Pre-Tactical Optimization of Runway Utilization
Under Uncertainty

Manu Kapolke*, Norbert Fürstenau†, Andreas Heidt*, Frauke Liers*, Monika Mittendorf† and Christian Weiß†

*Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)
Economics, Discrete Optimization, Mathematics
Cauerstr. 11, 91058 Erlangen, Germany
Email: name.surname@fau.de

†German Aerospace Center (DLR)
Institute of Flight Guidance
Lilienthalplatz 7, 38108 Braunschweig, Germany
Email: name.surname@dlr.de

Abstract—Efficient planning of runway utilization is one of the main challenges in Air Traffic Management (ATM). It is important because runway is the combining element between airside and groundside. Furthermore, it is a bottleneck in many cases. In this paper, we develop a specific optimization approach for the pre-tactical planning phase that reduces complexity by omitting unnecessary information. Instead of determining arrival/departure times to the minute in this phase yet, we assign several aircraft to the same time window of a given size. The exact orders within those time windows can be decided later in tactical planning. Mathematically, we solve a generalized assignment problem on a bipartite graph. To know how many aircraft can be assigned to one time window, we consider separation requirements for consecutive aircraft types. In reality, however, uncertainty and inaccuracy almost always lead to deviations from the actual plan or schedule. Thus, we present approaches to incorporate uncertainty directly in our model in order to achieve a stabilization with respect to changes in the data. Namely, we use techniques from robust optimization and stochastic optimization. Further, we analyze real-world data from a large German airport to obtain realistic delay distributions, which turn out to be two-parametric $\Gamma$-distributions. Finally, we describe a simulation environment to test our new solution methods.

I. INTRODUCTION

ATM systems are driven by economic interests of the participating stakeholders and, therefore, are performance oriented. As possibilities of enlarging airport capacities are limited, one has to enhance the utilization of existing capacities to meet the continuous growth of traffic demand. The runway system is the main element that combines airside and groundside of the ATM System. Therefore, it is crucial for the performance of the whole ATM System that the traffic on a runway is planned efficiently. Such planning is one of the main challenges in ATM. Uncertainty, inaccuracy and non-determinism almost always lead to deviations from the actual plan or schedule. A typical strategy to deal with these changes is a regular re-computation or update of the schedule. These adjustments are performed in hindsight, i.e. after the actual change in the data occurred. 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, more stable plans and a more efficient support for ATM controllers and stakeholders. Incorporating uncertainty into the ATM planning procedures further makes the total ATM System more resilient, because the impact of disturbances and the propagation of this impact through the system is reduced.

In the present paper, we investigate the problem of optimizing runway utilization under uncertainty. The goal is to incorporate uncertainties into the initial plan in order to retain its feasibility despite changes in the data. We focus on the pre-tactical planning phase, i.e. we assume the actual planning time to be several hours, or at least 30 minutes, prior to actual arrival/departure times. We develop an appropriate mathematical optimization model for this particular planning phase. The basic idea is that in pre-tactical planning we can reduce the complexity of the problem by not determining an exact arrival/departure sequence in terms of exact landing/take-off times for each aircraft, as we do later in tactical planning. Instead, we answer the question of how many aircraft can be scheduled to one time window of a given size without violating distance requirements. (For example, it is definitely possible to assign more than one aircraft to a time window from 12:00 pm to 12:10 pm.) Then, we consider a discretized time horizon consisting of such time windows and assign each aircraft to one of them.

This paper is an extension of [14], where the authors set up a mixed integer program (MIP) for the pre-tactical optimization of runway utilization. Afterwards, the impact of disturbances on the deterministic solutions was investigated. The results showed that it is crucial to enrich the optimization approach by protection against uncertainties, in order to produce less necessary replanning. In the current paper, we thus incorporate uncertainties directly into the model by using techniques from robust and stochastic optimization. The remainder of this paper is organized as follows: In Section II, we give an overview over the literature related to runway optimization and explain why our approach is different. We develop the pre-tactical runway
optimization model in Section III. In Section IV and V we describe our approaches to incorporate uncertainties into this model, and present some computational results in Section VI. In order to be able to test our approaches in a more realistic setting, we analyze real-world delay data from a large German airport in Section VII (extending the descriptions in [14]), where we also describe our simulation environment to test current and future solution methods. Finally, we conclude in Section VIII.

II. RELATED WORK

There are many different approaches that deal with the optimization of runway utilization in the literature. Most of them treat the runway scheduling problem in the tactical planning phase.

A. Deterministic Approaches

The most cited MIP model in this context is probably the one introduced by Beasley et al. [5]. Their linear objective function minimizes delay, the constraints come from the aircraft dependent separation times. They also present an integer program (IP) formulation where time is discretized, but they don’t explore it further because of disappointing computational experiences. Soomer and Frank [24] consider the problem from an airline point of view. They use Beasley’s MIP but allowing airlines to define their own cost functions for each flight. Bertsimas et al. [9] develop a comprehensive IP for Air Traffic Flow Management which integrates all phases of a flight, different costs for ground and air delays, rerouting, continued flights and cancellations. Kjenstad et al. [20] state a time-discretized model. They assign an aircraft to a time window and claim that a number of subsequent time windows (dependent on the aircraft type) remains unassigned. In their model, they also consider minimal taxiways and the possibility to drop departures. Their linear objective function minimizes delay and the number of dropped departures.

Many authors use heuristic methods aiming to provide solutions in close to real-time. To schedule aircraft in a first-come first-served order (FCFS) seems to be fair and also reduces the work of traffic controllers. However, such an approach doesn’t provide maximal throughput or minimal delay in general (Bennell et al. [6]). Dear [12] developed the concept of Constrained Position Shifting, were each aircraft can be scheduled only a limited number of steps away from the FCFS sequence. Balakrishnan and Chandran [3] solved this problem as a shortest path problem on a special network.

Anagnostakis and Clarke [2] formulate a two-stage heuristic algorithm for the outbound runway scheduling problem. In the first stage, candidate weight class sequences are determined w.r.t. distance requirements, ordered by the corresponding throughput. In the second stage individual aircraft are assigned using operational constraints (e.g. earliest and latest departure times of aircraft).

As mentioned, in our optimization model (described below and in [14]) we allocate time windows to aircraft. However, though many papers about runway optimization deal with "slot allocation", this term is used to describe different problems. Often, it is associated with the Ground-Holding Problem (GHP), where "slot" means a certain departure time which is assigned to an aircraft. Ball et al. [4] also address the GHP, but they assign arrival slots to aircraft which provide the corresponding departure delay in hindsight. They consider matchings in a bipartite graph which they call the "flight allocation graph". The main focus in this paper lies on the graph structure and matching algorithms.

None of the approaches above deal with "slots" as time windows to which several aircraft can be assigned. Thus, to the best of our knowledge there is no approach similar to ours in which the pre-tactical planning phase is modelled by assigning such time windows to aircraft.

B. Approaches that Incorporate Uncertainties

All runway optimization approaches presented above assume that all parameters are known with certainty. We found few works where uncertainties are incorporated. Chandran and Balakrishnan [11], e.g., develop an algorithm with Constrained Position Shifting that handles uncertainty in the estimated time of arrival. Hu and Paolo [17] formulate a genetic algorithm and compute solutions disturbing the estimated arrival time of 20% of the aircraft. Sölveling [23] presents a two-stage stochastic program for solving the mixed-mode runway scheduling problem with uncertain earliest times. In the first stage he determines the weight class sequence. An exact sequence of individual aircraft follows in the second stage.

III. THE MODELING

As mentioned above, we model the problem of optimizing runway utilization in the pre-tactical planning phase by assigning time windows to aircraft. Throughout this paper, we consider single-mode runways with only arriving aircraft. In the future, we will adjust our models to mixed-mode runways. But since the single-mode problem is already quite complex from a mathematical point of view, we decided to focus on arrivals for now. In our modeling approach we claim that each aircraft has to receive exactly one time window as each aircraft has to be scheduled. On the other hand, the number of aircraft that can be assigned to one time window depends on its size and the weight classes of the aircraft. The underlying idea is that, contrary to tactical planning, we don’t need to determine arrival times to the minute yet, because we are several hours (or at least 30 minutes) prior to the first scheduled time. Thus, the exact arrival sequences within the time windows can be decided later.

In this section, we develop a MIP for the described problem. The objective is the maximization of punctuality. In other words, the deviation from scheduled times in both directions shall be minimized. The MIP constraints consist of general assignment constraints and the modeling of minimal time distance requirement. Those minimum separation times between two consecutive aircraft depend on their corresponding weight classes. Hereof, we consider three different aircraft categories (Light, Medium and Heavy) and use Table I ([18]).
Before we can state our model, we have to analyze the underlying problem structure more precisely. For each aircraft, we consider several corresponding times:

- **Scheduled time of arrival (ST):** a fix time that yields a benchmark to identify delay and earliness of the aircraft. This may be the time the passenger finds on his flight ticket.
- **Earliest time of arrival (ET):** depends on operational conditions (and on the impact of disturbances).
- **Latest time of arrival (LT):** latest time the aircraft can land without holdings. It depends on the earliest time ET and on the actual planning time (or start time, respectively, if the aircraft is still on the ground).
- **Maximal latest time of arrival (maxLT):** a hard condition for landing which is calculated with respect to physical, operational and other relevant conditions (for instance, amount of fuel, prioritization, etc.).

Those times further determine the corresponding time windows ST$_W$, ET$_W$, LT$_W$ and maxLT$_W$ for each aircraft.

### A. Assignment Graph

To model the problem of assigning aircraft to time windows, we consider a bipartite graph $G = (A \cup W, E)$ consisting of a vertex set $A$ of aircraft and a vertex set $W$ of time windows of a given size in a given time period (ordered chronologically). An edge $(i, j) \in E$ corresponds to a possible assignment of aircraft $i$ to time window $j$. Possible assignments concerning a certain aircraft are all time windows from ET$_W$ to maxLT$_W$.

Now, a feasible solution for our assignment problem is a set of edges such that

- every aircraft vertex is linked with exactly one edge from this set, i.e. every aircraft is assigned to exactly one time window,
- every time window vertex is linked with a number of edges from this set, so that no separation time constraints are violated.

In Figure 1 we see a small example of a bipartite graph with a possible assignment of aircraft $a_1, \ldots, a_4 \in A$ to time windows $w_1, w_2, w_3 \in W$.

### B. Decision Variables

To solve our assignment problem, we have to decide whether to choose a certain edge or not. To model this decision in our MIP, we introduce a binary variables $x_{ij}$ for each edge $(i, j) \in E$:

$$x_{ij} = \begin{cases} 1, & \text{if aircraft } i \text{ is assigned to time window } j \\ 0, & \text{otherwise} \end{cases}$$

### C. Objective Function

Our objective is the minimization of delay and earliness, respectively. We model delay/earliness as edge weights. The weight $c_{ij}$ of an edge $(i, j) \in E$ results from the distance of time window $j$ to the ST$_W$ of aircraft $i$ (counted in number of time windows). Delay is penalized quadratically for reasons of fairness (e.g., a solution in which one aircraft has a delay of six time windows is worse than a solution in which two aircraft have a delay of three time windows each). Earliness is penalized linearly. If the assigned time window is after the LT$_W$ (i.e. between LT and maxLT), we add an extra penalization term, namely the squared distance from LT$_W$. Assume an aircraft $i$ with ST$_W$ $w_1$, ET$_W$ $w_1$, LT$_W$ $w_10$ and maxLT$_W$ $w_13$. Then we’d have, e.g., $c_{i2} = 3$, $c_{i8} = 3^2$ and $c_{i12} = 7^2 + 2^2$.

Now the objective function of our optimization model is the following:

$$\min \sum_{(i,j) \in E} c_{ij}x_{ij}$$

### D. Aircraft Constraints

First of all, we have to assert that each aircraft is assigned to exactly one time window. So we claim

$$\sum_{j \in W_i} x_{ij} = 1$$

for each $i \in A$, where $W_i = \{j \in W : (i, j) \in E\}$ describes the set of time windows that aircraft $i$ can be assigned to.

### E. Time Window Constraints

Further, we have to determine the number of aircraft that can be assigned to one time window. In order to do so, we need to consider the distance requirements, dependent on the weight classes of consecutive aircraft. We use the minimum separation times shown in Table I. Clearly, the maximum number of aircraft that fit in one time window is reached when a sequence from Light to Heavy is assumed. In more detail, to avoid separation times of 125 and 150 seconds, such a sequence contains sub-sequences of aircraft of the same type. First, all Lights are scheduled, followed by all Mediums, and finally by all Heavies. According to Table I, we therefore need a separation time of 75 seconds after each Light and

1 An adaptation of the results in this paper to other minimum separation time tables is possible as well.
each Medium, whereas we need 100 seconds after each Heavy except the last one (the needed separation time after the last aircraft in a time window models the distance requirements at the window boundary and is analyzed later in this section).

For each time window we get upper bounds on the number of aircraft by assuming such a sequence from Light to Heavy. Mathematically, it yields the following two constraints for each \( j \in W \):

\[
\sum_{i \in L_j} x_{ij} + 75 \sum_{i \in M_j} x_{ij} + 100 \sum_{i \in H_j} x_{ij} \leq s + 100 \quad (3)
\]

\[
75 \sum_{i \in L_j} x_{ij} + 75 \sum_{i \in M_j} x_{ij} \leq s + 75 \quad (4)
\]

Here, \( s \) is the size of the time windows (in seconds). Further, \( L_j = \{ i \in L : (i, j) \in E \} \) describes the set of Lights that may be assigned to time window \( j \) (and thus, the corresponding sum yields the number of Lights that are assigned to it). \( M_j \) and \( H_j \) are defined analogously.

If we assign aircraft to a time window without exceeding these bounds in (3) and (4), we know that there exists a sequence of those aircraft that fits in the time window. However, we do not determine how that sequence looks exactly in terms of concrete predecessors and successors. We are still flexible in (re)arranging different aircraft of the same type. And if the time window contains enough “empty space”, we can even deviate from the Light-Medium-Heavy order without changing the assignment.

Further, we extend (3) and (4), because they do not assert security distances at the time window boundaries yet. This means that the last aircraft in one time window and the first aircraft in the subsequent time window may be planned to land at the very same time. In order to obtain feasible solutions, we can generally claim 150 extra seconds as buffers in every time window. But of course, this approach only provides a heuristic procedure for solving the problem because those buffers will be unnecessarily large for some time windows. In [14], we describe a way to model distance requirements at the window boundaries precisely. For this purpose, we introduce additional variables for each time window which model the situation at the boundaries (dependent on corresponding aircraft types). Afterwards, we modify our constraints to assure suitable minimum separation times at the end of each time window. For instance, if we have a Heavy at the end of a time window \( j \) and a Medium at the beginning of the subsequent one (assuming a sequence from Light to Heavy in both windows), we assure 125 extra seconds of separation time at the end of \( j \) (according to Table I). If we have a Medium at the end of \( j \) instead, we assure 75 seconds and so on.

IV. INCORPORATING UNCERTAINTIES

In this section, we want to incorporate uncertainty into the model to receive a robustification of our solution plan. In general, robustification means to ensure that deviations in the input data do not have a large impact on the solution. Considering the optimal solution of the nominal problem, i.e.

the problem where uncertainties are ignored, small deviations in the input data could have the effect that the nominal optimum becomes infeasible for the disturbed problem, i.e. the problem where the input data suffers from deviations.

In mathematics, there exist different approaches to handle uncertainty in optimization. In stochastic optimization (e.g. [19]) the goal is to describe the uncertainty by probability distributions. Knowing these distributions, one can then optimize the expected values. A second approach to the problem of modelling uncertainty is located in robust optimization (e.g. [7], [8]), where the goal is to immunize against predefined worst-case scenarios. In contrast to stochastic optimization, the probability distributions of the uncertainties do not need to be known. However, one has to predefine uncertainty sets that determine the values of the uncertain parameters against which the optimization problem has to be protected. The task is to find robust feasible solutions, i.e. solutions that are feasible for all parameter values in the uncertainty set. Among all robust feasible solutions, the robust optimal solutions are those with the best guaranteed objective function values.

A. Robust Optimization Approach

In the setting for our model described in section III, the uncertain parameters are the ET windows \( ET_W \) and, dependent on those, \( LT_W \) and \( maxLT_W \). Hence, we have to predefine an uncertainty set for each aircraft. Therefore, we have to chose deviations of the earliest time we want to be protected against. For each aircraft this yields an interval of possible earliest times and thus a set of possible \( ET_W \)’s. These \( ET_W \)’s also determine the possible \( LT_W \)’s.

Now, we actually solve our optimization model from section III. But in the robust approach we assume an assignment graph that only contains edges corresponding to assignments which are feasible for every realization of our chosen uncertainty set. An example of feasible assignments for an aircraft in the robust model is illustrated in Figure 2. As mentioned, the robust model assumes the worst-case, i.e. the extreme cases for earliest time \( (w_1) \) and maximal latest time \( (w_7) \) in the predefined uncertainty set are taken into account, whereas the other time windows which lay within the uncertainty set for both times \( (w_2, w_3, w_8, w_9) \) are forbidden.

B. Stochastic Optimization Approach

We follow a single-stage stochastic optimization approach in which we optimize over all assignments which are "expected
to be possible” dependent on the underlying probability distribution. Therefore, we consider the expected values for ET and maxLT for each aircraft, or the corresponding time windows, respectively. Afterwards, we optimize the obtained “expected scenario”, i.e. we solve our mathematical model described above with decision variables (edges in the assignment graph) that correspond to the feasible assignments in this scenario. In Figure 3 we show an example of feasible assignments in the expected scenario for one aircraft.

V. ADVANCED ROBUSTNESS CONCEPTS

The robustification approach in the previous section was a strict one, which potentially can be too conservative and produce large delay values. In this section we take a look at less conservative robust approaches to avoid the over-conservatism of strict robustness.

A. Recoverable Robustness

In the context of railway, recoverable robustness ([21]) has been established. We now apply these approaches to our optimization model for pre-tactical planning.

Since a (nominal) solution might become infeasible by a realized scenario, the concept of recoverable robustness, introduced by Liebchen et al. [21], simultaneously considers the optimization of the problem and an algorithm, which repairs the solution to a feasible one, in cases of infeasibility. This algorithm is called recovery action. In contrast, the strict robust approach postulates that a strict robust solution is feasible for all possible uncertainties in the uncertainty set, which of course is a more strict assumption. However, a recoverable robust solution can be recovered by limited actions to a feasible one for all occurring scenarios (not necessarily the same feasible solution in each scenario).

We consider our nominal model and introduce first stage variables $x_{ij}$, which denote the assignment variables of the nominal solution (as defined in section III-B). Furthermore, we have to define additional recovery variables $y_{ij}^s$ for each aircraft $i \in A$, time window $j \in W_i^s$ and scenario $s \in S$. These variables denote second-stage assignments for each scenario (equal to 1 if aircraft $i$ is assigned to time window $j$ in scenario $s$).

Thus we want to minimize the delay costs of the nominal solution and the worst-case costs for the recovery action for each scenario. Since we want to plan a nominal solution $x$ as close as possible to a feasible solution $y^*$ for all scenarios, the recovery costs for each scenario $s \in S$ would be the difference between the computed nominal solution of the $x$-variables and the solution of the $y^*$-variables. As we require fairness for the recoverable robust solution in general, we have to optimize the following objective function

$$\min_x \sum_{i \in A} \sum_{j \in W_i} c_{ij} x_{ij} +$$

$$(5)$$

$$\max_{s \in S} \min_{y^s} \sum_{i \in A} \left( \sum_{j \in W_i} j \cdot x_{ij} - \sum_{l \in W_i^s} l \cdot y_{il} \right)^2$$

$$(6)$$

whereby the first summand (5) denotes the nominal optimization problem with delay costs $c_{ij}$ and the second summand (6) describes the recovery costs.

Now we require a feasible assignment on both first stage and second stage. Thus, besides the constraints defined in III-D and III-E for the $x$-variables, we further need the same constraints for the $y^s$-variables for each scenario $s \in S$ (using the corresponding assignment graphs). Although $x$- and $y$-variables decompose in the constraints, this problem is mathematically challenging. Beside the quadratic objective function we obtain a min-max-min-structure. Furthermore, it is a problem with $1 + |S|$ many assignments with quadratic objective function. Therefore we propose a simplification.

B. Recovery to Strict Robust Solution

In order to handle the problem of the min-max-min-structure, we can recover the nominal solution to a strict robust one. By that, we do not have $y^s$-variables for each scenario, but only consider the worst-case. Again, using the structure of uncertainty, the strict robust time horizon set can be defined as $W^R = \bigcap_{s \in S} W_i^s$, which is feasible for all scenarios $s \in S$. Thus we consider decision variables $y$, which are defined in the set $W_i^R$. We then can simplify the objective function, namely (6), by cutting off the $\max_{s \in S}$ part and minimizing over those $y$-variables instead of the $y^s$-variables.

We still have to consider the same constraints, but regarding the second stage, they reduce to constraints for only one scenario (the strict robust one). Thus, we have to solve two assignment problems, which are decomposed in the constraints and only coupled in the quadratic objective function. Furthermore, the min-max-min-structure is reduced to a minimization problem. However, this approach requires the existence of a feasible strict robust solution. Further, we do not get the optimal recovery costs for every realized scenario, but we get one recovery action which is always feasible.

Since the objective function above is still quadratic in the variables, we linearize it. Therefore, we introduce new binary $u$-variables for substitution $u = xy$ and use the fact, that for every binary variable $x \in \{0, 1\}$, $x^2 = x$ holds. We then achieve two assignment problems with a linear objective function. Furthermore the variables $x_{ij}$, $y_{il}$ and $u_{ijl}$ are linearly coupled within the constraints (modeling the fact $u = xy$).

![Fig. 3. Possible assignments for an aircraft $a_i$ in the stochastic model](image-url)
VI. COMPUTATIONAL RESULTS FOR THE ROBUST AND
STOCHASTIC OPTIMIZATION APPROACHES

The following results were obtained by the integer pro-
gramming solver Gurobi (version 5.6). For the experiments
we used a laptop with Intel i7 CPU, 4 cores (2.70 GHz)
and 8 GB RAM. We tested instances of 200 aircraft which
have to be assigned to 36 time windows of 600 seconds (10
minutes). The distribution of the weight classes was always
82% Medium, 11% Heavy and 7% Light. The ST\textsubscript{j}’s for all
aircraft are chosen randomly, i.e. uniformly distributed.
The ET\textsubscript{j}’s are assumed to be the predecessors of the ST\textsubscript{j}’s. We
further assumed LT\textsubscript{j} and maxLT\textsubscript{j} to be 40 min (4 time
windows) and 60 min (6 time windows), respectively, after
ET\textsubscript{j}. The disturbances on our ET windows were \(\Gamma\)-distributed
with \(\tau = 1.82\) windows (yields mean delay \(\mu = 0.73\) and
\(\sigma = 1.19\) (which is reasoned in section VII).

In this computational study we compare four approaches:
nominal, stochastic, strict robust, and recovery to strict robust
solution. The uncertainty set for the strict robust solutions is
determined by shifting ET/maxLT by \(\mu \pm k \cdot \sigma\) time windows,
with \(k = 1\) (note that \(k = 0\) yields the stochastic approach).
For each approach, we generated 100 random instances. In
Table II we see the averaged results.

The first observation considering Table II is that most
runtimes are very low. The parameter *infeasible assignments*
shows whether the optimal solution of the corresponding
approach is still feasible after the disturbances occurred. It
describes the number of aircraft that have been assigned to
time windows to which they cannot be assigned in the
disturbed situation. We see that the robust approaches have
the least infeasible assignments. In fact, these approaches are sig-
nificantly better than the stochastic approach which provides
about three times the number of infeasible assignments of the
strict robust approach. However, the stochastic approach still is
better than the nominal one. This satisfies our expectations,
because it shows that optimizing with robust or stochastic models
provides more stable plans that become "less infeasible" facing
disturbances. Further, the result for the 'recovery to strict'
approach is what we expect from recoverable robustness, i.e.
providing a trade-off between nominal delay and strict robust
stability. (To actually recover the computed solutions into the
strict robust scenario in case of infeasibility, we would replan
97.97 aircraft with a maximum replan distance of 1.72 time
windows.) However, we also notice that even the strict robust
approach still has some infeasible assignments, i.e. it is not
totally robust. This results from our chosen uncertainty set,
i.e. the relatively small chosen \(k = 1\). Anyway, because being
more robust means deleting more possible assignments, we
have to choose uncertainty sets carefully.

Concerning the average delay value (counted in time
windows) and the number of delayed aircraft, we observe
inverse relations: these values are smaller for the nominal
approach than for the stochastic one, and even larger for the
robust approaches. Considering the strict robust approach,
we see that all aircraft are delayed. However, this is not
surprising at all due to our setting. We chose ST\textsubscript{W} to follow
the nominal ET\textsubscript{W}. Hence, if we choose to protect us against a
deviation of more than one time window (which we did), we
can’t assign any aircraft to its ST\textsubscript{W} anymore. Considering the
'recovery to strict' approach, its reduced conservatism yields
the possibility to choose solutions with lower delay costs (in
comparison with the strict robust approach), but the recovery
condition still has a price (compared to the nominal approach).

So far, in this paper we have described a mathematical
approach for optimizing runway utilization in the pre-tactical
planning phase. Further we have enhanced our developed
optimization model by incorporating uncertainties in different
ways (robust and stochastic) and discussed some computa-
tional results. In the following section, we now analyze real-
world disturbances from our database from a large German
airport. Finally, we describe a simulation environment to
test our current and future approaches with those realistic
disturbances.

VII. EMPIRICAL DELAY DATA ANALYSIS
AND BASELINE SIMULATION

Understanding and modeling the statistics, dynamics, and
propagation of air-traffic arrival and departure delays is a
prerequisite of any attempt to optimize the punctuality of
schedules and airport capacity, and minimizing necessary
buffer times for required robustness of performance (e.g. [25],
[26]). That is why for validating the new scheduling models
by means of Monte Carlo simulations we start with the design
of an appropriate stochastic delay model.

A. Stochastic Delay Model

For validation of the new stochastic and robust optimization
algorithms we investigate a simple stochastic arrival and
departure delay model that is tested by means of empirical
delay data from a large German airport. Recently, Caccavale
et al. [10] presented a model for simulating inbound traffic
over a congested hub termed "pre-scheduled random arrivals"
(PSRA) where they defined the actual arrival time \(t_{i}^{\text{ATA}} := t_{i}\)
by a close to Poisson process with mean inter-arrival times \(\frac{i}{X}\)
of clients in a queuing line:

\[
t_{i} = \frac{i}{X} + \epsilon_{i}, \quad i = 1, \ldots, n \in \mathbb{Z}
\]
The model is represented by a continuous probability density function (PDF) \( f(t) \) of the random arrival time variable \( \epsilon \) with finite standard deviation \( \sigma \) and zero mean, without loss of generality. \( \frac{1}{\lambda} = \) expected inter-arrival time between two consecutive aircraft, \( \frac{1}{\lambda} = \mu = < \Delta t_{\text{ATA}}^{-1} >= < t_{\text{STA}}^{-1} - t_{\text{STA}}^{-1} >, \) with actual arrival times \( t_{\text{STA}}^{-1} = \) actual in-block time AIBT, in what follows. Guadagni et al. [15] prove that this process converges to the memoryless one-parametric Poisson process for large \( \sigma \). This approach overcomes the often used assumption of uncorrelated arrivals as precondition of the Poisson process, i.e. exponentially distributed inter-arrival times \( \Delta t_{\text{ATA}}^{-1} \). Empirical histograms of delay data exhibit a pronounced non-symmetry (e.g. [25]) that was modeled by Wu [27] by means of the two-parametric Beta-probability density function (limited to the open (0, 1) interval).

For our purpose the family of two-parametric Gamma (\( \Gamma \))-PDF's (limited to \( \mathbb{R}^+ \), with shape and scaling parameters \( a, b \)) appears more appropriate as analytical model of the empirical arrival delay statistics because it extends to \( +\infty \), i.e. like the Poisson process covers also large delays, and it contains the corresponding exponential distributions as a special case ([13]).

A realistic model of arrival delays, in addition to the asymmetry has to include a significant amount of early arrivals, i.e. delay \( t^D < 0 \). Furthermore, besides the statistics of the sequence of all different arrivals \( a_i \) (different flights) during single days of operation (single day statistics) also single flight (=airline) statistics (e.g. all arrivals \( j \) of the same flight \( a_i \) over a time interval of e.g. half a year) have to be modeled ([11]). The delay statistics naturally exhibits daily, weekly, and seasonal periodicities and trends, i.e. nonstationary behavior. Consequently any realistic model has to be a combination of deterministic and random components ([11], [25]) which is one reason for the inappropriateness of the Poisson model. For taking into account early arrivals \( t^D < 0 \) each histogram data set has to be transferred into \( \mathbb{R}^+ \) by subtracting the minimum delay \( \min(t^D) \) before data fitting with the \( \Gamma \)-model. The \( \Gamma \)-PDF as a generalization of the Poisson process of inter-arrival times \( t \) may be parametrized by the shape parameter \( a \) and the mean \( \tau \).

\[
\begin{align*}
f(t; \tau, a) &= \left( \frac{a}{\Gamma(a)} \right) a^{a-1} \tau^{-a} e^{-\frac{a}{\tau}}
\end{align*}
\]  

\begin{align*}
\text{with normalized time scale } \frac{\tau}{\lambda}, \text{ scaling parameter } b \text{ defined via } \tau = a \cdot b, \text{ and the } 2^\text{nd} \text{ and } 3^\text{rd} \text{ (central) moments } \\
\mu_2 = \text{variance } = a^2 = \frac{\tau^2}{\lambda}, \mu_3 = a b^2 = 2\sigma^2 b = 2 a^3, \\
\text{with skewness } \gamma = \frac{\mu_3}{\mu_2^{3/2}} = \frac{b}{\tau\sqrt{\lambda}}, \text{ and coefficient of variation } \frac{cv}{\tau} = \frac{1}{\sqrt{\lambda}} \text{ independent of } \tau. \text{ A residual linear correlation } cv \sim \tau \text{ would result in an inverse power law } a(b)\text{-anticorrelation. For } a = 1, \text{ (8) reduces to the Poisson case of maximum randomness, i.e. exponential } t\text{-distribution. For } a < 1, \text{ (8) models a process with larger variance than the random process due to clustering, i.e. non-independent clustered events. For large } a > 1, \text{ with the } \Gamma\text{-PDF approaches a } (\tau, \sigma)\text{-Normal distribution.}
\end{align*}

The \( \Gamma \)-model may be related to the PSRA model by splitting the average inter-arrival time \( \mu = \frac{1}{\lambda} \) of (7) into the deterministic (schedule) part \( \mu_{\text{STA}} \) and the stochastic delay contribution \( \mu^D = \mu + \mu_{\text{STA}} \), \( \mu^D = \tau + t^D_{\text{min}} \):

\[
t^D_i = \mu^D + \epsilon_i = \tau + t^D_{\text{min}} + \epsilon_i
\]  

where \( \epsilon_i \) collects the random contributions from \( \mu_2 \) and \( \mu_3 \). Usually \( t^D_{\text{min}} < 0 \) so that the \( \Gamma \)-PDF fit \( f(t^D) \) covers the required amount of early arrivals. The analysis of empirical arrival and departure delay histograms in the following section VII-B together with Monte Carlo (MC) computer experiments in section VII-C in fact indicate \( \Gamma \)-models to provide reasonable approximations for the arrival and departure delay statistics as one usable metric for the optimizer performance differences, with characteristic deviations from \( \Gamma \)-PDF due to the optimization.

B. Analysis of Empirical Arrival- and Departure Delays and Derivation of Disturbance Statistics

In this section we model the empirical arrival and departure delays of flights \( a_{ij} \) (i.e. delays \( t \)) with a stochastic \( \Gamma \)-process according to (8) and (9), with delays \( t \) = random deviations from scheduled arrival times \( \text{STA} \), flight plan, and we derive an empirical disturbance statistics for use with the MC-computer experiments. As proposed by Abdel-Aty et al. [1] we analyze and model daily delays observed within the time series of all flights \( a_{ij} \) (i = 1 - m > 200) during full days of operation, as well as delay data from a selection of single flights \( a_{ij} \) over a couple of months (with \( j = 1 - n \geq 150 \) monitored arrivals or departures).

Figure 4 shows an example of arrival delay distribution \( f(\text{ATA} - \text{STA}) \) for a single full day. We also analysed a sample of 33 flights (different callsigns) with \( \geq 150 \) arrivals each / half year (out of 1384 within 7 - 12/2013). The \( \chi^2 \)-tests of the maximum likelihood (ML) \( \Gamma \)-fits to the empirical delay histograms differ significantly between single days as well as between single flights. This is no surprise, of course, due to the neglect of any deterministic effect (correlations between consecutive flight arrival times or delays depending on traffic density, ATC-sequencing etc.).

The figure legend provides the fit results for the parameter estimates \( a, b \) with \( \Gamma \)-mean \( \tau \) (the same for empirical histogram and ML-estimate), \( a-b \) correlation coefficient, and \( \chi^2 \)-test of \( \Gamma \)-hypothesis (0-hypothesis rejection for \( p < 5\% \)). The fit example in the case in fact formally should be rejected at the \( p = 5\% \) level, basically due to the deviations around zero delay (AIBT - STA \( = t^D = 0 \)). Besides the necessity of considering the above mentioned deterministic effects, this deviation around \( t^D = 0 \) can be explained by active ATC interventions to minimize delays (replaced by the algorithmic scheduling optimization in the following section VII-C). Nevertheless we obtained many examples without 0-hypothesis rejection, i.e. \( p(\chi^2) > 5\% \). The average fit parameter estimates for the 33 single flights \( a_i \) are \( \pm 1 \) stddev: \( a > 3.5(1.3), \) \( b >= 8.7(3.4), \) \( \tau >= 27.5(7) \) min, with average minimum earliness \( t^D_{\min} >= -23.9(8.8) \) min (transformation
Performance Review Report [22]) departure delays represent 
activated by the fact that according to Eurocontrol statistics (see 
delays as the only disturbance during the flight. This is moti-

\[ \mu \text{mean departure delay} \]

\[ \sigma \text{mean departure delay variation} \]

\[ \epsilon \text{min.} \]

\[ D \text{arrival delay of} \]

\[ \tau \text{target time} \]

\[ \sigma \text{arrival variation of} \]

\[ \tau \text{earliness} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]

\[ \epsilon \text{min.} \]

\[ \epsilon \text{stderror of mean} \]
for the whole daily arrival sequence the deviations from the individual schedules ST Ai, or alternatively from ETi. An update of optimized ai-sequences is calculated for each Δt Sim, and the daily sequence will undergo changes as long as new flights are starting from their respective departure airports, with the individual departure delay drawn from the same average Γ-PDF (a = 2.5, b = 8, τ = 18.2 min; see previous section) and shifted back to the delay scale μ D. Typically, for 17 hours of daily operation of our empirical dataset we have ca. 260 simulation steps per MC-run. Runtime depends on the traffic density, time of operation and sequencing algorithm. With 200 MC-runs per experiment we typically have up to several hours of simulation time for a specific model and scenario. The simulations run on a high performance PC with 2xIntel 64 Bit E5645 12 core processors (24 cores with hyperthreading “on”), 2.4 GHz, 24 GB RAM.

2) Baseline Simulations: In order to establish a baseline, the MC-simulations as a first step were performed without considering a-priori knowledge of disturbance. The corresponding baseline simulations used the First-Come-First-Serve rule (FCFS) and a standard optimizer (Take Select 8-2 ([16]), requiring a monotonous version of the objective function with zero cost for early arrivals). In what follows we present results obtained with the FCFS and TS8-2 methods, for both the standard (S6.2) and dense traffic scenario (S7.2).

Figure 6 depicts an MC-simulation (MC057: S7.2) with the FCFS rule (i.e. no optimization) as an example for a single day (= single run) delay statistics for all flights of 8 hrs of operations. The figure shows the delay histogram with Γ-PDF fit that may be compared with the empirical PDF of Figure 4. In most cases the Γ-PDF fits to the delay histograms exhibit good χ2-test results (ρ(χ2) > 5%-rejection threshold of 0-hypothesis, i.e. for given confidence no rejection).

A corresponding result is obtained for the single flights ai-analysis with 200 repeated arrivals each. The runtime (±std(dev) of this experiment for S6 / S7 respectively was 1.7(0.6) / 1.1(0.4) s. The 209 individual histograms with Γ-PDF fits for each single flight and the summary plot exhibits results similar to the single days case, with the latter numbers (averages < of fit-mean values with standard deviations ()) for the 200 MC-runs (MC062(S6)/MC057(S7)):

- For the pre-tactical optimization of assigning time windows for runway utilization. In this model, several aircraft can be assigned to the same time window which reduces the complexity of the problem. Further, we enriched the model by protection against uncertainties using techniques from robust and stochastic optimization. Our computational study showed that such an incorporation of uncertainties has a large effect on the resulting solutions. The stochastic approach optimizes the expected scenario and, therefore, is more likely to remain feasible in the face of disturbances than the nominal approach.
Thus, on average it provides more stable plans and less necessary replanning. However, robust optimization methods provide even more stable solutions. Using the strict robust approach, we definitely know that a solution (if one exists) will be feasible for all scenarios within the pre-determined uncertainty set. Thus, it is the approach with the highest possible stability. However, this may come at the price of increased delay. Recoverable robustness on the other hand takes into account that a time window assignment might become infeasible in some scenario. In that case, it applies a recovery action, i.e. a replanning step, that makes the assignment feasible again. This potentially necessary recovery is already incorporated in the computation of the initial solution. Hence, recoverable robustness provides a promising trade-off between little delay (as nominal) and high stability (as strict robust). This is also true, if we consider the ‘recovery to strict robust solution’ approach (under the assumption that a strict robust solution exists). This approach is way easier to solve than the general recoverable robust approach, since we don’t have an exponentially large scenario set to consider and further we have an easier objective function. Our corresponding computed solutions remain quite stable while producing less delay than the strict robust approach.

We performed a statistical analysis of real-world data from a large German airport for deriving a departure delay model to generate realistic disturbances for Monte Carlo computer experiments. Furthermore, we described a simulation environment for these experiments to test current and future optimization approaches. The data analysis together with the baseline simulations indicate the two-parametric $G$-PDF to be a reasonable approach for modeling random disturbances.

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 Work Package E in the SESAR Programme. Opinions expressed 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.

We are grateful to A. Martin for many stimulating scientific discussions about mathematical optimization and for sharing valuable insights with us. Further, we are indebted to Olga Gluchshenko who prepared the initial dataset for the Monte Carlo simulations. Many thanks are due to S. Loth for paving the way to the airport data sources and to M. Helms for initial preparation of the large amount of empirical data from that source. We would like to thank H. Helmeke for valuable advice throughout this work concerning the optimization aspects and to J. Rataj for continuous support.

REFERENCES


