ARTICLE IN PRESS

Journal of Air Transport Management xxx (2016) 1–10

Contents lists available at ScienceDirect

Journal of Air Transport Management

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

Robust runway scheduling under uncertain conditions

Andreas Heidt [a,*], Hartmut Helmke [b], Manu Kapolke [a], Frauke Liers [a], Alexander Martin [a]

[a] Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Department of Mathematics, Discrete Optimization, Cauerstr. 11, 91058 Erlangen, Germany
[b] German Aerospace Center (DLR), Institute of Flight Guidance, Lilienthalplatz 7, 38108 Braunschweig, Germany

ARTICLE INFO

Article history:
Received 30 July 2015
Received in revised form
12 November 2015
Accepted 9 February 2016
Available online xxx

Keywords:
Scheduling
Uncertainty
Time-indexed model
MIP
Mixed-integer programming
Dynamic time-indexed model
Strict robustness
Light robustness

ABSTRACT

The runway is the main element that combines airside and groundside of the ATM System. Thus, it is crucial to develop efficient models and planning algorithms for its effective usage. The best planning algorithm, however, is useless if the resulting plans cannot be implemented in the real world. This often happens because the input data of the planning algorithms face disturbances or changes over time, respectively. For example, an estimated time of arrival/departure of an aircraft may be changed. It is usually not certain for the next ten hours.

In this work, we study the runway scheduling problem under uncertain conditions. First, we present mathematical optimization models that ignore uncertainties. In the most effective approach, we compute for every discretized point in time whether an aircraft is scheduled and if so, which one is. Then, in each planning step we take uncertainties into account. We then apply different robust optimization methods in order to devise solution approaches that lead to stable plans. These optimization approaches are integrated into a simulation tool and evaluated in different traffic scenarios.

The Monte-Carlo simulations for a mixed-mode runway system show that our robust approaches result in fewer sequence changes and target time updates, when compared to the usual approach in which the plan is simply updated in case of infeasibility. Thus, we show that protection against uncertainties by using robust optimization indeed leads to considerably more stable plans.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Planning, particularly scheduling of limited resources is one of the main tasks of Air Traffic Management (ATM). However, uncertainty, inaccuracy, and non-determinism almost always lead to deviations from the actual plan. Typical strategies to deal with these changes is to simply ignore them (plan freezing) or – slightly better – to regularly recompute or update the schedule $t(j)$ (see Fig. 1). $t(j)$ denotes the planned target time in the jth planning cycle. These adjustments are usually performed in *hindsight*, after the actual change in the data has occurred. This usually leads to schedules with reduced overall utilization and reduced throughput. The disadvantages of such an approach is obvious: A formerly optimum plan might not even be feasible any more after some disturbance has occurred. In this work, we follow a different route. First of all, we use mathematical optimization in order to design global optimum runway scheduling plans. Furthermore, in contrast to ignoring disturbances, we know there are disturbed scenarios that we integrate into the models a priori. Thus, the challenge is to incorporate uncertainty into the initial computation of the plans so that these plans are *stable* with respect to changes in the data. This leads to a better utilization of resources as well as to a more efficient support for ATM controllers and stakeholders.

Two different approaches currently exist to handle uncertainty in mathematical optimization: on the one hand *stochastic optimization* that can be used to compute good average solutions and on the other hand *robust optimization* to immunize against predefined worst-case scenarios. In *normal life* we intuitively act similarly. As a host of an invitation for 8 p.m. we know that there are some guests who often do not arrive before 8:30, but others will arrive in time. Depending on the invited guests, we start our preparation very early or we know that we still have time. In the stochastic case, our host tries to find a good compromise between his waiting time for the first guests and the waiting time of the guests for the host (still taking a shower and searching a pair of socks). In the robust case, however, the host tries to avoid the awkward situation that the first guest would arrive earlier than the host is prepared for.

* Corresponding author.
E-mail addresses: andreas.heidt@fau.de (A. Heidt), hartmut.helmke@dlr.de (H. Helmke), manu.kapolke@fau.de (M. Kapolke), frauke.liers@fau.de (F. Liers), alexander.martin@fau.de (A. Martin).

http://dx.doi.org/10.1016/j.jairtraman.2016.02.009
0969-6997/© 2016 Elsevier Ltd. All rights reserved.

Please cite this article in press as: Heidt, A., et al., Robust runway scheduling under uncertain conditions, Journal of Air Transport Management (2016), http://dx.doi.org/10.1016/j.jairtraman.2016.02.009

ARTICLE IN PRESS

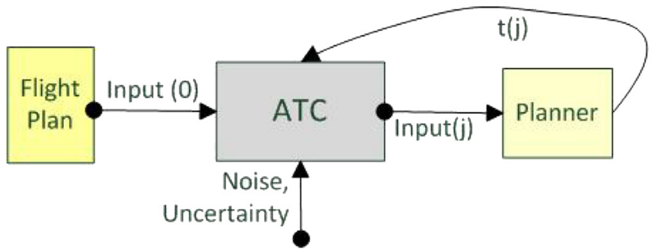

Fig. 1. The ATC planning and feedback cycle (Heidt et al., 2014).

In this paper we concentrate on robust optimization, i.e. the goal is to avoid improbable, but critical situations in order to obtain stable plans. Stochastic optimization and its combination with robustness is considered in the WP-E funded project *RobustATM* (Kapolke et al., 2016). However, an increase in stability naturally comes with the price that it might reduce the runway efficiency in general. In order to keep this price under control, we present different robust optimization concepts, a strict robust one and a light robust one. In the latter, the user specifies beforehand what reduction in efficiency is acceptable. Our methods then maximize the stability of the schedule while keeping the efficiency at the acceptable limit. Light robustness has been developed for railways (Fischetti and Monaci, 2009). We show here that it can be successfully applied for runway scheduling as well.

An extended abstract of this paper has already appeared in Heidt et al. (2014), in which two mathematical models for runway scheduling are used and evaluated in different traffic scenarios. Furthermore, the model that performs better in practice is extended to a strict robust model. It is investigated and validated within a simulation procedure. Finally, the results obtained from Monte Carlo simulations with the different models are compared and discussed. The remainder of this paper is structured as follows. Section 2 presents related works. In the following Section 3 we introduce the optimization models used for the runway scheduling problem and compare them experimentally. One approach computes the target times with an accuracy of one second, whereas the second approach discretizes the timeline into time slots with a width of *x* seconds and assigns aircraft to those. We get a trade-off between runtime performance and accuracy. In Section 4, uncertainties are taken into account and a strict robust model as well as a light robust model is introduced. Afterwards, the performed experiments are described in Section 5 and their results are compared in Section 6. We concentrate on runway scheduling only as an example to explain our approach, which is also applicable for other ATM planning problems. A MIP model of course is needed and we need explicit probabilistic knowledge of the uncertainties/possible disturbances.

2. Related work

The development of Arrival Managers for assisting air traffic controllers has started in the early eighties of the last century (see Neuman and Erzberger, 1990; Völkers, 1990). The development of Departure Managers (Böhme, 2005) goes back to the nineties. Bennell et al. (2013) wrote a survey on arrival and departure planning. They summarized techniques and tools of operation research and management sciences.

Usually, heuristic approaches are used for computing the air traffic sequences. Based on a first-come-first-served (FCFS) principle, constrained position shifting (CPS) was used by Balakrishnan and Chandran (2006). Here, only a limited number of aircraft can change their position. Hu and Chen (2005) used position shifting to determine an arrival sequencing. Hu and Di-Paolo (2008) furthermore proposed a genetic algorithm. Moreover, metaheuristics such as Tabu-Search have also been studied for the runway scheduling at London Heathrow Airport by Atkin et al. (2007).

Beasley et al. (2000) formulated an exact mixed-integer-program (MIP), where the variables are the exact target times of the aircraft. They used a linear cost function and developed heuristics and linear programming (LP) based on tree search. Clare and Richards (2011) used the same mixed-integer formulation and solved it by receding horizon methods. Considering a time-indexed (or time-discretized) formulation for runway scheduling, Anagnostakis and Clarke (2003) used a heuristic approach for departure planning. Here, a two-stage formulation was proposed. In the first stage, weight classes are assigned to slots, whereas in the second stage the specific aircraft are scheduled. In Kjenstad et al. (2013), the time-indexed formulation was used for Hamburg and Arlanda airport. The authors obtained short computational runtime (<20 s) for Hamburg Airport. In 2013, Frankovich and Bertsimas (2013) used a two-phase formulation including the time-indexed model over a time horizon of one hour and applied the model to historical data of Boston and Dallas Airport.

According to our knowledge, only few works exist in which the runway scheduling problem has been studied under uncertainties. Using stochastic programming, Corolli et al. (2014) tackled the strategic problem of departure and arrival slot allocation at airports under uncertainty. In Samà et al. (2014), a rolling horizon approach was developed for the problem of monitoring and controlling aircraft with disturbed take-off/landing times in the terminal control area. Jacquillat and Odoni (2014) optimized the airport's flight schedule at a strategic level. They also balanced the arrival and departure rates at a tactical level to mitigate airport congestion. Iterative, they used a stochastic queuing model for congestion, a dynamic programming model for capacity utilization and an integer programming model for scheduling interventions. They did not integrate robust optimization a priori. In the work of Chandran and Balakrishnan (2007) a heuristic approach was developed using FCFS with CPS for stochastic deviations in the earliest times. Agogino (2011) used a MIP model for departures, included normally distributed delay and proposed evolutionary algorithms for this robust scheduling problem. In the PhD thesis of Solveling (2012) a MIP model was solved by a 2-stage-stochastic approach using a stochastic branch & bound method. Apart from that, mathematically robust approaches have hardly been considered for solving the runway scheduling problem to optimality.

3. A mixed-integer model for the scheduling problem

In this section we model the runway scheduling problem as mixed-integer programs. First, we only plan once and ignore uncertainties. So we can simplify Fig. 1 in the way that *Input(0)* is just the input into the *Planner* and the output of the *Planner* has no effect on the *ATC* world. Thus, all aircraft land/depart at their planned time. We assume a given set of flights *F*. We abstract from a real airport with many constraints and terms in the objective function. Our simple objective function, which may be replaced by a more complex one in the future minimizes the sum of the deviations from the scheduled times. Furthermore, we want to avoid inbound holdings and want to meet outbound slot constraints, i.e. departure time constraints in order to meet an acceptable given latest time τ_i^{LT}. In our simulation we assume outbound slot sizes of 15 min. Slot sizes of this range are often used for balancing the demand capacities. Furthermore, each aircraft has a schedule time τ_i^{ST}. It is assigned an earliest possible time τ_i^{ET}, which is treated as a hard constraint. A violation of the latest time τ_i^{LT} just results in bad objective function value as well as holdings or departure slot losses.

Please cite this article in press as: Heidt, A., et al., Robust runway scheduling under uncertain conditions, Journal of Air Transport Management (2016), http://dx.doi.org/10.1016/j.jairtraman.2016.02.009

Due to wake vortexes and runway occupancy times, we have to maintain minimum separation times $\delta_{i,k}$ between two aircraft $i \in F$ and $k \in F$, which depend on the respective aircraft weight classes. Furthermore, fair schedules should be achieved. This means that it is worse if one aircraft has a delay of 30 min than three that are delayed 10 min each. We consider the following two different mathematical models: the first one is a mixed-integer model that decides the ordering of the aircraft together with their touch-down/take-off times. The second one uses a time discretization. The latter computes for every discretized point in time whether an aircraft is scheduled to it and if so, which one is. Based on this, we used a dynamic time-indexed model with flexible time discretization for each aircraft. As a next step, we compare the time-indexed models to the mixed-integer model. Many other heuristic and exact approaches exist to directly solve the runway scheduling problem (see Helmke (2011) for an overview). We will, however, benefit from our approach with a time-indexed model as soon as we consider uncertainty in Section 4.

3.1. A mathematical model with exact touch-down/take-off times (MIP)

In this section we use the model from Beasley et al. (2000) for the Runway Scheduling Problem with exact touch-down/take-off times. We however use a different objective function in order to ensure fairness. We introduce continuous variables $t_i \in \mathbb{R}$ that denote the touch-down or take-off time. As we want to schedule the set of all aircraft as close as possible to their schedule time, we minimize the sum of the absolute values of the differences between the touch-down/take-off time and schedule times. Thus, the first objective function would read as

$$\min \sum_{i \in F} \left| t_i - \tau_i^{ST} \right|.$$

In order to ensure fairness as well, we use the square deviation from the schedule time which yields

$$\min \sum_{i \in F} \left(t_i - \tau_i^{ST} \right)^2.$$

Planning aircraft after their latest time is possible, but then undesirable flight maneuvers have to be performed, e.g., airborne or ground holdings. Thus, we additionally penalize if an aircraft is planned after its latest time. This can be achieved by an auxiliary variable t_i^{LT+} for every aircraft that satisfies

$$t_i^{LT+} = max\left\{0, t_i - \tau_i^{LT}\right\}.$$

This expression enforces t_i^{LT+} to take the value 0, if the target time is smaller than the latest time. It takes the difference between target time and latest time in case aircraft i is planned later than its latest time.

Combining these expressions, the overall objective function then reads as

$$\min \sum_{i \in F} \left(\omega_{ST} \left(t_i - \tau_i^{ST} \right)^2 + \omega_{LT} \left(t_i^{LT+} \right)^2 \right), \tag{1}$$

with weights $\omega_{ST}, \omega_{LT} \in \mathbb{R}$.

The constraints then model that an aircraft cannot land/depart before its earliest time. Furthermore, safety distances need to be maintained. For the details about the constraints and how they are modeled, we refer to Heidt et al. (2014).

Computationally, it turns out that this formulation is solvable to optimality only for small instances of aircraft sets <15 elements within acceptable time, i.e., within less than 10 min. In the following, we introduce a different optimization model for the runway scheduling problem. We show experimentally that this time-discretized formulation leads to more effective solution algorithms, when compared to the model introduced above.

3.2. Runway scheduling as an assignment problem with side constraints

Motivated by Dyer and Wolsey (1990), we focus on a time-indexed model (TIM) for the runway scheduling, similar to what has been done in Kjenstad et al. (2013).

The time horizon $T = \{t_1, \ldots, t_m\}$ is discretized into time slots $t_j \in T$. We use equidistant time intervals for each aircraft. For each aircraft $i \in F$, the interval $[\tau_i^{ET}, \tau_i^{LT}]$ denotes the target time window in which the touch-down and take-off times may vary. Thus, for each aircraft $i \in F$ the time horizon reduces to

$$T_i = T \cap \left[\tau_i^{ET}, \tau_i^{LT} \right]. \tag{2}$$

For example, if the discretization is given by $T = \{1000, 1075, 1150, \ldots, 1600\}$ and the earliest/latest time interval for aircraft i by [1111,1582], the time horizon for aircraft i would be $T_i = \{1150, 1225, \ldots, 1525\}$. If aircraft $i \in F$ proceeds aircraft $k \in F$, $\delta_{i,k}$ is the minimum separation time. For each aircraft $i \in F$ and time slot $t_j \in T_i$, binary variables $b_{i,j}$ are introduced to declare whether an aircraft is scheduled on a slot or not:

$$b_{i,j} = \begin{cases} 1, & \text{if aircraft } i \in F \text{ scheduled on slot } t_j \\ 0, & \text{otherwise} \end{cases}$$

The goal is to minimize the sum of deviation from the schedule time τ_i^{ST}, We furthermore penalize all aircraft which are scheduled later than their latest time $t_i^{LT+} = max\{0, t_j - \tau_i^{LT}\}$. Additionally, we use quadratic deviations to reduce unfairness in the schedules. Thus, we want to find an assignment $b_{i,j}$ which minimizes the objective function with weights $\omega_{ST}, \omega_{LT} \in \mathbb{R}$:

$$\min \sum_{i \in F} \sum_{j \in \mathcal{T}_i} b_{ij} \left(\omega_{ST} \left(t_j - \tau_i^{ST} \right)^2 + \omega_{LT} \left(t_i^{LT+} \right)^2 \right). \tag{3}$$

It is worth mentioning that the quadratic term now consists of exclusively input data, since t_j are fixed slots - the variables b_{ij} are linear in the objective. Thus, there is only a contribution to the objective function value if an aircraft i is assigned to slot t_j and thus $b_{i,j} = 1$. Hence, we are able to model arbitrarily complex objective functions without losing performance, if we model the scheduling task as a slot assignment problem. The first set of constraints models that each aircraft has to be scheduled:

$$\sum_{j \in T_i} b_{i,j} = 1 \quad \forall i \in F \tag{4}$$

We have to ensure that each slot can be used at most once.

$$\sum_{i \in F} b_{i,j} \leq 1 \quad \forall j \in T \tag{5}$$

For each pair of aircraft and each time slot one constraint considers minimum separation times. If aircraft i is scheduled on time slot t_j, it is forbidden for aircraft k to be scheduled on following time slots until the separation time $\delta_{i,k}$ is reached. This yields

$$b_{i,j} + \sum_{l=j+1}^{j+\left\lceil \frac{\delta_{i,k}}{\Delta t} \right\rceil} b_{k,l} \leq 1 \quad \forall i \in F, \ \ \forall j \in T_i, \ \ \forall k \neq i, \tag{6}$$

where Δt describes the length of one time slot. Since $\delta_{i,k}/\Delta t$ can be fractional, we have to round up this value to ensure that the minimum separation times holds. By this, the buffer between two aircraft can be larger than the minimum separation times $\delta_{i,j}$. The optimization thus computes for every discretized point in time whether an aircraft is scheduled and if so, which one is. Mathematically, without constraint (6), we get an assignment problem, which is well understood and solvable in polynomial time (Ahuja et al., 1993). In general, the problem with constraints (6), however, is difficult to solve in practice for large instances. For the application of runway scheduling we have minimum separation between 75 and 150 s. Hence, solutions for example for slot sizes of $\Delta t = 75$s have good runtime performance, but often result in low objective function values with respect to runway utilization. The reason is that we plan at most one aircraft into a slot which might mean that two aircraft are planned further apart from each other than necessary. In contrast, solutions for slot sizes of $\Delta t = 5$s compute good objective function values, but have poor computational runtime. The challenge is to reduce the number of time-indexed variables so that the problem is still computationally tractable. On the other hand, we have to compute acceptable objective function values.

In ATM business it is, however, seldom necessary to plan take-off or touch down times with an accuracy of five or even one second, if the event is still far in the future, e.g. in 30 or even in 120 min from now on. This idea is elaborated in the next section by the dynamic time-indexed model approach with variable slot sizes.

3.3. Dynamic time-indexed model for runway scheduling

The dynamic time-indexed model (DTIM) is constructed like the time-indexed formulation in Subsection 3.2 with the difference that for each aircraft $i \in F$ we may enable a different time discretization. Thus, the nearer an aircraft gets to the airport, the smaller are the slot sizes. For aircraft with more remaining flight time, it is sufficient to know the target time roughly according to the slot sizes. The same holds for outbounds, i.e. the nearer the schedule departure time is, the more accurate values are required. Thus, we may reduce the number of variables used in the optimization. This leads to smaller computational runtimes, improving the objective function value simultaneously. The time horizon T_i for each aircraft $i \in F$ is, therefore, defined as

$$T_i = \quad \{t_1, t_1 + \Delta t_i, t_1 + 2\Delta t_i, \ldots, t_m\} \quad \cap \quad \left[\tau_i^{ET}, \tau_i^{LT}\right].$$

Thus, the slot sizes Δt_i can vary for different aircraft $i \in F$.

3.4. Computational results for the different models

In order to evaluate the performance of the models we choose 10 different randomly generated instances for 50 aircraft and test them once for each model. The Table 1 reads as follows: comp. runtime denotes the computational runtime in seconds, whereas avg. dev. is an approximation of the average deviation from schedule time per aircraft in seconds and can be computed by the formula

$$\text{avg. dev.} = \sqrt{\frac{\text{obj. func. value}}{\text{number of aircraft}}}.$$

Table 1
Computational results for ten different instances of 50 aircraft.

Instances		MIP	TIM		DTIM
			$\Delta t = 5$	$\Delta t = 75$	
1	Comp. runtime [s]	375	504	<1	14
	Avg. dev. [s]	34.6	37.4	117.5	64.8
2	Comp. runtime [s]	>3600	836	<1	20
	Avg. dev. [s]	44.7	61.6	159.4	95.9
	Gap %	26.0			
3	Comp. runtime [s]	>3600	>3600	<1	23
	Avg. dev. [s]	76.2	120.8	170.3	91.7
	Gap %	60.1	17.7		
4	Comp. runtime [s]	>3600	520	<1	23
	Avg. dev. [s]	31.6	31.6	187.6	64.8
	Gap %	58.3			
5	Comp. runtime [s]	>3600	>3600	<1	19
	Avg. dev. [s]	73.5	110.5	234.1	128.8
	Gap %	57.9	50		
6	Comp. runtime [s]	1714	1971	<1	11
	Avg. dev. [s]	34.6	42.4	140	60
	Comp. runtime [s]	245	482	<1	11
7	Avg. dev. [s]	29.3	34.6	127.3	44.7
	Comp. runtime [s]	>3600	468	<1	8.3
8	Avg. dev. [s]	33.8	33.8	199.5	98.9
	Gap %	20.6			
	Comp. runtime [s]	>3600	621	<1	13
9	Avg. dev. [s]	46.9	58.3	192.9	98.9
	Gap %	34.7			
	Comp. runtime [s]	>3600	>3600	<1	11
10	Avg. dev. [s]	51.0	72.1	194.9	78.7
	Gap %	2.3	20.9		
	Comp. runtime [s]			<1	15.3
avg	Avg. dev. [s]	58.3	67.5	176.0	86.0

Comparison between MIP, TIM and DTIM with respect to runtime and average deviation from schedule time.

We thereby compare the MIP model, the time-indexed model with time discretization steps Δt of 5 and 75 s and the dynamic time-indexed model. We choose Δt_1 of 5 s for the first 10 aircraft and Δt_2 of 75 s for the remaining aircraft. The weights ω_{ST} and ω_{LT} in the objective function (1) and (3) are set to ½. Thus, the deviation from schedule time and planning behind latest time are equally important. It should be noted that from a mathematical point of view is irrelevant how these weights are chosen. Gap denotes the relative gap between the computed lower bound and the computed minimum, if we interrupt the computation after a time limit of 60 min CPU time is reached.

First of all, the MIP model computes the objective function value precisely, but unfortunately runs out of time (>1 h) for 7 of 10 instances. The TIM model with time discretization size of 5 s exceeds the time limit only in 3 cases. With increasing time discretization sizes for TIM, the computational runtime decreases. For discretization size of 75 s, the solution for every instance is computed in less than 1 s, but the average deviation from schedule increases rapidly. This arises out of the fact that the greater the time discretization size is, the less variables are computed, but the more imprecise one gets because the time is only determined within the discretization size. Another fact is that one might get losses in the criteria of average deviation when two consecutive aircraft are scheduled beyond their minimum separation times. This occurs when the minimum separation times are bigger than one slot length, but much smaller than two. To avoid these two extreme cases, the dynamic model DTIM is considered. We thereby obtain fast runtimes, which are also suitable in practice and the average deviation increases only by a factor of 1.3 compared to the cases in which all aircraft are computed with discretization size of 5 s. In comparison an overall discretization size of 75 s results in an increase by a factor of 2.6. Going into details of each instance we

ARTICLE IN PRESS

obtain that DTIM is fast (≤23 s), whereas MIP needs at least 245 s. When the number of aircraft having comparable schedule times rises, like instance 3, 5 and 10, the problem gets difficult to solve. Both MIP and TIM with discretization size of 5 s run out of time. Also the average deviation from schedule is increased in these cases. This occurs due to the fact that many schedule times are similar and thus the gap between target time and schedule time increases. TIM, respectively DTIM has another advantage. Since we use binary assignment variables, the objective function stays linear in the variables whatever we want to optimize. Thus, more complex objective functions can be considered without loss in performance. DTIM is also very flexible with respect to runtime and objective function value: it is thereby possible to compute a good solution very fast or spend more time to get an even more beneficial solution. In operational services it is required to compute solutions fast, because the position and speed of each aircraft change. Based on these results, the (dynamic) time-indexed model was chosen for robustification.

4. Robust scheduling considering uncertainties

4.1. Strict robustness approach for ATM schedules

In reality, we have to face disturbances and uncertainties in the input data that usually lead to deviations from the actual plan. From a mathematical point of view robust optimization approaches have to be considered (see Ben-Tal and Nemirovski, 1999; Bertsimas and Sim, 2004). Here, we protect the model against data uncertainties by first specifying an uncertainty set that contains all scenarios against which protection is sought. We only consider a solution, i.e., a schedule, feasible if it remains feasible no matter how the uncertainties manifest themselves within the uncertainty set. Among all these robust feasible solutions, we determine a robust optimal one, i.e., one with the best guaranteed objective function value. The following robustification focuses on TIM, but can analogously be obtained for DTIM. The strict robust counterparts are denoted by SRTIM and SRDTIM. In the robust runway scheduling problem the uncertainties lie in the deviations of earliest and latest times. For both models the uncertainty set is a discrete one, because the assignment of a slot is possible or not, i.e. the discrete variable $b_{i,j}$ either exists or the variable vanishes. Hence, the set T_i in (2) changes according to the variations in the interval of earliest and latest times. Thus, the strict robust time horizon reads as

$$T_i^R = T \cap \left[\tau_i^{ET} + prot_i, \tau_i^{LT} - prot_i\right], \tag{7}$$

where $prot_i$ denotes the deviation in the earliest and latest times for each aircraft against which we want to protect. The choices for values $prot_i$ and $buff_i$ are not necessarily related. The next section specifies the choices made here. The set T_i^R is the resulting discretized time horizon, to which aircraft i can always be assigned. Thus, for each aircraft the feasible time horizon set T_i is replaced by the strict robust time horizon set T_i^R. The robustification of the constraint (4)) and (6) then reads as

$$\sum_{j \in T_i^R} b_{i,j} = 1 \quad \forall i \in F$$

and

$$b_{i,j} + \sum_{l=j+1}^{j+\lceil \frac{\delta_{i,k}}{\Delta t} \rceil} b_{k,l} \le 1, \quad \forall i \in F, \ \ \forall j \in T_i^R \forall k \ne i,$$

Inequality (5) is maintained from the nominal time-indexed model, because each slot can still be assigned at most once. Furthermore, the minimum separation times between two aircraft might be violated, which leads to go-arounds or departure slot losses. So we protect the constraints (6) by the use of additional buffer:

$$b_{i,j} + \sum_{l=j+1}^{j+\lceil \frac{\delta_{i,k}}{\Delta t_k} \rceil + buff_{i,k}} b_{k,l} \le 1, \ \ \forall i \in F, \ \ \forall j \in T_i^R, \ \ \forall k \ne i.$$

where buffer $buff_{i,k}$ can for example be computed by the knowledge of the expected delay/earliness of aircraft i and k. We will detail this in the next section.

4.2. Light robustness model for runway scheduling problem

Clearly, depending on the size of the additional security buffers, the strict robustification from Section 4.1 might reduce the throughput and increase the delay considerably. This has also been observed in other contexts, for example by Ben-Tal and Nemirovski (1999) or Bertsimas and Sim (2004). In this subsection, we introduce a less conservative robustness concept that has been developed for timetabling in railways. Using the concept of light robustness (Fischetti and Monaci, 2009; Schöbel, 2014), the price of robustness, i.e., the reduction in throughput and increase in delay, can be controlled.

The goal of light robustness is to achieve a trade-off between the stability of a solution under disturbances and the price of robustness. First, the optimization model is solved by ignoring uncertainties. In a second step, the optimization model is modified and solved again. In this light robust optimization problem, we allow an increase in the costs by a certain percentage κ that is specified beforehand. We now aim at determining a solution that maximizes robustness among all solutions that do not increase the cost by more than κ percent.

As in Subsection 4.1 we focus on the time discretized method and apply the idea of light robustness here. The light robust time-indexed counterpart is denoted by LRTIM. As described above, we solve the nominal problem first (see Subsection 3.2) and denote the nominal optimal solution as z_{NOM}^*. Afterwards, we have to solve a second optimization problem. This problem contains again the constraints (4), (5) and (6). Additionally, to fix the quality standard of the nominal solution, we add the constraint

$$\sum_{i \in F} \sum_{j \in T_i} b_{ij} \left(\omega_{ST} \left(t_j - \tau_i^{ST} \right)^2 + \omega_{LT} \left(t_i^{LT+} \right)^2 \right) \le (1+\kappa) \ \ z_{NOM}^*, \tag{8}$$

which allows an increase in the nominal costs of κ with $\kappa \geq 0$. The left hand side of this constraint (8) is in fact the objective function (3) of the nominal problem. Since (3) computes an optimal nominal solution, we now look for solutions, which may be at most $\kappa \cdot 100\%$ worse than the nominal one z_{NOM}^*. Among these solutions, we want to find one that maximizes robustness. Thus, we have to consider a new objective function, which maximizes robustness. Since uncertainty effects earliest and latest times, planning very near to the earliest or latest times usually does not lead to stable schedules. In contrast, scheduling all aircraft into its strict robust time horizon $\mathcal{T}_i^R$ is most robust, because it contains all time slots that are possible in each considered realization of uncertainties.

For example, let the nominal objective function value z_{NOM}^* be 5000, which means for a scenario with 50 aircraft an average delay

Please cite this article in press as: Heidt, A., et al., Robust runway scheduling under uncertain conditions, Journal of Air Transport Management (2016), http://dx.doi.org/10.1016/j.jairtraman.2016.02.009

of 10 s per aircraft. The feasible time horizon of an aircraft i is $\mathscr{T}_i = \{100, 125, 150, 175, 200\}$ with its schedule time settled at 110. If we want to protect against deviations in earliest and latest times of 25 s, the strict robust time horizon of aircraft i reads as $T_i^R = \{125, 150, 175\}$. The nominal solution for this aircraft is then the slot 100 and the strict robust solution is 125. If we choose κ=0.1, we may allow an objective function value of 5500. In the light robust approach aircraft i will be computed into its strict time horizon on slot 125. The left hand side of (8) will increase to 5125, which is smaller than 5500 and thus feasible. But this approach does not allow to assign aircraft i on slot 150, because it violates constraint (8). It is also obvious that depending on the value of κ, possibly not all 50 aircraft can be planned into their strict robust time horizon.

In summary, the light robust objective function minimizes the assignment of aircraft to slots that lie outside of their strict robust time horizon, i.e., that will become infeasible in some scenario. Every assignment within the strict time horizon is considered robust and is thus not penalized in the objective. The objective counts only the delay costs of those assignments which do not lie in $T_i \backslash T_i^R$. Of course, planning an aircraft into its strict robust time slots might increase its delay which however is restricted by the value of parameter κ in (8).

Thus, the light robust objective is given by:

$$\min \sum_{i \in F} \sum_{j \in T_i \backslash T_i^R} b_{i,j} \left(t_j - \tau_i^{ST}\right)^2. \tag{9}$$

5. Validation setup

Up to now, we have only considered the results of the static case. We are, however, interested in the algorithmic behavior when input data are disturbed. Therefore we use random initial data for each aircraft and for the uncertainties in the earliest and latest times. The different robust models are compared to the nominal algorithm. This is done within a simulation for a planning horizon up to two hours before touch-down/take-off. In the simulation we will additionally use the quadratic deviation from the last schedule τ_i^{LS} in the objective function weighted equally to the other both terms. Hence, we avoid jumps from one simulation step to another, if e.g. two different solutions with the same objective function value exist. Thus, the sequence does not change until an improvement in the objective function value is achieved. To be precise, we only adapt a solution if the previous (best) solution is not feasible anymore due to disturbed input data. We compare four different planners ($M = 4$ in Fig. 2) in the simulation (see Table 2):

The TIM model with discretization size of 75 s works as a nominal planner without considering robustness (III-B). Its strict robust counterpart (SRTIM) with discretization size of 75 s denotes robust planner 1. As strict robust planner 2 we choose the strict robust DTIM model (SRDTIM) and the light robust planner is LRTIM. The omniscient planner knows all uncertainties beforehand and computes the best sequence with the same discretization size of the robust planner but without robustification. Thereby it is possible to compare the results of both planners with the optimal values of a planner who already knows all deviations in advance.

We will first compare different discretization sizes for DTIM to evaluate which one fits best for planner 2B. Also we vary the quality standard parameter κ to find out the best configuration of the light robust planner.

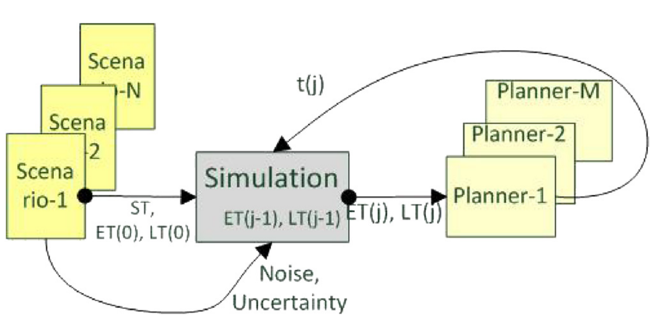

Fig. 2. Simulation and planning (Heidt et al., 2014).

Algorithm 1: High-level description of the simulation procedure.

Data: earliest, latest, schedule times for all aircraft
Result: target times for all aircraft, statistics of criteria values
calculate initial sequence;
choose planner;
while *not all aircraft landed/took-off* **do**
  increase simulation time;
  disturb earliest and latest times;
  for *all chosen planners* **do**
    decrease window of earliest and latest times;
    test for go-arounds, departure slot losses;
    serve landed/departed aircraft;
    if *any target time drops out of its earliest latest time interval* **then**
      optimize the model (update sequence);
    else
      keep last sequence;
    end
    get computed target times;
    update statistics;
  end
end

Algorithm 1 gives an overview over each step of the simulation. We will detail this in the following. Fig. 2 shows that each planning takes effect on the simulation and thus on the input data for the next simulation step. Fig. 2 also shows that each simulation step consists of two parts. On the one hand, it contains the simulation of the operator who tries to implement the plan. Therefore, the earliest and latest times converge towards the planned target times, because this interval shrinks the nearer an aircraft gets to its planned time. On the other hand, it generates the disturbances. Here, we add a random value to earliest (ET) and latest times (LT). We update every three minutes ET and LT and therefore replan also every three minutes if the previous sequence is not feasible anymore, i.e., the simulation step size $\Delta\tau^{Sim}$ is always 180 s. The increment of the earliest time $inc_i^{ET,j+1}$ of aircraft a_i in simulation

Table 2
The different considered planners.

#	Planner	Model
1	Nominal	TIM
2A	Strict robust planner 1	SRTIM
2B	Strict robust planner 2	SRDTIM
3	Light robust	LRTIM

Please cite this article in press as: Heidt, A., et al., Robust runway scheduling under uncertain conditions, Journal of Air Transport Management (2016), http://dx.doi.org/10.1016/j.jairtraman.2016.02.009

step $j+1$ converges linearly to the planned target time $\tau_i^{TT,j}$ of the previous simulation step j by the following formula:

$$\mathrm{inc}_i^{ET,j+1} = c \cdot \Delta\tau^{Sim} \cdot \frac{\tau_i^{TT,j} - \tau_i^{ET,j}}{\tau_i^{TT,j} - \tau^{Sim}}$$

The convergence factor c depends on the difference of the planned target time $\tau_i^{TT,j}$ and the simulation time τ^{Sim}. It varies from 0.2 (more than 20 min before planned touch-down/take-off time) to 1.0 (3 min before planned touch-down/take-off time). We randomly choose the disturbance dist_i^{j+1} of aircraft a_i in simulation step $j+1$ from a normal distribution

$$\mathscr{N}\left(n_i^{j+1}, \left(\chi_i^{j+1} \cdot \sigma_i\right)^2\right). \quad (10)$$

$n_i^0 = n_i^1 = \mu_i$ is determined by the used scenario (low or high uncertainty). For $j>1$ the mean values n_i^{j+1} are calculated from the previous disturbances:

$$n_i^{j+1} = \frac{2}{3} \cdot \mathrm{dist}_i^j + \frac{1}{3} \cdot \mathrm{dist}_i^{j-1}$$

Thus, the mean value and the standard deviation of the normal distribution changes for each aircraft and each simulation step. Factor χ_i is set to 1.0 for outbounds and for inbounds it ensures that the standard deviation decreases the nearer an aircraft gets to its target touch-down time and thus the uncertainty reduces:

$$\chi_i^{j+1} = \frac{1800 + \tau_i^{ET,j} - \tau^{Sim}}{3600}$$

In each simulation step j we calculate the convergence of the earliest time to the planned target time $\mathrm{inc}_i^{ET,j}$ and the disturbance dist_i^j.

$$\tau_i^{ET,j} = \begin{cases} \tau_i^{ET,j-1} + \mathrm{inc}_i^{ET,j} + \mathrm{dist}_i^j & \text{if } \mathrm{dist}_i^j < 0 \\ \tau_i^{ET,j-1} + \max\left(\mathrm{inc}_i^{ET,j}, \mathrm{dist}_i^j\right) & \text{if } \mathrm{dist}_i^j \geq 0 \end{cases}$$

The latest times similarly converge towards the planned times. Similar randomly chosen disturbance dist_i^{j+1} are added.

After updating the earliest and latest times we decide whether an update of the sequence, denoted by the previous target times, is necessary (see Fig. 3). This is necessary if the inequality $\tau_i^{ET,j+1} \leq \tau_i^{TT,j} \leq \tau_i^{LT,j+1}$ is not satisfied any more, i.e., in case the planned target time is outside the earliest/latest time interval.

6. Computational results for the different planners

We use three different scenarios (N = 3 in Fig. 2):

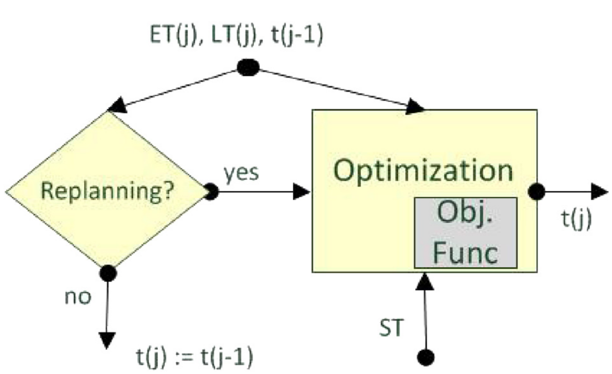

Fig. 3. Replanning decision and optimization (Heidt et al., 2014).

1) high traffic demand with low uncertainty
2) high traffic demand with high uncertainty
3) medium traffic demand with high uncertainty

The case of medium traffic demand with low uncertainty is not considered because this one is very stable. Each scenario (low or high uncertainty, medium or high traffic) contains 20 runs with 50 different randomly chosen aircraft. Schedule times are randomly chosen, so that each five minutes slot (for example 12:00, 12:05, 12:10, …) except potentially the last one contains the same number of aircraft S. For medium traffic $S=3$ is chosen and for high traffic scenarios we use $S=5$. The earliest time of each aircraft is randomly chosen out of a range of 5–10 min before its schedule time. Weight classes as well as operation types (inbound or outbound) are randomly chosen. We protect the earliest and latest times against deviations, which are smaller than the sum of mean value and l times standard deviation:

$$prot_i = \mu_i + l \cdot \sigma_i \quad \forall i \in F, l \in \mathbb{Z} \quad \text{and}$$
$$\mathrm{buff}_{i,k} = max(0, prot_i) + max(0, -prot_k) \quad \forall i, k \in F.$$

Thus, a buffer is only installed, when the predecessor is supposed to be later or the follower earlier. For low uncertainty we choose for all aircraft as mean value $\mu_i = 2\,s$ and standard deviation $\sigma_i = 2s$, and at high uncertainty the values are $\mu_i = 10s$ for all aircraft, $\sigma_i = 4s$ for arrivals and $\sigma_i = 6s$ for departures. Recall that these values are not the predicted delay, which is computed with formula (10). $\mu_i = 10$s and $\sigma_i = 4$s means that in each simulation time step of size $\Delta\tau^{Sim}$ we have to expect an additional delay of 10 s. We choose $l = 3$ in all computations. For an aircraft with a remaining flight time of 3600 s and with $\Delta\tau^{Sim} = 180$ s we expect an average delay of 200 s (10•3600/180) with a standard deviation of 120 s (1800+3600/3600•4•3600/180).

This comparison is done with respect to the following average criteria values:

***GoAround*:** number of go-arounds in each simulation; A go-around with an increase of the earliest time by 15 min is assigned to an aircraft if the simulated disturbances result in a separation loss to the predecessor. Then the latest time of an aircraft violates the minimum separation of the predecessor inbound resp. outbound.

***Dep. loss*:** number of departure slot losses in each simulation; A departure slot loss with an increase of the earliest time by 60 s is assigned to an outbound if the simulated disturbances result in a separation loss to its predecessor or immediate successor inbound.

***Makespan [s]*:** difference in seconds between the target times, except the first two and the last two of the resulting final simulation sequence; we thereby avoid effects on the makespan due to go-around or departure slot losses (+15 min) of the first or last aircraft.

***Changed Pos/SimStep*:** number of position changes per simulation step $\Delta\tau^{Sim}$.

***Changed TT/acft [min]*:** sum of absolute target time changes per aircraft in minutes.

avg. dev. *[s]:* First we calculate the objective function of the final sequence and subtract the objective function value of the omniscient planner. We only consider the absolute quadratic deviation of the target time and the schedule time. We divide this difference by the number of aircraft and calculate the square root.

***Comp. runtime [s]*:** computational runtime of the chosen algorithm for all optimization steps without simulation time itself.

The test runs are executed on a Windows-7-Notebook with Intel Core i5-2410M processor with 4 kernels, 2.3 GHz and 4GB RAM. The code is written in C++. As a commercial solver for mixed integer optimization problems we use GUROBI (Gurobi Optimization and Inc, 2014), version 5.6.

ARTICLE IN PRESS

6.1. Tests for different discretization sizes of SRDTIM

We compare three different configurations of discretization sizes for SRDTIM. The first configuration takes a discretization size of 5s for the last 10 min before touch-down/take-off and discretization size of 75s otherwise. The second one chooses 5s as discretization size for the last 10 min before touch-down/take-off, 25s for target times between 10 and 20 min before planned touch-down/take-off and 75s otherwise. The third has a discretization size of 25s for the last 10 min and 75s otherwise. Thereby we use the scenario of high traffic and high uncertainty for 20 runs with 50 different randomly chosen aircraft. All three planners compute no go-arounds and no departure slot losses, see Table 3. The changed position and target times are slightly higher for planner 1, because by decreasing the discretization size, few more changes are possible and the aircraft can be scheduled closer to each other. We will focus on planner 2 for the computational study, because it yields better solutions with respect to makespan and average deviation than planner 3 and is more robust (changed Pos/TT) than planner 1.

6.2. Test for the quality standard parameter κ in LRTIM

We compare different quality standard values κ for LRTIM, namely 10%, 30% and 50%. In eq. (8) κ denotes the allowed deviation from the nominal objective function value. Thereby we use the scenario of medium traffic and high uncertainty for 20 runs with 50 different randomly chosen aircraft. In Table 4, we observe for the parameter $\kappa=0.1$ the best makespan and the best average deviation, because the solution is close to the nominal one. But for $\kappa=0.5$, there is no departure loss necessary. The advantage lies in a low number of position changes as well as target time changes. Hence, we choose for robustness purposes $\kappa=0.5$ for the computational study. Analogous results are obtained choosing one of the other scenarios.

6.3. Scenario 1: high traffic, low uncertainty

In this first scenario, Table 5 shows the comparison between the strict robust (SRTIM), the nominal time-discretized with no robustness consideration (TIM), the strict robust dynamic time-indexed (SRDTIM) and the light robust time-indexed model (LRTIM). Comparing the strict robust time-indexed and the nominal time-indexed model, the criteria go-around and departure slot losses are profitable in the robust case. By the robustification we achieve that no go-arounds and no departure slot losses are obtained. Usually, the price of robustness would increase the makespan. However, the contrary is true here. The reason is that because of less go-arounds and departure slot losses the robust planner reveals seven seconds less makespan than the nominal one. The robust results show that one protects not only against undesired flight maneuvers such as go-arounds, but also achieves stable plans. In the simulation we observe 0.05 position changes per simulation step, i.e., about 1.5 position changes per simulation on average. In comparison, the nominal planner needs to change positions more often. Considering the changed target times the robust planner with 0.11 min per aircraft on average is more stable than its nominal counterpart, which has an average change of 2.33 min. But the average deviation from schedule time is 46 s higher than in the nominal case. This is due to the price of robustness, which we have to pay at this low level of uncertainty. We immunize against the predefined uncertainty set, but only few deviations from schedule time happen. The computational runtime is the same. This shows that the strict robust time-indexed model is not harder to solve than the nominal one.

To accomplish a better makespan one can use the light robust time-indexed model. It computes a slightly better makespan by still having a low number of changes in position and target time, but does not prohibit the high value for average deviation. Compared to the strict robust dynamic time-indexed model, the makespan decreases. The reason is the smaller discretization sizes near take-off/touch-down that reduce the average deviation value, when compared to the strict robust planner.

We also performed runs with a higher robustification of 5σ ($l=5$) and thus computed a plan with no changes in target time and no position changes. As expected, the makespan raised to 3641 s and the average deviation value is with 552 s very high. This demonstrates that our approach also enables extremely stable plans, but we have to pay a prize for it as well, i.e. aircraft delay increases.

6.4. Scenario 2: high traffic, high uncertainty

In the second scenario the uncertainty is larger. First of all, we look at the strict robust and the nominal planner in Table 6. It shows that go-arounds and departure slot losses are still not needed in the robust case. The makespan is the same in both cases. Thus, the price of robustness in makespan is compensated, which is comparable to

Table 3
Average results for different SRDTIM planners for 20 instances of 50 aircraft.

Criteria	SRDTIM1	SRDTIM2	SRDTIM3
GoAround	0	0	0
Dep. loss	0	0	0
Makespan [s]	2795	2795	2816
Changed Pos/SimStep	0.27	0.21	0.18
Changed TT/acft [min]	0.50	0.39	0.36
Avg. dev. [s]	468.1	466.8	475.7
Comp. runtime [s]	17.8	45.3	9.3

Comparison between different SRDTIM planners with respect to the criteria introduced in Section 6.

Table 4
Average results for different LRTIM planners for 20 instances of 50 aircraft.

Criteria	$\kappa=0.1$	$\kappa=0.3$	$\kappa=0.5$
GoAround	0	0	0
Dep. loss	0.2	0.1	0
Makespan [s]	4186	4210	4211
Changed Pos/SimStep	0.14	0.13	0.09
Changed TT/acft [min]	0.69	0.90	0.62
Avg. dev. [s]	89.3	92.9	93.8
Comp. runtime [s]	14.3	14.3	15.9

Comparison between three different quality standard values κ with respect to the criteria introduced in Section 6.

Table 5
Average results of high traffic, low uncertainty scenario for 20 instances of 50 aircraft.

Criteria	SRTIM	TIM	SRDTIM2	LRTIM
GoAround	0	1.8	0	0
Dep. loss	0	0.4	0	0
Makespan [s]	2832	2839	2818	2820
Changed Pos/SimStep	0.05	0.58	0.09	0.05
Changed TT/acft [min]	0.11	2.33	0.16	0.08
Avg. dev. [s]	474.5	428.1	464.1	475.3
Comp. runtime [s]	13.6	13.1	32.9	47.1

Comparison between SRTIM, TIM, SRDTIM2 and LRTIM planners with respect to the criteria introduced in Section 6.

Please cite this article in press as: Heidt, A., et al., Robust runway scheduling under uncertain conditions, Journal of Air Transport Management (2016), http://dx.doi.org/10.1016/j.jairtraman.2016.02.009

ARTICLE IN PRESS

the first scenario. The number of changed positions reduces drastically in the robust case, which is also true for the changes in target time. Considering the criteria of objective function value, the strict robust and the nominal planner differ by about 30 s. The reason for this is that a buffer between two aircraft and a protection against earliest time deviations is incorporated. In contrast, more go-arounds and departure slot losses are needed in the nominal case. Again the computational runtime is practicable.

To avoid the mismatch in average deviation, one can choose the light robustness planner, which computes almost the same average deviation level as the nominal one. In fact, it obtains a low number of changes and a better makespan, but needs more computational runtime (90.1 s).

The strict robust dynamic time-indexed model computes the best makespan and the best average deviation, whereas the number of changes in position and target time is only slightly bigger than of the strict robust one.

Next, we analyze how the strict robust planner changes in terms of increased uncertainty at constant high traffic (Table 7). We have to mention that the buffer between all aircraft also increases by the same amount. Thus, the number of go-arounds and departure slot losses is kept at zero.

The criteria changed position and changed target time rise by a factor of about 3–4 because of the larger uncertainties. These values, however, are still reduced, when compared to the nominal planner. Considering the computational runtime, higher uncertainty leads to smaller time windows for each aircraft by constant high traffic, which makes it more difficult to find a feasible solution.

6.5. Scenario 3: medium traffic, high uncertainty

Finally, we study the behavior of the planner when reducing the traffic to a medium level (see Table 8). Due to high uncertainty the buffer is still high and thus protects against go-arounds and departure slot losses in contrast to the nominal planner. Due to medium traffic only, the makespan of the robust and the nominal planner is almost the same. Here, in contrast to high traffic, there is a natural buffer in the scenario itself. Comparable to both scenarios before, the robust planner “wins” the criteria changed position and changed target time. The average deviation reduces, because go-arounds and departure slot losses affect the nominal planner more when compared to high traffic, because fewer other aircraft can reduce the average deviation value. The runtime is again viable. Again the light robust planner succeeds in the measurement average deviation.

The behavior of the strict robust dynamic and the light robust time-indexed model is similar to scenario 2. The light robust time-indexed model is fast in handling medium traffic. Concerning the strict robust dynamic time-indexed model, one difference is recognized in computational runtime. Here, the strict robust dynamic time-indexed model needs more computational time, because it still has to manage small discretization sizes.

How does the robust planner react for reduced traffic? Table 9 shows that there is a significant increase in makespan value. This is due to the fact that at medium traffic three instead of five aircraft are assigned to the same schedule time. By always considering 50 aircraft, the makespan increases automatically, because the deviation from schedule time is optimized. The reduction in changed position stems from the fact that changes at medium traffic are easier to control and do not have as large effects on other aircraft, when compared to high traffic. The objective function value increases rapidly, because at high traffic each aircraft has a higher impact on the others than at medium traffic. Finally, the computational runtime reduces because less aircraft are in conflict state of being scheduled at the same time.

7. Conclusion

The goal of this work was to study runway scheduling when explicit knowledge of aircraft uncertainty is available. To this end, we used a time-indexed optimization model for the mixed-mode runway scheduling problem that is able to cope with uncertainties in the input data. Using this model, we set-up a

Table 6
Average results of high traffic, high uncertainty scenario for 20 instances of 50 aircraft.

Criteria	SRTIM	TIM	SRDTIM2	LRTIM
GoAround	0	2.4	0	0
Dep. loss	0	0.9	0	0
Makespan [s]	2861	2861	2795	2833
Changed Pos/SimStep	0.13	3.13	0.21	0.20
Changed TT/acft [min]	0.38	8.56	0.39	0.42
Avg. dev. [s]	517.1	484.6	466.8	487.9
Comp. runtime [s]	20.1	26.4	45.3	90.1

Comparison between SRTIM, TIM, SRDTIM2 and LRTIM planners with respect to the criteria introduced in Section 6.

Table 7
Average results for low and high uncertainty at high traffic for 20 instances of 50 aircraft.

Criteria	LowU	HighU	HighU − lowU
GoAround	0	0	0
Dep. loss	0	0	0
Makespan [s]	2832	2861	29
			HighU/lowU
Changed Pos/SimStep	0.05	0.13	2.6
Changed TT/acft [min]	0.11	0.42	3.8
Avg. dev. [s]	474.5	517.1	1.09
Comp. runtime [s]	13.6	20.1	1.48

Comparison of SRTIM planner at different uncertainty levels with respect to the criteria introduced in Section 6.

Table 8
Medium traffic, high uncertainty scenario for 20 instances of 50 aircraft.

Criteria	SRTIM	TIM	SRDTIM2	LRTIM
GoAround	0	1.1	0	0
Dep. loss	0	0.2	0	0
Makespan [s]	4232	4241	4194	4210
Changed Pos/SimStep	0.06	0.62	0.15	0.09
Changed TT/acft [min]	0.47	4.86	0.75	0.62
Avg. dev. [s]	99.6	157.2	90.2	93.8
Comp. runtime [s]	14.3	26.9	34.2	14.1

Comparison between SRTIM, TIM, SRDTIM2 and LRTIM planners with respect to the criteria introduced in Section 6.

Table 9
Average results for medhigh traffic at high uncertainty for 20 instances of 50 aircraft.

Criteria	MedTr	HighTr	HighTr − medTr
GoAround	0	0	0
Dep. loss	0	0	0
Makespan [s]	4232	2861	−1371
			HighTr/medTr
Changed Pos/SimStep	0.06	0.13	2.2
Changed TT/acft [min]	0.47	0.42	0.90
Avg. dev. [s]	99.6	517.1	5.19
Comp. runtime [s]	14.3	20.1	1.41

Comparison of SRTIM planner at different traffic level with respect to the criteria introduced in Section 6.

Please cite this article in press as: Heidt, A., et al., Robust runway scheduling under uncertain conditions, Journal of Air Transport Management (2016), http://dx.doi.org/10.1016/j.jairtraman.2016.02.009

simulation approach in which we determined optimum schedules in each time step. Especially, we studied the question whether the robust approach leads to more stable plans with fewer go-arounds and departure slot losses.

We evaluated our method on three different and relevant scenarios. In more detail, we considered the scenarios of high traffic and low uncertainty, high traffic and high uncertainty, medium traffic and high uncertainty. We compared our approach with the corresponding results for the nominal approach that does not take uncertainties into account.

The computational results show that the strict robust model indeed computes more stable sequences: the number of sequence changes is at least decreased by a factor of 10! Depending on the needed protection against replanning we could reduce the number of go-arounds and departure slot losses to zero. Usually, one expects that a high level of protection against uncertainty leads to lower throughput, i.e. one has to pay a certain price of robustness. However, in our tests the contrary was the case. Due to the fact that the robust planner significantly reduces the number of go-arounds and departure slot losses, the makespan value is comparable to the nominal case. The average deviation from schedule, however, decreases when robustness is considered. We investigated a light robust time-indexed model. We observed an improvement with respect to the strict robust time-indexed model in makespan and average deviation. This behavior is also shown by the strict robust dynamic time-indexed model. Concluding the computational results of our study, we would recommend the light robust planner at medium traffic and high uncertainty and the strict robust dynamic during high traffic situations with high uncertainty. In case of high traffic and low uncertainty there is a trade-off between the light robust and the strict robust dynamic planner. The user has to set the priority between number of position changes and changes in the target time on the one hand and computational runtime as well as average deviation from schedule on the other hand.

We have chosen the deviation from schedule as one objective function example, but other, also more complex ones, are usable. Since we use assignment variables in our models the objective function stays linear in the variables, no matter what objective the user wants to optimize. Even the light robust time-indexed model only needs runtime of 10 s for one optimization step.

We also showed that our approach enables stable sequences without replanning. The prize potentially, but not necessarily, is a lower objective function value. However, by tuning the parameter l the user can decide the needed trade-off between sequence stability (robustness) on the one hand and efficiency (objective function value) on the other hand. We can benefit from our knowledge of aircraft uncertainty already when aircraft sequences are calculated. Modeling techniques together with efficient algorithms are available.

Acknowledgment

This work is co-financed by EUROCONTROL acting on behalf of the SESAR Joint Undertaking (the SJU) and the EUROPEAN UNION as part of ComplexWorld Research Network in the SESAR Programme.

References

Agogino, A., 2011. Robustness of two air traffic scheduling approaches to departure uncertainty. In: 30th Digital Avionics Systems Conference (DASC), Seattle.
Ahuja, R., Magnanti, T., Orlin, J., 1993. Network Flows: Theory, Algorithms, and Applications. Prentice Hall Inc, Upper Saddle River, New Jersey.
Anagnostakis, I., Clarke, J.P., 2003. Runway operations planning: a two-stage solution methodology. In: Proceedings of the 36th Hawaii International Conference on System Sciences.
Atkin, J., Burke, E., Greenwood, J., Reeson, D., 2007. Hybrid metaheuristics to aid runway scheduling at London Heathrow Airport. Transp. Sci. 41 (1), 90–106.
Balakrishnan, H., Chandran, B., 2006. Scheduling aircraft landings under constrained position shifting. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado.
Beasley, J., Krishnamoorthy, M., Sharaiha, Y., Acramson, D., 2000. Scheduling aircraft landings - the static case. Transp. Sci. 34 (2), 180–197.
Bennell, J., Mesgarpour, M., Potts, C., 2013. Airport runway scheduling. Ann. Oper. Res. 204 (1), 249–270.
Ben-Tal, A., Nemirovski, A., 1999. Robust solutions of uncertaint linear programs. Oper. Res. Lett. 25, 1–13.
Bertsimas, D., Sim, M., 2004. The price of robustness. Oper. Res. 52 (1), 35–53.
Böhme, D., 2005. Tactical departure management with the Eurocontrol/DLR DMAN. In: 6th EUROCONTROL/FAA ATM R& D Seminar, Baltimore, MD, USA.
Chandran, B., Balakrishnan, H., 2007. A Dynamic Programming Algorithm for Robust Runway. American Control Conference, New York City.
Clare, G., Richards, A., 2011. Optimization of taxiway routing and runway scheduling. IEEE Trans. Intell. Transp. Syst. 12 (4), 1000–1013.
Corolli, L., Lulli, G., Ntaimo, L., 2014. The time slot allocation problem under uncertain capacity. Transp. Res. Part C Emerg. Technol. 46, 16–29.
Dyer, M., Wolsey, L., 1990. Formulating the single machine sequencing problem with release dates as a mixed-integer program. Discrete Appl. Math. 26 (2–3), 255–270.
Fischetti, M., Monaci, M., 2009. Light robustness. In: Ahuja, R.K., et al. (Eds.), Robust and Online Large Scale Optimization. Springer.
Frankovich, M., Bertsimas, D., 2013. Air traffic Flow management at airports: a unified optimization approach. In: ATMSeminar.
Gurobi Optimization, Inc, 2014. Gurobi Optimizer Reference Manual.
Heidt, A., Helmke, H., Liers, F., Martin, A., 2014. Robust runway scheduling using a time-indexed model. In: Schaefer, Dirk (Ed.), Proceedings of the SESAR Innovation Days (2014) EUROCONTROL. ISBN 978-2-87497-077-1.
Helmke, H. Scheduling algorithms for ATM applications tools and Toys, 30th Digital Avionics Systems Conference, Seattle, October 16-20, 2011.
Hu, X.-B., Chen, W.-H., 2005. Receding horizon control for aircraft arrival sequencing and scheduling. IEEE Trans. Intell. Transp. Syst. 6 (2), 189–197.
Hu, X.-B., Di-Paolo, E., 2008. Binary-representation-based genetic algorithm for aircraft arrival sequencing and scheduling. IEEE Trans. Intell. Transp. Syst. 9 (2), 301–310.
Jacquillat, A., Odoni, A., 2014. An Integrated Scheduling and Operations Approach to Airport Congestion Mitigation, Operations Research, to Appear.
Kapolke, M., Fürstenau, N., Heidt, A., Liers, F., Mittendorf, M., Weiß, C., 2016. Pre-tactical optimization of runway utilization under uncertainty. J. Air Transp. Manag. http://dx.doi.org/10.1016/j.jairtraman.2016.02.004.
Kjenstad, D., Mannino, C., Nordlander, T.E., Schittekat, P., Smedsrud, M., 2013. Optimizing AMAN-SMAN-DMAN at Hamburg and Arlanda airport. In: Proceedings of the SID, Stockholm.
Neuman, F., Erzberger, H., 1990. Analysis of sequencing and scheduling methods for arrival traffic. In: NASA Technical Memorandum, Moffett Field, California.
Samà, M., D'Ariano, A., D'Ariano, P., Pacciarelli, D., 2014. Optimal aircraft scheduling and routing at a terminal control area during disturbances. Transp. Res. Part C Emerg. Technol. 47 (P1), 61–85.
Schöbel, A., 2014. Generalized light robustness and the trade-off between robustness and nominal quality. Math. Methods Oper. Res. 80, 161–191.
Solveling, G., 2012. Stochastic Programming Methods for Scheduling of Airport Runway Operations under Uncertainty. Dissertation. George Institute of Technology.
Völkers, U., 1990. Arrival planning and sequencing with COMPAS-OP at the Frankfurt ATC-Center. In: The 1990 American Control Conference, San Diego, California.