Robust Runway Scheduling Using a Time-indexed Model

Andreas Heidt[1], Hartmut Helmke[2], Frauke Liers[1] and Alexander Martin[1]

[1]Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Department of Mathematics, Discrete Optimization
Cauerstr. 11, 91058 Erlangen, Germany
Email: name.surname@fau.de

[2]German Aerospace Center (DLR)
Institute of Flight Guidance
Lilienthalplatz 7, 38108 Braunschweig, Germany
Email: Hartmut.Helmke@dlr.de

***Abstract*—The runway system is the main element that combines airside and groundside of the ATM System. Efficient models and planning algorithms are required. 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 is disturbed respectively it changes. For example, an estimated time of an aircraft is not stable for the next ten hours. These disturbances are not deterministic, but often their stochastic distributions with mean values and standard deviations are known.**

We present a robust model together with an optimization algorithm which explicitly incorporates the knowledge of uncertainty into each planning step. Our approach transforms the planning problem into an assignment problem with side constraints. We compute for every discretized point in time whether an aircraft is scheduled and if so, which one is. Experimentally, we show that runtimes are better than when using a different non-linear integer optimization model. Our Monte-Carlo simulation for a mixed-mode runway system shows that our approach results in fewer sequence changes and target time updates compared to the usual approach of just updating the plan if the actual plan is not feasible any more.

***Keywords*—scheduling, uncertainty, time-indexed model, MIP, mixed-integer programming, dynamic time-indexed model, mixed mode runway scheduling**

I. 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 a deviation from the actual plan or schedule. 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). These adjustments are performed after the actual change in the data occurred in *hindsight*, leading to schedules with reduced overall utilization. The goal of robust plans is to reduce rescheduling. The challenge is to incorporate uncertainty into the initial computation of the plans so that these plans are *robust* with respect to changes in the data leading to a better utilization of resources and a more efficient support for ATM controllers and stakeholders. However, the price of robustness has to be paid by a reduction of efficiency in general.

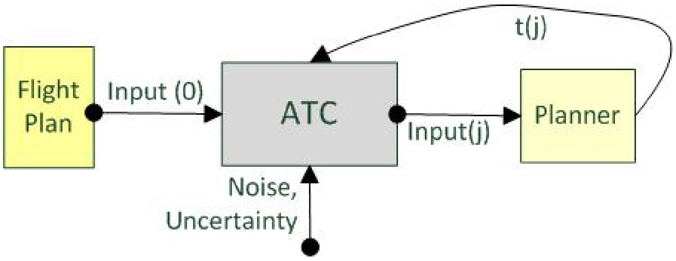

Fig. 1: The ATC planing and feedback cycle

In *normal life* we intuitively act in the same way. As a host of an invitation for 8 p.m. we know that there are some guest *always* arriving not before 8:30, but others will arrive in time (*German sharp* or even Swiss sharp). Depending on the invited guests, we start our preparation very early or we know that we still have time.

Two different mathematical approaches currently exist to handle uncertainty in optimization: *stochastic optimization* that can be used to compute good average solutions and *robust optimization* to immunize against predefined worst-case scenarios with a desired probability. In the stochastic case our host tries to find a good compromise between his waiting time for the first guests and the waiting 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 he has expected.

Section II presents related work. In the following section III we model the scheduling problem as a mixed-integer program (MIP) and an integer (binary) program (IP) without explicit consideration of knowledge about uncertainties and compared both approaches. In section IV uncertainties are taken into account and a robust model (not a stochastic model) is used. Results are compared with those of a non-linear mixed-integer model. As the time-discretized model has several advantages, it is the one that is used in the simulation. Afterwards the performed experiments are described in section V and their results are given in section VI.

II. Related Work

The development of Arrival Managers start in the early eighties ([19], [21]) and of Departure Managers ([9]) in the nineties of the last century to assist air traffic controller.

Heuristic approaches were used to compute the sequencing of the air traffic. Based on a first-come-first-served (FCFS) principle, constrained position shifting (CPS) was used by Balakrishnan and Chandran ([5]). Here, only a limited number of aircraft can change their position. Hu and Chen ([16]) use position shifting to determine an arrival sequencing. Hu and Di-Paolo ([17]) furthermore propose a genetic algorithm. Moreover metaheuristics, such as TabuSearch, have also been studied for the runway scheduling at London Heathrow Airport by Atkin et al. ([4]).

Beasley et al. [6] formulated a mixed-integer-program (MIP), where the variables are the exact target times of the aircraft. They used a linear cost function and developped heuristics and LP-based tree search. Clare and Richards ([11]) use the same mixed-integer formulation and solved it by receding horizon methods. Considering a time-indexed formulation for runway scheduling, Anagnostakis and Clarke [3] declared a heuristic approach for the planning of departures. Thereby a two-stage formulation is proposed. In the first stage, weight classes are assigned to slots, whereby in the second stage the specific aircraft are scheduled. In [18], the time-indexed formulation was used for Hamburg and Arlanda airport. They obtain short computational runtime ($<$ 20s) for Hamburg Airport. In 2013, Frankovich and Bertsimas [13] 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. Nevertheless, according to our knowledge, there was no literature which tackles dynamic time-indexed formulation for the runway scheduling problem, which varies the size of time slots for every aircraft depending on the remaining flight time.

Considering robustness, [10] developed a heuristic approach using FCFS with CPS for stochastic deviations in the earliest times. Agogino [1] 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 [20] a MIP model is solved by a 2-stage-stochastic approach using a stochastic branch & bound method. Apart from that, mathematically robust approaches have hardly been considered to solve the runway scheduling problem to optimality.

III. Modelling the Scheduling Problem with Mixed-Integer Programming

In this section we model the runway scheduling problem as different mixed-integer programs. In the moment we do not consider uncertainty and we only plan once. 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. We focus on the benefits of the explicit modeling of the knowledge of uncertainty. Therefore, our simple objective function tries to meet (for each aircraft) the given scheduled time τ_i^{ST} at the runway. Furthermore we want to avoid inbound holdings and want to meet outbound slot constraints, so we assume a given latest time τ_i^{LT}. Furthermore each aircraft 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 and holdings.

Due to wake vortexes and runway occupancy times, we have to consider minimum separation times $\delta_{i,k}$ between two aircraft $i \in F$ and $k \in F$, which depend on the weight classes of the two aircraft. Furthermore, fair schedules should be achieved. This means, that it is worse if one aircraft has a delay of 30 minutes than three that are delayed 10 minutes each. We consider the following two different mathematical models: the first one is a mixed-integer model, deciding the ordering of the aircraft together with the touch-down/take-off time. The second uses a time discretization. This yields an assignment problem with side constraints that computes for every discretized point in time whether an aircraft is scheduled 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 with the mixed-integer model. With it we obtain which nominal model is solved fast, which is a necessary condition in practical ATC application. As we find experimentally that the time-indexed model has several advantages when compared to the mixed-integer model, we use it subsequently. Many other heuristic and exact approaches exist to directly solve the runway scheduling problem (see [15] for an overview). We will, however, benefit from our approach with a time-indexed model as soon as we consider uncertainty in section IV.

A. Modelling with Exact Touch-down/Take-off Times (MIP)

In this section we formulate a mixed-integer non-linear program for Runway Scheduling, like in [6], but using a quadratic objective value. With the standard linearization we achieve a mixed-integer linear program. To accomplish a schedule, which fulfills the above described problem statement, the variables $t_i \in \mathbb{R}$, which denote the touch-down or take-off time, are introduced. We optimize the objective function

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

with schedule time τ_i^{ST}, $t_i^{LT+} = max\{0, t_i - \tau_i^{LT}\}$ and weights $\omega_{ST}, \omega_{LT} \in \mathbb{R}$.
Furthermore the touch-down or take-off time has to be bigger or equal than its earliest time:

$$t_i \geq \tau_i^{ET} \qquad\qquad \forall i \in F$$

We now introduce for each pair of aircraft $i, j \in F$ a binary variable $m_{i,j}$ that models the order:

$$m_{i,j} = \begin{cases} 1, & \text{if aircraft } i \in F \text{ scheduled before } j \in F \\ 0, & \text{otherwise.} \end{cases}$$

Minimum separation times $\delta_{i,j}$ have to be fulfilled between two aircraft $i \in F$ and $j \in F$. We thus have to assure that the following statement holds if aircraft $i \in F$ lands/starts before aircraft j:

$$m_{i,j} = 1 \Rightarrow t_j \geq t_i + \delta_{i,j} \qquad \forall\ i,j \in F$$

This yields

$$m_{i,j} \cdot (t_i + \delta_{i,j} - t_j) \leq 0 \qquad \forall\ i,j \in F,$$

and the standard linearization results in the formulation

$$\begin{aligned} t_i + \delta_{i,j} - t_j &\leq M_{i,j} \cdot (1 - m_{i,j}), \quad && \forall\ i,j \in F, i < j \\ t_j + \delta_{j,i} - t_i &\leq M_{j,i} \cdot m_{i,j}, && \forall\ i,j \in F, i < j, \end{aligned}$$

where $M_{i,j}, M_{j,i}$ are 'large' constants.

In general, it will turn out in the computational results that this formulation is solvable to optimality only for small instances. Next, we introduce a different optimization model for the runway scheduling problem. In the computational results we show that the time-discretized formulation leads to more effective solution algorithms.

B. Runway Scheduling as an Assignment Problem with Side Constraints

We focus on the time discretization method, suggested by Dyer and Wolsey ([12]), and transfer it to the runway scheduling problem as a time-indexed model (TIM), similar to what has been done in [18].
Let $F = A \cup D$ be the set of aircraft to be scheduled in the time horizon T. We denote the set of arrivals as A and the set of departures as D. The time horizon $T = \{t_1, ..., t_m\}$ is discretized in time slots $t_j \in T$. 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. We use equidistant time intervals for each aircraft.

From a mathematical point of view, the holdings act like a recovery to make the solution feasible. Thus for each aircraft $i \in F$ the time horizon reduces to

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

For example, if the discretization is given by $T = \{1000, 1075, 1150, ..., 1600\}$ and the earliest/latest time intervall for aircraft i by $[1111, 1582]$, the time horizon for aircraft i would be $T_i = \{1150, 1225, ..., 1525\}$. 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 if an aircraft is scheduled on a slot or not:

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

We want to minimize the sum of deviation from the schedule time τ_i^{ST} and penalize the sum over 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 unfair schedules. The objective function with weights $\omega_{ST}, \omega_{LT} \in \mathbb{R}$ can then be formulated as

$$\min \quad \sum_{i=1}^{n} \sum_{j \in T_i} b_{ij} \left(\omega_{ST} \left(t_j - \tau_i^{ST} \right)^2 + \omega_{LT} (t_i^{LT+})^2 \right) \tag{3}$$

The first constraint 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

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

where Δt describes the length of one time slot. 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 ([2]). In general, the problem with constraints (6), however is difficult to solve in practice for large instances. Solutions for slot sizes of $\Delta t = 75$s have good runtime performance, but result often in low objective function values with respect to runway utilization. The challenge is to reduce the number of time-indexed variables so that the problem is still computationally tractable.

C. Dynamic Time-Indexed Model for Runway Scheduling

The dynamic time-indexed model (D-TIM) is constructed like the time-indexed formulation in subsection III-B 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 scheduled time roughly according to the slot sizes. Thus we may reduce the number of variables used in the optimization, leading to smaller computational runtimes. The time horizon T_i for each aircraft $i \in F$ is, therefore, defined as

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

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

TABLE I: Comparison between MIP, TIM and DTIM

Instances		MIP	TIM		DTIM
			$\Delta t = 5$	$\Delta t = 75$	
1	Runtime [s]	375	504	<1	14
	Obj. value 10^5	0.6	0.7	6.9	2.1
2	Runtime [s]	>3600	836	<1	20
	Obj. value 10^5	1.0	1.9	12.7	4.6
	Gap %	26.0			
3	Runtime [s]	>3600	>3600	<1	23
	Obj. value 10^5	2.9	7.3	14.5	4.2
	Gap %	60.1	17.7		
4	Runtime [s]	>3600	520	<1	23
	Obj. value 10^5	0.5	0.5	17.6	2.1
	Gap %	58.3			
5	Runtime [s]	>3600	>3600	<1	19
	Obj. value 10^5	2.7	6.1	27.4	8.3
	Gap %	57.9	50		
6	Runtime [s]	1714	1971	<1	11
	Obj. value 10^5	0.6	0.9	9.8	1.8
7	Runtime [s]	245	482	<1	11
	Obj. value 10^5	0.43	0.6	8.1	1.0
8	Runtime [s]	>3600	468	<1	8.3
	Obj. value 10^5	0.57	0.57	19.9	4.9
	Gap %	20.6			
9	Runtime [s]	>3600	621	<1	13
	Obj. value 10^5	1.1	1.7	18.6	4.9
	Gap %	34.7			
10	Runtime [s]	>3600	>3600	<1	11
	Obj. value 10^5	1.3	2.6	19	3.1
	Gap %	2.3	20.9		
avg	Runtime [s]			<1	15.3
	Obj. value 10^5	1.7	2.28	15.5	3.7

D. Results for the Different Approaches

To check the performance of the models we choose 10 different randomly instances for 50 aircraft and test them once for each model. In table I the results with respect to computational runtime in seconds and objective function value scaled by 10^5 are shown. We thereby compare the MIP model, the time-indexed model with time discretization steps Δt of 5 and 75 seconds and the dynamic time-indexed model. As we do not consider a whole simulation, but only one simulation step, we have to determine the time discretization steps for the dynamic time-indexed model: Δt_1 of 5 seconds for the first 10 and Δt_2 of 75 seconds for the remaining aircraft. The weights ω_{ST} and ω_{LT} in the objective function (1) and (3) are chosen equally to $\frac{1}{2}$. Gap denotes the relative gap between the computed lower bound and the computed minimum.

First of all the MIP model computes the objective function value precisely, but unfortunately runs out of time (>1h) for 7 of 10 instances, while the TIM model with time discretization size of 5 seconds exceeds the time limit only in 3 cases. With increasing time discretization sizes for TIM the computational runtime decreases, For discretization size of 75 seconds, the solution for every instance is computed in less than 1 second, but the objective function value 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 objective function when two following aircraft are scheduled beyond their minimum separation times. This occurs when the minimum separation times are bigger than one slot length, but much more 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 objective function value increases only by a factor of 1.7 of that case where all aircraft are computed with discretization size of 5 seconds. In comparison an overall discretization size of 75 seconds results in a rise by a factor of 6.8.

Going into details of each instance we can obtain that DTIM is fast ($\leq$ 23seconds), whereby MIP needs at least 245 seconds. When many aircraft have similar schedule times, the problem gets hard to solve, like instance 3, 5 and 10. Both MIP and TIM with discretization size of 5 seconds run out of time. Also the objective function value is increased in this cases. This occurs due to fact that many schedule times are similar and thus the gap between target time and schedule time increases. TDIM 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.

IV. Robust Scheduling Considering Uncertainties

In reality, we have to face disturbances and uncertainties in the input data that usually lead to deviations from the actual plan or schedule. From a mathematical point of view robust optimization approaches have to be considered (see [7], [8]). Here, we protect the model against data uncertainties by incorporating a predefined uncertainty set into the nominal model in order to avoid expensive or even infeasible (worst-case) solutions for the disturbed problem. We thereby obtain among all robust feasible solutions, a robust optimal one, which guarantees the best objective function value. The following robustification focuses on TIM, but can analogously be obtained for DTIM. The robust counterparts are denoted by RTIM and RDTIM. 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 discrete variable $b_{i,j}$ either exists or the variable vanishes. Thus the set T_i in (2) changes according to the variations in the interval of earliest and latest times. Thus the robust time horizon reads as

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

where $prot_i$ denotes the deviation in the earliest and latest times for each aircraft, we want to protect against.

The uncertainty affects moreover some constraints in the considered optimization model, such that an optimal nominal solution might become infeasible in case of uncertainty. So we have to robustify the constraints (4) and (6), whereas inequality (5) maintains.

For each aircraft we ensure that it lands or departs within its robust earliest and latest time interval. So constraint (4) has to be modified to

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

Furthermore the minimum separation times between two aircraft might be violated, which leads to go-arounds or lost departure drops. So we protect the constraints (6) by the use of additional buffer:

$$b_{i,j} + \sum_{l=j+1}^{j+\lceil \frac{\delta_{i,k}}{\Delta t_k} \rceil + \mathit{buff}_{i,k}} b_{k,l} \leq 1, \quad \forall\ i \in F, \forall\ j \in T_i^R, \forall k \neq i.$$

whereat buffer $\mathit{buff}_{i,k}$ 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.

V. Validation Setup

Using 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 to avoid jumps from one simulation step to another. Thus the sequence does not change until significant improvement in the objective function value is achieved. We compare three different planners in the simulation (M=3 in Fig.2): 1) nominal planner, 2) robust planner 1, 3) robust planner 2. The TIM model with discretization size of 75 seconds works as a nominal planner without considering robustness (III-B). Its robust counterpart (RTIM) also with discretization size of 75 seconds denotes robust planner 1. As robust planner 2 we chose the robust DTIM model (RDTIM) with discretization size of 5s for the last 10 minutes before touch-down/take-off and discretization size of 75s otherwise. 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 planner with the optimal values of a planner knowing already all the deviations in advance.

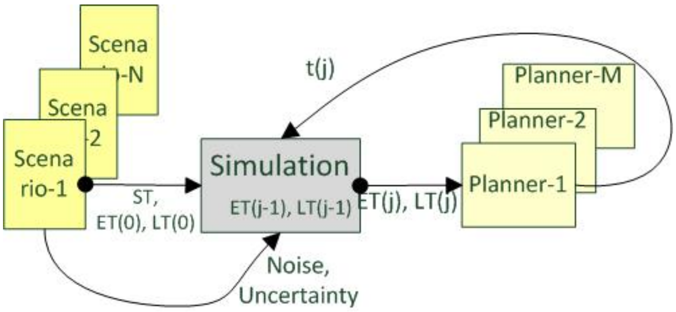

Fig. 2: Simulation and Planing

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

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. Fig. 2 furthermore 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 the simulation of the operator 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 arrival. On the other hand we have the simulation of the disturbances, in which we add a random value to ET and LT.

The increment of the earliest time $\text{inc}_i^{ET,j+1}$ of aircraft a_i in simulation step $j+1$ converges linearly to the planned target time $\tau_i^{TT,j}$ of the previous simulation step j by the following formula:

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

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 1.0 (3 minutes before planned touch-down/take-off time) to 0.2 (more than 20 minutes 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

$$\mathcal{N}(n_i^{j+1}, (\kappa_i^{j+1} \cdot \sigma_i)^2). \tag{7}$$

$n_i^0 = n_i^1 = \mu_i$ 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:

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

Factor κ_i is set to 1.0 for outbounds and for inbound it ensures that the standard deviation decreases the nearer an aircraft gets to its target touch-down time and thus the uncertainty reduces:

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

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

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

The latest time are changed in a similar way: convergence towards the planned times and adding the same randomly chosen disturbance dist_i^{j+1}.

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 $\tau_i^{ET,j+1} \leq \tau_i^{TT,j} \leq \tau_i^{LT,j+1}$ does not hold anymore.

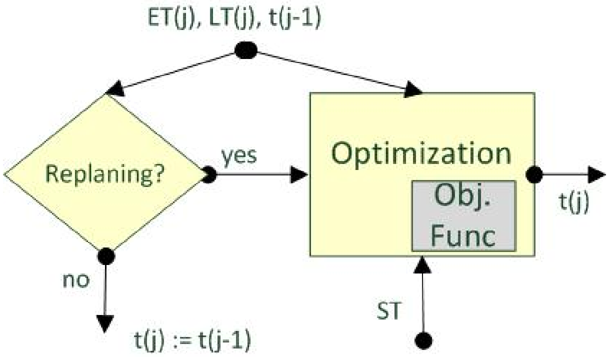

Fig. 3: Replanning Decision and Optimization

VI. Results

Each scenario (low, high uncertainty, medium, high traffic) contains 20 runs with 50 different randomly chosen instances. Schedule times are randomly chosen, so that each five minutes slot (for example 12:00, 12:05, 12:10, ...) except perhaps 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 to 10 minutes 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 two times standard deviation:

$$prot_i = \mu_i + 2 \cdot \sigma_i \quad \forall \imath \in F \quad \text{and}$$

$$buff_{i,k} = max(0, prot_i) + max(0, -prot_k) \quad \forall \imath, k \in F.$$

Thus a buffer is only installed, when the predecessor is supposed to be later or the follower earlier. For low uncertainty we chose for all aircraft as mean value $\mu_i = 2s$ and standard deviation $\sigma_i = 2s$, and at high uncertainty the values are $\mu_i = 10s$ for all aircraft, $\sigma_i = 4s$ for arrival and $\sigma_i = 6s$ for departure. Recall that these values are not the predicted delay, which is computed with formula (7). $\mu_i = 10$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 seconds. For an aircraft with a remaining flight time of 3,600 seconds and with $\Delta\tau^{Sim} = 180$s we expect an average delay of 200 seconds $(10 \cdot \frac{3600}{180})$ with a standard deviation of 120 seconds $(\frac{1800+3600}{3600} \cdot 4 \cdot \frac{3600}{180})$.

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

GoAround: number of go-arounds in each simulation; A go-around with an increase of the earliest time by 15 minutes 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. output.

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

TABLE II: Results scenario 1: RTIM vs. TIM

Criteria	RTIM	TIM	TIM - RTIM
GoAround	0.2	1.8	1.6
Dep. drop	0.15	0.4	0.25
Makespan [s]	3064	3144	80
			TIM/RTIM
Changed Pos / SimStep	0.29	0.69	2.38
Changed TT / acft [min]	1.95	3.24	1.67
Obj. func. value / acft [s]	276.5	359	1.29
Comp. runtime [s]	12.5	13.1	1.05

TABLE III: Results scenario 1: RDTIM vs. RTIM

Criteria	RDTIM	RTIM	RTIM - RDTIM
Makespan [s]	2878	3064	186
			RTIM/RDTIM
Changed Pos / SimStep	0.91	0.29	0.31
Changed TT / acft [min]	3.34	1.95	0.58
Comp. runtime [s]	236.4	12.5	0.05

Makespan [s]: difference in seconds between the target time of the last aircraft and the first aircraft in the resulting final simulation sequence

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

Obj. func. value / acft [s]: First we calculate the objective function of the final sequence and subtract the objective function value of the omniscient planner. We only consider here 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 kernel, 2,3 GHz and 4 GB RAM. The code is written in C++. As a commercial solver for integer problems we use GUROBI ([14]), version 5.6.
The tables read as follows: in the second and third column the average values per simulation of the corresponding planner are listed. In the forth column the difference between the two considered planner is illustrated for the first three criteria and the quotient for the remaining criteria.

A. Scenario 1: high traffic, low uncertainty

In this first scenario, table II shows the comparison between the robust and the nominal time-discretized model. The criteria go-around and departure drop are profitable in the robust case by a factor of 9 for go-arounds and 2.7 for departure drops respectively. In general, the price of robustness would increase the makespan. But because of less go-arounds and departure drops the robust planner reveals 80 seconds less makespan than the nominal one. The robust results show that one protects not only against undesired maneuver in flight, like go-arounds, but

TABLE IV: Results scenario 2: RTIM vs. TIM

Criteria	**RTIM**	**TIM**	**TIM - RTIM**
GoAround	0	2.4	2.4
Dep. drop	0	0.9	0.9
Makespan [s]	3269	3274	5
			TIM/RTIM
Changed Pos / SimStep	1.26	3.13	2.48
Changed TT / acft [min]	6.67	8.56	1.28
Obj. func. value / acft [s]	484.7	484.6	1.0
Comp. runtime [s]	31.8	26.4	0.83

TABLE V: RTIM for low and high uncertainty at high traffic

Criteria	**lowU**	**highU**	**highU - lowU**
GoAround	0.2	0	-0.2
Dep. drop	0.15	0	-0.15
Makespan [s]	3064	3269	205
			highU/lowU
Changed Pos / SimStep	0.29	1.26	4.34
Changed TT / acft [min]	1.95	6.67	3.42
Obj. func. value / acft [s]	276.5	484.7	1.75
Comp. runtime [s]	12.5	31.8	2.54

TABLE VI: Results scenario 3: RTIM vs. TIM

Criteria	**RTIM**	**TIM**	**TIM - RTIM**
GoAround	0	1.1	1.1
Dep. drop	0	0.2	0.2
Makespan [s]	3982	4015	33
			TIM/RTIM
Changed Pos / SimStep	0.31	0.62	2.0
Changed TT / acft [min]	2.5	4.86	1.94
Obj. func. value / acft [s]	96.9	157.2	1.62
Comp. runtime [s]	15.2	26.9	1.76

TABLE VII: RTIM for med./high traffic at high uncertainty

Criteria	**medTr**	**highTr**	**highTr - medTr**
GoAround	0	0	0
Dep. drop	0	0	0
Makespan [s]	3982	3269	-713
			highTr/medTr
Changed Pos / SimStep	0.31	1.26	4.06
Changed TT / acft [min]	2.5	6.67	2.68
Obj. func. value / acft [s]	96.9	484.7	5.0
Comp. runtime [s]	15.2	31.8	2.1

also achieves stable plans. In the simulation of all runs less than a third position changes are observed in average, such that only in less than every third run one position change is required. In comparison the nominal planner needs to change positions 2.38 times more often. Considering the changed target times the robust planner with 1.95 minutes per aircraft in average is more stable than its nominal counterpart, which has an average change of 3.24 minutes. The average deviation from schedule time is 1.29 times higher in the nominal case, whereby the computational runtime is the same. This shows that the robust model is not harder to solve than the nominal one.

To accomplish a better makespan one can use the robust dynamic time-indexed model. Thereby the inaccuracy is reduced by decreasing the discretization from 75 seconds to 5 seconds 10 minutes before touch-down/take-off (see table III). The disadvantage thereby is the increase in position and target time changes because of the jump in the discretization size. Also the computational runtime increases rapidly, because more binary variables are computed in the dynamic model.

B. Scenario 2: high traffic, high uncertainty

In the second scenario the uncertainty is increased. First of all, table IV shows, that no go-arounds and departure drops are required in the robust case because of the high protection, which depends on the uncertainty. The makespan is slightly the same in both cases. Thus, the price of robustness is compensated like in the first scenario. The number of changed positions reduces in the robust case by a factor of 2.48 and the changes in target time by 1.28. Considering the criteria of objective function value, robust and nominal planner compute the same. This is planner, that in the robust case, a buffer between two aircraft and a protection against earliest time deviations is incorporated. Whereas in the nominal case, go-arounds and departure drops are needed. Again the computational runtime is practicable.

As a second part, we want to analyse how the robust planner changes in terms of increased uncertainty at constant high traffic (table V). We have to mention that the buffer between all aircraft also increases by this. Thus the number of go-arounds and departure drops is kept at a minimum. Notice, that by increasing the buffer in the first scenario by an addition of σ, the number of go-arounds and departure drops would vanish. In contrast the makespan would increase. The buffer in combination with the protection against deviations in the earliest times lead to bigger makespan value as well as to higher objective function value. The criteria changed position and changed target time rise by a factor of about 4 because of the larger uncertainties. These values, however, are still lower in comparison to the nominal planner. Considering the computational runtime, higher uncertainty lead to smaller time windows for each aircraft by constant high traffic, which makes it more difficult to find a feasible solution.

C. Scenario 3: medium traffic, high uncertainty

Having considered high traffic and high uncertainty, we are now interested in the behaviour of the robust and nominal planner when reducing the traffic to a medium level (see table VI). Due to high uncertainty the buffer is still high and thus protects against go-arounds and departure drops in contrast to the nominal planner. As well the criteria makespan is profitable in the robust case with 33 seconds in average. Comparable to both scenarios before, the robust planner ”wins” the criteria changed position and changed target time by a factor of 2. The objective function value reduces, because go-arounds and departure drops affect the nominal planner. The runtime is

again viable.
How does the robust planner react, when traffic reduces? Table VII 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 considering constantly 50 aircraft, the makespan increases automatically, because the deviation from schedule time is optimized. The reduction in changed position and changed target times stems from changes at medium traffic are easier to control and do not have as high effects on other aircraft as at high traffic. The criteria objective function value points into the same direction. Finally the computational runtime reduces because less aircraft are in conflict state of being scheduled at the same time.

VII. Conclusion

The goal of this work was to study runway scheduling under disturbances. 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 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.
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 compare our approach with the corresponding results for the nominal approach that does not take uncertainties into account.
The computational results show that the robust model indeed computes less go-arounds and less departure drops in every scenario. Furthermore, the number of changed positions and changed target time per aircraft are profitable in the robust setting.
Usually, one expects that a high level of protection leads to low throughput. In other words, 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 reduces the number of go-arounds and departure drops significantly, the makespan is equal or slightly better, when compared to the nominal case. The same argument is valid for the objective function value. With our approach it is also possible to choose more complex objective functions, without having significant effects on runtime efficiency. For demonstration purposes we have concentrated on the deviation from the scheduled time. In our scenarios, also the achieved computational runtime for the robust time-indexed model is practicable. Therefore, taking uncertainties into account already in the model, it is indeed possible to calculate more stable schedules.
In order to reduce the makespan further, we investigated a robust dynamic time-indexed model in which aircraft with less remaining flight time use a smaller discretization size than before. Although such an approach makes sense algorithmically, our preliminary computational results lead to some disadvantages. Currently, the number of position changes and target time changes as well as the runtimes increase. In order to eliminate these disadvantages, we will conduct further research in context of the dynamic time-indexed model in the future. Special care has to be taken in the choice of the discretization size in order to make sure that the number of variables does not grow too large.

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. Opinions in this work reflect the authors views only, and EUROCONTROL and/or the SJU shall not be considered liable for them or for any use that may be made of the information contained herein.

References

[1] Agogino, A.: Robustness of two Air Traffic Scheduling Approaches to Departure Uncertainty, 30th Digital Avionics Systems Conference (DASC), Seattle, 2011

[2] Ahuja, R.; Magnanti, T.; and Orlin, J. : Network Flows: Theory, Algorithms, and Applications, Prentice Hall Inc., Upper Saddle River, New Jersey, 1993

[3] Anagnostakis, I.; Clarke J.P.: Runway operations planning: a two-stage solution methodology, Proceedings of the 36th Hawaii International Conference on System Sciences, 2003

[4] Atkin, J.; Burke, E.; Greenwood, J. and Reeson D.: Hybrid metaheuristics to aid runway scheduling at London Heathrow Airport, Transp. Sci., vol. 41, no. 1, pp.90 -106, 2007

[5] Balakrishnan, H.; Chandran, B.: Scheduling Aircraft Landings under Constrained Position Shifting. AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado, 2006

[6] Beasley, J.; Krishnamoorthy, M.; Sharaiha, Y.; and Acramson, D.: Scheduling Aircraft Landings - the Static Case. Transportation Science, Vol. 34, Issue 2, p.180-197, 2000

[7] Bertsimas, D.; Sim, M.: The price of robustness, Operations Research, 52(1):35-53, 2004

[8] Ben-Tal, A.; Nemirovski, A.: Robust Solutions of Uncertaint Linear Programs", Operations Research Letters , Vol. 25, pp. 1-13, 1999

[9] Böhme, D.: Tactical Departure Management with the Eurocontrol/DLR DMAN, 6th EUROCONTROL/FAA ATM R& D Seminar, Baltimore, MD, USA, 2005

[10] Chandran, B.; Balakrishnan, H.: A Dynamic Programming Algorithm for Robust Runway. American Control Conference, New York City, 2007

[11] Clare, G.; Richards A.: Optimization of taxiway routing and runway scheduling, IEEE Trans. Intell. Transp. Syst., vol. 12, no. 4, pp.1000 -1013, 2011

[12] Dyer, M.; Wolsey, L.: Formulating the single machine sequencing problem with release dates as a mixed-integer program, Discrete Applied Mathematics, no. 26 (2-3), pp. 255-270, 1990

[13] Frankovich, M.; Bertsimas, D.: Air Traffic Flow Management at Airports: A Unified Optimization Approach, ATMSeminar 2013

[14] Gurobi Optimization, Inc.: Gurobi Optimizer Reference Manual, 2014

[15] Helmke, H.: Scheduling Algorithms for ATM Applications Tools and Toys, 30th Digital Avionics Systems Conference, Seattle, October 16-20, 2011

[16] Hu, X.-B.; Chen, W.-H.: Receding horizon control for aircraft arrival sequencing and scheduling, IEEE Trans. Intell. Transp. Syst., vol. 6, no. 2, pp.189 -197, 2005

[17] Hu, X.-B.; Di-Paolo, E.: Binary-representation-based genetic algorithm for aircraft arrival sequencing and scheduling, IEEE Trans. Intell. Transp. Syst., vol. 9, no. 2, pp.301 -310, 2008

[18] Kjenstad, D.; Mannino, C.; Nordlander, T.E.; Schittekat, P.; and Smedsrud, M.: Optimizing AMAN-SMAN-DMAN at Hamburg and Arlanda airport, Proceedings of the SID, Stockholm, 2013

[19] Neuman, F.; Erzberger, H.: Analysis of Sequencing and Scheduling Methods for Arrival Traffic. NASA Technical Memorandum, Moffett Field, California, 1990

[20] Solveling, G.: Stochastic programming methods for scheduling of airport runway operations under uncertainty, Dissertation, George Institute of Technology, 2012

[21] Völkers, U.: Arrival Planning and Sequencing with COMPAS-OP at the Frankfurt ATC-Center. The 1990 American Control Conference, San Diego, California, 1990