



Jena Research Papers in Business and Economics

Scheduling Aircraft Landings to Balance Workload of Ground Staff

Nils Boysen, Malte Fliedner

08/2008

Jenaer Schriften zur Wirtschaftswissenschaft

Working and Discussion Paper Series
School of Economics and Business Administration
Friedrich-Schiller-University Jena

ISSN 1864-3108

Publisher:

Wirtschaftswissenschaftliche Fakultät
Friedrich-Schiller-Universität Jena
Carl-Zeiß-Str. 3, D-07743 Jena
www.jbe.uni-jena.de

Editor:

Prof. Dr. Hans-Walter Lorenz
h.w.lorenz@wiwi.uni-jena.de
Prof. Dr. Armin Scholl
armin.scholl@wiwi.uni-jena.de

www.jbe.uni-jena.de

Scheduling Aircraft Landings to Balance Workload of Ground Staff

Nils Boysen^{a,*}, Malte Fliedner^b

^a*Friedrich-Schiller-Universität Jena, Lehrstuhl für Operations Management, Carl-Zeiß-Straße 3, D-07743 Jena, Germany, nils.boysen@uni-jena.de*

**Corresponding author, phone +49 3641 943-100.*

^b*Universität Hamburg, Institut für Industrielles Management, Von-Melle-Park 5, D-20146 Hamburg, Germany, fliedner@econ.uni-hamburg.de*

Abstract

Scheduling landings of aircrafts is an essential problem which is continuously solved as part of the daily operations of an airport control tower. All planes in the airspace of an airport are to be assigned to landing slots by the responsible air-traffic controller. The support of this decision problem with suited optimization approaches has a long lasting tradition in operations research. However, none of the former approaches investigates the impact of the landing sequence on the workload of ground staff. The paper on hand presents three novel objectives for the aircraft landing problem, which aim at leveling the workload of ground staff by evenly spreading: (1) number of landed passengers, (2) landings per airline, and (3) number of landed passengers per airline over the planning horizon. Mathematical models along with complexity results are developed and exact and heuristic solution procedures are presented.

Keywords: Aviation; Aircraft Landing; Scheduling; Workload Balancing

1 Introduction

With increasing levels of air traffic, an efficient planning and execution of airport operations becomes more and more important. An essential problem in this context is the aircraft landing problem (ALP), which aims at supporting air-traffic controllers in scheduling landings of all planes in the airspace of an airport to its runway(s). For each single plane a landing time within a prespecified time window, which depends on the remaining distance to be covered, the plane's maximum and minimum velocity and

the remaining fuel, is to be determined such that landing separation criteria due to air turbulence specified for each pair of planes are observed. A detailed description of the operational characteristics of ALP are provided by Beasley et al. (2000). Preceding research on the ALP focuses on one of the following objectives:

- Minimize the remaining fuel costs of all planes to be landed by meeting their most economic target landing times at preferred speed (Ernst et al., 1999; Beasley et al., 2000; Pinol and Beasley, 2006).
- Minimize the deviation to target landing times, which are planned in a mid-term horizon and published in the flight schedule (Beasley et al., 2001; Bianco et al., 2006; Pinol and Beasley, 2006).
- Minimize the completion time (makespan) of the schedule or maximize the throughput (Psaraftis, 1980, Bianco et al., 1999; Bianco et al., 2006; Atkin et al., 2007).
- Reduce perturbation of successively determined plans in a rolling horizon (online problem) by minimizing a displacement function (Beasley et al., 2004).

Thus far, none of the previous approaches considers the impact of the landing schedule on the workload of ground staff. In general, the ground staff working at the airport can be separated into two groups with regard to their affiliation. The first group are airport employees which are, for instance, engaged with unloading baggage, the fueling of planes and security checks. Whenever several big planes which carry plenty of passengers are assigned to landing slots in direct succession, workload of all operators can increase dramatically leading to an increase in waiting queues of passengers (diminishing customer satisfaction) or additional manpower and thus increasing wage costs. On the other hand, a sequence of small aircrafts with only few passengers causes idle time. The second group of airline employees is engaged with operations like cleaning planes, refilling of catering supply and maintenance checks. Here, successive landings of planes of the same airline (especially those carrying plenty passengers) alternated with periods without landings of the respective airline likewise cause high workload and idle time, respectively. It thus seems suggestive to generate landing schedules which lead to balanced workloads of ground staff. Note that even if the services are subcontracted to third-party service providers, the airport and airlines will also profit from balanced workloads, as the resulting cost advantages should lower service prices at least in the mid-term. In order to yield such balanced schedules, we define a target rate, which is based on the assumption that planned passenger arrivals and/or landings can be evenly spread over the planning horizon, so that actual landings should approximate this rate as close as possible.

The basic idea of leveling is borrowed from the famous “Toyota Production System” (see Monden, 1998), where a “level scheduling” (see Kubiak, 1993; Boysen et al., 2007a) of the final production stage (here, the runway schedule) facilitates the Just-in-Time principle. Subassemblies (here, ground staff services) are smoothly “pulled” off preceding (here, also succeeding) production stages, so that enlarged safety stocks (here, additional manpower) become obsolete. Three different objectives minimizing the deviation of actual from ideal schedules are investigated:

- To level the workload of airport employees a runway schedule is to be determined, such that the *number of passengers* carried by landing aircraft are evenly spread over the planning horizon.
- Whenever the planes of an airline carry a comparable number of passengers, it is sufficient to spread the *landings per airline* equally over the planning horizon to achieve a leveling of airline staff's workload.
- If the number of passengers per plane diverges considerably, then, to balance the workload of airline staff, the *number of passengers per airline* are to be evenly spread over time.

To establish this new class of leveling objectives and to derive basic insights on the objectives' impact on the structure of the decision problem, we model the aircraft landing problem in its very basic form. A given static set of planes is to be scheduled at a single runway. The modifications necessary to run such a static model to solve the underlying online problem is covered by Beasley et al. (2004). Moreover, as is a common premise in ALP research we restrict our investigation to aircraft landings, although mixed schedules incorporating take-offs can be covered as well (see Beasley et al., 2000). The separation time due to air turbulence between adjacent planes is assumed to be equidistant, which, in the real world, only occurs when all aircrafts are of the same plane model. Otherwise, it is a simplifying assumption, which reduces the scheduling problem to a sequencing problem. Finally, earliest and latest landing times associated with each single plane are not considered. With these reductions on hand, the core problem with regard to these objectives is extracted, so that the isolated impact of the objectives, i.e. on the complexity of the problem, can be investigated.

The paper is organized as follows. First, the landing problem covered in this paper is formalized in Section 2. Subsequently all three objectives are addressed in a separate section (Sections 3 to 5), each of which presents a mathematical model, states complexity, and develops exact and heuristic solution procedures. Finally, Section 6 concludes the paper with insights on how to relax some of the simplifying assumptions to solve real-world aircraft landing problems.

2 Problem description

To extract the core problem of leveled aircraft landing we restrict the problem as follows:

- We assume that the set P of planes remains unaltered during processing a derived plan. Thus, only the static version of ALP is considered.
- Only landings of aircraft (no take-offs) at a single runway are considered.
- The separation time between all pairs of planes is assumed to be equidistant. In the real world, this premise holds true if all aircraft P are of the same basic plane model and is an approximation of reality otherwise.

A	set of airlines (index a)
P	set plane (index p)
P_a	subset of planes ($P_a \subset P$) belonging to airline a (index p)
T	number of time slots for landings (index t)
g_p	number of passengers in plane p
r	ideal rate
x_{pt}	binary variable: 1, if plane p lands during slot t ; 0, otherwise
y	auxiliary integer variable

Table 1: Notation

- Earliest and latest landing times of planes are not considered. Thus, it is assumed that no assignment restrictions between planes and slots exist.

Following these assumptions we can now deduce a general set of mathematical constraints which are shared by all ALP versions covered in this paper. The notation is summarized in Table 1.

The input data of ALP is a given set P of planes each of which is to be assigned to a landing slot $t = 1, \dots, T$, where $|P| = T$. The assignment decision is represented by binary variables x_{pt} , which are 1, if plane p is scheduled to land during slot t and 0 otherwise:

$$x_{pt} \in \{0, 1\} \quad \forall p \in P; t = 1, \dots, T \quad (1)$$

Each plane p is further assigned to exactly one landing slot t in the planning horizon:

$$\sum_{t=1}^T x_{pt} = 1 \quad \forall p \in P \quad (2)$$

On the other hand, during each slot t exactly one plane is allowed to land:

$$\sum_{p \in P} x_{pt} = 1 \quad \forall t = 1, \dots, T \quad (3)$$

For all problem versions, we will further determine a target rate by distributing the overall number of passengers or landings evenly over the planning horizon. In order to balance the number of arriving passengers, for instance, the respective target rate is obtained by dividing the total number $\sum_{p \in P} g_p$ of passengers, where g_p denotes the number of passengers on plane p , by the number of slots T : $r = \frac{\sum_{p \in P} g_p}{T}$. Hence, a landing sequence is sought where actual landing rates of passengers are as close as possible to the target rate, so that the deviation aggregated over all slots is minimized. Figure 1 exemplifies the basic principle of leveled landing schedules.

In order to measure the overall deviation we first need to determine a metric which quantifies the actual deviation at a slot. Among the most prominent choices in the literature are absolute (also known as Manhattan or rectilinear), Euclidean or squared deviations (see Boysen et al., 2007a). In this paper, we will focus on absolute deviations, although our results likewise hold for the other forms. The single deviations need to

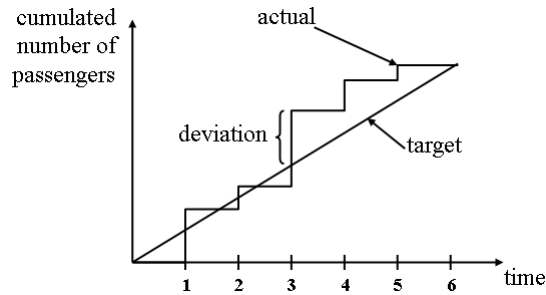


Figure 1: Basic principle of leveled ALP schedules

be further aggregated over all slots to a single objective value. Typically, either the sum of deviations (min-sum objective) or the maximum deviation (min-max objective) is minimized. In this work we will pursue the min-max objective, since it minimizes the extent of the deviations while preventing that single deviations become extraordinarily high, as might occur in the min-sum case. Thus, the min-max objective has a more direct economic impact compared to min-sum, as it reduces workload peaks during a shift, so that the number of permanent staff and/or stand-by workers do not need to cover these amplitudes during shift planning.

In the following three sections we will differentiate between three objectives and provide solution procedures for each of the problems separately.

3 Balancing the number of landed passengers

3.1 Mathematical model

To level the workload of airport staff we introduce the model ALP_1 , which aims at evenly distributing the number of landed passengers over time. Therefore, a target rate r is calculated as follows: $r = \frac{\sum_{p \in P} g_p}{T}$. With the help of this target rate, model ALP_1 can be formulated as a mathematical program with objective function (4) and constraints (1)-(3) and (5):

$$ALP_1: \text{Minimize } C(X, Y) = \max_{t=1, \dots, T} |y_t - t \cdot r| \quad (4)$$

subject to (1)-(3) and

$$y_t = \sum_{\tau=1}^t \sum_{p \in P} x_{p\tau} \cdot g_p \quad \forall t = 1, \dots, T \quad (5)$$

Equations (5) define auxiliary integer variables y_t to be the cumulative number of passengers landed up to period t . This number y_t minus the optimal number of landed passengers ($t \cdot r$) denotes the deviation of slot t . The maximum deviation over all slots t

is to be minimized within objective function (4).

Note that this problem has not been covered by level scheduling research for mixed-model assembly lines, thus far. However, it can be seen as a special version of the so called Output Rate Variation problem (see Bautista et al., 1996). The problem corresponds to a model sequencing problem, where the processing times (number of passengers g_p) of different models (planes $p \in P$) are to be evenly spread over the production cycles (landing slots $t = 1, \dots, T$) to balance the workload at an assembly line with a single station. In its structure, the problem is also similar to the unconstrained maximum job cost sequencing problem (e.g. see Monma, 1980), unlike the latter however, ALP_1 is NP-hard in the strong sense as is shown in the following section.

3.2 Complexity

In the following we will proof NP-hardness for ALP_1 . For this purpose we show how to transform instances of the 3-Partition problem to aircraft landing. 3-Partition is well known to be NP-hard in the strong sense (see Garey and Johnson, 1979) and can be summarized as follows:

3-Partition Problem: Given $3q$ positive integers a_p ($p = 1, \dots, 3q$) and a positive integer B with $B/4 < a_p < B/2$ and $\sum_{p=1}^{3q} a_p = qB$, does there exist a partition of the set $\{1, 2, \dots, 3q\}$ into q sets $\{A_1, A_2, \dots, A_q\}$ such that $\sum_{p \in A_i} a_p = B \quad \forall i = 1, \dots, q$?

Transformation of 3-Partition: Consider $4q + 2$ aircrafts where the first $3q$ “small” aircrafts have passenger numbers equal to $g_p = r - a_p \forall p = 1, \dots, 3q$, the following q “large” aircrafts carry $g_p = B + r \forall p = 3q + 1, \dots, 4q$ and the last two aircrafts have $g_{4q+1} = r - B/2$ and $g_{4q+2} = r + B/2$ passengers where a_p and B are positive integers with $B/4 < a_p < B/2$ and $r > B/2$ is the desired integer target rate. The length of such an instance is polynomially bounded in q , so that any instance of 3-Partition can be transformed to such an instance of ALP_1 in polynomial time. Note that in order to ensure integer numbers of passengers in ALP_1 , B and all a_j can be multiplied with a given even constant in the transformation w.l.o.g. and further that r can be any number greater than $B/2$ and will always result to the actual target rate for the given instance.

A simple lower bound \underline{C} for ALP_1 bases on the consideration that each plane $p \in P$ with $g_p > r$ ($g_p < r$) cases least deviation, when the deviation at the previous sequence position $t - 1$ is $\frac{1}{2} \cdot (r - g_p)$ ($\frac{1}{2} \cdot (g_p - r)$), because, then, scheduling plane p at position t causes a deviation of $\frac{1}{2} \cdot (g_p - r)$ ($\frac{1}{2} \cdot (r - g_p)$). Any bigger or smaller deviation at position $t - 1$ causes additional deviation at either slot t or $t - 1$. Obviously, the one plane with maximum deviation from target rate r constitutes the lower bound:

$$\underline{C} = \frac{\max_{p \in P} |g_p - r|}{2} \quad (6)$$

Note that in the considered instances the maximum absolute deviation from the target rate is B , so that $\underline{C} = B/2$ constitutes a lower bound in this case. We will now show

that finding an answer to the question of whether a solution with an objective value of $B/2$ actually exists is as hard as 3-Partition.

We can transform any solution to a YES-instance of 3-Partition to a solution of aircraft landing by simply ordering the sets A_i arbitrarily and scheduling them in the following fashion:

$$\langle 4q+1 \ 3q+1 \ A_1 \ 3q+2 \ A_2 \ 3q+3 \ A_3 \ \dots \ 4q \ A_q \ 4q+2 \ \rangle$$

At the beginning and the end of the sequence the two aircrafts carrying $r - B/2$ and $r + B/2$ passengers are assigned. In between a large aircraft is followed by a set of small aircrafts in an alternating fashion. It can be easily verified that such a sequence yields an objective value of $B/2$.

Let us assume that there exists a feasible sequence with $C \leq B/2$. In fact the existence of such a sequence depends critically on the assignment of the large airplanes. A large airplane can only be scheduled at a slot t if the previous slot has a deviation of $d_{t-1} = -B/2$, where $d_t = \sum_{\tau=1}^t \sum_{p \in P} x_{p\tau} \cdot g_p - t \cdot r$ denotes the actual deviation at slot t . Any smaller value $d_{t-1} < -B/2$ would immediately lead to a contradiction with $C \leq B/2$, any larger value would lead to a deviation at t of $d_t = d_{t-1} + r + B - r > B/2$ with $d_{t-1} > -B/2$ and also contradict $C \leq B/2$. We can thus conclude that the deviation after the assignment of a large airplane at t is $d_t = B/2$. As a consequence in between of any two large airplanes there needs to be a subsequence of other airplanes whose cumulated deviation is exactly $-B$. As there are q large airplanes, at least $q - 1$ subsequences of airplanes with a cumulated deviation of $-B$ are required. Note that before the first large airplane and after the last large airplane is assigned, the cumulated deviation needs to be brought from 0 to $-B/2$ and from $B/2$ to 0 respectively. It follows that the sequence needs to begin and end with a subsequence of planes with a cumulated deviation of $-B/2$.

It can be readily checked that a subsequence with a cumulative deviation of $-B/2$ cannot consist of small airplanes alone, as they show a deviation of $-a_p$ with $B/4 < a_p < B/2$, so that the deviation of any single small plane is strictly larger than $-B/2$ while any two small planes already have a cumulated deviation strictly smaller than $-B/2$. It follows that plane $4q+1$ has to be assigned to the beginning of a sequence with $C \leq B/2$, if $4q+2$ is assigned to the end and vice versa. While $4q+1$ immediately yields a deviation of $-B/2$, plane $4q+2$ requires an additional subset of small airplanes whose cumulated deviation is $-B$ to yield a total cumulated deviation of $-B/2$. Together with the $q - 1$ subsets of small planes with a cumulated deviation of B in between the large planes this yields the required partition.

An instance of 3-Partition is thus a YES-instance if and only if there exists a solution with $C \leq B/2$ for the corresponding instance of ALP_1 , which means that ALP_1 is NP-hard in the strong sense.

Reduction rule: Note that a problem instance of ALP_1 can be reduced by all planes carrying a number of passengers g_p which equals target rate r because, independent of their landing position, these planes only restore the previous deviation and can, thus, not

lead to an increased maximum absolute deviation. After having determined an optimal solution with the reduced input data these planes can be scheduled at arbitrary sequence position without altering the objective value.

3.3 Solution Algorithms

In this section we develop an exact Dynamic Programming approach and two heuristic start procedures for solving instances of ALP_1 .

3.3.1 Dynamic Programming approach

The Dynamic Programming (DP) approach to solve ALP_1 is based on an acyclic digraph $G = (V, E, w)$ with a node set V divided into $T + 1$ stages, a set E of arcs connecting nodes of adjacent stages and a node weighting function $w : V \rightarrow \mathbb{R}$ (see Bautista et al., 1996; Boysen et al., 2007b, for related approaches to scheduling mixed-model assembly lines). Each position t of the landing sequence is represented by a stage which contains a subset $V_t \subset V$ of nodes representing *states* of the partial landing sequence up to position t . Additionally, a start level 0 is introduced. Each index $i \in V_t$ identifies a state (t, i) defined by the vector \mathbf{X}_{ti} of binary indicators X_{tip} of all planes $p \in P$ already scheduled up to sequence position t . It is sufficient to store the numbers of planes already landed instead of their exact partial sequence, because the actual number of landed passengers at sequence position t and, thus, the deviation from the ideal number only depends on the aircraft scheduled up to position t irrespective of their order.

The following conditions define all *feasible* states to be represented as nodes of the graph:

$$\sum_{p \in P} X_{tip} = t \quad \forall t = 0, \dots, T; i \in V_t \quad (7)$$

$$X_{tip} \in \{0, 1\} \quad \forall p \in P; t = 0, \dots, T; i \in V_t \quad (8)$$

Obviously, the node set V_0 contains only a single node (initial state $(0, 1)$) corresponding to the vector $\mathbf{X}_{01} = [0, 0, \dots, 0]$. Similarly, the node set V_T contains a single node (final state $(T, 1)$) with $\mathbf{X}_{T1} = [1, 1, \dots, 1]$. The remaining stages have a variable number of nodes depending on the number of different plane vectors \mathbf{X}_{ti} possible.

Two nodes (t, i) and $(t + 1, j)$ of two consecutive stages t and $t + 1$ are connected by an arc if the associated vectors \mathbf{X}_{ti} and \mathbf{X}_{t+1j} differ only in one element, i.e., exactly one plane is additionally scheduled in position $t + 1$. This is true if $X_{tip} \leq X_{t+1jp}$ holds for all $p \in P$, because both states are feasible according to (7) and (8). The overall arc set is defined as follows:

$$E = \{((t, i), (t + 1, j)) \mid t = 0, \dots, T - 1; i \in V_t; j \in V_{t+1} : X_{tip} \leq X_{t+1jp} \forall p \in P\} \quad (9)$$

Finally, node weights w_{ti} assign the actual deviation of the partial sequence presented by state (t, i) . For this purpose, the cumulative number of landed passengers $\sum_{p \in P} X_{tip}$.

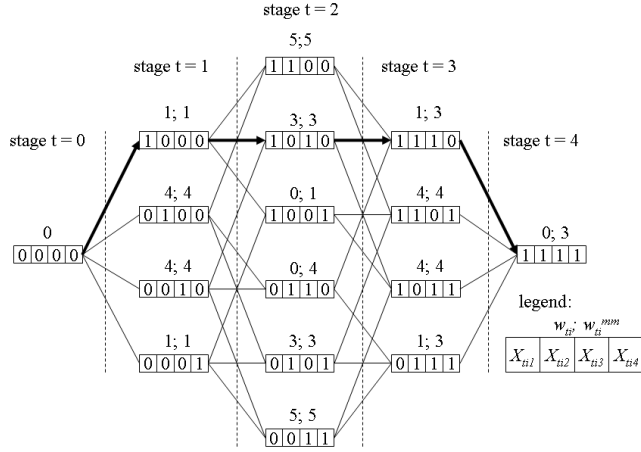


Figure 2: Example graph of DP for ALP_1

g_p are to be compared with the ideal number $(t \cdot r)$, so that node weights are calculated as follows:

$$w_{ti} = \left| \sum_{p \in P} X_{tip} \cdot g_p - t \cdot r \right| \quad \forall t = 0, \dots, T; i \in V_t \quad (10)$$

With this graph on hand, the problem reduces to finding a path from the source node at level 0 to the unique sink note at level T , which minimizes the maximum node weight (min-max weight path). This path can be easily determined during the stage-wise construction of the graph by updating the min-max weight w_{ti}^{mm} utilized up to the actual node according to the following recursion formula, where P_{ti} denotes the set of predecessor nodes of node (t, i) :

$$w_{ti}^{mm} = \max \{ \min_{(t-1,j) \in P_{ti}} \{w_{t-1j}\}; w_{ti} \} \quad \forall t = 1, \dots, T; i \in V_t \quad (11)$$

The DP-approach does not need to store the complete graph but only the reference to a predecessor node $(t-1, j) \in P_{ti}$ with minimum min-max weight w_{t-1j}^{mm} for each single node (t, i) of actual stage t . Any other node of the previous stage can be deleted. The optimal objective value corresponds to the min-max weight w_{T1}^{mm} of sink node $(T, 1)$. The respective optimal landing sequence can be determined by backwards recursion along the stored predecessor nodes (along the optimal path). The plane to be assigned at sequence position $t+1$ is the only one for which $X_{t+1jp} - Y_{tip} = 1$ holds.

Example: Given a set P of planes consisting of $|P| = 4$ aircrafts, which are supposed to carry 7, 10, 2 and 5 passengers, respectively. Thus, target rate r amounts to $r = 6$. The resulting graph along with a bold-faced optimal path is depicted in Figure 2. The corresponding optimal landing sequence is $\pi = \{1, 3, 2, 4\}$ resulting to a minimum maximum

absolute deviation of $C^* = 3$.

To further speed-up the procedure two extensions of basic DP are applied. The first extension employs a global upper bound, which is calculated upfront by some heuristic procedure(s), to decide whether an actual node can be fathomed or needs to be stored in the graph. Whenever a node weight w_{ti}^{mm} equals or exceeds upper bound \bar{C} the node can not be part of a solution with a better objective value than the incumbent upper bound and can, thus, be discarded. Such an extension of DP is also known as *Bounded Dynamic Programming*, which was introduced by Morin and Marsten (1976), Marsten and Morin (1978) and later on successfully applied, e.g., by Carraway and Schmidt (1991), Bautista et al. (1996) as well as Boysen et al. (2007b). Additionally, we apply a global lower bound to check whether optimality of the upper bound solution can be proven prior to constructing the graph. In Section 3.2 we showed that $\underline{C} = \frac{\max_{p \in P} |g_p - r|}{2}$ constitutes a possible lower bound for ALP_1 .

The second extension of our basic DP approach utilizes the symmetry of landing sequences. It can be shown (see appendix) that any partial sequence leads to the same maximum deviation as its reverted counterpart. Furthermore a unification of two subsequences π and π' with landing slots $t = 1, \dots, t'$ and $t = t' + 1, \dots, T$ filled with plane of sets $P^* \subset P$ and $P' = P \setminus P^*$, respectively, leads to maximum deviation that is equal to the maximum objective values of both subsequences. Consequently, the DP-graph merely needs to be constructed to its half, because for any node (t^m, i) in medium stage $t^m = \lceil \frac{T}{2} \rceil$ a complementary node $(t^m, i)^c$ of complementary stage $t^c = \lfloor \frac{T}{2} \rfloor$ covering all planes not in (t, i) has already been generated, so that both subsequences can be unified to a complete solution. Note that for an even slot number T medium stage t^m and complementary stage t^c are identical ($t^m = t^c = \frac{T}{2}$), whereas an odd T results to diverging stages with $t^m = t^c + 1$. For each node $i \in V_{t^m}$ the complementary node $(t^m, i)^c$ can be determined as follows:

$$(t^m, i)^c = \{j \in V_{t^c} : X_{t^m i p} + X_{t^c j p} = 1 \forall p \in P\} \quad (12)$$

Thus, the optimal objective value C^* amounts to:

$$C^* = \min_{i \in V_{t^m}} \left\{ \max \left\{ w_{(t^m, i)}^{mm}; w_{(t^m, i)^c}^{mm} \right\} \right\} \quad (13)$$

It further holds that whenever the complement of a node $i \in V_{t^m}$ has already been fathomed on the basis of an upper bound, it follows that also node (t^m, i) leads to an objective value higher than the upper bound and can be discarded.

Example (cont.): The potential of the aforementioned extensions of basic DP to reduce the graph is depicted in Figure 3 for our example. The graph merely needs to be constructed up to stage $t^m = 2$ with only 6 nodes remaining (instead of 16 with basic DP).

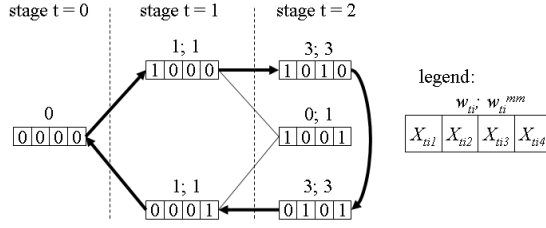


Figure 3: Example graph of DP with extensions for ALP_1

3.3.2 Heuristic start procedures

In spite of the considered extensions, the number of states in the DP approaches raises exponentially with the number of planes $|P|$, so that two heuristic start procedures (HSP) is developed to solve large ALP_1 -instances and/or to derive upper bounds. The first method, called HSP_1 simply fills the solution vector π of elements π_t ($t = 1, \dots, T$) from left to right by fixing an unscheduled plane $p \in OUT$ at the actual decision point t . Each myopic sequencing decisions aims at avoiding an increase of the maximum absolute deviation, which especially impends from those planes whose passenger number g_p deviates from target rate r , considerably. It seems desirable to minimize additional deviations caused by these planes, which is the easier the earlier these planes are scheduled. At the beginning of the sequence the degrees of freedom are higher to find preceding planes, which enable an efficient scheduling of high deviation planes. Thus, we consecutively determine target planes tp ordered by decreasing deviation from the ideal rate: $|g_p - r|$, which are prefixed by respective planes (determined by myopic choice) until an efficient sequence position for the actual target plane tp is found. A formal description of HSP_1 is as follows:

- (0) Initialize the following data: $OUT := P$; $t := 1$; $maxdev := 0$; $actdev := 0$
- (1) Determine the actual target plane tp , which is the one out of remaining planes $p \in OUT$ deviating most from target rate r :

$$tp := \operatorname{argmax}_{p \in OUT} |g_p - r| \quad (14)$$

- (2) If scheduling target plane tp does not exceed the actual maximum deviation: $|actdev + g_{tp} - r| \leq maxdev$, then select the target plane to be scheduled next: $sel := tp$ and go to step (5).
- (3) Select a preceding plane sel , which if scheduled at actual position t and target plane tp in position $t + 1$ causes least actual deviation:

$$sel := \operatorname{argmin}_{p \in OUT \setminus \{tp\}} \{\max\{|actdev + g_p - r|, |actdev + g_p + g_{tp} - 2r|\}\} \quad (15)$$

- (4) If the selected plane sel causes more deviation at the actual position than target plane tp : $|actdev + g_{sel} - r| > |actdev + g_{tp} - r|$, then select target plane: $sel := tp$.

(5) $OUT \setminus \{sel\}; \pi_t := sel; actdev := actdev + g_{sel} - r; maxdev := \max\{maxdev, |actdev|\};$
 $t := t + 1$

(6) If all planes are assigned then end the procedure, else proceed with step (1).

Example (cont.): For our example, the first target plane tp is the one carrying 10 passengers (deviation from target rate +4). The best preceding plane is the one with 5 passengers, so that scheduling both planes at the first two slots leaves behind an actual deviation of $actdev = 3$. The next target plane tp with 2 passengers can be directly scheduled without increasing maximum deviation, so that scheduling the remaining plane at the last position results to landing sequence $\pi = \{4, 2, 3, 1\}$, which is an optimal solution with $C = maxdev = 3$.

The second heuristic is based on a similar consideration, but more directly focuses on target planes. Note that according to the lower bound argumentation, the desired deviation before sequencing the target plane is exactly equal to $\frac{g_{tp} - r}{2}$. Any deviation from this value will result in an increased maximum deviation. Once a target plane has been identified, we could thus solve a special subset sum problem, which aims at identifying the subset of planes which comes as close as possible to this target deviation. Unfortunately the subset sum problem is well-known to be NP-hard, so that in the following heuristic HSP_2 , we will once again aim for a greedy solution. HSP_2 starts out the total set of planes identifies the target plane. The set is then subdivided into a set of predecessors, whose cumulated deviation is as close as possible to the target deviation and a set of successors which contains all remaining planes. This process is then repeated for all generated sets until the total set of planes has been divided into an ordered 1-partition, which provides the sequence.

(0) Initialize the following data: A list $L := \langle P \rangle$ containing the set of planes P , an empty list L^{New} ,

(1) For all $k = 1, \dots, K$ elements of list L do set $OUT := L_k$, if $|OUT| = 1$ then chose next k else set $PREV := \bigcup_{j=1}^k L_j$ and do the following

(2) Determine the target plane tp in OUT according to:

$$tp := \operatorname{argmax}_{p \in OUT} |g_p - r| \quad (16)$$

(3) Set $OUT := OUT \setminus \{tp\}$, $BEST := \emptyset$, $ACT := \emptyset$, $bestDiff := 0$ and calculate the targeted difference according to

$$tarDiff := - \left(\frac{g_{tp} - r}{2} \right) - \sum_{j \in PREV} g_j - r \quad (17)$$

(4) Retrieve the plane sel that comes as close as possible to the targeted difference

$$sel := \operatorname{argmin}_{p \in OUT \setminus ACT} \left\{ |tarDiff - g_p + r - \sum_{j \in ACT} g_j - r| \right\} \quad (18)$$

and add this plane to the current set $ACT := ACT \cup \{sel\}$

(5) If the current set ACT comes closest to the targeted difference $tarDiff - \sum_{j \in ACT} g_j - r < tarDiff - bestDiff$ then save the set as new best set: $BEST := ACT, bestDiff := \sum_{j \in ACT} g_j - r$

(6) If $|ACT| < |OUT|$ proceed with (3) else append the following sets to the new list in the fashion $L^{New} := \langle L^{New}, BEST, \{tp\}, OUT \setminus BEST \rangle$ while empty sets are ignored.

(7) If $k < K$ proceed with (1) else set $L := L^{New}$

(8) If list L contains $|P|$ elements then end procedure, else proceed with step (1).

The heuristic returns an ordered list of $|P|$ sets each containing one plane, which yields the required sequence.

3.4 Computational study

The computational study shall evaluate the maximum size of input data up to which the DP approach can be reasonably applied as well as the solution quality of the heuristic approaches HSP_1 and HSP_2 to solve ALP_1 . As there exists no established test bed for ALP_1 , we first elaborate on the generation of test instances:

As input parameter for instance generation, we vary the number of planes given: $|P| \in \{5, 6, \dots, 22\}$. Per instance, each plane p in set P receives its number of passengers g_p by randomly drawing an equally distributed integer number out of the interval $[1; 1000]$. For each set size $|P|$ this procedure is repeated 20 times, so that in total 360 test instances are derived.

The results of the computational study are listed in Table 2. For the optimal DP approach we report CPU-seconds (avg cpu) averaged over all 20 instances with equal number of planes $|P|$. Results of the heuristic HSP approaches consist of the average gap (avg gap) and maximum gap (max gap) from the optimum, where each single deviation is defined by: $\frac{C(HSP) - C(DP)}{C(DP)}$, and $C(HSP)$ ($C(DP)$) denotes the objective value of the HSP approach (DP). The minimum objective value ($\min\{HSP_1; HSP_2\}$) of both HSP approaches is utilized as the upper bound solution for DP. All methods have been implemented in C# (Visual Studio 2003) and run on a Pentium IV, 1800 MHz PC, with 512 MB of memory.

The results show that DP solves all 360 instances to optimality within a given time frame of 300 CPU-seconds with an average of only 13.4 CPU-seconds. $|P| = 22$ can be seen as an upper limit for reasonably applying DP, as with 23 planes no instance can be solved within the given time frame. On the other hand, the pure DP approach without the aforementioned extensions, i.e., bounded dynamic programming and graph reduction, can only solve instances with up to 18 planes within 300 CPU-seconds. Our heuristic HSP approaches perform satisfactorily as HSP_1 and HSP_2 result in an average (maximum) gap of 8.0% (42.5%) and 5.5% (38.8%), respectively. Interestingly, the performances of both heuristics are supplementary in the sense that the average gap can be

$ P $	DP	HSP_1		HSP_2		$\min\{HSP_1; HSP_2\}$	
	avg cpu	avg gap	max gap	avg gap	max gap	avg gap	max gap
5	<0.1	10.3	39.7	0.0	0.0	0.0	0.0
6	<0.1	3.4	15.0	0.1	0.8	0.1	0.8
7	<0.1	9.1	27.1	3.6	27.1	3.6	27.1
8	<0.1	5.9	15.9	1.1	4.1	1.1	4.1
9	<0.1	11.0	42.5	5.2	14.9	4.7	10.1
10	<0.1	5.4	15.0	3.6	15.0	3.6	15.0
11	<0.1	8.3	29.0	5.5	17.1	4.2	17.1
12	<0.1	7.8	24.1	6.1	29.1	3.4	10.2
13	<0.1	10.0	21.3	3.8	16.7	3.8	16.7
14	0.1	9.3	21.4	5.5	20.5	5.0	16.0
15	0.1	11.9	34.5	7.9	29.8	5.1	25.4
16	0.3	7.5	18.2	10.7	38.8	5.8	18.2
17	0.9	6.8	17.5	8.7	33.5	3.9	7.9
18	1.5	7.2	20.4	5.7	24.1	3.3	20.4
19	7.2	7.0	20.3	6.0	25.9	2.4	6.8
20	14.7	9.4	22.6	8.5	35.0	5.3	14.2
21	55.5	9.6	23.6	11.7	28.4	7.7	20.0
22	161.0	4.2	10.6	6.1	38.5	2.2	7.5
total	13.4	8.0	42.5	5.5	38.8	3.6	27.1

Table 2: Results for ALP_1

decreased to merely 3.6% by choosing the better heuristic objective value per instance ($\min\{HSP_1; HSP_2\}$). Moreover, the computational time is near to nothing for both HSP approaches as it falls below 0.1 CPU-seconds for any instance.

4 Balancing of landings per airline

4.1 Mathematical model

To balance the workload of airline ground staff, the passengers of the respective airline need to be evenly distributed over the planning horizon. Assumed that all planes of an airline are approximately of the same size and the number of carried passengers is comparable, it is sufficient to level the landings per airline over time. The individual target rate r_a per airline a is hence $r_a = \frac{|P_a|}{T} \forall a \in A$. The following mathematical program ALP_2 with objective function (19) and constraints (1)-(3) and (20) hence minimizes target rate deviations:

$$ALP_2: \text{Minimize } Z(X, Y) = \max_{t=1, \dots, T; a \in A} |y_{at} - t \cdot r_a| \quad (19)$$

subject to (1)-(3) and

$$y_{at} = \sum_{\tau=1}^t \sum_{p \in P_a} x_{p\tau} \quad \forall a \in A; t = 1, \dots, T \quad (20)$$

The auxiliary integer variables y_{at} in equations (20) denote the number of landings per airline a up to slot t . The maximum absolute difference between the actual number of landings per airline and the ideal number $(t \cdot r_a)$ per slot t and airline a is minimized by objective function (19).

4.2 Solution Algorithm

ALP_2 can be shown to be equivalent to the well known product rate variation problem (PRV), which deals with evenly spreading the copies of different models over the production cycles of a mixed-model assembly line. This problem was introduced by Miltenburg (1989), further prominent contributions stem from Inman and Bulfin (1991), Kubiak and Sethi (1991, 1994) as well as Steiner and Yeomans (1993) and detailed reviews can be found at Kubiak (1993) and Boysen et al. (2007a). The equivalence between the PRV and ALP_2 becomes immediately obvious by exchanging the following terms: Instead of evenly distributing the landings per airline over landing slots the PRV aims at leveling the copies of models to be produced over production cycles, so that the terms “planes” are to be replaced by “copies”, “airlines” by “models” and “landing slots” by “production cycles”. For the PRV with min-sum objective and absolute as well as squared deviations Kubiak and Sethi (1991, 1994) introduce an exact solution procedure, which is based on a transformation to a linear assignment problem, with a runtime complexity of $O(T^3)$. For the min-max objective with absolute deviations, which is also investigated within the paper on hand, Steiner and Yeomans (1993) develop an exact procedure with runtime complexity $O(T \log T)$. Note that the runtime of both procedures for the PRV is only polynomially bounded in the number of slots T , which is by itself only pseudo-polynomially bounded in the length of a reasonably encoded input. Consequently, just as for many high-multiplicity problems (see Grigoriev and van de Klundert, 2006) the complexity of ALP_2 remains open, as it is unclear whether it belongs to NP.

The Steiner and Yeomans procedure decomposes the overall problem into a set of feasibility problems, each of which answering the question of whether there exists a landing sequence whose maximum deviation does not exceed a given maximum deviation level D . With a given D , for the i -th plane (with $i = 1, \dots, |P_a|$) of airline a the set of feasible landing slots T_{ai} can be determined by:

$$T_{ai} = \{t = 1, \dots, T \mid |i - t \cdot r_a| \leq D \text{ and } |i - 1 - (t - 1) \cdot r_a| \leq D\} \quad \forall a \in A; i = 1, \dots, |P_a| \quad (21)$$

To take up slot t in set T_{ai} the i -th plane scheduled in slot t may not exceed the given maximum deviation D (first term of the condition) furthermore the postponement of this plane must be possible without causing an infeasible deviation of the preceding plane $i - 1$ in the preceding slot $t - 1$ (second term of the condition). If these sets T_{ai} are determined the feasibility problem reduces to a perfect matching problem in a bipartite graph (see Hopcraft and Karp, 1973). The two node sets are represented by the planes $i = 1, \dots, |P_a|$ of the airlines $a \in A$ and the landing slots $t = 1, \dots, T$, respectively. A

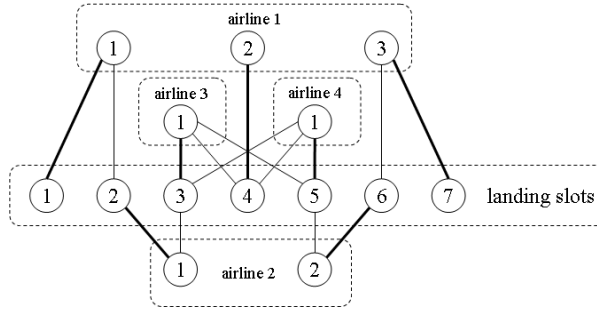


Figure 4: Perfect matching to solve the feasibility version of ALP_2

node representing plane i of airline a is connected to a landing slot t with an arc, if $t \in T_{ai}$ holds. If a perfect matching exists, the respective feasibility problem is a yes-instance and the edges chosen in the perfect matching represent the landing sequence of airlines. Furthermore, Steiner and Yeomans (1993) prove the so called *order preserving* property, which means that in a perfect matching the i -th plane can never be scheduled in a period ahead of plane $i - 1$, which indeed proves equivalence between the matching problem and the feasibility version of ALP_2 .

By deriving upper and lower bounds for the optimization version of the problem, Steiner and Yeomans show that it is sufficient to restrict the search for the optimal deviation level D^* to the following set D^D :

$$D^D = \{D \in \mathbb{R} : 1 - \max_{a \in A} \{r_a\} \leq D \leq 1 \wedge D \cdot T \in \mathbb{Z}^+\} \quad (22)$$

Closer bounds are introduced by Kubiak (2004). The lowest $D \in D^D$ for which the feasibility problem holds true is the minimum maximum deviation D^* and the optimal objective value of ALP_2 .

Example: Given $|P| = 4$ airlines ($P = \{1, 2, 3, 4\}$) with 3, 2, 1 and 1 planes to be landed, respectively. The resulting bipartite graph for a given maximum deviation of $D = 4/7$ is depicted in Figure 4 along with a bold faced perfect matching. The corresponding succession of airlines is $\pi = \{1, 2, 3, 1, 4, 2, 1\}$. The set of deviations D^D , for which a matching problem is to be solved is $D^D = \{4/7, 5/7, 6/7, 1\}$. As there exists a perfect matching for the minimum maximum deviation of $D = 4/7$ it is also the optimal solution value $D^* = 4/7$.

5 Balancing the number of landed passengers per airline

5.1 Mathematical model

Finally, we aim at leveling the workload of airline staff if the planes per airline carry different numbers of passengers. In such a setting the individual target rate r_a per airline a amounts to: $r_a = \frac{\sum_{p \in P_a} g_p}{T} \forall a \in A$. The resulting mathematical program ALP_3

consists of objective function (23) and is subject to constraints (1)-(3) and (24):

$$\begin{aligned} ALP_3: \text{ Minimize } Z(X, Y) &= \max_{t=1, \dots, T; a \in A} |y_{at} - t \cdot r_a| & (23) \\ &\text{subject to (1)-(3) and} \end{aligned}$$

$$y_{at} = \sum_{\tau=1}^t \sum_{p \in P_a} x_{p\tau} \cdot g_p \quad \forall a \in A; t = 1, \dots, T \quad (24)$$

The auxiliary variable y_{at} of equations (24) denote the cumulated number of passengers of planes of airline a assigned up to slot t . This airline specific number of passengers should approximate the ideal number of passengers per airline ($t \cdot r_a$) as is expressed in objective function (23).

ALP_3 can be seen as a special version of a multi-output rate variation problem. Since ALP_3 further contains ALP_1 as a special case, the problem is likewise NP-hard in the strong sense.

5.2 Solution Algorithms

To solve ALP_3 , we first show how to adopt the Dynamic Programming (DP) approach of ALP_1 and then discuss some heuristic procedures, which are based on a decomposition of ALP_3 into the both versions of ALP (ALP_1 and ALP_2) previously considered.

5.2.1 Dynamic Programming approach

To solve ALP_3 to optimality, the DP approach developed for ALP_1 can be applied nearly unmodified. The structure of the graph remains identical, only the weighting function (10) to calculate the resulting deviation for each node (t, i) needs to be adopted:

$$w_{ti} = \max_{a \in A} \left| \sum_{p \in P_a} X_{tip} \cdot g_p - t \cdot r_a \right| \quad \forall t = 0, \dots, T; i \in V_t \quad (25)$$

Furthermore, both extension of basic DP designed for ALP_1 can be applied to ALP_3 , as well. An upper bound can be derived, e.g. by the heuristic approaches described in Section 5.2.2, which serves as a standard of comparison to exclude nodes with higher weights from the graph. since the symmetry property proven for ALP_1 , likewise holds for ALP_3 , the same principles for a graph reduction can be applied.

Example: Given are $|A| = 2$ airlines with $|P_a| = 3$ planes each to be landed. The number of passengers on the planes 1, 2 and 3 of airline $a = 1$ are 3, 8 and 7, respectively. Planes 4, 5 and 6 belong to airline $a = 2$ with passenger numbers 3, 2 and 7, respectively. For these given input data the target rates result to $r_1 = 3$ and $r_2 = 2$. The belonging DP-graph is depicted in Figure 5. An upper bound value of $UB = 4$ is assumed, which is

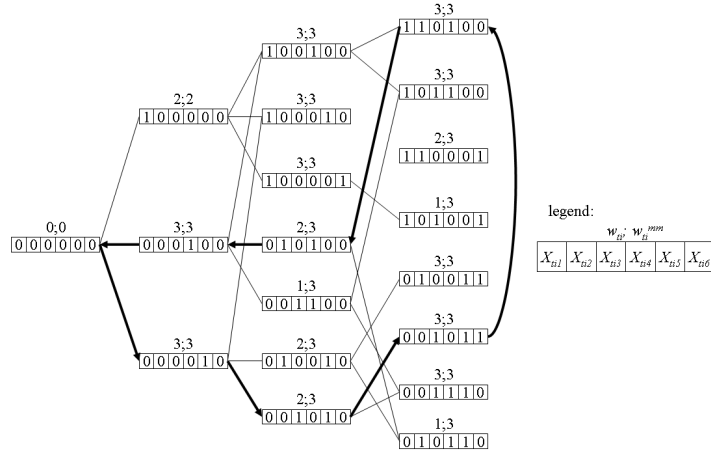


Figure 5: Example graph of DP with extensions for ALP_3

applied to reduce the number of nodes to be constructed. A bold faced optimal solution with objective value $C^* = 3$ represents the landing sequence $\pi = \{5, 3, 6, 1, 2, 4\}$.

5.2.2 Decomposition approach

To derive a heuristic start procedure, ALP_3 can be decomposed into $|A|$ ALP_1 problems to determine the sequence of planes for each airline separately (first step) and a single ALP_2 problem to assign the landing slots to airlines (second step). For both steps two slightly different alternatives exist, which are described in the following:

Step 1: For each airline a separate sequence of planes is determined by solving $|A|$ ALP_1 problems with the heuristic start procedures HSP introduced in Section 3.3.2. Note that determining exact solutions for the ALP_1 problems, for instance by our DP approach, is no serious alternative, as $|A|$ NP-hard problems need to be solved. However, there are two alternatives of how to derive the respective input data for ALP_1 from a given ALP_3 instance:

- An ALP_1 instance for any airline $a \in A$ can be extracted by considering exclusively the planes P_a of the respective airline. Consequently, the number of possible landing slots T is reduced to the number $|P_a|$ of actual planes. This advancement, which we label as *time-reduced* (abbreviated by ALP_1^{TR}), requires the following preparation of data: $P(a) := P_a$, $T(a) := |P_a|$ and $r(a) := \frac{\sum_{p \in P_a} g_p}{|P_a|}$.

Example (cont.): The target rates for the two time-reduced ALP_1^{TR} instances result to $r(1) = 6$ (airline 1) and $r(2) = 4$ (airline 2), respectively. If the HSP_2 -heuristic is applied the resulting landing sequences are $\pi(1) = \{2, 1, 3\}$ (airline 1) and $\pi(2) = \{4, 6, 5\}$ (airline 2).

- As can be shown by example, an optimal solution of ALP_1^{TR} is not order-preserving, if empty slots, i.e., further landings of other airlines, are inserted. This means that

an insertion of empty planes (with $g_p = 0$) may alter the original optimal landing sequence of planes (with $g_p > 0$) within the modified instance. Note that the optimal solution is only order-preserving when adding the neutral element with $g_p = r$ (see Section 3.2). Thus, there might be a considerable difference between optimal landing sequences of two ALP_1 instances, which either operate on a reduced planning horizon (see alternative 1 above) or preserve the number of landing slots of the original ALP_3 instance. The latter alternative is labeled as *time-preserving* (abbreviated by ALP_1^{TP}) and bases on the following preparation of ALP_3 input data for any airline a : any input data of ALP_3 is transferred to ALP_1 , except for passenger numbers of all planes not belonging to airline a , which are overwritten by zero, so that $g_p = 0 \forall p \notin P_a$.

Example (cont.): The target rates of both time-preserving ALP_1^{TP} instances are $r(1) = 3$ (airline 1) and $r(2) = 2$ (airline 2), respectively. For airline 1, the solution obtained by HSP_2 is $\{0, 2, 0, 3, 0, 1\}$, which is to be reduced to the respective landing sequence $\pi(1)$ of airline 1: $\pi(1) = \{2, 3, 1\}$. The ALP_1 solution for airline 2 amounts to $\{0, 6, 0, 0, 4, 5\}$ and a landing sequence $\pi(2) = \{6, 4, 5\}$.

Step 2: Furthermore, there are also two alternatives of how to solve the remaining problem, which allocates landing slots to airlines:

- The problem can be solved *independent* of the airlines' landing sequences obtained first step, which means that an original ALP_2 problem is extracted from an ALP_3 instance. This requires the preparation of the following input data: $A := A$, $T := T$ and $r_a := \frac{|P_a|}{T} \forall a \in A$. This advancement is labeled as ALP_2^I .

Example (cont.): An independent ALP_2 instance solved to optimality with the matching based approach of Steiner and Yeomans (1993) (see Section 4.2) leads to a succession of airlines 1 and 2 of $\pi = \{1, 2, 1, 2, 1, 2\}$.

- The remaining problem of slot allocation can also be solved by considering the determined landing sequences of any airline a (denoted by vectors π_a with elements π_{ai} storing the i -th plane landing of airline a) as additional input data. This problem can be solved by an adoption of the matching based procedure of Steiner and Yeomans (1993) for ALP_2 described in Section 4.2:

Again, the procedure bases on a decomposition into a set of feasibility problems with diverging maximum deviations D given. The bipartite graph, for which it is to be decided of whether a perfect matching exists, contains of two node sets on the one hand representing the planes in the given sequence π_a for any airline a and on the other hand the landing slots. The set of edges connecting both node sets can be determined with the help of a modified formula (21), which derives the sets T_{ai} of feasible landing slots for the i -th plane in sequence π_a of airline a :

$$T_{ai} = \left\{ t = 1, \dots, T : \left| \sum_{\tau=1}^i g_{\pi_{a\tau}} - t \cdot r_a \right| \leq D \text{ and } \left| \sum_{\tau=1}^{i-1} g_{\pi_{a\tau}} - (t-1) \cdot r_a \right| \leq D \right\} \\ \forall a \in A; i = 1, \dots, |P_a| \quad (26)$$

Whenever there exists a perfect matching for the aforementioned graph, the respective feasibility problem is a yes-instance, so that there exists an assignment of slots to airlines in the given landing sequence π_a per airline a not exceeding given maximum deviation D . Furthermore, we need to adopt the calculation of lower and upper bounds of possible deviation levels. Lower bound \underline{D} is set to the minimum deviation possible, when fixing the first plane:

$$\underline{D} = \min_{a \in A} \left\{ \max\{|g_{\pi_{a1}} - r_a|; \max_{a' \in A \setminus \{a\}} \{r_{a'}\}\} \right\} \quad (27)$$

The actual deviation at slot $t = 1$, when assigning the first plane π_{a1} out of the given landing sequence of airline a amounts to the maximum of the deviations directly caused by scheduling the first plane of airline a and the deviations arising by not scheduling a plane of the other airlines $a' \in A \setminus \{a\}$. The minimum over all airlines $a \in A$ determines lower bound \underline{D} .

On the other hand, upper bound \overline{D} bases on the consideration that in the worst case all planes $p \in P_a^+$ of an airline a , whose passenger number g_p exceed target rate r_a , follow directly one after another alternated by all other planes with $g_p < r_a$. The plane sets P_a^+ are defined as follows: $P_a^+ = \{p \in P_a | g_p > r_a\} \forall a \in A$. In this worst-case scheduling pattern the maximum deviation always occurs in the $|P_a^+|$ -th position irrespective of the actual sequence of the planes in P_a^+ , so that upper bound \overline{D} can be determined as follows:

$$\overline{D} = \max_{a \in A} \left\{ \sum_{p \in P_a^+} g_p - |P_a^+| \cdot r_a \right\} \quad (28)$$

Within interval $[\underline{D}; \overline{D}]$ a binary search is performed and for each actual maximum deviation D a perfect matching problem is solved. The procedure is aborted, when the difference between actual upper bound (for which no perfect matching exists) and the actual lower bound (for which a perfect matching exists) of the binary search procedure falls below a predefined parameter V , for which we choose $V = 0.1$. This procedure solves ALP_2 in *dependence* of the landing sequences determined in the first step of the decomposition approach in is, thus, abbreviated by ALP_2^D .

Example (cont.): If the airline specific succession of planes obtained by ALP_1^{TR} are passed over to the second step, ALP_2^D determines the following succession of airlines $\pi = \{2, 1, 1, 2, 1, 2\}$.

Finally, the results of both algorithmic steps of the decomposition approach must be unified to the overall landing sequence π , which is to be evaluated with the objective function (23) of ALP_3 to determine the respective objective value. Our decomposition approach can be executed in four different modes: $ALP_1^{TR} + ALP_2^I$, $ALP_1^{TR} + ALP_2^D$, $ALP_1^{TP} + ALP_2^I$ and $ALP_1^{TP} + ALP_2^D$. All four settings are to be evaluated with regard to their solution performance by the following computational study.

decomposition approach	sequence	objective value
$ALP_1^{TR} + ALP_2^I$	{2, 4, 1, 6, 3, 5}	5
$ALP_1^{TR} + ALP_2^D$	{4, 2, 1, 6, 3, 5}	3
$ALP_1^{TP} + ALP_2^I$	{2, 6, 3, 4, 1, 5}	6
$ALP_1^{TP} + ALP_2^D$	{2, 6, 4, 3, 1, 5}	5

Table 3: Example for ALP_3 - results of the decomposition approach

Example (cont.): Table 3 summarizes the resulting landing sequences and the belonging objective values for all four settings of our decomposition approach.

5.3 Computational study

To generate test instances, we varied the number of planes $|P| = \{5, 6, \dots, 22\}$ as an input parameter to determine the number of airlines $|A|$, the number of passengers g_p per plane p and the assignment of planes to airlines as output, altogether defining an ALP_3 instance. For each different plane number $|P|$ given, the procedure is repeated 20 times so that in total 360 instances emerge. For a given $|P|$, instance generation iterates through the following steps: First, the number $|A|$ of airlines is determined by randomly drawing an integer number out of interval $[2; \frac{1}{2}|P|]$. Then, by randomly drawing an integer number per plane p out of interval $[1; 1000]$ the number of passenger g_p is set. Finally, we assign planes to airlines by randomly determining the airline number per plane. For all random numbers equal distribution is chosen.

The results are listed in Table 4. We only report the results of DP and both ALP_2^D based settings of the decomposition approach, because an allocation of slots to airlines independent (ALP_2^I) of the results obtained by the first step of the decomposition approach shows not competitive. Their average relative deviation over all 360 test instances amounts to 78.9% ($ALP_1^{TP} + ALP_2^I$) and 110.3% ($ALP_1^{TR} + ALP_2^I$), respectively. Thus, results are reported for DP, $ALP_1^{TR} + ALP_2^D$, $ALP_1^{TP} + ALP_2^D$ and the minimum objective value obtained by both settings of the decomposition approach ($\min\{(1); (2)\}$). The employed evaluation criteria are described in Section 3.4.

Just like for ALP_1 , the DP approach can solve all instances up to $|P| = 22$ within a given time-frame of 300 CPU-seconds. In total the DP approach performs slightly worse for ALP_3 than for APL_1 , which is explained by the less tight upper bounds for this problem. We apply the minimum objective value of the decomposition approaches as the upper bound for DP ($\min\{(1); (2)\}$). Both settings of the heuristic decomposition approach perform satisfactorily with an average gap of 11.8% ($ALP_1^{TR} + ALP_2^D$) and 18.2% ($ALP_1^{TP} + ALP_2^D$). When applying the minimum objective value of both heuristic approaches ($\min\{(1); (2)\}$) per instance the average gap decreases to 6.3%, which can be interpreted as a promising result the more so as both heuristic approaches require negligible computational time (< 0.1 CPU-seconds in any instance).

P	DP avg cpu	(1): $ALP_1^{TR} + ALP_2^D$		(2): $ALP_1^{TP} + ALP_2^D$		min{(1); (2)}	
		avg gap	max gap	avg gap	max gap	avg gap	max gap
5	<0.1	23.7	73.5	2.3	23.4	2.3	23.4
6	<0.1	8.5	23.8	5.6	53.9	0.2	1.6
7	<0.1	16.4	43.7	10.5	66.7	7.1	31.9
8	<0.1	7.0	28.3	9.4	58.9	2.1	17.5
9	<0.1	20.3	52.2	18.8	53.6	11.2	52.2
10	<0.1	15.2	37.6	12.6	51.8	5.9	30.9
11	<0.1	11.1	59.8	12.4	37.9	5.0	18.0
12	<0.1	9.3	30.5	10.3	53.8	4.7	25.9
13	<0.1	9.8	35.2	21.4	83.2	6.0	35.2
14	0.1	9.4	39.3	27.4	122.2	8.8	39.3
15	0.2	6.4	27.1	14.6	38.4	6.0	27.1
16	0.4	14.3	42.3	29.6	116.3	10.8	40.6
17	1.2	12.2	56.6	31.6	163.0	11.1	56.6
18	2.0	12.7	40.2	38.6	115.5	11.4	40.2
19	10.4	6.7	18.9	20.2	70.4	6.3	18.9
20	19.1	15.1	72.0	42.5	216.6	6.0	18.8
21	70.6	8.7	64.5	8.5	50.8	6.4	50.8
22	202.0	5.1	12.5	10.5	48.5	2.6	10.0
total	17.0	11.8	73.5	18.2	216.6	6.3	56.6

Table 4: Results for ALP_3

6 Conclusions

The paper on hand investigates a novel class of objective functions for ALP, which aims at balancing the workload of ground staff at airports. Three different objective functions are considered, for each of which complexity results and exact as well as heuristic solution procedures are presented. However, the findings rest on a set of simplifying assumptions which limit a direct application to real-world settings. This is discussed in more detail in the following:

- In real-world landing problems, planes are bound to earliest and latest landing times. Earliest landing times e_p can be calculated on the basis of the remaining distance to be covered by a plane p and its maximum velocity. Latest landing times l_p are determined on the basis of the fuel level and the most fuel-efficient speed the plane can take while circling in the airspace of the airport. Earliest and latest landing times can be easily incorporated in the models for ALP_1 to ALP_3 by adding the following constraint:

$$e_p \leq \sum_{t=1, \dots, T} x_{pt} \cdot t \leq l_p \quad \forall p \in P \quad (29)$$

To solve the resulting problems, our exact DP approach can be easily extended. As each arc represents a scheduling of a plane at a specific landing slots, all arcs,

which would result in an untimely landing, merely need to be excluded from the graph with the eligible side effect of reducing the graph structure and accelerating the solution process.

- Furthermore, the controllers typically need to consider minimum separation times between consecutive aircrafts to account for air turbulences and ensure safe landings. These separation times are sequence-dependent as the extent of evoked (by predecessors) and tolerable (by successors) turbulence depends on the dimensions of the respective planes. In such a setting, ALP is no longer a sequencing problem but becomes a scheduling problem, where the length T of the schedule depends on the succession of planes. Consequently, the target rates r can not be calculated prior to determining a solution. This complicates the solution process as it is impossible to exactly quantify the contribution of solution parts to the objective value, so that the consequences of single scheduling decisions while constructing a solution can not be exactly valuated. Thus, it is a challenging task for future research to combine leveling objectives with sequence-dependent separation times.

As was presented in Section 1, there are other very important objectives to be regarded in ALP. Thus, especially multi-objective optimization approaches seem an important contribution of future research, in which the leveling objectives presented within the paper on hand should be a valuable component to facilitate efficient airport operations.

Appendix

Symmetry of Solution Sequences

In the following we will investigate the symmetry of obtained sequences and point out consequences with regard to employed solution methods. The results are shown to hold for ALP_1 , but can be easily shown to hold for ALP_3 with just minor modifications. We will differentiate between “complete” sequences which have a length of T and represent an ordered assignment of all planes in P and “partial” sequences of length $t^* < T$, which only contain a subset of planes $P^* \subset P$. For simplicity, we will denote all sequences with the same symbol π . We can now redefine d_t as the deviation at slot t of sequence π recursively as

$$d_t = d_{t-1} + g_{\pi_t} - r \quad \text{for } t = 1, \dots, T \quad (30)$$

so that the deviation at a slot t is the result of the deviation at the previous slot increased by the difference between the passengers of the plane at slot t and the target rate r . The equation requires the determination of a starting deviation d_0 , which will be useful in the following argumentation. Note, however, that a starting deviation can also be of practical relevance whenever the model is employed as part of a sequential rolling horizons approach.

In the proposed model formulation of Section (3.1) it is assumed that the starting deviation is zero, so that the objective of ALP_1 is equal to minimizing $\max_{t=0, \dots, T} |d_t|$

with $d_0 = 0$. We can now make some observations regarding the symmetry of solutions. Note that due to the structure of the model formulation, it holds for any complete sequence π that the observed deviation is zero at the last slot

$$d_T = d_0 + \sum_{\tau=1}^T g_{\pi_\tau} - r \cdot T = \sum_{p \in P} g_p - \frac{\sum_{p \in P} g_p}{T} \cdot T = 0 \quad (31)$$

which follows directly from the determination of the target rate r . Now, consider two complete sequences π and π' , where π' is the inverted sequence of π , so that $\pi_t = \pi'_{T-t+1}$ for $t = 1, \dots, T$. It holds that

$$-d_t = d'_{T-t} \quad \text{for } t = 0, \dots, T \quad (32)$$

where d'_t is the deviation of sequence π' at slot t . By considering (31) equation (32) certainly holds for $t = 0$ and $t = T$ with $d_0 = d'_0 = 0$, but it also holds for all other cases as is easily shown by insertion

$$\begin{aligned} d'_{T-t} &= d'_0 + \sum_{\tau=1}^{T-t} g_{\pi'_\tau} - r \cdot (T-t) \\ &= \sum_{p \in P} g_p - \sum_{\tau=1}^t g_{\pi_\tau} - r \cdot T + r \cdot t \\ &= -\sum_{\tau=1}^t g_{\pi_\tau} + r \cdot t \\ &= -d_t \end{aligned}$$

It directly follows from (32) that $|d_t| = |d'_{T-t}|$ for $t = 0, \dots, T$, so that also $\max_{t=0, \dots, T} |d_t| = \max_{t=0, \dots, T} |d'_t|$, which means that both sequences π and π' have the same objective value. An interesting consequence of (32) is thus, that there are always at least two optimal solution sequences for non-trivial problems. Once an optimal solution has been found, a second optimal solution can be readily generated by simply inverting the corresponding sequence.

The symmetry is, however, also valid for partial sequences as we will show in the following. Consider a partial sequence π with length t^* to which only a subset of planes $P^* \subset P$ has been assigned. Irrespective of the order of planes in the sequence, the deviation at the last slot will amount to $d_{t^*} = \sum_{p \in P^*} g_p - r \cdot t^*$ which is not necessarily zero. Based on (32) we can conclude for such a partial sequence π with $d_0 = 0$ and its reverted counterpart π' that

$$-d_t = d'_{t^*-t} \quad \text{for } t = 0, \dots, t^* \quad (33)$$

with $d'_0 = r \cdot t^* - \sum_{p \in P^*} g_p$ for π' . Again the relationship obviously holds for $t = 0$ and $t = t^*$, but likewise holds also for all other slots as

$$\begin{aligned}
d'_{t^*-t} &= d'_0 + \sum_{\tau=1}^{t^*-t} g_{\pi'_\tau} - r \cdot (t^* - t) \\
&= r \cdot t^* - \sum_{p \in P^*} g_p + \sum_{p \in P^*} g_p - \sum_{\tau=1}^t g_{\pi_\tau} - r \cdot t^* + r \cdot t \\
&= - \sum_{\tau=1}^t g_{\pi_\tau} + r \cdot t \\
&= -d_t
\end{aligned}$$

which in turn means that both partial sequences will yield the same objective values. We can thus conclude that a partial sequence which is optimal for a given subset of planes and a starting deviation of zero, can be inverted to yield the optimal sequence for the corresponding problem of optimally sequencing the same subset with a starting deviation of $r \cdot t^* - \sum_{p \in P^*} g_p$.

This insight has direct consequences for the determination of optimal sequences as part of the dynamic programming approach. Each node of the presented graph yields the optimal partial sequence π for a particular subset of planes P^* with a length t^* and a deviation at this slot of $d_{t^*} = \sum_{p \in P^*} g_p - r \cdot t^*$. In order to obtain a solution, the remaining subproblem constitutes in constructing an optimal partial sequence π' of length $T - t^*$ of the remaining planes $P \setminus P^*$ with a starting deviation of $d_0 = \sum_{p \in P^*} g_p - r \cdot t^*$. Once this partial sequence π' is found it can be appended to π to yield a solution. Obviously, this solution is not necessarily optimal for the overall problem, since there is no compelling reason why sequencing the subset of P^* planes first should result to an optimal overall sequence. As the dynamic program, however, ensures that all subsets of planes are considered at every stage, the best determined sequence over all pairs needs to be optimal.

Now, consider a problem with an even number of planes T^E . We pick a random node N at stage $t^* = T^E/2$ of the graph with a sequence π of planes P^* a deviation at slot $d_{T^E/2} = \sum_{p \in P^*} g_p - r \cdot T^E/2$ and an objective value of C_π . At the same stage of the graph we can find a corresponding node N' with an optimal partial sequence π' of planes $P \setminus P^*$ with an objective value of $C_{\pi'}$. This sequence has a deviation of $d'_{T^E/2} = \sum_{p \in P \setminus P^*} g_p - r \cdot T^E/2$. Due to (33) sequence π' can be reverted to yield the optimal partial sequence for the corresponding subproblem of sequencing the same planes with a starting deviation of $-d'_{T^E/2}$. We can use this information to determine the optimal objective value of any solution which starts with a partial sequence of π . In fact it holds that $-d'_{T^E/2} = d_{T^E/2}$ as is easily shown by insertion:

$$\begin{aligned}
d_{T^E/2}^l &= \sum_{p \in P \setminus P^*} g_p - r \cdot T^E/2 \\
&= \sum_{p \in P} g_p - \sum_{p \in P^*} g_p - r \cdot T^E/2 \\
&= \sum_{p \in P} g_p - \sum_{p \in P^*} g_p - \frac{\sum_{p \in P} g_p}{T^E} \cdot T^E/2 \\
&= - \sum_{p \in P^*} g_p + \frac{\sum_{p \in P} g_p}{2} \\
&= - \sum_{p \in P^*} g_p + r \cdot T^E/2 \\
&= -d_{T^E/2}
\end{aligned}$$

It follows that the inversion of pi' directly yields the optimal solution to the subproblem of node N . Due to the symmetry, the inversion of π likewise yields the optimal solution of the subproblem of N' . N and N' can thus be seen as partner nodes, since their partial sequences can be combined to yield two solutions of the same objective value. Since both partial sequences are optimal regarding their corresponding subproblems, the best objective value of any possible solution which begins with a partial sequence of π or π' , respectively, is simply determined by $Max\{C_\pi, C_{\pi'}\}$. Instead of continuing the generation of nodes, we can abort the procedure at this stage and determine the solution values by finding the corresponding partner node to each node N . The optimal solution for the overall problem is hence retrieved by finding the combination of partner nodes, which yields the lowest objective value.

Elaborated search techniques, such as hash tables, allow an inspection of a vast amount of nodes in short computation times. Over more, if a corresponding partner node has already been fathomed on the basis of a lower bound and can thus not be found in the stage, the node on hand can be discarded as well.

If the number of planes is uneven T^U , the graph can be constructed until stage $1 + \lfloor T^U/2 \rfloor$ and the partner nodes are found in stage $\lfloor T^U/2 \rfloor$. With the help of this method the effort for node construction can be considerably reduced which should speed up the procedure significantly.

References

- [1] Atkin, J.A.D., Burke, E.K., Greenwood, J.S., Reeson, D., 2007. Hybrid metaheuristics to aid runway scheduling at London Heathrow airport, *Transportation Science* 41, 90–106.
- [2] Bautista, J., Companys, R., Corominas, A., 1996. Heuristics and exact algorithms for solving the Monden problem, *European Journal of Operational Research* 88, 101–113.

- [3] Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D., 2000. Scheduling aircraft landings – the static case, *Transportation Science* 34, 180–197.
- [4] Beasley, J.E., Sonander, J., Havelock, P., 2001. Scheduling aircraft landings at London Heathrow using a population heuristic, *Journal of the Operational Research Society* 52, 483–493.
- [5] Beasley, J.E., Krishnamoorthy, M., Sharaiha, Y.M., Abramson, D., 2004. Displacement problem and dynamically scheduling aircraft landings, *Journal of the Operational Research Society* 55, 54–64.
- [6] Bianco, L., Dell’Olmo, P., Giordani, S., 1999. Minimizing total completion time subject to release dates and sequence-dependent processing times, *Annals of Operations Research* 86, 393–415.
- [7] Bianco, L., Dell’Olmo, P., Giordani, S., 2006. Scheduling models for air traffic control in terminal areas, *Journal of Scheduling* 9, 223–253.
- [8] Boysen, N., Fliedner, M., Scholl, A., 2007a. Sequencing mixed-model assembly lines to minimize part inventory cost, *European Journal of Operational Research* (to appear).
- [9] Boysen, N., Fliedner, M., Scholl, A., 2007b. Sequencing mixed-model assembly lines to minimize part inventory cost, *OR Spectrum* (to appear).
- [10] Carraway, R.L., Schmidt, R.L., 1991. An improved discrete dynamic programming algorithm for allocating resources among interdependent projects, *Management Science* 37, 1195–1200.
- [11] Ernst, A.T., Krishnamoorthy, M., Storer, R.H., 1999. Heuristic and exact algorithms for scheduling aircraft landings, *Networks* 34, 229–241.
- [12] Garey, M.R., Johnson, D.S., 1979. *Computers and intractability: A guide to the theory of NP-completeness*, Freeman, New York.
- [13] Grigoriev, A., van de Klundert, J.J., 2006. On the high multiplicity traveling salesman problem, *Discrete Optimization* 3, 50–62.
- [14] Hopcroft, J., Karp, R., 1973. An $n^{2.5}$ algorithm for maximum matching in bipartite graphs, *SIAM Journal on Computing* 2, 225–231.
- [15] Inman, R.R., Bulfin, R.L., 1991. Sequencing JIT mixed model assembly lines, *Management Science* 37, 901–904.
- [16] Kubiak, W., 1993. Minimizing variation of production rates in just-in-time systems: A survey, *European Journal of Operational Research* 66, 259–271.
- [17] Kubiak, W., 2004. Fair Sequences, Leung, J.Y.-T., Anderson, J.H. (eds.) *Handbook of scheduling: Algorithms, models, and performance analysis*, Boca Raton.

- [18] Kubiak, W., Sethi, S.P., 1991. A note on "level schedules for mixed-model assembly lines in just-in-time production systems", *Management Science* 37, 121–122.
- [19] Kubiak, W., Sethi, S.P., 1994. Optimal just-in-time schedules for flexible transfer lines, *The International Journal of Flexible Manufacturing Systems* 6, 137–154.
- [20] Marsten, R.E., Morin, T.L., 1978. A hybrid approach to discrete mathematical programming, *Mathematical Programming* 14, 21–40.
- [21] Miltenburg, J., 1989. Level Schedules for mixed-model assembly lines in just-in-time production systems, *Management Science* 35, 192–207.
- [22] Monden, Y., 1998. *Toyota Production System: an integrated approach to just-in-time*, 3rd edition, Norcross, 1998.
- [23] Monma, C. L., 1980. Sequencing to Minimize the Maximum Job Cost, *Operations Research* 28, 942–951.
- [24] Morin, T.L., Marsten, R.E., 1976. Branch-and-bound strategies for dynamic programming, *Operations Research* 24, 611–627.
- [25] Pinol, H., Beasley, J.E., 2007. Scatter search and bionomic algorithms for the aircraft landing problem, *European Journal of Operational Research* 171, 439–462.
- [26] Psaraftis, H.N., 1980. A dynamic programming approach for sequencing groups of identical jobs, *Operations Research* 28, 1347–1359.
- [27] Steiner, G., Yeomans, J.S., 1993. Level schedules for mixed-model, just-in-time processes, *Management Science* 39, 728–735.