighborhood and adaptive large neighborhood search methods were proposed by Salehipour, Moslemi Naeni, & Kazemipoor (2009) and Vadlamani & Hosseini (2014). Salehipour, Modarres, & Naeni (2013) and Bennell, Mesgarpour, & Potts (2017) suggested simulated annealing approaches to solve the aircraft landing problem. Various population based heuristics were proposed as well. Abela, Abramson, Krishnamoorthy, De Silva, & Mills (1993), Hansen (2004), and Liu (2010) developed genetic search algorithms and Pinol & Beasley (2006) presented a scatter search and bionomic algorithms to solve the aircraft landing problem.

Exact approaches for the RSP mainly use dynamic programming (DP) or mixed-integer programming (MIP) techniques. 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, which reduces the solution space of the DP by prohibiting large deviations from a First-Come-First-Served (FCFS) order. Furini, Kidd, Persiani, & Toth (2014) built up on that and presented state space reduction techniques to further improve computational times for DP approaches.

MIP models optimize runway schedules either in continuous time using big-M formulations or in discrete time using time-indexed models. Abela et al. (1993) presented a time-continuous MIP formulation for the single runway case. Beasley, Krishnamoorthy, Sharaiha, & Abramson (2000) developed a MIP model for the multiple runway case, which has become the groundwork for many succeeding model formulations. Pohl et al. (2021) extended the MIP approach of Beasley et al. (2000) to reflect winter operations and proposed a time-continuous MIP formulation for the WRSP.

Table 1
Separation requirements based on aircraft operation classes according to FAA (2017) (in seconds).

Leading	Trailing							
	Landing				Take-off			
	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

For realistic instance sizes, time-discrete formulations were long considered to be computationally intractable due to a very large number of binary variables. However, thanks to algorithmic advances of MIP solvers and increasing computational power, they received more attention in recent years. Kjenstad, Mannino, Nordlander, Schittekat, & Smedsrud (2013) proposed a three-step approach for air traffic management at airports where, in the second step, a time-indexed formulation is used to optimally assign arrival and departure times to aircraft. Heidt, Helmke, Liers, & Martin (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 our 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. The authors developed a new family of clique inequalities which is a generalization of clique inequalities presented in Nogueira, Carvalho, Ravetti, & Souza (2019). In the introduction of their paper, Avella et al. (2017) also discussed properties of time-indexed formulations describing advantages and drawbacks of this model type.

In a method-independent publication, Maere, Atkin, & Burke (2017) presented pruning rules for the RSP to decrease corresponding solution or search spaces. Their generic pruning rules are model-independent and can be applied to different (meta) heuristics and exact algorithms.

With regard to CP, Allignol, Barnier, Flener, & Pearson (2012) compiled a survey of CP approaches for air traffic management including but not restricted to runway scheduling. For the aircraft sequencing problem on a single runway, Fahle, Feldmann, Götz, Grothklags, & Monien (2003) compared different exact and heuristic methods including a CP model. For the same single runway problem, Díaz & Mena (2005) presented a CP implementation and preprocessing techniques. Junker et al. (1999) and Yunes, Moura, & De Souza (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 (Gualandi & Malucelli, 2013), specifically to the tail assignment problem (Gabteni & Grönkvist, 2008; Grönkvist, 2006) and the crew assignment problem (Fahle et al., 2002; Junker et al., 1999).

Our work in this paper builds up on Avella et al. (2017), extends their formulation to consider multiple runways as well as winter operations, and presents a novel solution algorithm combining a column generation scheme with a CP start heuristic.

4. Model formulations

In Section 4.1, we introduce a time-discrete BP for the WRSP considering multiple homogeneous or 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. In Section 4.2, we present additional constraints for the BP based on precedence relations. We present an equivalent CP model formulation in Section 4.3 and show additional precedence constraints for the CP model in Section 4.4. While our solution approach and computational study focus on convex cost functions which are piecewise linear with integer break points, both model formulations presented in Sections 4.1 and 4.3 are also valid for arbitrary cost functions.

4.1. Time-discrete binary program based on clique inequalities

In this section, we introduce a time-discrete mathematical problem formulation for the WRSP. We consider 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 the 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, this is the latest time at which an aircraft can still safely take-off or land and the runway has to be closed immediately afterwards.

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.

We consider 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. The preferred take-off or landing time of aircraft a on runway r, i.e., the time avoiding any earliness or tardiness cost, is denoted as the aircraft's runway-specific target time T_{ar} with $E_{ar} \leq T_{ar} \leq L_{ar}$. If aircraft a cannot be assigned to runway r, then $\mathcal{T}_{ar} = \emptyset$. In practice, time window differences between runways are caused by slightly different flight paths and runway approach paths and, thus, are often minor. Similarly, we consider 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 can neither start before E_{rg} nor after L_{rg}. Here, snow removal time windows of a snow removal group can differ between runways depending on the current position of the snow removal group within

Table 2
Sets, parameters and decision variables.

Sets and related parameters	
Notation	Definition
$t \in \mathcal{T}$	Set of considered time points within planning horizon
$r \in \mathcal{R}$	Set of runways
U_r	Time at which aircraft 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 transfer time from runway r to runway s)
$a \in \mathcal{A}$	Set of aircraft
$\mathcal{C} \in \mathfrak{C}$	Set of aircraft operation classes
$c(a)$	Operation class of aircraft a
E_{ar}	Earliest possible 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
C_{art}	Cost coefficient for scheduling aircraft a on runway r at time t
$S_{\mathcal{C}\mathcal{C}'}$	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
$\mathcal{V} \in \mathfrak{V}_{rt}$	Set of cliques required to represent incompatibilities on runway r at time t
$\mathcal{W} \in \mathfrak{W}_{gt}$	Set of cliques required to represent incompatibilities for snow removal group g at time t
Decision variables	
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}	$= \begin{cases} 1 & \text{if snow removal group } g \text{ starts to clear runway } r \text{ at time } t \\ 0 & \text{otherwise} \end{cases}$
z_r	$= \begin{cases} 1 & \text{if runway } r \text{ is not cleared during the planning horizon} \\ 0 & \text{otherwise} \end{cases}$

the airport area, which impacts the driving time to the respective runways.

We denote the operation class of aircraft a by $c(a)$. We also enforce 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 a and following snow removals on the same runway, and runway-specific snow removal durations P_r. If an aircraft a is scheduled at time t, the take-off or landing operation starts at t and is completed at $t + S_{c(a)c(b)}$ (if aircraft b is scheduled next) or at $t + O_{c(a)}$ (if a snow removal is scheduled next). If a snow removal is scheduled at time t, snow removal activities start at t and are finished at $t + P_r$. Additionally, we consider that each runway r can become unsafe due to snow, ice, or slush at time U_r if it has not been cleared before in the planning horizon. If $U_r \geq \mathcal{T}^{\max}$, runway r does not become unsafe during the planning horizon. At the moment a runway becomes unsafe, the previous flight operation must have been completed. An unsafe runway must be cleared before it can be re-opened and used again. We assume that, due to the operational planning horizon of up to two hours, at most one snow removal must be scheduled for each runway. For snow removal groups, we consider sequence-dependent setup times Q_{rs} between runways r and s, which, for the ease of notation, include the time required to clear runway r and the transfer (driving) time from runway r to runway s. Our model minimizes the weighted deviation of aircraft from their target take-off or landing times and, therefore, the overall lateness cost of the schedule. The notation we use for our BP is summarized in Table 2. Note that the model formulation allows early starts of snow removals (before runways become unsafe) as well as late starts of snow removals (after runways have become unsafe). The objective function implicitly computes the optimal trade-off between early starts and late starts of snow removals considering all implications within the planning horizon, e.g., a loss of runway capacity due to late starts of snow removals or insufficient runway availability in case too many runways are cleared at the same time.

The presented 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 underlying idea of our model formulation is to generate constraints which ensure that at most one runway operation (aircraft take-off or landing or snow removal activity) utilizes a specific runway at a given point in time, taking also runway closures into account. For this, we formulate clique inequalities limiting the number of parallel operations for every runway at every point in time to one. Similarly, we generate clique inequalities for snow removal groups, limiting the number of parallel snow removal activities by the same snow removal group to one.

In our model formulation, we consider required separation times between two operations i and j as part of preceding operation i. We define subsets (cliques) of operations, i.e., of aircraft and snow removal groups, as $\mathcal{V} \subseteq (\mathcal{A} \cup \mathcal{G})$ with $|\mathcal{V}| \geq 2$. 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

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

The set of all cliques required for a specific combination of runway r and time t is denoted as $\mathfrak{V}_{rt}$. Similarly, we define 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 period t is denoted as $\mathfrak{W}_{gt}$.

With this, our BP is as follows:

$$\textbf{minimize} \quad \sum_{a\in\mathcal{A}} \sum_{r\in\mathcal{R}} \sum_{t\in\mathcal{T}_{ar}} C_{art} x_{art} \tag{1}$$

subject to

$$\sum_{r\in\mathcal{R}} \sum_{t\in\mathcal{T}_{ar}} x_{art} = 1 \qquad \forall\, a\in\mathcal{A} \tag{2}$$

$$\sum_{g\in\mathcal{G}} \sum_{t\in\mathcal{T}_{rg}} y_{grt} + z_r = 1 \qquad \forall\, r\in\mathcal{R} \tag{3}$$

$$\sum_{a\in\mathcal{V}\cap\mathcal{A}} \sum_{l\in[t-v_a(\mathcal{V})+1,t]\cap\mathcal{T}_{ar}} x_{arl} + \sum_{g\in\mathcal{V}\cap\mathcal{G}} \sum_{m\in[t-P_r+1,t]\cap\mathcal{T}_{rg}} y_{grm} \le 1 \qquad \forall\, r\in\mathcal{R}; t\in\mathcal{T}: t\le U_r; \mathcal{V}\in\mathfrak{V}_{rt} \tag{4}$$

$$\sum_{a\in\mathcal{V}\cap\mathcal{A}} \sum_{l\in[t-v_a(\mathcal{V})+1,t]\cap\mathcal{T}_{ar}} x_{arl} + \sum_{g\in\mathcal{V}\cap\mathcal{G}} \sum_{m\in[t-P_r+1,T^{\max}]\cap\mathcal{T}_{rg}} y_{grm} + z_r \le 1 \qquad \forall\, r\in\mathcal{R}; t\in\mathcal{T}: t> U_r; \mathcal{V}\in\mathfrak{V}_{rt} \tag{5}$$

$$\sum_{r\in\mathcal{W}} \sum_{l\in[t-w_r(\mathcal{W})+1,t]\cap\mathcal{T}_{rg}} y_{grl} \le 1 \qquad \forall\, g\in\mathcal{G}; t\in\mathcal{T}; \mathcal{W}\in\mathfrak{W}_{gt} \tag{6}$$

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

The Objective (1) minimizes the overall aircraft lateness cost of the schedule. Constraints (2) assign exactly one runway and one take-off or landing time to each aircraft. Constraints (3) assign a snow removal group and clearing time to a runway r if this runway is cleared during the planning horizon (and z_r is zero). If runway r is not cleared during the planning horizon, z_r equals one. Constraints (4) and (5) make sure that, at any point in time, each runway is occupied by at most one aircraft or snow removal group respecting all separation requirements. Constraints (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 (6) secure that, at any point in time, each snow removal group is conducting at most one snow removal respecting setup times between runways.

The number of Constraints (4)–(6) mainly depends on the cardinalities of sets $\mathfrak{V}_{rt}$, $\mathfrak{W}_{gt}$, and $\mathcal{T}$ since the cardinalities of sets $\mathcal{R}$ and $\mathcal{G}$ are comparatively small.

Our model formulation requires that all potential conflicts between aircraft operations, snow removal activities, and unsafe runways are reflected in at least one of the clique inequalities (4)–(6). Conditions (7)–(11) define such potential conflicts formally and state that, for each potential conflict, a corresponding clique inequality must be present in the model:

$$\forall\, a,b\in\mathcal{A}: a\neq b; r\in\mathcal{R}; t\in\{E_{ar},\ldots,L_{ar}+S_{c(a)c(b)}-1\}\cap\mathcal{T}_{br}$$
$$\exists\, \mathcal{V}\in\mathfrak{V}_{rt}: a,b\in\mathcal{V}\wedge S_{c(a)c(b)} = v_a(\mathcal{V}) \tag{7}$$

$$\forall\, a\in\mathcal{A}; g\in\mathcal{G}; r\in\mathcal{R}; t\in\{E_{ar},\ldots,L_{ar}+O_{c(a)}-1\}\cap\mathcal{T}_{rg}$$
$$\exists\, \mathcal{V}\in\mathfrak{V}_{rt}: a,g\in\mathcal{V}\wedge O_{c(a)} = v_a(\mathcal{V}) \tag{8}$$

$$\forall\, a\in\mathcal{A}; g\in\mathcal{G}; r\in\mathcal{R}; t\in\{E_{rg},\ldots,L_{rg}+P_r-1\}\cap\mathcal{T}_{ar}$$
$$\exists\, \mathcal{V}\in\mathfrak{V}_{rt}: a,g\in\mathcal{V} \tag{9}$$

$$\forall\, r,s\in\mathcal{R}: r\neq s; g\in G; t\in\{E_{rg},\ldots,L_{rg}+Q_{rs}-1\}\cap\mathcal{T}_{sg}$$
$$\exists\, \mathcal{W}\in\mathfrak{W}_{gt}: r,s\in\mathcal{W}\wedge Q_{rs} = w_r(\mathcal{W}) \tag{10}$$

$$\forall\, a\in\mathcal{A}; r\in\mathcal{R}; t\in\mathcal{T}_{ar}: t>U_r \qquad \exists\, \mathcal{V}\in\mathfrak{V}_{rt}: a\in\mathcal{V} \tag{11}$$

Condition (7) covers incompatibilities between two aircraft on the same runway due to separation requirements. Condition (8) covers incompatibilities between aircraft and following snow removals, while Condition (9) covers incompatibilities between snow removals and aircraft ensuring that no aircraft are scheduled during a snow removal. Condition (10) covers incompatibilities between two snow removals conducted by the same snow removal group ensuring sufficient setup times between snow removals. Condition (11) covers incompatibilities between aircraft and unsafe runways making sure that no aircraft is scheduled on a runway which is unsafe.

To create sets $\mathfrak{V}_{rt}$ and $\mathfrak{W}_{gt}$ satisfying Conditions (7)–(11), we use algorithms Construct-$\mathfrak{V}_{rt}$ (Algorithm 1) and Construct-$\mathfrak{W}_{gt}$ (Algorithm 2). They enumerate all potential conflicts and add corresponding cliques $\mathcal{V}$ and $\mathcal{W}$ to the model, while keeping the cardinalities of sets $\mathfrak{V}_{rt}$ and $\mathfrak{W}_{gt}$ small. 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 1) and on required separation times $O_{\mathcal{C}}$ between aircraft and following snow removals. Construct-$\mathfrak{W}_{gt}$ is based on the matrix of sequence-dependent setup times Q_{rs} between runways. Both algorithms enumerate all potential conflicts in a systematic and structured way ensuring that all separation requirements between aircraft and snow removals (Construct-$\mathfrak{V}_{rt}$) and all setup times between pairs of snow removals (Construct-$\mathfrak{W}_{gt}$) are reflected, and, therefore, Conditions

Algorithm 1 Construct-$\mathfrak{V}_{rt}$.

```
initialize set of cliques 𝔙_rt = ∅
for all leading operation classes 𝓛 ∈ ℭ do
    create set of separation times 𝒮 = {S_𝓛𝓕 : 𝓕 ∈ ℭ, O_𝓛}
    for all separation times s ∈ 𝒮 do
        if s ≤ S_𝓛𝓛 then
            create new clique 𝒱 of leading aircraft a, trailing
            aircraft b, and snow removal groups g
            with 𝒱 = {a : a ∈ 𝓛 ∧ t ∈ {E_ar, ..., L_ar + s − 1},
                      b : S_𝓛c(b) ≥ s ∧ t ∈ 𝒯_br,
                      g : g ∈ 𝒢 ∧ O_𝓛 ≥ s ∧ t ∈ 𝒯_rg}
            add 𝒱 to 𝔙_rt
        if s > S_𝓛𝓛 then
            for all leading aircraft a ∈ 𝓛 : t ∈ {E_ar, ..., L_ar + s − 1}
            do
                create new clique 𝒱 of leading aircraft a, trailing
                aircraft b, and snow removal groups g
                with 𝒱 = {a,
                          b : S_𝓛c(b) ≥ s ∧ t ∈ 𝒯_br,
                          g : g ∈ 𝒢 ∧ O_𝓛 ≥ s ∧ t ∈ 𝒯_rg}
                add 𝒱 to 𝔙_rt
remove dominated cliques from 𝔙_rt
```

Algorithm 2 *Construct-* $\mathfrak{W}_{gt}$.

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

(7)–(11) are fulfilled. Both algorithms potentially generate dominated cliques, leading to redundant or non-binding constraints in the BP. Here, a clique $\mathcal{V}$ dominates a 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, a clique $\mathcal{W}$ dominates a 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). In the last step of both algorithms, we identify such dominated cliques through a pairwise comparison of cliques and remove them from sets $\mathfrak{V}_{rt}$ and $\mathfrak{W}_{gt}$. This reduces the number of constraints in our model and helps keeping the size of the model manageable.

4.2. Additional constraints for the time-discrete binary program

Additional constraints which decrease the solution space can be derived from compulsory precedence relations. We exploit such compulsory precedence relations between snow removals and between aircraft of the same class and present aggregated-time and disaggregated-time variants for both precedence constraints.

Precedence constraints for snow removals on homogeneous runways based on dominant snow removal patterns

Pohl et al. (2021) showed that, for a set of homogeneous runways, i.e., runways allowing for the same flight operations, it is often possible to precompute a dominant snow removal pattern $\mathfrak{P}^* = \left((r_1, g_1, e_1), (r_2, g_2, e_2), \dots, (r_{|\mathcal{R}|}, g_{|\mathcal{R}|}, e_{|\mathcal{R}|})\right)$ and, thereby, an optimal sequence of snow removals. In pattern $\mathfrak{P}^*$, the i-th triple (r_i, g_i, e_i) denotes the i-th snow removal in chronological order with runway r_i being cleared by snow removal group g_i. e_i denotes the earliest possible start time of the i-th snow removal activity, which is constrained by preceding snow removals and resulting setup times. $e_i \leq e_j$ holds for $i < j$. A snow removal pattern defines a clear precedence for each pair of snow removals (r_i, g_i, e_i) and (r_j, g_j, e_j) such that snow removal (r_i, g_i, e_i) must not start after snow removal (r_j, g_j, e_j) if $i < j$. If a dominant, i.e., optimal, snow removal pattern for homogeneous runways is known, Constraints (12) express all corresponding precedence constraints in an aggregated-time variant.

$$\sum_{t \in \mathcal{T}_{r_j g_j}} (t \cdot y_{g_j r_j t}) + \max_{t \in \mathcal{T}_{r_i g_i}}(t) \cdot z_{r_j} - \sum_{t \in \mathcal{T}_{r_i g_i}} (t \cdot y_{g_i r_i t}) \geq 0$$
$$\forall\, (r_i, g_i, e_i), (r_j, g_j, e_j) \in \mathfrak{P}^* : i < j \qquad (12)$$

A disaggregated-time variant of the same precedence constraints for snow removals can be formulated as Constraints (13).

$$\sum_{l \in [0,t] \cap \mathcal{T}_{r_i g_i}} y_{g_i r_i l} - \sum_{m \in [0,t] \cap \mathcal{T}_{r_j g_j}} y_{g_j r_j m} \geq 0$$
$$\forall\, (r_i, g_i, e_i), (r_j, g_j, e_j) \in \mathfrak{P}^* : i < j; \quad t \in \mathcal{T}_{r_j g_j} \qquad (13)$$

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 those aircraft. For the general RSP, 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 have the same convex cost function. Briskorn & Stolletz (2014) showed that such complete orders within aircraft operation classes also exist if a time window order exists for all pairs of aircraft within a class and if class-specific piecewise linear convex cost functions are assumed. For the RSP without earliness (where aircraft can not be accelerated beyond their target time and $E_{ar} = T_{ar}$) and with aircraft-specific cost coefficients, Pohl et al. (2021) 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 RSP with aircraft acceleration (where $E_{ar} \leq T_{ar}$).

Theorem 1. *For two aircraft a and b of the same operation class, it is always optimal to schedule a not after b if $E_{ar} \leq E_{br}, L_{ar} \leq L_{br}\ \forall\, r \in \mathcal{R}$ (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). We denote such a compulsory precedence with $a \preceq b$.*

To prove Theorem 1, a swap argument can be applied. 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 can not be improved.

For pairs of aircraft a and b, for which it is known that a must not be scheduled after b, Constraints (14) express the corresponding precedence constraints in an aggregated-time variant.

$$\sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}_{br}} (t \cdot x_{brt}) - \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}_{ar}} (t \cdot x_{art}) \geq 0 \qquad \forall\, a, b \in \mathcal{A} : a \preceq b \quad (14)$$

A disaggregated-time variant of the same aircraft precedence constraints can be formulated as Constraints (15).

$$\sum_{r \in \mathcal{R}} \sum_{l \in [0,t] \cap \mathcal{T}_{ar}} x_{arl} - \sum_{r \in \mathcal{R}} \sum_{m \in [0,t] \cap \mathcal{T}_{br}} x_{brm} \geq 0$$
$$\forall\, a, b \in \mathcal{A} : a \preceq b; t \in \bigcup_{r \in \mathcal{R}} \mathcal{T}_{br} \qquad (15)$$

By adjusting the set $\mathcal{R}$ in Constraints (14) and (15), these constraints can be adapted to situations where aircraft precedence constraints only hold for a certain runway or a subset of runways.

In our computational experiments, we found the aggregated-time variants, especially for snow removal precedence constraints, to be more efficient because the disaggregated-time variants yield too many constraints.

4.3. Equivalent constraint programming model

In this section, we present a CP model formulation for the time-discrete WRSP. To denote our CP model, we use common scheduling concepts from CP and a terminology which is similar to the one used in IBM ILOG CPLEX CP Optimizer and in Goel, Slusky, van Hoeve, Furman, & Shao (2015) and Novara, Novas, & Henning (2016). We briefly introduce and define the used constructs. Specifically, we describe used variable types, expressions, and cumulative functions.

We use *interval variables* to model 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 $\tilde{x} \in \{[s, e[: s, e \in \mathbb{Z} \wedge s \leq e\} \cup \{\perp\}$. The set of possible values for an interval variable constitutes its domain. We use the following expressions and constraints for interval variables $\tilde{x}$ and $\tilde{y}$ and sets $\tilde{\mathcal{A}}$ of interval variables.

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

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

lengthOf($\tilde{x}$) equals 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.

We use *sequence variables* to model 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 ordering 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 $\Omega : \tilde{\mathcal{A}} \to \Theta$. A transition matrix can be specified as a function $M : \Theta \times \Theta' \to \mathbb{Z}$ 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. We use 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.

We use *cumulative functions* to model runways and snow removal groups as constrained resources. Hereby, a cumulative function's value over time represents a resource availability over time. Intervals individually contribute to a cumulative function by changing the function's value through elementary step functions, which represents the use and release of a resource. We use 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.

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

To facilitate constraint propagation and domain reduction in our CP model, we split required separation times $S_{\mathcal{C}\mathcal{C}'}$ between aircraft of operation class $\mathcal{C}$ and aircraft of operation class $\mathcal{C}'$ in a constant processing time $p = \min_{\mathcal{C},\mathcal{C}' \in \mathfrak{C}}\{S_{\mathcal{C}\mathcal{C}'}\}$ and a setup time $S^{\text{CP}}_{\mathcal{C}\mathcal{C}'} = S_{\mathcal{C}\mathcal{C}'} - p$.

We define two mappings of interval variables to types and two transition matrices accordingly. With regard to specific runway sequences, mapping $\Omega^R : \tilde{\mathcal{A}} \to \mathfrak{C} \cup \{srg\}$ 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 $M^R : \mathfrak{C} \cup \{srg\} \times \mathfrak{C} \cup \{srg\} \to \mathbb{Z}$ 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 $\Omega^G : \tilde{\mathcal{A}} \to \mathcal{R}$ 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 $M^G : \mathcal{R} \times \mathcal{R} \to \mathbb{Z}$ represents required transfer times between runways accordingly:

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

The function $Cost_{ar} : \mathcal{T} \to \mathbb{R}$ denotes an aircraft- and runway-specific cost function and determines the resulting earliness cost or tardiness cost if aircraft a is scheduled on runway r at time t. For cost factors C_{art} in the binary program and cost function $Cost_{ar}$ of the CP model, $C_{art} = Cost_{ar}(t)$ holds.

With this, our CP model formulation is as follows:

$$\textbf{minimize} \quad \sum_{a \in \mathcal{A}} \sum_{r \in \mathcal{R}} Cost_{ar}(\text{startOf}(\tilde{x}_{ar})) \tag{16}$$

subject to

$$\tilde{x}_{ar} \in [E_{ar}, L_{ar} + p) \qquad \forall\, a \in \mathcal{A}; r \in \mathcal{R} \tag{17}$$

$$\text{lengthOf}(\tilde{x}_{ar}) = p \qquad \forall\, a \in \mathcal{A}; r \in \mathcal{R} \tag{18}$$

$$\text{presenceOf}(\tilde{x}_a) = \texttt{TRUE} \qquad \forall\, a \in \mathcal{A} \tag{19}$$

$$\text{alternative}(\tilde{x}_a,\ \{\tilde{x}_{ar} : r \in \mathcal{R}\}) \qquad \forall\, a \in \mathcal{A} \tag{20}$$

$$\text{lengthOf}(\tilde{y}_{gr}) = P_r \qquad \forall\, g \in \mathcal{G}; r \in \mathcal{R} \tag{21}$$

$$\text{alternative}(\tilde{y}_r,\ \{\tilde{y}_{gr} : g \in \mathcal{G}\}) \qquad \forall\, r \in \mathcal{R} \tag{22}$$

$$\hat{\sigma}_r = \text{sequenceVar}(\{\tilde{x}_{ar} : a \in \mathcal{A}, \tilde{y}_{gr} : g \in \mathcal{G}\},\ \Omega^R) \qquad \forall\, r \in \mathcal{R} \tag{23}$$

$$\text{noOverlap}(\hat{\sigma}_r,\ M^R) \qquad \forall\, r \in \mathcal{R} \tag{24}$$

$$\hat{\tau}_g = \text{sequenceVar}(\{\tilde{y}_{gr} : r \in \mathcal{R}\},\ \Omega^G) \qquad \forall\, g \in \mathcal{G} \tag{25}$$

$$\text{noOverlap}(\hat{\tau}_g,\ M^G) \qquad \forall\, g \in \mathcal{G} \tag{26}$$

$$\sum_{a \in \mathcal{A}} \text{pulse}(\tilde{x}_{ar},\ 1) + \text{step}(U_r,\ 1) + \sum_{g \in \mathcal{G}} \text{stepAtEnd}(\tilde{y}_{gr},\ -1) \le 1 \qquad \forall\, r \in \mathcal{R} \tag{27}$$

Objective (16) minimizes the overall aircraft lateness cost of the schedule depending on the take-off and landing times of aircraft. Constraints (17) and (18) define the time windows and processing times for all aircraft. Constraints (19) and (20) ensure that each aircraft is scheduled on exactly one runway. Constraints (21) define the duration of snow removals on runways and Constraints

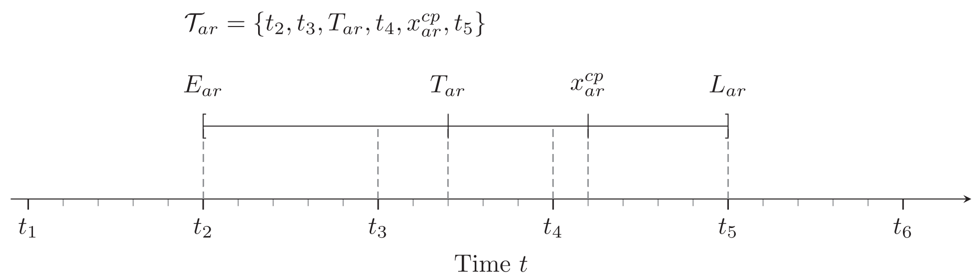

Fig. 1. Enhanced time discretization and resulting time window $\mathcal{T}_{ar}$ of aircraft a on runway r.

(22) assign exactly one snow removal group to each runway which is cleared. Constraints (23) declare runway-specific sequences of aircraft and snow removals and Constraints (24) ensure that all separation times between aircraft and snow removals on the same runway are met. Similarly, Constraints (25) declare a snow removal sequence for each snow removal group and Constraints (26) ensure sufficiently large setup times between snow removals for each snow removal group. Finally, Constraints (27) represent the availability of each runway and make sure that aircraft are only scheduled on a runway when it is safe.

4.4. Additional constraints for the constraint programming model

As for the BP, additional constraints which decrease the solution space can be derived from compulsory precedence relations between snow removals and between aircraft of the same operation class.

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 (28) assign to each runway a corresponding snow removal group and Constraints (29) express all derivable precedence constraints between snow removals.

$$\text{presenceOf}(\tilde{y}_{g_i r_i}) = \texttt{TRUE} \qquad \forall\, (r_i, g_i, e_i) \in \mathfrak{P}^* \tag{28}$$

$$\text{startBeforeStart}(\tilde{y}_{g_i r_i}, \tilde{y}_{g_j r_j}) \qquad \forall\, (r_i, g_i, e_i), (r_j, g_j, e_j) \in \mathfrak{P}^* : i < j \tag{29}$$

Precedence constraints for aircraft of the same operation class

For pairs of aircraft a and b with $a \preceq b$, Constraints (30) express the corresponding precedence relations.

$$\text{startBeforeStart}(\tilde{x}_a, \tilde{x}_b) \qquad \forall\, a, b \in \mathcal{A} : a \preceq b \tag{30}$$

5. Enhanced time discretization

To apply the time-discrete BP, we discretize the continuous planning horizon $[1, T^{\max}]$ by defining a set $\mathcal{T}$ of considered time points $t \in \mathcal{T} : 1 \leq t \leq T^{\max}$. The size of our 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 is increased, the number of variables x_{art} and y_{grt} as well as the number of constraints (4)–(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.

We propose the following time discretization approach to keep $|\mathcal{T}|$ small while, at the same time, enabling improved objective function values (cf. Fig. 1):

1. We 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, we add the aircraft's target time T_{ar} to the set $\mathcal{T}_{ar}$ and to the set $\mathcal{T}$. Thus, we open the possibility to schedule aircraft a on runway r exactly at its target time T_{ar} avoiding any earliness or tardiness cost, i.e., we do not have to deviate from T_{ar} if an optimal solution with $x_{art} = 1$ exists.
3. Similarly, for each aircraft a and runway r, we add the time $x_{ar}^{\text{CP}} = \text{startOf}(\tilde{x}_{ar})$ to the sets $\mathcal{T}_{ar}$ and $\mathcal{T}$ if the CP start heuristic (which solves the time-discrete problem for a step size of one) assigns aircraft a to runway r. Thus, we include the solution of the CP start heuristic for a step size of one, referred to as step-size-one solution, in our BP solution space. Notably, we include the optimal step-size-one solution for aircraft a on runway r to our BP solution space if it is found by the CP start heuristic.

For step sizes greater than one, this enhanced time discretization approach often enables significantly better solutions or even the optimal step-size-one solution while adding only marginal complexity (in terms of variables and constraints) to the model. Note that the optimal step-size-one solution equals the optimal time-continuous solution if all time parameters in the model are integer and the cost function is piecewise linear with integer break points.

6. Exact approach using constraint programming and column generation

In this section, we detail all steps of our algorithm to solve the time-discrete WRSP to optimality. Fig. 2 gives an overview of our proposed approach. During preprocessing, we compute dominant snow removal patterns to decrease the number of variables, which reduces our solution space. We use our CP model to reduce variable domains through constraint propagation and to generate an initial (incumbent) solution heuristically. This incumbent solution from the CP start heuristic provides the basis for the enhanced time discretization. We also use this heuristic solution to initialize a column generation scheme which solves the LP relaxation of our BP optimally. We identify all variables which are potentially required to solve the BP to integer optimality resulting in a column-reduced BP. In the last step, we use a branch-and-bound procedure to solve this column-reduced BP.

A key feature of the proposed 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. Preprocessing, 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

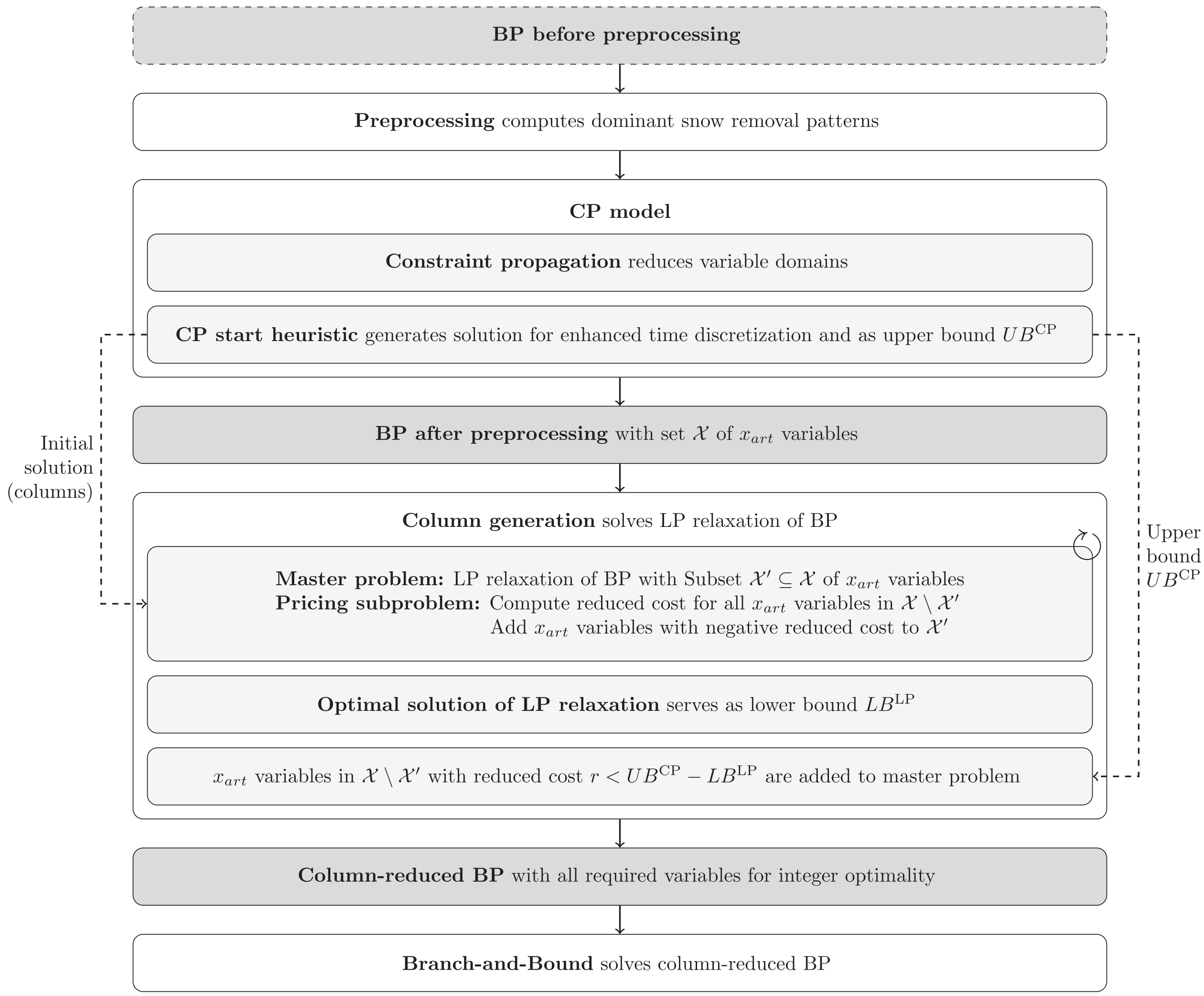

Fig. 2. Overview of our exact approach using constraint programming and column generation.

usually decreases the sizes of sets $\mathcal{T}_{ar}$, $\mathcal{T}_{rg}$, $\mathfrak{V}_{rt}$, and $\mathfrak{W}_{gt}$. This, subsequently, also reduces the number of constraints (4)–(6).

6.1. Preprocessing

In a preprocessing step, we use the concept of dominant snow removal patterns 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 precalculated combinations of runways, snow removal groups, and earliest possible snow removal times have to be considered in order to solve the WRSP optimally. Therefore, we restrict the y_{grt} variables to all tuples (g, r, t) with $(r, g, e) \in \mathfrak{P}^*$ and $e \leq t$.

6.2. Constraint programming for constraint propagation and as a start heuristic

We apply constraint propagation to our CP model in order to compute possible assignments of aircraft to runways and to tighten 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. We only include x_{art} and y_{grt} binary variables in our BP which correspond to $\tilde{x}_{ar}$ and $\tilde{y}_{gr}$ interval variables in the CP model after constraint propagation. This significantly reduces the number of y_{grt} and x_{art} variables. Furthermore, we exclude from $\mathcal{T}$ all time points t which can not be the start time of any interval variable. This reduces the number of constraints (4)–(6) in our BP and, thus, its model size and complexity.

We use a CP optimization engine to solve the CP model of the WRSP heuristically in order to derive an initial solution and upper bound UB^{CP} for the problem. We terminate the CP solution procedure 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, we compute these limits from the number of considered aircraft and runways. We use the resulting best found solution of the CP model for our enhanced time discretization approach and include the heuristically computed time-continuous solution in our 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. We also use the solution from the CP start heuristic as initial columns for our column generation scheme.

6.3. Column generation scheme to solve the linear relaxation of the binary program

After reducing the number of variables during preprocessing, the resulting BP still has a large number of x_{art} variables. Most of them, however, are non-basic in the optimal solution. We denote the full set of x_{art} variables after preprocessing as $\mathcal{X}$. We use column generation to identify a subset $\mathcal{X}' \subseteq \mathcal{X}$ which includes all x_{art}

variables required to solve the LP relaxation of the BP to optimality.

The master problem of our 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, we restrict our column generation scheme to the generation of x_{art} variables. We initialize the master problem 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, our pricing mechanism computes the 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. We add x_{art} variables with negative reduced cost to the variable set $\mathcal{X}'$ for the next iteration of the column generation and terminate the column generation 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 our BP.

6.4. Solving the binary program optimally

To compute the optimal integer solution for our BP, we apply a method initially developed and proven by Baldacci, Christofides, & Mingozzi (2007) to solve the capacitated vehicle routing problem. In the following, we describe how we adapt their approach to our problem and sum up the method and the rationale behind it.

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, we know that variables $x_{art} \in \mathcal{X}\backslash\mathcal{X}'$ with $LB^{\text{LP}} + \bar{c}_{x_{art}} \geq UB^{\text{CP}}$ can not 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 has been initialized with these variables. To secure that also all variables potentially required for integer optimality of the BP are present in the master problem, we add all variables $x_{art} \in \mathcal{X}\backslash\mathcal{X}'$ with $LB^{\text{LP}} + \bar{c}_{x_{art}} < UB^{\text{CP}}$ to the master problem, resulting in the column-reduced BP. Finally, we solve the integer variant of the column-reduced BP using a standard branch-and-bound procedure of a MIP solver.

7. Computational study

In our computational study, we test and evaluate our approach with real world data from Munich International Airport. We show that our enhanced time discretization approach enables good solutions even for larger step sizes. By analyzing model sizes in terms of variables and constraints, we show that preprocessing, constraint propagation, and our column generation approach significantly reduce the model complexity. Finally, we analyze the computational times of our approach. We show that our method outperforms a time-continuous model formulation for more complex instances and solves all instances of our computational study significantly faster than a pure CP approach. We computed all instances on an Intel i7-8700K 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 as CP optimization engine.

Table 3
Transfer times between runways (in seconds).

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

7.1. Description of test instances

Based on the data set of Pohl et al. (2021), we consider 24 realistic instances which differ in their characteristics and parameters. All instances consider flight operations of a winter day with considerable snow fall in November 2017 at Munich International Airport. We tested our approach with various parameter combinations and assumptions:

Aircraft time windows We consider two types of aircraft time windows which both are established in literature on runway scheduling. First, we consider instances without earliness, where an aircraft's target time constitutes its earliest possible take-off or landing time. Second, we consider instances allowing earliness where aircraft can be scheduled up to 10 minutes ahead of their target time causing respective earliness cost. In both cases, we allow aircraft tardiness of up to 20 minutes after target time. For each aircraft, we assume the same aircraft time window for all runways.

Cost function We consider two different types of cost functions, which both are used in literature (cf. Fig. 3). For the cost function type "linear", we assume that earliness cost as well as tardiness cost are linearly related to the deviation from an aircraft's target time. In practice, earliness often causes higher cost than tardiness, e.g., due to higher fuel consumption at higher aircraft speeds. We assume that the cost increase rate (slope of the cost curve) for earliness is 50% higher than for tardiness. To take into account that delays of larger aircraft with a higher number of passengers are more critical, we use cost increases for tardiness of 1, 2, and 3 monetary units per second for aircraft of the classes "Large", "Boeing 757", and "Heavy" respectively. For the cost function type "double", we assume that the cost increase rate of the linear cost function doubles every minute for earliness cost as well as for tardiness cost.

Snowfall scenario We consider two scenarios regarding snowfall. For the scenario "beginning snowfall", we assume that, at the start of the planning horizon, all runways have the same snow conditions. Due to snowfall, operations become unsafe at the same point in time on all runways, i.e., $U_1 = U_2 = \ldots = U_{|\mathcal{R}|}$. In our computational study, we assume that runways become unsafe 25 minutes after the beginning of the planning horizon. For the scenario "continuous winter operations", we assume that 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, i.e., $U_1 \neq U_2 \neq \ldots \neq U_{|\mathcal{R}|}$. In our computational study, we assume that the first runway becomes unsafe 10 minutes after the beginning of the planning horizon and that the second runway becomes unsafe after 25 minutes. In case of three runways, we assume that the third runway becomes unsafe after 40 minutes.

Table 4
Configurations and parameters of the instances.

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

Table 5
Improving objective function values through enhanced time discretization.

Instance	TD1 (as benchmark)			TD3			TD5			TD5e		
	Variables	Constraints	*Obj. Val.*	Variables	Constraints	*Obj. Val.*	Variables	Constraints	*Obj. Val.*	Variables	Constraints	*Obj. Val.*
T / l / begin / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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 / 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)*

All objective values rounded to integer.

Table 6
Reducing the model size through preprocessing, constraint propagation, and column generation.

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

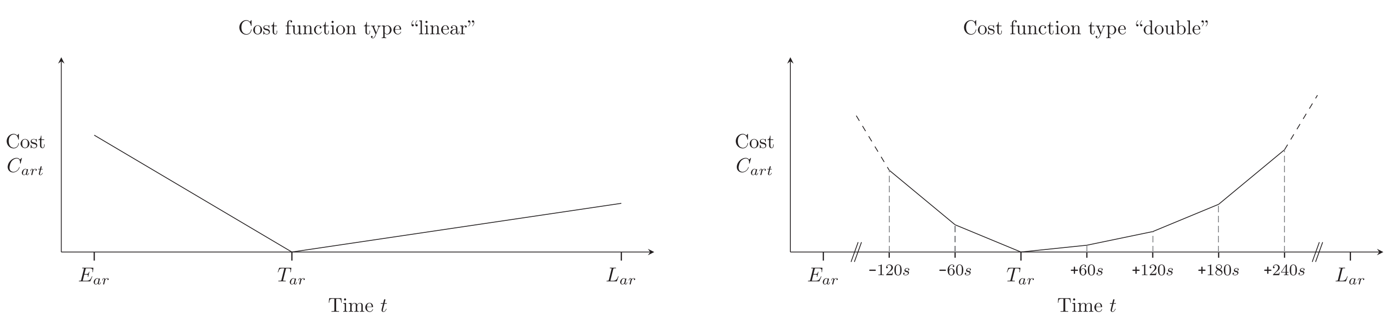

Fig. 3. Cost functions.

- **Runways and snow removal groups** We consider instances with either two runways and one snow removal group or with three runways and two snow removal groups. We consider sets of homogeneous and independent runways, i.e., runways allow for the same flight operations and flight operations on one runway do not affect flight operations on other runways. Table 3 lists considered transfer times between runways. For the first two runways, these are based on the road infrastructure of Munich International Airport. For the third runway, we consider an extension of the existing runway system as currently planned by the airport. In our experiments, we do not limit snow removal time windows and allow for all runways r and snow removal groups g the maximum set $\mathcal{T}_{rg} = \{0, \ldots, T^{\max}\}$ of potential snow removal times.
- **Aircraft and flight density** We compute all instances for 45 and 75 aircraft to be scheduled. For instances with two runways and one snow removal group, we consider 45 flight operations per hour. For instances with three runways and two snow removal groups, we assume 60 flight operations per hour.

Table 4 lists the complete configurations and parameters for all considered instances.

For the CP start heuristic, we used a limit of $2{,}000 \times |\mathcal{R}| \times |\mathcal{A}|$ fails for instances without earliness and a limit of $8,000 \times |\mathcal{R}| \times |\mathcal{A}|$ fails for instances with earliness to account for their larger solution space.

7.2. Balancing model size and solution quality through enhanced time discretization

In Table 5, we show that our enhanced time discretization approach enables high quality solutions while keeping the model size comparatively small. Therefore, we compare the number of variables and constraints and the optimal objective function values of four different time discretization variants. We compare step sizes of one second (TD1), three seconds (TD3), and five seconds (TD5) using a standard time discretization (without consideration of target times T_{ar} and heuristic solutions x_{ar}^{CP}) to our enhanced time discretization approach which uses a step size of five seconds and includes target times T_{ar} and heuristic solutions x_{ar}^{CP} in the solution space (TD5e). We use the most granular time discretization variant TD1 as a benchmark. We generated all four time discretization variants for all instances to determine the respective model sizes. When solving instances with TD1, for some of the instances, we ran out of memory and had to abort the solution procedure (indicated by "n/a" in column "Obj. Val.").

Table 7
Computational Times.

Instance	Time-continuous WRSP implementation (cf. Pohl et al., 2021)		Pure CP model of the WRSP			Our approach (Time-discrete WRSP using TD5e)									
						CP start heuristic			Column generation to solve LP relaxation			BP			
	Obj. Val.	*Time (s)*	Obj. Val.	CP internal Gap (%)	*Time (s)*	Obj. Val. (UB^{CP})	CP internal Gap (%)	*Time (s)*	Bound (LB^{LP})	Iterations	*Time (s)*	Obj. Val.	*Time (s)*	*Overall Time (s)*	*Improvement*
T / l / begin / 2 / 45	2,933	*6*	2,933	87	*> 300*	2,933	87	*2*	2,469	9	*1*	2,933	*4*	*7*	-
T / l / begin / 2 / 75	3,142	*40*	3,142	90	*> 300*	3,142	90	*5*	2,852	16	*4*	3,142	*11*	*20*	50%
T / l / begin / 3 / 45	1,683	*601*	1,683	100	*> 300*	1,683	100	*2*	977	9	*1*	1,683	*45*	*48*	92%
T / l / begin / 3 / 75	1,763	*1,680*	1,763	100	*> 300*	1,763	100	*6*	1,070	16	*4*	1,763	*63*	*73*	96%
T / l / cont / 2 / 45	2,933	*5*	2,933	87	*> 300*	2,933	87	*2*	2,857	12	*1*	2,933	*1*	*4*	20%
T / l / cont / 2 / 75	3,142	*26*	3,142	88	*> 300*	3,142	88	*5*	3,082	17	*4*	3,142	*3*	*12*	54%
T / l / cont / 3 / 45	664	*32*	664	100	*> 300*	664	100	*3*	533	16	*5*	664	*7*	*15*	53%
T / l / cont / 3 / 75	744	*90*	744	100	*> 300*	744	100	*6*	604	15	*7*	744	*23*	*36*	60%
E+T / l / begin / 2 / 45	1,966	*7*	2,070	92	*> 300*	2,206	92	*14*	1,637	8	*2*	2,004	*20*	*36*	-
E+T / l / begin / 2 / 75	2,133	*46*	2,508	94	*> 300*	2,492	94	*40*	1,919	13	*6*	2,170	*50*	*96*	-
E+T / l / begin / 3 / 45	1,086	*1,454*	1125	1,00	*> 300*	1,167	100	*17*	748	11	*3*	1,093	*42*	*62*	96%
E+T / l / begin / 3 / 75	1,166	*2,486*	1,916	100	*> 300*	1,238	100	*48*	831	16	*7*	1,175	*101*	*156*	94%
E+T / l / cont / 2 / 45	1,966	*6*	2,136	92	*> 300*	2,349	93	*11*	1,875	8	*1*	1,999	*8*	*20*	-
E+T / l / cont / 2 / 75	2,133	*40*	2,511	94	*> 300*	2,530	94	*33*	2,078	11	*4*	2,171	*19*	*56*	-
E+T / l / cont / 3 / 45	428	*123*	435	100	*> 300*	448	100	*17*	351	13	*4*	430	*8*	*29*	76%
E+T / l / cont / 3 / 75	586	*1,503*	590	100	*> 300*	618	100	*49*	481	18	*15*	592	*45*	*109*	93%
E+T / d / begin / 2 / 45	2,373	*3*	2,917	91	*> 300*	2,686	90	*12*	1934	8	*1*	2,413	*5*	*18*	-
E+T / d / begin / 2 / 75	2,540	*24*	3,254	93	*> 300*	3,050	93	*34*	2,206	10	*4*	2,585	*36*	*74*	-
E+T / d / begin / 3 / 45	1,247	*2,779*	1,292	100	*> 300*	1,507	100	*17*	756	10	*2*	1,261	*55*	*74*	97%
E+T / d / begin / 3 / 75	1,327	*> 3,600*	1,379	100	*> 300*	1,544	100	*49*	843	14	*6*	1,336	*104*	*159*	96%
E+T / d / cont / 2 / 45	2,373	*3*	3,113	92	*> 300*	2,839	91	*10*	2,231	9	*1*	2,429	*3*	*14*	-
E+T / d / cont / 2 / 75	2,540	*29*	3,391	93	*> 300*	3,288	93	*28*	2,461	11	*4*	2,602	*12*	*44*	-
E+T / d / cont / 3 / 45	428	*140*	436	100	*> 300*	458	100	*15*	351	15	*5*	432	*7*	*27*	81%
E+T / d / cont / 3 / 75	594	*> 3,600*	625	100	*> 300*	647	100	*46*	484	15	*12*	602	*55*	*113*	97%

All objective values rounded to integer.

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 each 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 each 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 losing granularity and solution quality compared to a step size of one second. A standard time discretization of three seconds (TD3) yields optimal objective function values which are up to 14.5% (on avg. 4.3%) higher than objective function values of variant TD1. If a standard time discretization of five seconds (TD5) is used, respective optimal objective function values increase up to 25.0% (on avg. 8.7%). With our enhanced time discretization (TD5e), we achieve optimal objective function values which are only up to 2.4% (on avg. 0.9%) higher than for TD1 while realizing almost the full model size reduction of factor 25.

Considering operational circumstances at airports, a scheduling accuracy of five seconds is often not achievable in practice. Given that, during peak periods, aircraft are scheduled every 80 to 90 seconds on airport runways (cf. Flughafen München, 2014; Kobie, 2018) and considering the distribution of required separation times between aircraft (cf. Table 1), step sizes of up to 20 seconds seem reasonable. While, in our computational study, we are showcasing the capabilities of our method using a step size of up to five seconds (TD5e), our approach is also applicable and valid for larger step sizes. Larger step sizes would further reduce the problem size and accelerate the solution procedure.

7.3. Reducing the number of variables through preprocessing, constraint propagation, and column generation

Table 6 shows the impact of preprocessing, constraint propagation, and the column generation approach on the size of the resulting model. For this analysis, we use time discretization variant TD5e due to its favorable trade-off between model size and solution quality. For all considered instances, we report the number of variables and constraints of the original BP before preprocessing (corresponding to columns “TD5e” of Table 5). We also show the number of variables and constraints after preprocessing and constraint propagation. In the last two columns, we report the number of variables and constraints which remain after the column generation scheme in the column-reduced BP. This includes all variables x_{art} with $LB^{\mathrm{LP}} + \bar{c}_{x_{art}} < UB^{\mathrm{CP}}$ and, thus, all variables which are required to solve the BP to integer optimality. Our column generation scheme generates only 14.7% to 56.0% (on avg. 29.6%) of all variables of the original BP. This corresponds to 15.4% to 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% to 27.4% (on avg. 9.6%) of the matrix size of the original BP before preprocessing.

7.4. Analysis of computational times

In Table 7, we report in detail the computational performance of our proposed algorithm using enhanced time discretization variant TD5e. We present details on the three main components of our 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, we report the best found solution as upper bound UB^{CP}, the CP internal optimality gap at termination of the heuristic, and the computational time at termination (due to reaching the given fail limit in seconds). For the column generation phase, we report the number of required iterations before no more variables with negative reduced cost are found and the corresponding solution as LB^{LP}. As computational time, we report the combined time over all iterations to compute the LP solutions. For the final branch-and-bound procedure, we report the computational time to solve the column-reduced BP to integer optimality and the objective value of the final solution. In the last column, we show the sum of computational times over all three components as overall computational time. We also compare our algorithm to a pure CP approach and report the objective value of the best found solution and the remaining CP internal optimality gap of the CP model after a time limit of five minutes. Additionally, we compare our algorithm to an implementation of the time-continuous WRSP model presented in Pohl et al. (2021) using a time limit of one hour. Where applicable, we report the relative decrease of computational times through our new approach (compared to the time-continuous WRSP model) in column *Improvement*.

Our algorithm based on TD5e solves all 24 considered instances of the time-discrete WRSP to optimality within three minutes. For 15 out of 24 instances, we compute the optimal solution in less than one minute. Within the chosen fail limits, our CP start heuristics calculates good solutions deviating at most 26.4% (8.4% on avg.) from the optimal solution in 2 to 49 seconds. For instances without aircraft earliness, the CP start heuristic finds the optimal solution but can not prove its optimality and terminates with significant optimality gaps above 87%. For all instances, we require less than 20 iterations to generate all required columns and, in each iteration, we calculate the solution of the LP in less than one second on average.

Our algorithm outperforms a pure CP approach, which can not solve any instance to proven optimality within a time limit of five minutes. The computational experiments also show that, for many instances, our approach outperforms the computational times of a time-continuous model formulation. Overall, the time-discrete formulation has lower computational times for 15 of the 24 instances. Remarkably, the time-discrete model performs significantly better for large problem sizes. Our approach solves much faster all instances with more than two runways and reduces computational times in these cases by up to 97%.

8. Conclusion

Previous research showed that an integrated scheduling of aircraft and snow removals on runways enables large efficiency gains. The existing time-continuous model for the WRSP struggles to solve larger and more complex instances to proven optimality. Therefore, we presented a time-discrete variant of the WRSP and an exact solution algorithm to solve it. We proposed a CP start heuristic and a BP, which we solved using a column generation scheme and a branch-and-bound procedure. We demonstrated the efficiency of our algorithm by applying it to realistic instances from Munich International Airport and showed that our algorithm regularly outperforms the time-continuous model formulation and a pure CP approach in terms of computational times. We also proposed an enhanced time discretization method which enables high-quality solutions while maintaining small model sizes. We achieve this through a model formulation which allows time discretization with variable step sizes and by utilizing results from the CP start heuristic. Preprocessing, constraint propagation, and column generation further reduce the size of our time-discrete model resulting in a reduced BP whose matrix size is, on average, less than 10% of the matrix size of the original BP. Based on this reduction of model size, we hope that our proposed algorithm combining CP techniques and column generation is transferable and adaptable also to time-discrete optimization models with high numbers of variables in other domains and encourage further research in that direction.

While this paper solved the static and deterministic version of the WRSP, many aspects of the problem setting, including aircraft time windows, weather conditions, and potential operational disruptions, are subject to change or uncertainty. Therefore, it might be worthwhile to account for the stochasticity of the problem and to explore how concepts of robust optimization can be applied to the WRSP.

References

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).

Allignol, C., Barnier, N., Flener, P., & Pearson, J. (2012). Constraint programming for air traffic management: A survey. *The Knowledge Engineering Review, 27*(3), 361–392.

Avella, P., Boccia, M., Mannino, C., & Vasilyev, I. (2017). Time-indexed formulations for the runway scheduling problem. *Transportation Science, 51*(4), 1196–1209.

Balakrishnan, H., & Chandran, B. G. (2010). Algorithms for scheduling runway operations under constrained position shifting. *Operations research, 58*(6), 1650–1665.

Baldacci, R., Christofides, N., & Mingozzi, A. (2007). An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming, 115*(2), 351–385.

Beasley, J. E., Krishnamoorthy, M., Sharaiha, Y. M., & Abramson, D. (2000). Scheduling aircraft landings - the static case. *Transportation Science, 34*(2), 180–197.

Bennell, J. A., Mesgarpour, M., & Potts, C. N. (2011). Airport runway scheduling. *4OR, 9*(2), 115–138.

Bennell, J. A., Mesgarpour, M., & Potts, C. N. (2017). Dynamic scheduling of aircraft landings. *European Journal of Operational Research, 258*(1), 315–327.

Bertsimas, D., & Frankovich, M. (2015). Unified optimization of traffic flows through airports. *Transportation Science, 50*(1), 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*(3), 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*(1), 31–45.

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*(1), 59–81.

Faye, A. (2015). Solving the aircraft landing problem with time discretization approach. *European Journal of Operational Research, 242*(3), 1028–1038.

Flughafen München (2014). Facts and figures - munich airport in brief. https://web.archive.org/web/20160304023703/http://www.munich-airport.de/media/download/general/publikationen/en/facts_and_figures.pdf.

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).

Gabteni, S., & Grönkvist, M. (2008). Combining column generation and constraint programming to solve the tail assignment problem. *Annals of operations research, 171*(1), 61–76.

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*(3), 662–673.

Grönkvist, M. (2006). Accelerating column generation for aircraft scheduling using constraint propagation. *Computers & Operations Research, 33*(10), 2918–2934.

Gualandi, S., & Malucelli, F. (2013). Constraint programming-based column generation. *Annals of Operations Research, 204*(1), 11–32.

Hansen, J. V. (2004). Genetic search methods in air traffic control. *Computers & Operations Research, 31*(3), 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*.

Kobie, N. (2018). The wild logistics of heathrow airport will instantly devour its much-needed third runway. https://www.wired.co.uk/article/heathrow-third-runway-plans-expansion.

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*(2), 229–245.

Maere, G. D., Atkin, J. A. D., & Burke, E. K. (2017). Pruning rules for optimal runway sequencing. *Transportation Science, 52*(4), 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*(1), 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*(2), 439–462.

Pohl, M., Kolisch, R., & Schiffer, M. (2021). Runway scheduling during winter operations. *Omega, 102*, 102325.

Psaraftis, H. N. (1980). A dynamic programming approach for sequencing groups of identical jobs. *Operations Research, 28*(6), 1347–1359.

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*(1), 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*(1), 39–49.

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*(3), 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.

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).