# An Optimal Fleet Assignment and Flight Scheduling Problem for an Airline Company

Yurii A. Mezentsev[1] and Igor V. Estraykh[2]

[1] Novosibirsk State Technical University, Novosibirsk, Russia
mesyan@ yandex.ru
[2] Novosibirsk State Technical University, Novosibirsk, Russia
ive7@yandex.ru

**Abstract.** An original problem statement and solution algorithms are presented for an applied problem in the scheduling theory. The idea of the optimal fleet assignment and Flight Scheduling problem considered in this paper is to find a scheduling control method that minimizes the losses of the airline company from aircraft schedule disruptions. The problem is NP-complete and cannot be solved accurately for any real-life number of dimensions. An efficient parametric algorithm is proposed for finding an approximate solution of the problem. The proposed algorithm is an extension of the schedule optimization algorithm for a system of unrelated parallel machines with job release dates, which is based on the makespan criterion (Cmax). A substantial example is presented of applying the algorithm, as well as statistics of testing it on the data of a generating problem by the Cmax criterion.

**Keywords:** Optimal scheduling · Airline fleet assignment · Makespan criterion ·Efficient parametric algorithm

## Introduction

One of the areas where optimization methods are traditionally applied in practice is the planning of airline operations. In this case, planning involves several stages, the most important being aircraft scheduling, fleet assignment, routing, and crew planning. A detailed review on this topic was published, e.g., by Grönkvist (2005) [1]; the formal problem statements and approaches to solving the basic problems were discussed by Sherali, Bish, and Zhu (2006) [2] and in the numerous publications following individual lines of applied research from among those listed above [3–22].

For instance, the problems and algorithms of airline fleet assignment modeling (FAM) are examined in [3–7]. The main focus in [8–16] is on the aircraft routing problem (ARP). Simultaneous solving of both problems (FAM and ARP) is considered, e.g., in [17–19]. Finally, the studies closest to the subject of the present paper [1, 5, 20–22] consider the optimal scheduling problem, including route changes and fleet assignment.


In: S. Belim et al. (eds.): OPTA-SCL 2018, Omsk, Russia, published at http://ceur-ws.org

Obviously, the above listed problems are closely related. All their relevant formal representations belong to the class of intractable problems of mixed programming. The approaches that are used to find their approximate solutions build on classical schemes such as the Lagrangian relaxation methods, column generation, and Benders decomposition and apply the well-known computational tools of combinatorial optimization, methods of cuts, and programming in constraints [1, 2].

The core of this work is an original problem statement in the form of an optimal scheduling problem for a system of unrelated parallel machines (aircraft) with job release dates (flight delays), which is adapted to the airline flight scheduling problem proposed by the authors in [23, 24], together with a special efficient parametric algorithm for its approximate solution [24].

## 1 Conceptual and Formal Statement for the Optimal Fleet Assignment and Flight Scheduling Problem

The input data are airline flight schedules, standard flight times for of all types of aircraft, and standard times for ground handling and flight preparations for all types of aircraft.

The real-time information is flight delays at any given time at all airports.

Then, conceptually, the scheduling problem consists in finding, for the flights in the planning period, such an airline fleet assignment that will minimize the maximum total deviation from the initial schedule for the entire fleet while satisfying all the constraints of the initial schedule in terms of the passenger flow, number of flights, and flight standards.

We use the following notation:

$l$ is the airport number, $l \in L$;

$i$ is the flight number, $i \in I_l$, $\bigcup_{l \in L} I_l = I$, $I_l \bigcap I_{l'} = \varnothing, \forall l, l' \in L$;

$s$ is the type of aircraft, $s \in S$;

$j$ is the tail number, $j \in J_s$, $\bigcup_{s \in S} J_s = J$, $J_s \bigcap J_{s'} = \varnothing, \forall s, s' \in S$;

$\tau_i^0$ is the actual delay of flight $i$ at the time of scheduling, $\tau_i^0 \geq 0$, $i \in I_l$, $\forall l \in L$, $\mathrm{T}^0 = \left\| \tau_i^0 \right\|$. Hereinafter, $\left\| \cdot \right\|$ denotes a vector, matrix, or tensor corresponding to the context of dimension;

$t_i^0$ is the scheduled departure time of flight $i$, $i \in I_l$, $\forall l \in L$, $T^0 = \left\| t_i^0 \right\|$;

$t_i^0 + \tau_i^0$ is the possible actual departure time of flight $i$ at the initial time of scheduling;

$t_{i,j}$ is the time of ground handling, preparation, and air travel of flight $i$ of aircraft $j$, $T = \left\| t_{i,j} \right\|$, $i \in I_l$, $\forall l \in L$, $j \in J_s, \forall s \in S$.

We need to find $x_{i,j}$ under the constraints:

$$x_{i,j} = \begin{cases} 1, & \text{if aircraft } j \text{ is assigned to flight } i, \\ 0 & \text{otherwise,} \end{cases} \quad i \in I_l, \ \forall l \in L, \ j \in J_s, \forall s \in S \tag{1}$$

$$\sum_{j \in J} x_{i,j} = 1, \ i \in I_l \ \forall l \in L, \tag{2}$$

(constraint (2) means that only one aircraft is assigned to flight i);

$$\underline{b}_j \le \sum_{i \in I_l} x_{i,j} \le \bar{b}_j, \ \forall l \in L, j \in J_s \ \forall s \in S, \tag{3}$$

(constraint (3) means that aircraft with tail number $j$ can be assigned to no less than $\underline{b}_j$ and no more than $\bar{b}_j$ flights);

$\tau_{i,j}$ is a possible delay in the departure of aircraft $j$ on flight $i$, $i \in I_l$, $\forall l \in L$, $j \in J_s, \forall s \in S$ ($\tau_{i,j}$ can be negative, which is taken into account in constraints (5) and (6))

$$\tau_{i,j} = -t_i^0 - \tau_i^0 + \sum_{k \in I_k} (\tau_{k,j} + t_{k,j}) x_{k,j}, \ \forall i \notin I_k, \ j \in J_s, \ s \in S, \tag{4}$$

(constraint (4) means that the delay of aircraft $j$ at the current step (on flight $i$) is a recursive function of the delays accumulated in the previous flights of this aircraft);

$$\hat{\tau}_{i,j} = \tau_{i,j} + y_{i,j} \ge 0, \ \ i \in I_l, \ \forall l \in L, \ j \in J_s, \ \forall s \in S; \tag{5}$$

$$y_{i,j} \ge 0, \ \ i \in I_l, \ \forall l \in L, \ j \in J_s, \ \forall s \in S. \tag{6}$$

Constraints (5) and (6) neutralize negative delays through the compensating variables $y_{i,j} \ge 0$; then, $\hat{\tau}_{i,j} \ge 0$ is the dependent variable, having the meaning of adjusted delay between the arrival of the aircraft $j$ and its flight $i$, taking into account the required service time on the ground, and

$$\sum_{i \in I_l} \hat{\tau}_{i,j} x_{i,j} + \sum_{i \in I_l} t_{i,j} x_{i,j} \le \lambda, \ \forall l \in L, j \in J_s, \ s \in S, \tag{7}$$

$$\lambda \to \min \tag{8}$$

Relations (7) and (8) represent the minimax makespan criterion. The use of this criterion helps achieve a uniform distribution of load on the fleet by minimizing the maximum total downtime for any aircraft from the whole set of aircraft of the airline.

Another variant of constraint (7) is

$$\sum_{i \in I_l} \hat{\tau}_{i,j} x_{i,j} \le \lambda \tag{7'}$$

Instead of, or together with (7), one can apply an additive criterion of minimization of the total delays

$$\sum_{j\in J}\sum_{i\in I}\hat{\tau}_{i,j}x_{i,j} \to \min \tag{9}$$

Relations (4), which mediate constraints (5) and (7), contain recursions because any subsequent (in time) values of $\tau_{i,j}$ and $\hat{\tau}_{i,j}$ depend on the previous ones.

Calculating the delays $\tau_{i,j}$ in all the $i$ previous steps is associated with considerable difficulties because, first, due to the multiplicity of the variants of their formation with the subsequent choice of the best, and, second, the expansion of recursions and reduction of the statement (1) - (8) to the one-stage mixed programming problem leads to an increase in the number of Boolean variables and constraints in the problem by a factor of $\bar{I}/2$, where $\bar{I} = \sup I$ [23]. The structural complexity of (1)–(8) is thereby reduced to the computational complexity of the resulting problem statement, which remains intractable given that the initial dimension increases by a large factor.

For more detail on the expansion of recursions for a statement identical to (1)–(8) with the formation of a one-step problem (which we call, for brevity, a *direct reduction*) and the subsequent formation of a simplified (relaxed) problem with two criteria (called a *bicriteria relaxation*), which allows one to find close-to-optimal schedules in terms of makespan, see [23, 24]. These works also provide experimental proof of inefficiency of using the direct reduction. Thus, e.g., finding even an approximate solution with no more than a six-percent deviation from the optimum for a problem instance with 20 flights and 5 aircraft took more than 16 hours of computing time using a 6-core processor and the latest version of the IBM ILOG CPLEX optimization studio. In [24], one can also find the results of applying the bicriteria relaxation using CPLEX. Below we compare the accuracy and computing time in the approach developed in our publication with the results achieved through the application of the bicriteria relaxation (Table 9).

## 2 Parametric Algorithm for Finding Suboptimal Solutions

Since the problem contains recursions, DP is, most likely, the only computational method directly applicable to solving problem (1)–(8). However, the direct application of DP is inefficient, partly because the problem in question is NP-complete. In attempts at finding an accurate solution of (1)–(8), DP leads to an exhaustive search through all possible options. It is easy to calculate the number $N$ of these options. For example, if $k$ is the step number and we assume in (2) that $\underline{b}_j = 0$ and $\bar{b}_j = \sup I$, $\bar{I} = \sup I$, $\bar{J} = \sup J$, then, as shown below, considering that the number of options grows in a geometric progression with DP steps, we have $N = \left(\bar{J}^{\bar{I}+1} - \bar{J}\right)/2$. Therefore, the DP method in (1)–(8) has a complexity greater than the exponential one and is not applicable in its pure form to problems with an actual number of dimensions.

To construct an efficient approximate algorithm, we use a general DP scheme with the sifting of locally worst options at certain DP steps. We tested this approach previously in solving optimal scheduling problems for unrelated parallel machines with job start delays [24]; the tests showed good results in terms of accuracy and speed.

We assume that all flights $i \in I_l$, $\forall l \in L$, are arranged in the order of the initial delays (the initial schedule $\left\|\tau_i^0\right\|$), considering the aircraft locations at the time of scheduling. Then, based on the DP procedure, we determine the step number $\eta = \overline{1, \bar{I}}$. We denote the time when aircraft $j$ completes flight $\eta$ at step $\eta$ as $f_{\eta,j}(\hat{\tau}_{\eta,j}, t_{\eta,j}, x_{\eta,j})$, $j \in J_s$, $\forall s \in S$, and the conditional minimum time of completion of all flights at steps from 1 to $\eta$ as $\varphi_\eta(\hat{\tau}_{\eta,j}, t_{i,j}, x_{i,j})$ $i = \overline{1, \bar{I}}$, $j = \overline{1, \eta}$

$$f_{\eta,j}(\hat{\tau}_{\eta,j}, t_{\eta,j}, x_{\eta,j}) = \max\{0, [\tau_{\eta,j} x_{\eta,j} - \varphi_{\eta-1,j}(\hat{\tau}_{\eta-1,j}, t_{i,j}, x_{i,j})]\} + t_{\eta,j} x_{\eta,j}, \ j = \overline{1, \bar{J}}, \quad i = \overline{1, \eta - 1}. \tag{10}$$

The recurrent Bellman relation for this problem is

$$\varphi_{\eta,j}(\hat{\tau}_{\eta-1,j}, t_{i,j}, x_{i,j}) = \{f_{\eta,j}(\hat{\tau}_{\eta,j}, t_{\eta,j}, x_{\eta,j}) + \varphi_{\eta-1,j}(\hat{\tau}_{\eta-1,j}, t_{i,j}, x_{i,j})\}, i = \overline{1, \eta - 1}, \tag{11}$$

$$\varphi_\eta(\hat{\tau}_{\eta,j}, t_{i,j}, x_{i,j}) = \max_j \{\varphi_{\eta,j}(\hat{\tau}_{\eta-1,j}, t_{i,j}, x_{i,j})\}, \ j \in J_s, s \in S, i = \overline{1, \eta}. \tag{12}$$

To achieve the minimum makespan in (7)–(8), we should select in the last step the minimum value of $\varphi_{\bar{I}}(\hat{\tau}_{\bar{I},j}, t_{i,j}, x_{i,j})$, i.e., find. $\lambda = \min\{\varphi_{\bar{I}}(\hat{\tau}_{\bar{I},j}, t_{i,j}, x_{i,j})\}$ $j \in J_s, s \in S$, $i \in I_l, l \in L$. The total number of scheduling options that we need to find to ensure the best schedule is

$$N = \bar{J} + \bar{J}^2 + \ldots + \bar{J}^k + \ldots + \bar{J}^{\bar{I}} = \left(\bar{J}^{\bar{I}+1} - \bar{J}\right) \Big/ 2 \tag{13}$$

We can sift out intermediate schedules in DP in different ways. If we discard all intermediate schedules at step $k$ except the locally best one, we have a greedy algorithm. If we keep all the intermediate schedules, we have an exhaustive search through all the options. In the latter case, we have $\bar{J}$ intermediate schedule options at step 1, $\bar{J}^2$ options at step 2, and $\bar{J}^k$ options at step $k$. If we look for a compromise between accuracy and speed, then, considering that we seek to construct an efficient algorithm, the number of intermediate schedules should be polynomially dependent on the number of Boolean variables in (1)–(8).

Let us now consider one such compromise. First, we determine the maximum number $K$ of the options retained at stage $k$ for further analysis. For convenience of description, we assume that $K$ is a constant. For example, we assume that $K = 1024$ and determine the maximum number $K' = \overline{J}^k \le K$. Since the number of possible intermediate schedules increases by a factor of $\overline{J}$ at each step, we suggest sifting out $1 - 1/\overline{J}$ of the locally worst options at each step starting from $k+1$.

Obviously, $k$ : $k = \left[ \frac{\ln(K)}{\ln(\overline{J})} \right]$, where $[\cdot]$ is the integer part of the number.

If we calculate the total number of schedule options generated in the algorithm, we obtain $\overline{J}$ options at step 1, $\overline{J}^k$ options at step $k$, and also $\overline{J}^k$ options at steps from $k+1$ to $\overline{I}$. This scheme is implemented by sifting out $\overline{J}^{k-1}$ intermediate schedule options at all the steps from $k+1$ to $\overline{I}$. Then, the number of options that remains for further consideration at each step beginning from $k+1$ is exactly $\overline{J}$, and the total number of intermediate schedules $N'$ is

$$N' = \overline{J} + \overline{J}^2 + \ldots + \overline{J}^{k-1} + \overline{J}^k + \ldots + \overline{J}^k = \left( \overline{J}^{k-1} - \overline{J} \right) \Big/ 2 + \left( \overline{I} - k + 1 \right) \cdot \overline{J}^k \qquad (14)$$

Since $k$ is a constant, relation (14) represents a polynomial dependence of the complexity of the parametric DP algorithm with option-sifting on the dimension of problem (1)–(8). In this case, $k$ is the degree of this polynomial. For clarity, we compare $N$ with $N'$, assuming $k = 3$, $\overline{I} = 1000$, and $\overline{J} = 100$.

Then, $N = \left(100^{1001} - 100\right)/2$, $N' = \left(100^2 - 100\right)/2 + (100 - 2) \cdot 100^3 = 98009900$. These circumstances underlie the ordinary complexity of the parametric algorithm (its complexity is defined by the parameter $k$) and the virtually infinite complexity of the DP method.

Based on (14), we can estimate the total complexity of the parametric algorithm. To this end, it is sufficient to determine the complexity of the step beginning from $k$, which directly depends on the number of combinations of the variables $x_{i,j}$ at step $k$. If we denote this value as $P_k$, then, obviously, $P_k = \overline{J}^k$. In fact, this means that at each step beginning from $k$, the algorithm requires calculating $P_k$ variants of constrains (7) for all the possible values of $x_{k,j}$. In total, we have at all steps: $P_1 = \overline{J}$, $P_2 = \overline{J}^2$, $P_l = \overline{J}^k$, $l = \overline{k, \overline{I}}$. In the above example, $P_k = 100^3$, and, considering (14), we obtain a high complexity for a problem with an actual number of dimensions. We can overcome this difficulty either by reducing $k$ or by decomposition.

Below we describe the parametric DP algorithm with the sifting of the locally worst intermediate options.

***Algorithm*** $A_P$

1. Enter the input data $(\tau_i^0, t_{i,j})$, $j \in J_s$, $s \in S$, $i \in I_l$, $l \in L$, and the parameters $k$ and $N'$. Assuming that $\varphi_{0,j}(\tau_0^0, t_{i,j}, x_{i,j}) \equiv 0$, determine the initial step number $\eta := 0$.
2. $\eta := \eta + 1$.
3. Check the step number. If $\eta > \bar{I}$, proceed to point 7; otherwise, proceed to the next point.
4. At step $\eta$, determine the sequence of the subsequent steps (rearrange the flight list), calculate the delays $\hat{\tau}_{\eta,j}$, generate from (10)–(12) all the feasible fleet assignment options, and calculate $f_{\eta,j}(\hat{\tau}_{\eta,j}, t_{\eta,j}, x_{\eta,j})$ and the schedule lengths $\varphi_{\eta,j}(\hat{\tau}_{\eta,j}, t_{i,j}, x_{i,j})$.
5. Check $N^{\eta}$, i.e., the number of options of $\varphi_{\eta,j}(\hat{\tau}_{\eta,j}, t_{i,j}, x_{i,j})$ at step $\eta$. If $\eta < k$, i.e., $N^{\eta} \le N'$, then proceed to point 2; otherwise, proceed to the next point.
6. Sift out $\bar{J}^{k-1}$ of all the options generated at point 4 with the largest schedule lengths $\varphi_{\eta,j}(\hat{\tau}_{\eta,j}, t_{i,j}, x_{i,j})$. Return to point 2.
7. Choose schedule options with the minimum length. Construct the final schedules using the inverse DP procedure.

A note on the $A_P$ algorithm regarding estimates for the delays $\hat{\tau}_{i,j}$:

At each step of the algorithm, one needs to estimate the delays $\hat{\tau}_{i,j}$ for aircraft that have not yet arrived at the airport of departure. The estimates for $\hat{\tau}_{i,j}$ are found by solving the subproblems of finding all the shortest paths in a graph composed of the possible connections between the airports. In general, one should find these estimates at each step because the delays can vary from step to step, depending on the previous local fleet assignments.

## 3 Illustrative Example

Below is an illustrative fragment demonstrating the application of the proposed algorithm for constructing a close-to-initial flight schedule and fleet assignment for three aircraft (designated by their tail numbers 1, 2, and 3) of two different types.

The input data on the flights and time costs are given in Table 1.

In Table 1, the Time column contains $t_{i,j}$; the Location column shows the presence or absence of aircraft at the airport of departure at each step of scheduling; and the initial schedule corresponds to $\tau_i^0$. The Tail Number column shows the numbers

$j$. The data in Table 1 are sorted by $\tau_i^0$, taking into account the aircraft location. The flights are numbered in the same order.

The algorithm parameters are $k=1$, ($N'=3$), $\overline{J}=3$, and $\overline{I}=11$.

Since $k=1$, we can reduce the number of dimensions (the eliminated options are highlighted by filling).

After the third step, both the aircraft locations and the current delays $\hat{\tau}_{i,j}$ change, necessitating a new tail assignment sequence, i.e., a change in the sequence of the algorithm steps (see point 4 in $A_P$).

**Table 1.** Flight Data

| Flight | Departure airport | Destination airport | Aircraft data | | | Type and | initial |
|---|---|---|---|---|---|---|---|
| | | | Type | Time | Tail Number | location | schedule |
| 1 | 1 | 2 | 1т | 5 | 1 | 1т | 1 |
| 2 | 2 | 3 | 1т | 3 | 2 | 1т | 2 |
| 3 | 2 | 1 | 2т | 4 | 3 | 2т | 1 |
| 4 | 3 | 1 | 2т | 2 | | | 1 |
| 5 | 1 | 3 | 1т | 2 | | | 2 |
| 6 | 3 | 2 | 1т | 4 | | | 2 |
| 7 | 1 | 2 | 2т | 4 | | | 4 |
| 8 | 1 | 3 | 2т | 2 | | | 4 |
| 9 | 3 | 1 | 1т | 3 | | | 4 |
| 10 | 2 | 1 | 1т | 5 | | | 4 |
| 11 | 2 | 3 | 1т | 3 | | | 5 |

**Table 2.** Algorithm: Step One

| $\eta=1$ | $x_{1,1}$ | $x_{1,2}$ | $x_{1,3}$ | $f_{1,j}$ | $\varphi_{1,j}=\{f_{1,j}\}$ $\varphi_1=\max_j\{\varphi_{1,j}\}$ |
|---|---|---|---|---|---|
| Flight 1 | 1 | 0 | 0 | (1+5,0,0) | $\varphi_1=\max\{6;0;0\}=6$ |
| | 0 | 1 | 0 | (0,~,0) | $\varphi_1=\max\{0,\tau_{1,2},0\}=\tau_{1,2}$ |
| | 0 | 0 | 1 | (0,0 ,~) | $\varphi_1=\max\{0,0,\infty\}=\infty$ |

**Table 3.** Algorithm: Step Two

| $\eta=2$ | $x_{1,1}$ | $x_{1,2}$ | $x_{2,1}$ | $x_{2,2}$ | $x_{2,3}$ | $f_{2,j}$ | $\varphi_{2,j}=\{f_{2,j}+\varphi_{1,j}\}$, $\varphi_2=\max_j\{\varphi_{2,j}\}$ |
|---|---|---|---|---|---|---|---|
| Flight 1 + | 1 | 0 | 1 | 0 | 0 | (6+3,0,0) | $\varphi_2=\max\{6+3;0+0;0+0\}=9$ |
| Flight 2 | 1 | 0 | 0 | 1 | 0 | (0, 2+3, 0) | $\varphi_2=\max\{6+0;0+5, 0+0\}=6$ |
| | 1 | 0 | 0 | 0 | 1 | (0, 0 , ~) | $\varphi_2=\max\{6+0,0,\infty\}=\infty$ |
| | 0 | 1 | 1 | 0 | 0 | (13+3,0,0) | $\varphi_2=\max\{13+3;0+0;0+0\}=16$ |
| | 0 | 1 | 0 | 1 | 0 | (0, 2+3, 0) | $\varphi_2=\max\{13+0;0+5, 0+0\}=13$ |
| | 0 | 1 | 0 | 0 | 1 | (0, 0 , ~) | $\varphi_2=\max\{13+0,0,\infty\}=\infty$ |

The change in the tail assignment is given in Table 4, whose rows are arranged in the order of increasing $\widehat{\tau}_{i,j}$, considering the current aircraft locations.

**Table 4.** Ordered Data

| Flight | Departure airport | Destination airport | Aircraft data: Type | Time | Number | Type, location | Initial schedule |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 1т | 5 | 1 | 1т | 1 |
| 2 | 2 | 3 | 1т | 3 | 2 | 1т | 2 |
| 3 | 2 | 1 | 2т | 4 | 3 | 2т | 1 |
| 4 | 3 | 1 | 2т | 2 | | | 1 |
| 5 | 1 | 3 | 1т | 2 | | | 2 |
| 6 | 3 | 2 | 1т | 4 | | | 2 |
| 7 | 1 | 2 | 2т | 4 | | | 4 |
| 8 | 1 | 3 | 2т | 2 | | | 4 |
| 9 | 3 | 1 | 1т | 3 | | | 4 |
| 10 | 2 | 1 | 1т | 5 | | | 4 |
| 11 | 2 | 3 | 1т | 3 | | | 5 |

**Table 5.** Step Four

| $\eta = 4$ | $x_{2,1}$ | $x_{2,2}$ | $x_{4,1}$ | $x_{4,2}$ | $f_{4,j}$ | $\varphi_{4,j} = \{f_{4,j} + \varphi_{3,j}\},\ \varphi_4 = \max_j \{\varphi_{4,j}\}$ |
|---|---|---|---|---|---|---|
| Flights 1,2,3,6 | 1 | 0 | 1 | 0 | (4,0,0) | $\varphi_4$ = max{6+3+0+4; 0+0+0+0; 0+0+5+0} =13 |
| | 1 | 0 | 0 | 1 | (0,4,0) | $\varphi_4$ = max{6+3+0+0; 0+0+0+40; 0+0+5+0} =9 |
| | 0 | 1 | 1 | 0 | (4,0,0) | $\varphi_4$ = max{6+0+0+4; 0+5+0+0; 0+0+5+0} =10 |
| | 0 | 1 | 0 | 1 | (0,4,0) | $\varphi_4$ = max{6+0+0+0; 0+5+0+4; 0+0+5+0}=9 |
| | 0 | 1 | 1 | 0 | (4,0,0) | $\varphi_4$ = max{13+0+0+4; 0+5+0+0;0+0+5+0} =17 |
| | 0 | 1 | 0 | 1 | (0,4,0) | $\varphi_4$ = max{13+0+0+0;0+5+0+4; 0+0+5+0} =13 |

The calculations at the fourth step are given in Table 5.

Consistent with the parameter $k = 1$, we cross out the locally worst current schedule options (for $\varphi_4 \geq 13$).

The results of the calculations at the final eleventh step are summarized in Table 6. The tail assignment options excluded from consideration are highlighted by filling.

The square brackets in the sums of the aircraft handling times highlight the options when the aircraft with the corresponding tail numbers are absent at the airport of departure. In this case, the aircraft needs to be delivered from another nearby airport.

The delivery time is indicated inside the square brackets.

Thus, we have obtained four tail assignment options and flight schedules, equivalent in terms of criterion (7)–(8) ($\lambda = 17$ for all the options).

Tables 7 and 8 show all the best aircraft assignment options and the corresponding flight schedules. The Start columns show the departure times, and the End columns show a total of the arrival time plus the time on ground handling and preparations for the next flight.

**Table 6.** Final Step

| $\eta = 11$ | $x_{8,1}$ | $x_{8,2}$ | $x_{11,1}$ | $x_{11,2}$ | $f_{11,j}$ | $\varphi_{11,j} = \{f_{11,j} + \varphi_{10,j}\}$, $\lambda = \varphi_{11} = \max\{\varphi_{11,j}\}$ |
|---|---|---|---|---|---|---|
| Flights: 1,2,3,6, 7,10,11, 5,8,4,9 | 1 | 0 | 1 | 0 | (3,0,0) | $\varphi_{11}$ = max{6+0+0+0+0+0+3+[2+2]+0+0+3; 0+5+0+4+0+5+0+0+0+0+0; 0+0+5+0+4+0+0+0+[4+2]+2+0} =17 |
| | 1 | 0 | 0 | 1 | (3,0,0) | $\varphi_{11}$ = max{6+0+0+0+0+5+0+2+0+0+3; 0+5+0+4+0+0+3+0+0+0+0; 0+0+5+0+4+0+0+0+[4+2]+2+0} =17 |
| | 1 | 0 | 0 | 0 | (3,0,0) | $\varphi_{11}$ = max{6+0+0+0+0+0+3+0+0+0+3; 0+5+0+4+0+5+0+2+0+0+0; 0+0+5+0+4+0+0+0+[4+2]+2+0} =17 |
| | 0 | 1 | 1 | 0 | (0, 3, 0) | $\varphi_{11}$ = max{6+0+0+0+0+0+3+[2+2]+0+0+0; 0+5+0+4+0+5+0+0+0+0+[3+2]; 0+0+5+0+4+0+0+0+[4+2]+2+0}=19 |
| | 0 | 1 | 0 | 1 | (0, 3, 0) | $\varphi_{11}$ = max{6+0+0+0+0+5+0+2+0+0+0; 0+5+0+4+0+0+3+0+0+0+3; 0+0+5+0+4+0+0+0+[4+2]+2+0} =17 |
| | 0 | 1 | 0 | 0 | (0, 3, 0) | $\varphi_{11}$ = max{6+0+0+0+0+0+3+0+0+0+0; 0+5+0+4+0+5+0+2+0+0+3, 0+0+5+0+4+0+0+0+[4+2]+2+0} =19 |

**Table 7.** Shortest Schedule: Options 1 and 2

| i= | j=1 | j=2 | j=3 | Start | End |
|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 6 |
| 2 | 0 | 1 | 0 | 2 | 5 |
| 3 | 0 | 0 | 1 | 1 | 5 |
| 4 | 0 | 1 | 0 | 5 | 9 |
| 5 | 0 | 0 | 1 | 5 | 9 |
| 6 | 0 | 1 | 0 | 9 | 14 |
| 7 | 1 | 0 | 0 | 6 | 9 |
| 8 | 1 | 0 | 0 | 11 | 13 |
| 9 | 0 | 0 | 1 | 13 | 15 |
| 10 | 0 | 0 | 1 | 15 | 17 |
| 11 | 1 | 0 | 0 | 13 | 16 |

| j=1 | j=2 | j=3 | Start | End |
|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 6 |
| 0 | 1 | 0 | 2 | 5 |
| 0 | 0 | 1 | 1 | 5 |
| 0 | 1 | 0 | 5 | 9 |
| 0 | 0 | 1 | 5 | 9 |
| 1 | 0 | 0 | 6 | 11 |
| 0 | 1 | 0 | 9 | 12 |
| 1 | 0 | 0 | 11 | 13 |
| 0 | 0 | 1 | 13 | 15 |
| 0 | 0 | 1 | 15 | 17 |
| 1 | 0 | 0 | 13 | 16 |

This example clearly demonstrates the universal applicability of the $A_P$ algorithm. It is also suitable for solving both the FAM problem and the mixed FAM + ARP problem. The same feature allows finding for the target optimal scheduling problem a $k$-best solution that minimizes the airline company losses from disruptions in the initial aircraft flight schedules.

**Table 8.** Shortest Schedule: Options 3 and 4

| i= | j=1 | j=2 | j=3 | Start | End | j=1 | j=2 | j=3 | Start | End |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 6 | 1 | 0 | 0 | 1 | 6 |
| 2 | 0 | 1 | 0 | 2 | 5 | 0 | 1 | 0 | 2 | 5 |
| 3 | 0 | 0 | 1 | 1 | 5 | 0 | 0 | 1 | 1 | 5 |
| 4 | 0 | 1 | 0 | 5 | 9 | 0 | 1 | 0 | 5 | 9 |
| 5 | 0 | 0 | 1 | 5 | 9 | 0 | 0 | 1 | 5 | 9 |
| 6 | 0 | 1 | 0 | 9 | 14 | 1 | 0 | 0 | 6 | 11 |
| 7 | 1 | 0 | 0 | 6 | 9 | 0 | 1 | 0 | 9 | 12 |
| 8 | 0 | 1 | 0 | 14 | 16 | 1 | 0 | 0 | 11 | 13 |
| 9 | 0 | 0 | 1 | 13 | 15 | 0 | 0 | 1 | 13 | 15 |
| 10 | 0 | 0 | 1 | 15 | 17 | 0 | 0 | 1 | 15 | 17 |
| 11 | 1 | 0 | 0 | 9 | 12 | 0 | 1 | 0 | 12 | 15 |

## 4 Exchange-Based Improving Algorithm

The solution obtained by the parametric $A_P$ algorithm can be improved using a procedure based on an exchange of flights between aircraft. Below we describe this algorithm.

At the first step, we select an aircraft with the maximum total flight time and try reassigning one of its flights to another aircraft. If the exchange reduces the value of the objective function, we repeat the process; otherwise, we move to the next flight.

At the second step, we select an aircraft with the maximum total flight time and consistently review the flights assigned to this aircraft. In each case, we search for flights that this aircraft makes in less time yet assigned to another aircraft. Then, the flights are exchanged between the aircraft. If the exchange reduces the value of the objective function, we repeat the second step; otherwise, we cancel the exchange and search for another flight suitable for reassignment. We denote the general exchange-based algorithm, as well as the one implemented at step two, as $A_C$. The application of the $A_C$ algorithm at step two can be detailed as follows:

### *Algorithm* $A_C$

1. Select aircraft *m* with the maximum total flight time $\lambda_m = \lambda$.
2. Assume $i := 1$.
3. If $i := m$, then go to point 14.
4. Assume $l := 1$.
5. If $x_{m,l} = 0$, then go to point 12.
6. Assume $j := 1$.
7. If $j = l$ or $x_{i,j} = 0$ or $t_{m,j} \geq t_{m,l}$ or $t_{i,l} - t_{i,j} \geq \lambda_m - \lambda_i$, go to point 10.
8. Assume $x_{i,j} = 0$, $x_{m,j} = 1$, $x_{i,l} = 1$, $x_{m,l} = 0$.

9. Calculate the value of the objective function. If it has decreased, go to point 1; otherwise, assume $x_{i,j}=1$, $x_{m,j}=0$, $x_{i,l}=0$, $x_{m,l}=1$.

10. Assume $j:=j+1$.

11. If $j \le \overline{J}$, go to point 7.

12. Assume $l:=l+1$.

13. If $l \le \overline{J}$, go to point 5.

14. Assume $i:=i+1$.

15. If $i \le \overline{I}$, go to point 3; otherwise, stop the algorithm.

Let us estimate the complexity of the $A_C$ algorithm, which searches for flight exchange options between aircraft for any finite schedule obtained by the $A_P$ algorithm. Since the former conducts an exhaustive search among all aircraft and all flights and calculates the value of the objective function for every exchange option, the complexity of the exchange-based algorithm for one schedule is $O\left(\overline{I}\cdot\overline{J}^2\right)$. But since the $A_P$ algorithm generates $\overline{J}$ schedules at the final step, the total upper-bound estimate for the complexity of $A_C$ is $O\left(\overline{I}\cdot\overline{J}^3\right)$.

## 5 Results of Testing the Algorithms

A software implementation of the parametric $A_P$ and exchange-based $A_C$ algorithms allowed us to investigate their properties for instances with a close-to-actual number of dimensions. The algorithms were tested on the data of the optimal scheduling problem for a system of unrelated parallel machine with job start delays [23]. Table 9 contains the results of testing the instances of problem (1)–(8) with the use of the above-mentioned means of solving the *bicriteria relaxation* of problem (1)–(8) [23] and the $A_P$ and $A_C$ algorithms. All the tests have the same number of dimensions. The number of flights is $\overline{I}=100$, and the number of aircraft is $\overline{J}=5$. The algorithms $A_P$ and $A_C$ were applied with two values of the parameter: $k=4$ and $k=5$.

In Table 9, $t_{da}$ and $\lambda_{da}$ are, respectively, the solution time (hh:mm:ss) and the value of the efficiency criterion, which were obtained by applying the basic algorithm based on bicriteria relaxation and IBM ILOG CPLEX [24].The values $t_{dp}$, $\lambda_{dp}$, $p_{dp}$, and $\Delta_{dp}$ are, respectively, the solution time (in seconds); the value of the criterion; and the relative and absolute worsening (improvement at a negative value) of the criterion achieved by the $A_P$ algorithm, compared with the basic algorithm. The corresponding $\lambda^c_{dp}$, $p^c_{dp}$, and $\Delta^c_{dp}$ values were obtained by the $A_C$ algorithm.

In general, there is an evident absolute gain in speed due to the efficiency of the $A_P$ algorithm and its combination with $A_C$. Moreover, the solutions obtained show almost complete superiority over the basic algorithm in terms of closeness to the optimal solutions.

**Table 9.** Comparative Characteristics of the Algorithms

| No. | $t_{da}$ | $\lambda_{da}$ | $t_{dp}$ | $\lambda_{dp}$ | $p_{dp}$ | $\Delta_{dp}$ | $\lambda^c_{dp}$ | $p^c_{dp}$ | $\Delta^c_{dp}$ | $t_{dp}$ | $\lambda_{dp}$ | $p_{dp}$ | $\Delta_{dp}$ | $\lambda^c_{dp}$ | $p^c_{dp}$ | $\Delta^c_{dp}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | hh:mm:ss | $\bar{I}=100$, $\bar{J}=5$, | | | | | $A_C$ | | | $\bar{J}=5$, $k=5$ | | | | $A_C$ | | |
| 1 | 00:07:18 | 389 | 17 | 368 | -5.40 | -21 | 364 | -6.43 | -25 | 435 | 363 | -6.68 | -26 | 359 | -7.71 | -30 |
| 2 | 00:15:29 | 335 | 18 | 337 | 0.60 | 2 | 335 | 0.00 | 0 | 436 | 331 | -1.19 | -4 | 329 | -1.79 | -6 |
| 3 | 00:09:21 | 401 | 17 | 382 | -4.74 | -19 | 382 | -4.74 | -19 | 438 | 375 | -6.48 | -26 | 375 | -6.48 | -26 |
| 4 | 00:01:42 | 356 | 18 | 370 | 3.93 | 14 | 366 | 2.81 | 10 | 437 | 365 | 2.53 | 9 | 361 | 1.40 | 5 |
| 5 | 00:12:11 | 334 | 17 | 337 | 0.90 | 3 | 333 | -0.30 | -1 | 435 | 337 | 0.90 | 3 | 337 | 0.90 | 3 |
| 6 | 04:43:14 | 363 | 17 | 377 | 3.86 | 14 | 369 | 1.65 | 6 | 434 | 375 | 3.31 | 12 | 361 | -0.55 | -2 |
| 7 | 00:11:56 | 403 | 18 | 404 | 0.25 | 1 | 404 | 0.25 | 1 | 434 | 404 | 0.25 | 1 | 401 | -0.50 | -2 |
| 8 | 01:34:02 | 395 | 17 | 398 | 0.76 | 3 | 397 | 0.51 | 2 | 433 | 386 | -2.28 | -9 | 386 | -2.28 | -9 |
| 9 | 00:03:33 | 364 | 17 | 372 | 2.20 | 8 | 372 | 2.20 | 8 | 434 | 371 | 1.92 | 7 | 371 | 1.92 | 7 |
| 10 | 00:15:37 | 395 | 17 | 392 | -0.76 | -3 | 366 | -7.34 | -29 | 434 | 389 | -1.52 | -6 | 376 | -4.81 | -19 |
| | Average | | | | 0.16 | 0.2 | | -1.14 | -4.7 | | | -0.93 | -3.9 | | -1.99 | -7.9 |

Tables 10 and 11 show the algorithm testing statistics to estimate the solution times for instances of the optimal airline fleet assignment and flight scheduling problem with an actual number of dimensions.

**Table 10.** Tests: $\bar{I}=100$, $\bar{J}=10$, $k=3$

| Test No. | $t_{dp}$ | $\lambda_{dp}$ | $\lambda^c_{dp}$ |
|---|---|---|---|
| 1 | 325 | 163 | 154 |
| 2 | 322 | 126 | 122 |
| 3 | 322 | 145 | 145 |
| 4 | 338 | 139 | 136 |
| 5 | 329 | 157 | 152 |
| 6 | 322 | 134 | 132 |
| 7 | 324 | 147 | 147 |
| 8 | 322 | 135 | 132 |
| 9 | 323 | 153 | 153 |
| 10 | 326 | 137 | 137 |

**Table 11.** Tests: $\bar{I}=100$, $\bar{J}=30$, $k=2$

| Test No. | $t_{dp}$ | $\lambda_{dp}$ | $\lambda^c_{dp}$ |
|---|---|---|---|
| 1 | 18 | 106 | 106 |
| 2 | 18 | 105 | 105 |
| 3 | 18 | 107 | 107 |
| 4 | 18 | 106 | 106 |
| 5 | 17 | 108 | 108 |
| 6 | 18 | 114 | 114 |
| 7 | 18 | 105 | 104 |
| 8 | 18 | 108 | 106 |
| 9 | 18 | 106 | 106 |
| 10 | 19 | 105 | 105 |

Concerning the accuracy (i.e., closeness of the flight assignments and aircraft schedules generated by the $A_P$ and $A_C$ algorithms to the optimal ones) estimates, we note the following points. There are no a priori accuracy estimates for $A_P$ and for the combination $A_P + A_C$, but there are a posteriori ones at small dimensions, which are as follows [24]: in approximately 82% of cases, an accurate solution was obtained in the generated tests. In the other cases, the deviation from the optimum was no more than 6%. This conclusion was derived from a comparison of the $A_P + A_C$ testing results with the solutions of the same tests in CPLEX by the expansion of recursions and the direct reduction of problem (1)–(8) in milp.

The dimensions of the tests (100 flights and 30 aircraft) and the solution time offer hope that the designed toolkit would be efficient in solving real-life problems of airline planning.

## Conclusions

The results obtained demonstrate the efficiency of the proposed approaches in solving real-life problems of air transportation planning, including aircraft fleet assignment, routing, and, if necessary, flight scheduling. Thus, our approach would make a promising a contribution to the planning practice of an airline company of any size. A posteriori estimates for the accuracy and speed of the algorithms lead us to conclude that the developed toolkit has evident advantages over its analogs.

The testing confirms experimentally, in terms of computing time, the efficiency of the $A_P + A_C$ pair. Noteworthy is the insignificant contribution of the $A_C$ algorithm to the total complexity. The use of this algorithm adds no more than a second to the total computing time $t_{dp}$ for the tests in Tables 9–11.

## References

1. Grönkvist, M.: The Tail Assignment Problem. PhD thesis, Chalmers University of Technology and Göteborg University, Göteborg, Sweden (2005)
2. Sherali, H.D., Bish, E.K., Zhu, X.: Airline fleet assignment concepts, models, and algorithms. European Journal of Operational Research 172, 1-30 (2006)
3. Hane, C. A., Barnhart, C., Johnson, E. L., Marsten, R. E., Nemhauser, G. L., Sigismondi, G.: The fleet assignment problem: solving a large-scale integer program. Mathematical Programming 70, 211-232 (1995)
4. Gu, Z., Johnson, E. L., Nemhauser, G. L., Wang, Y.: Some Properties of the Fleet Assignment Problem. Technical report, School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA (1993)
5. Subramanian, R., Scheff, Jr. R. P., Quillinan, J. D., Wiper, D. S., Marsten, R. E. Coldstart: Fleet Assignment at Delta Air Lines. Interfaces 24(1), 104-120 (1994)
6. Ozdemir, Y., Basligil, H., Nalbant, K.G.: Optimization of fleet assignment: a case study in Turkey. An International Journal of Optimization and Control: Theories & Applications 2(1), 59-71 (2012)
7. Blegur, F.M.A., Bakhtiar, T., Aman, A.: Scenarios for fleet assignment: a case study at Lion Air. IOSR Journal of Mathematics 10(5) Ver. I, 64-68 (2014)
8. Kabbani, N. M., Patty, B. W.: Aircraft routing at American Airlines. In: Proceedings of the Thirty-Second Annual Symposium of AGIFORS (1992)
9. Clarke, L. W., Johnson, E. L., Nemhauser, G. L., Zhu, Z.: The aircraft rotation problem. Annals of Operations Research 69, 33-46 (1997)

10. Cordeau, J. F., Stojkovi´c, G., Soumis, F., Desrosiers, J.: Benders decomposition for simultaneous aircraft routing and crew scheduling. Transportation Science 35(4), 55-76 (2001)
11. Barnhart, C., Boland, N.L., Clarke, L.W., Johnson, E.L., Nemhauser, G.L., Shenoi, R.G. Flight string models for aircraft fleeting and routing. Transportation Science 32(3), 208-220 (1998)
12. Elf, M., J¨unger, M., Kaibel, V.: Rotation planning for the continental service of a European Airline. In: W. Jager and H.-J. Krebs, editors, Mathematics – Key Technologies for the Future. Joint Projects between Universities and Industry. pp. 675-689, Springer (2003)
13. Sarac, A., Batta, R., Rump, C. M.: A Branch-and-Price Approach for Operational Aircraft Maintenance Routing. Working Paper, Department of Industrial Engineering, University of Buffalo, Buffalo, NY, USA, ( 2003)
14. Gopalan, R., Talluri, K. T.: The aircraft maintenance routing problem. Operations Research, 46(2), 260-271 (1998)
15. Ahuja, R. K., Goodstein, J., Mukherjee, A., Orlin, J. B., Sharma, D.: A Very Large-Scale Neighborhood Search Algorithm for the Combined Through and Fleet Assignment Model. Working Paper 4388-01, MIT Sloan School of Management, Cambridge, USA (2001)
16. Birbil, S.I., Frenk, J.B.G., Gromicho, J.A.S., Zhang, S.: A Network Airline Revenue Management Framework Based on Decomposition by Origins and Destinations. Submitted manuscript (2015)
17. Yan, S., Tseng, C.H.: A passenger demand model for airline flight scheduling and fleet routing. Computers & Operations Research 29, 1559-1581 (2002)
18. Barnhart, C., Boland, N.L., Clarke, L.W., Johnson, E.L., Nemhauser, G.L., Shenoi, R.G.: Flight string models for aircraft fleeting and routing. Transportation Science 32(3), 208-220 (1998)
19. Sandhu, R., Klabjan, D.: Integrated Airline Planning. Working paper, submitted for publication, Department of Mechanical and Industrial Engineering, University of Illinois at Urbana-Campaign, Urbana, IL, USA (2004)
20. Lettovsky, L.: Airline Operations Recovery: An Optimization Approach. PhD thesis, School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA (1997)
21. Kohl, N., Larsen, A., Larsen, J., Ross, A., Tiourine, S.: Airline Disruption Management - Perspectives, Experiences and Outlook. Research and Technology Report CRTR-0407, Carmen Systems AB, Gothenburg, Sweden (2004)
22. Rosenberger, J. M.: Topics in Airline Operations. PhD thesis, School of Industrial & Systems Engineering, Georgia Institute of Technology, Atlanta, GA, USA (2002)
23. Avdeenko, T. V., Mesentsev, Y. A.: Efficient approaches to scheduling for unrelated parallel machines with release dates. IFAC-Papers Online (IFAC Proceedings Volumes). 49(12), 1743-1748 (2016)
24. Avdeenko, T. V., Mezentsev, Y. A., Estraikh, I.V.: Heuristic approach to unrelated parallel machines scheduling under availability and resource constraints. IFAC-PapersOnline, 50(1), 13096-13101 (2017)
25. Mezentsev, Y. A.: Binary cut-and-branch method for solving linear programming problems with boolean variables. In: Proc. DOOR 2016, Vladivostok, Russia, September 19-23, CEUR-WS 1623. Pp. 72-85 (2016)