Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 36 (2009) 176 – 195
computers & operations research
www.elsevier.com/locate/cor

Integrated airline scheduling

Nikolaos Papadakos[a, b, *]

[a]Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK
[b]Decisal Ltd, 20 Purcell Crescent, London SW6 7PB, UK

Available online 25 August 2007

Abstract

Airline scheduling is composed of fleet assignment, aircraft maintenance routing, and crew scheduling optimization subproblems. It is believed that the full optimization problem is computationally intractable, and hence the constituent subproblems are optimized sequentially so that the output of one is the input of the next. The sequential approach, however, provides an overall suboptimal solution and can also fail to satisfy the maintenance constraints of an otherwise feasible full problem. In this paper several integrated models for the optimization of airline scheduling are presented for the first time, and solved by applying an enhanced Benders decomposition method combined with accelerated column generation. Solutions of several realistic data sets are computed using the integrated models, which are compared with solutions of the best known approaches from the literature. As a result, the integrated approach significantly reduces airline costs. Finally, a comparison of alternative formulations has shown that keeping the crew scheduling problem alone in the Benders subproblem is much more efficient than keeping the aircraft routing problem.

Keywords: Integrated scheduling; Fleet assignment; Aircraft maintenance routing; Crew scheduling; Plane-count constraints; Accelerated Benders decomposition; Column generation

1. Introduction

Airlines commence their tactical planning with *schedule generation* where the timetable of profitable legs is devised—see, e.g. [1]. A *leg* is a flight from a specific origin to a destination at a given departure time. Based on the timetable, airlines proceed onto solving the *airline scheduling* problem months before the day of operation. Since this problem is considered computationally intractable [2], it is typically decomposed into its constituent stages, and the full problem is solved in a sequential manner where the output of one stage is the input of the next [1]. In this manner, airlines initially solve the *fleet assignment* (FA) stage, deciding which fleet should fly each scheduled leg, while using the available aircraft and maximizing revenue. After this, the *maintenance routing* (MR) stage is solved, ensuring that aircraft will be periodically scheduled for maintenance, by devising individual aircraft routes. The obtained routes are used in the *crew pairing* (CP) stage, which devises the series of legs crew have to fly, while respecting labor rules and minimizing crew costs.

The knowledge of aircraft routes is important for CP in order to determine whether crew should remain on the same aircraft for their following leg, rather than using precious time commuting within the airport to board a different aircraft. The time needed for commuting is known as sit-time or connection, and a connection shorter than the minimum sit-time

* Corresponding author at: Department of Computing, Imperial College London, 180 Queen's Gate, London SW7 2AZ, UK.
E-mail address: nikolaos.papadakos01@imperial.ac.uk.

0305-0548/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2007.08.002

allowed is known as a *short-connection*. When the maintenance is guaranteed—i.e. maintenance can always be done during the night—the MR is relaxed to the *aircraft routing* problem, which devises aircraft routes without maintenance constraints. In this case *plane-count constraints* can be added in the CP to count the number of available aircraft on the ground at any time to make sure that the aircraft routing problem will be feasible with the set of forced turns [3,4].

Although the sequential procedure reduces computational complexity, because of the interdependence of each stage, the resulting solution is suboptimal. Even worse, in some cases feasible problems might not be solvable. As an example, in [5] after solving the FA problem with the approximate MR considerations of [6], it was not possible to find MR solutions to an otherwise feasible scheduling problem. In order to circumvent the previous obstacles, over the past years models considering several stages simultaneously have been proposed. The aim of these *semi-integrated models* was to achieve better quality results, with the ultimate goal being to integrate all the stages. Several models and a solution methodology that achieve complete integration of all stages are presented in this paper for the first time. Before going into detail about this paper's contributions, a summary of related work on semi-integrated airline scheduling and relevant solution methods is initially presented.

1.1. Literature review

The first exact semi-integration attempt included FA and aircraft routing, where the departure times of legs were flexibly defined within time-windows [7]. An integration of FA with MR was presented by [5]. Both papers solved their models by employing column generation in a branch-and-bound tree, known as branch-and-price. Furthermore, time-windows were integrated with FA, resulting in important savings [8]. This model was solved using a specialized algorithm which iteratively added "beneficial" time-windows. Additionally, CP was integrated with time windows and plane-count constraints [3]. That model was also solved with a specialized algorithm selecting a "good" subset of pairings and strong branching during the branch-and-bound algorithm.

Although plane-count constraints assist integrating aircraft routing with CP, MR feasibility is not always guaranteed, a fact speculated by [4,9] and also verified by the experiments of the present paper in Section 5.2.4. For this reason, in [10] an exact model combining MR and CP was introduced. The great number of constraints in that model was handled by Benders decomposition, where MR was the master problem. Furthermore, MR was integrated with CP by [11], where it was argued that since crew short-connections are involved one could reduce the problem size by identifying key MR decisions. To be more specific, many routes that are maintenance and short-connection feasible are equivalent, and could be represented by a unique route. Additionally, from the collection of these unique routes there exists a maximal subset that covers most possible legs. Thus, the integrated model was solved using column generation where the columns generated had to be unique and maximal. Concerning the efficiency of the previously mentioned methodology the interested reader is also referred to the theoretical and experimental arguments of [12]. The latter paper also presented an integration of MR and CP, extending the work of [10] with the inclusion of special constraints for robust planning. In that paper it was also proved that it is more efficient reversing the decomposition of [10] by having CP on the Benders master problem. In [9] a model merging time-windows, MR, and CP was solved by Benders decomposition, leading to important cost savings. For the integrated MR and CP with Benders decomposition one typically employs artificial variables to obtain an always feasible primal Benders problem, and avoid feasibility cuts stemming from extreme rays. As [13] stressed, the choice of these artificial variables has an impact on the dual subproblem polyhedron and therefore on the generated cuts.

The closest attempt to integrate all airline scheduling stages was that of [4], integrating FA and CP with plane-count constraints, and not considering MR. They did, however, report important profit increase due to this integration. The authors demonstrated two solution methodologies, in the first they used Benders decomposition and in the second a combination of Lagrangian relaxation and column generation. As mentioned, however, at the beginning of the previous paragraph, the model of [4] is typically not MR feasible.

Finally, in the present paper novel methods are used to speed-up Benders decomposition and column generation, and are described respectively in [14,15].

1.2. Paper contributions and outline

This paper contributes to airline scheduling optimization since:

- fully integrated models are presented for the first time, and a novel, generic, and efficient algorithm is devised for their solution;

- based on realistic data of European and North American airlines, computational results are reported, comparing the integrated with the best known methods from the literature, proving that the integrated significantly reduces costs;
- although different integrated models are introduced, one of them outperforms the rest, showing that there is probably a unique choice amongst them;
- novel plane-count constraints are presented, tighter than those of [4], with the further advantage that, instead of adding them, one can incorporate them in models and maintain the number of total constraints invariant;
- some of the integrated formulations include plane-count constraints, and the computational results demonstrate that these constraints alone cannot ensure MR feasibility.

Regarding the structure of this paper, in Section 2 the most efficient integrated formulation is introduced and Benders decomposition is employed to reduce its size. In Section 3, several techniques are used to accelerate Benders decomposition. Then, in Section 4, the best known methods from the literature are discussed and several alternative integrated formulations are also introduced. Results of the performed computational experiments are reported in Section 5, comparing the most efficient integrated model with the best known methods from the literature as well as with the alternative integrated formulations. Finally, in Section 6, conclusions are drawn and future research directions are suggested.

2. The original integrated model and Benders decomposition

This section introduces the most efficient integrated model of the performed experiments presented in Section 5. The integration consists of *fleet assignment* (FA), *maintenance routing* (MR), and *crew pairing* (CP) problems. Concerning (FA) there are models considering passenger demand of specific legs [16], as well as passenger itineraries [17]. However, the former modeling is used in the present paper for simplicity. Moreover, concerning MR the model considered here is very similar to the one presented by [11]. Finally, for a review of CP the interested reader is referred to [18–20].

2.1. Mathematical formulation

An *aircraft route* is a series of legs to be flown by an individual aircraft of a specific fleet. Therefore, the aircraft flow conservation and count constraints, needed for FA, are typically already implemented in the MR problem. For this reason from the FA constraints of [16], one only needs to include the one restricting each leg to be flown by a single aircraft of a particular fleet. Next, the concepts for modeling MR and CP are summarized below.

2.1.1. Maintenance routing

A *maintenance route* is made of an aircraft route preceded and followed by maintenance procedures, called hereafter *initial* and *terminal maintenance* respectively. The initial and terminal maintenance serve to determine the period during which the aircraft operates without maintenance. According to regulations this period has to be at most 3 or 4 days, while maintenance itself typically lasts for 8 h. It should be noted that [5,11] considered as maintenance opportunities any activity in the network, i.e. landing or taking-off. In this paper however, the freedom to choose from a subset of these opportunities is offered.

Different aircraft have different velocity influencing the arrival time. Moreover, after arrival each aircraft needs a period of time to be prepared for the following flight. The time at which the aircraft is ready for take-off is known as *ready-time*, and is usually fleet dependent. For this reason the maintenance routes of each fleet f are distinct, and therefore each route r_f will belong to a particular set of routes R^f. If a route is known to belong to that set, the fleet subscript may be dropped and then simply written $r \in R^f$.

For each route $r \in R^f$ there is a cost c_r made of maintenance costs minus the revenue generated by through flights. A *through flight* is one where an aircraft is scheduled to fly to a final destination after an intermediate short stop, which is more expensive as passengers do not have to change aircraft. One may also include c_{fl}, the cost associated with the assignment of each leg l to fleet f, which is composed of: fuel and oil costs, landing fees, and loss of revenue for spilling passengers. In this case the joint FA and MR cost will be denoted as $c_r^+ := c_r + \sum_{l \in L} c_{fl} e_{lr}$ for $r \in R^f$—cf. Table 1.

Table 1
Notation of the original model (1)–(9)

F	set of fleets
L	set of legs
M^f	set of maintenance procedures of fleet f (distinct for each fleet)
$\hat{M}^f$	($\subseteq M^f$) maintenance procedures crossing the scheduling horizon
P^f	set of pairings of fleet f (distinct for each fleet)
R^f	set of routes of fleet f (distinct for each fleet)
S^f	set of leg pairs that can be short-connected for fleet f (distinct for each fleet)
q_m	variable counting aircraft on the ground between starting times of m and m^+
v_r	binary variable representing the usage of route r
w_p	binary variable representing the usage of pairing p
a_{lp}	=1 if leg l is in pairing p
c_{fl}	cost of assigning fleet $f \in F$ to leg $l \in L$
c_p, c_r	costs of pairing p and of route r, respectively
c_p^+, c_r^+	costs c_p and c_r that also include the sum of c_{fl} of legs $l \in p$ and $l \in r$, respectively
e_{lr}	=1 if leg l is in route r
e_{mr}^+, e_{mr}^-	=1 if m are, respectively, the initial and terminal maintenance of route r
$\hat{e}_r$	= number of times route r crosses the scheduling horizon (excluding terminal maintenance)
m_f	a maintenance opportunity at a specific station for fleet f
$m \in M^f$	a simplified notation for a maintenance m_f (as M^f are distinct)
m^+, m^-	maintenance on the same station as m, respectively, postdating and predating m
n_f	number of available aircraft for fleet f
p_f	a pairing of fleet f
$p \in P^f$	a simplified notation for a pairing p_f (as P^f are distinct)
r_f	a route of fleet f
$r \in R^f$	a simplified notation for a route r_f (as R^f are distinct)
s_f^{ij}	=1 if MR solution can short-connect legs i and j for fleet f
s_p^{ij}	=1 if pairing p short-connects legs i and j
s_r^{ij}	=1 if route r can short-connect legs i and j
x_{fl}	=1 if fleet $f \in F$ is assigned to leg $l \in L$

2.1.2. *Crew pairing*

Based on the MR solution, the CP problem can be solved for each fleet, where each leg is covered by exactly one pairing. A *crew pairing* is a multi-day schedule starting and terminating at the same crew base. Pairings must respect labor and contractual regulations which usually guarantee maximum: flying time per duty, duration of a duty, and *time away from base*. Additionally, there are regulations on the flying hours per day and on the maximum number of duties per pairing. Furthermore, there are minimum and maximum rest periods between duties as well as between legs of the same duty. The rest between legs is known as *sit-time* or *connection*, and the minimum is here assumed to be 30 min. The minimum sit-time is the time required for crew to travel within the airport to board to their next leg, and it is computed on top of the ready-time [10,12]. The minimum sit-time can be violated when crew remain on the same aircraft for their next flight, and such information is provided by the MR. The crew connection in violation of the minimum sit-time rule is known as a *short-connection*.

A pairing p_f will belong to a set of pairings P^f of fleet f, which sets are distinct for the same reason sets R^f were. Thus, once known that a pairing belongs to a specific set P^f the fleet subscript may be dropped by simply writing $p \in P^f$. Concerning crew costs, for each individual pairing there is a minimum guaranteed pay per duty and per flying time, a guaranteed percentage of duty time counting as flying time, and a per diem away from base payment. The cost of a pairing p will be noted as c_p, and if the FA costs of the legs in p are also accounted, one will write $c_p^+ := c_p + \sum_{l \in L} c_{fl} a_{lp}$—cf. Table 1.

2.1.3. *The model*

The original integrated airline scheduling model introduced here is

$$\min \sum_{f \in F} \sum_{r \in R^f} c_r^+ v_r + \sum_{f \in F} \sum_{p \in P^f} c_p w_p, \tag{1}$$

$$\text{subject to} \quad \sum_{f\in F}\sum_{r\in R^f} e_{lr}v_r = 1, \quad \forall l \in L, \tag{2}$$

$$\sum_{p\in P^f} a_{lp}w_p - \sum_{r\in R^f} e_{lr}v_r = 0, \quad \forall l \in L, \ \forall f \in F, \tag{3}$$

$$\sum_{p\in P^f} s_p^{ij}w_p - \sum_{r\in R^f} s_r^{ij}v_r \leqslant 0, \quad \forall (i,j) \in S^f, \ \forall f \in F, \tag{4}$$

$$q_m - q_{m^-} + \sum_{r\in R^f} (e_{mr}^+ - e_{mr}^-)v_r = 0, \quad \forall m \in M^f, \ \forall f \in F, \tag{5}$$

$$\sum_{m\in \hat{M}^f} q_m + \sum_{r\in R^f} \hat{e}_r v_r \leqslant n_f, \quad \forall f \in F, \tag{6}$$

$$v_r \in \{0,1\}, \quad \forall r \in R^f, \ \forall f \in F, \tag{7}$$

$$q_m \geqslant 0, \quad \forall m \in M^f, \ \forall f \in F, \tag{8}$$

$$w_p \in \{0,1\}, \quad \forall p \in P^f, \ \forall f \in F, \tag{9}$$

where all the symbols used above are explained in Table 1. The goal (1) of the model is minimizing the total costs, and constraint (2) ensures that each leg is assigned to exactly one route of one fleet. Furthermore, it is easy to see that

$$x_{fl} := \sum_{r\in R^f} e_{lr}v_r, \quad f \in F, \ l \in L \tag{10}$$

is a binary constant indicating whether fleet f is assigned to leg l. Therefore, constraint (3) is assigning leg l to be flown by exactly one crew pairing of fleet f, if and only if that leg is assigned to that fleet. Additionally, constraint (4) allows a crew short-connection from leg i to leg j if and only if there is an aircraft flying those legs consecutively. Finally, constraint (5) conserves flow of routes consecutively operated by the same aircraft, and constraint (6) restricts the used aircraft to be at most the number of available ones.

2.2. Benders subproblem

The integrated model (1)–(9) has a large number of constraints and Benders decomposition [21] is employed for its reduction. This method has recently been successfully used for the integrated MR with CP problem [10,12], and its extension that included time-windows [9]. There are two natural decompositions of the integrated model (1)–(9), one having MR in the Benders subproblem and the other the CP. In the present section the latter is demonstrated, while the former, that was not that efficient in the performed experiments, is discussed in Section 4.2.

Therefore, the Benders subproblem for each fleet $f \in F$ is here

$$\min \sum_{p\in P^f} c_p w_p, \tag{11}$$

$$\text{subject to} \quad \sum_{p\in P^f} a_{lp}w_p = \bar{x}_{fl}, \quad \forall l \in L, \tag{12}$$

$$\sum_{p\in P^f} s_p^{ij}w_p \leqslant \bar{s}_f^{ij}, \quad \forall (i,j) \in S^f, \tag{13}$$

$$w_p \geqslant 0, \quad \forall p \in P^f, \tag{14}$$

where $\bar{\mathbf{x}} \equiv (\bar{x}_{fl} | \forall l \in L, \forall f \in F)$ is computed with the assistance of definition (10) and $\bar{\mathbf{s}} \equiv (\bar{s}_f^{ij} | \forall (i,j) \in S^f, \forall f \in F)$ with the assistance of

$$\bar{s}_f^{ij} := \sum_{r \in R^f} s_r^{ij} \bar{v}_r, \quad (i,j) \in S^f, \quad f \in F, \tag{15}$$

given $\bar{v}_r$ that satisfy constraints (2) and (5)–(8).

Model (11)–(14) is a CP problem with short-connections, and is typically solved using the column generation decomposition algorithm [3,9,10,12]. The problem is decomposed into a column generation master problem and a subproblem. On the ith iteration the former is basically problem (11)–(14) with a restricted set of pairings $P_i^f \subseteq P^f$ in the place of P^f. The column generation subproblem is a resource-constrained shortest path problem in the so-called *crew-connection network*—see, e.g. [12]. This network contains all legs to be flown by the crew; paths in this network are potentially valid crew pairings. In order for the algorithm to verify whether any path is indeed a valid crew pairing, for each path reaching a node, labels are used. These labels model the resource usage of CP and enforce the respective constraints on them [18,19,22]. Fortunately, not all possible paths need to be stored, but by considering a *dominance relation* between the paths' labels and costs, one eliminates those paths whose extension cannot lead to shorter paths than the rest [23,24].

The column generation algorithm iterates between the master problem and the subproblem. If in the subproblem no pairing has a negative reduced cost, then the optimal solution of the column generation master problem is the optimal one for the full problem. Otherwise, the pairings with negative reduced cost need to be introduced in P_{i+1}^f, as with the simplex method. The reduced cost is obtained by $\bar{c}_p = c_p - \sum_{l \in L} \alpha_l a_{lp} - \sum_{(i,j) \in S^f} \varsigma^{ij} s_p^{ij}$, where $\alpha_l, \forall l \in L$, and $\varsigma^{ij}, \forall (i,j) \in S^f$, are the dual values corresponding to constraints (12) and (13). In column generation, typically one uses the Dantzig pricing, where pairings with the most negative reduced cost are introduced.

2.2.1. Accelerating column generation

There are two heuristic methods employed here to speed-up column generation. The first method relaxes the dominance relation for some of the labels. To be more specific, two predetermined labels are not considered in the dominance comparison of paths. The first label is the sum of the time flown in the previous duties, and the second is the flown time during the last 24 h. This method is used every time the column generation subproblem of CP is solved.

The second heuristic method accelerating column generation concerns the utilization of deepest-cut pricing [25,26], instead of Dantzig. In deepest-cut pricing one adds the pairings with the most negative:

$$\bar{c}_p^{\mathrm{DC}} := \frac{\bar{c}_p}{\sqrt{1 + \sum_{l \in L} (a_{lp})^2 + \sum_{(i,j) \in S^f} (s_p^{ij})^2}}, \quad p \in P^f. \tag{16}$$

To understand the intuition behind the deepest-cut, one should view the problem in the dual space. The deepest-cut method reduces the dual space as much as possible, by finding the dual cut whose hyperplane has the maximum distance from the dual solution. Criterion (16) is easy to compute dynamically given a pairing p, since $\sum_{l \in L} (a_{lp})^2$ and $\sum_{(i,j) \in S^f} (s_p^{ij})^2$ are, respectively, equal to the number of legs and the number of short-connections in pairing p. The deepest-cut pricing is heuristically implemented in the resource-constrained shortest path algorithm, where $\bar{c}_p^{\mathrm{DC}}$ can be computed for each non-dominated path p during the process. This way, upon termination of the algorithm, one firstly knows the pairing with the most negative reduced cost, and secondly can find a pairing with a relatively "good" deepest-cut value, although probably not optimal. The possibility of using the deepest-cut pricing in column generation was also discussed by [25,26]. In the present implementation deepest-cut pricing proved to be on average three times faster than the classic Dantzig pricing, for the LP solution of the Benders master problem presented next [14].

2.3. Benders master problem

The dual Benders subproblem is utilized to obtain the Benders cuts [21], which for problem (11)–(14) and each fleet $f \in F$ is

$$\max \sum_{l \in L} \bar{x}_{fl} \alpha_l + \sum\nolimits_{(i,j) \in S^f} \bar{s}_f^{ij} \varsigma^{ij} \tag{17}$$

$$\text{subject to} \quad \forall(\boldsymbol{\alpha}, \varsigma) \in \Pi^f, \tag{18}$$

where $\boldsymbol{\alpha} \equiv (\alpha_l | \forall l \in L)$ and $\varsigma \equiv (\varsigma^{ij} | \forall (i, j) \in S^f)$, and Π^f is the polyhedron defined by the constraints of the dual subproblem. The Benders master problem is formulated by replacing in the airline scheduling model (1)–(9) the CP constraints (3), (4) and (9), with Benders cuts. One should also replace the CP cost for each fleet $f \in F$ in objective (1) with a variable z_f. Thus, the Benders master problem is

$$\min \sum_{f \in F} \sum_{r \in R^f} c_r^+ v_r + \sum_{f \in F} z_f, \tag{19}$$

$$\text{subject to} \quad \sum_{f \in F} \sum_{r \in R^f} e_{lr} v_r = 1, \quad \forall l \in L, \tag{20}$$

$$q_m - q_{m^-} + \sum_{r \in R^f} (e_{mr}^+ - e_{mr}^-) v_r = 0, \quad \forall m \in M^f, \ \forall f \in F, \tag{21}$$

$$\sum_{m \in \hat{M}^f} q_m + \sum_{r \in R^f} \hat{e}_r v_r \leqslant n_f, \quad \forall f \in F, \tag{22}$$

$$-z_f + \sum_{r \in R^f} \left(\sum_{l \in L} \alpha_l e_{lr} + \sum_{(i,j) \in S^f} \varsigma^{ij} s_r^{ij} \right) v_r \leqslant 0, \quad \forall (\boldsymbol{\alpha}, \varsigma) \in \Pi^f_{\text{points}}, \ \forall f \in F, \tag{23}$$

$$\sum_{r \in R^f} \left(\sum_{l \in L} \alpha_l e_{lr} + \sum_{(i,j) \in S^f} \varsigma^{ij} s_r^{ij} \right) v_r \leqslant 0, \quad \forall (\boldsymbol{\alpha}, \varsigma) \in \Pi^f_{\text{rays}}, \ \forall f \in F, \tag{24}$$

$$v_r \in \{0, 1\}, \quad \forall r \in R^f, \ \forall f \in F, \tag{25}$$

$$q_m \geqslant 0, \quad \forall m \in M^f, \ \forall f \in F, \tag{26}$$

where $\Pi^f_{\text{points}} \subseteq \Pi^f$ and $\Pi^f_{\text{rays}} \subseteq \Pi^f$ are, respectively, the sets of extreme points and extreme rays of Π^f. The Benders cuts (23) and (24) are imposed by weak duality theorem applied on objectives (11) and (17), and by using definitions (10) and (15). Notice that $\bar{\mathbf{x}}$ and $\bar{\mathbf{s}}$ were expanded from objective (17), using the definitions (10) and (15) to acquire cuts (23) and (24). The Benders master problem (19)–(26) is solved with the column generation method of [5], which is here accelerated using the heuristic deepest-cut pricing described in Section 2.2.1.

More specifically, the column generation master problem is formulated by substituting R^f with a subset R_0^f. Furthermore, the column generation subproblem is a resource-constrained shortest path problem in the so-called *aircraft-connection network* for each fleet— see, e.g. [12]. The latter network is similar to the crew-connection network, but it additionally includes the maintenance procedures; paths in that network are potential aircraft routes.

Whenever the full IP problem (19)–(26) is solved, a branch-and-bound tree is formed and on each node of the tree the LP relaxation is solved with column generation. The IP solution is accomplished in two stages. In the first stage, with the assistance of definition (10) the x_{fl} with a value closer to 1 is found, and two branches are created: one where that particular fleet f is assigned to leg l ($x_{fl} = 1$), and another where this assignment is disallowed ($x_{fl} = 0$). The two branches are created by setting the appropriate variables v_r to null and deleting the corresponding legs from the aircraft-connection network [5]. When the first stage of the IP algorithm is terminated the sets of legs assigned to each fleet are determined. Therefore, one has to solve individual MR problems for each fleet. For each individual fleet, one finds all fractional v_r variables and determines the *follow-on legs*, which are pairs of successive legs in the routes. After choosing the most attractive follow-on pair one branches: on one side by imposing these legs to be followed-on, and on the other by disallowing them. This is accomplished by setting the appropriate variables v_r to null and deleting the corresponding arcs connecting these legs in the aircraft-connection network [5].

In the implemented algorithm, the search is depth-first and terminates upon the first encountered solution. Such IP heuristics are not uncommon in the airline scheduling literature, and have already been successfully applied by

[5,9,10,12]. Their success lies on the fact that the experiments have shown short IP solution-times with surprisingly low optimality gaps.

2.4. The Benders algorithm

The Benders algorithm is dual to column generation in the sense that "beneficial" constraints are progressively added to a restricted master problem. Thus, one starts with empty $\Pi^f_{\text{points},0}$ and $\Pi^f_{\text{rays},0}$ sets of cuts, and on the ith iteration the restricted Benders master problem is initially solved. Then, the acquired solution $\bar{v}_r$, with the assistance of definitions (10) and (15), gives $\bar{\mathbf{x}}$ and $\bar{\mathbf{s}}$, which are passed to the Benders subproblems. In cases where column generation is utilized in the Benders subproblem, however, one solves the primal one—here (11)–(14)—instead of the dual [9,10,12]. From the solution of this subproblem, the dual values $(\boldsymbol{\alpha}, \varsigma)$ are acquired, and if the problem is feasible they are added in $\Pi^f_{\text{points},i+1}$ generating an *optimality cut*; otherwise they are added in $\Pi^f_{\text{rays},i+1}$ generating a *feasibility cut*. The Benders subproblem of each fleet is here the CP problem (11)–(14) and it is typically transformed to an always feasible problem by adding artificial columns with high costs, and solved with column generation—cf. [13]. If these artificial columns are in the optimal basis of the column generation master problem, then the original problem is infeasible. In this case the acquired dual values are not extreme rays but rather extreme points approximating them. For the rest of this paper the cuts derived from an originally infeasible subproblem will still be named feasibility cuts.

The Benders algorithm terminates when no more cuts need to be generated or if the Benders master problem is infeasible; the latter case implies that the full problem is infeasible [21]. In airline scheduling one is usually content with near-optimal solutions and the algorithm can terminate when the gap between the upper and lower bound is less than a user defined constant [12,27]. In the present implementation the algorithm terminates when no more cuts have to be generated or enough savings have been achieved, as discussed next.

2.4.1. Algorithm termination criteria

An *optimality cut* for a specific fleet $f \in F$ is generated when the ratio of the difference between the upper and lower bound of the corresponding CP problem to the lower bound of the Benders master problem is more than a user defined constant h_f,

$$\frac{z(\bar{w}^i_{p_f}) - z_f}{\max_i z(\bar{v}^i_r)} > h_f, \tag{27}$$

where $z(\bar{w}^i_{p_f})$ and $z(\bar{v}^i_r)$ are, respectively, the objective values of CP problem (11)–(14) and of the Benders master problem (19)–(26), on the ith Benders iteration.

Furthermore, one can terminate the Benders algorithm when there are relatively small potential savings due to the integration. A rough estimate of such savings is obtained by observing that the solution of the Benders master problem on the first iteration corresponds to that of a sequential method, due to the absence of any cuts. In that sequential method the integrated FA with MR problem is followed by the solution of each CP with short-connections problem. The potential savings that can be generated through the rest of the iterations are given by the sum of the differences between the upper and lower bounds of the CP problems of each fleet. The criterion to continue the Benders iterations in the present implementation is

$$\frac{\sum_{f\in F}[z(\bar{w}^i_{p_f}) - z_f]}{z^{\text{Seq}} - z(\bar{v}^1_r)} > h, \tag{28}$$

where $z^{\text{Seq}} := \sum_{f\in F}\sum_{r\in R^f} c^+_r \bar{v}^1_r + \sum_{f\in F}\sum_{p\in P^f} c_p \bar{w}^1_p$ is the integrated cost of the first iteration obtained by objective (1), and h is a user defined constant.

3. Accelerating Benders decomposition

3.1. The Magnanti–Wong method

The Benders subproblem (11)–(14) has a set partitioning structure, making it degenerate, and as a result it has multiple dual solutions. Hence, it is possible to choose among different cuts (23) that might actually have different strengths [28]. One therefore needs to define a relation comparing the strength of cuts corresponding to different dual values $(\boldsymbol{\alpha}, \boldsymbol{\varsigma})$. Given two cuts $(\boldsymbol{\alpha}^1, \boldsymbol{\varsigma}^1)$ and $(\boldsymbol{\alpha}^2, \boldsymbol{\varsigma}^2)$, it is said that the first *dominates* the second if and only if

$$\sum_{l \in L} \alpha_l^1 x_{fl} + \sum_{(i,j) \in S^f} \varsigma^{1,ij} s_f^{ij} \geqslant \sum_{l \in L} \alpha_l^2 x_{fl} + \sum\nolimits_{(i,j) \in S^f} \varsigma^{2,ij} s_f^{ij}, \quad \forall (\mathbf{x}, \mathbf{s}) \in (X, S), \tag{29}$$

with strict inequality for at least one point $(\mathbf{x}, \mathbf{s}) \in (X, S)$, where sets X and S are made of points defined in (10) and (15) for $v_r \geqslant 0$. A cut is said to be *Pareto-optimal* if no other cut dominates it. Notice that in definition (29) the notation provided in (17) was used in the place of that in (23).

Magnanti and Wong introduced a problem that assists in computing a Pareto-optimal cut [28]. Their problem requires:

- an optimal solution of the Benders subproblem;
- a Benders master problem core point.

A point $x \in X$ is a *core point* if $x \in ri(X^c)$, where $ri(X)$ and X^c are, respectively, the relative interior and the convex hull of set X. Both requirements are too stringent for the efficiency of the Magnanti–Wong method solving the decomposed model presented in Sections 2.2–2.4.

3.2. Practical enhancements to the Magnanti–Wong method

3.2.1. Optimal subproblem solution requirement

Concerning the former requirement, it should be noticed that the Magnanti–Wong problem is similar to the dual Benders subproblem. As explained in Section 2.4, however, the Benders primal subproblem is solved instead of the dual. For the same reason, the dual Magnanti–Wong problem [28] will be solved here, as [12] did for the integrated MR with CP problem and [29] for a problem with a similar structure. The dual Magnanti–Wong problem is for efficiency reasons also solved with column generation which is typically terminated to a near-optimal solution, due to the *tailing-off* effect [30]. It was shown however, both mathematically and experimentally, that if one provides the dual Magnanti–Wong problem with a suboptimal solution of the Benders subproblem, the former problem becomes numerically unbounded [15].

This numerical unboundedness was circumvented by introducing a variation of the Magnanti–Wong problem [15]. It was proven that given a Benders master problem core point $(\mathbf{x}^0, \mathbf{s}^0) \in (X, S)$, the optimal solution w_p^0 of the following problem, for each fleet $f \in F$:

$$\min \sum_{p \in P^f} c_p w_p, \tag{30}$$

$$\text{subject to} \quad \sum_{p \in P^f} a_{lp} w_p = x_{fl}^0, \quad \forall l \in L, \tag{31}$$

$$\sum_{p \in P^f} s_p^{ij} w_p \leqslant s_f^{0,ij}, \quad \forall (i,j) \in S^f, \tag{32}$$

$$w_p \geqslant 0, \quad \forall p \in P^f, \tag{33}$$

corresponds to a dual solution $(\boldsymbol{\alpha}^0, \boldsymbol{\varsigma}^0)$ whose cut is Pareto-optimal. Notice that the above Magnanti–Wong problem (30)–(33) is independent of the Benders subproblem solution, allowing the more efficient near optimal solutions.

3.2.2. Master problem core point requirement

Concerning the core point requirement, it should be stressed that in practice there are no generic methods available for computing it [4,9,12,31], and one usually has to rely on approximations instead. The approximation chosen in this paper is computed by

$$x_{fl}^0 \leftarrow g \cdot x_{fl}^0 + (1-g) \cdot \bar{x}_{fl}, \quad \forall l \in L, \ \forall f \in F, \tag{34}$$

$$s_f^{0,ij} \leftarrow \min\{g \cdot s_f^{0,ij} + (1-g) \cdot \bar{s}_f^{ij}, \min\{x_{fi}^0, x_{fj}^0\}\}, \quad \forall (i,j) \in S^f, \ \forall f \in F, \tag{35}$$

where $(\mathbf{x}^0, \mathbf{s}^0)$ is the core-point approximation of the previous iteration, and $(\bar{\mathbf{x}}, \bar{\mathbf{s}})$ is the current solution. Initially $(\mathbf{x}^0, \mathbf{s}^0)$ is set equal to some approximate solution; for instance $\mathbf{x}^0$ is quickly found by solving the LP relaxation of the FA problem, and $s_f^{0,ij} \leftarrow \min\{x_{fi}^0, x_{fj}^0\}$; the latter is similar to the approximation of [12]. Given the convexity of the LP relaxation, with the above approximation one obtains a point between two solutions: $(\mathbf{x}^0, \mathbf{s}^0)$ and $(\bar{\mathbf{x}}, \bar{\mathbf{s}})$, where an interior point is most likely to be found. Experiments show that $g = \frac{1}{2}$ is the most efficient choice.

Instead of solely relying on core point approximations, it was proven [15] that for the present problem, by setting $\mathbf{x}^0 \leftarrow \mathbf{1}$ in the Magnanti–Wong, one can still get a Pareto-optimal cut. Furthermore, for the integrated MR with CP problem it was found [12] that if one sets $\mathbf{s}^0 \leftarrow \mathbf{1} - \boldsymbol{\epsilon}$ (where $\boldsymbol{\epsilon}$ is a vector with numbers close to zero) it is possible to acquire an efficient algorithm. For these reasons, in the present implementation the following initial conditions are used:

$$x_{fl}^0 \leftarrow 1, \quad \forall l \in L, \ \ \forall f \in F, \tag{36}$$

$$s_f^{0,ij} \leftarrow 1, \quad \forall (i,j) \in S^f, \ \ \forall f \in F. \tag{37}$$

Notice that the above, usually, is not even a point—a solution of the Benders master problem—as typically not all aircraft can be assigned to a single fleet.

It is important to stress that the methodology described by equations (34)–(37) was compared with approximations of core points—including the independent Magnanti–Wong problem (30)–(33)—and it was found to reduce *CPU* times by 70% on average [15].

3.3. The three–phase algorithm

Column generation typically has a large number of variables which makes the IP search tree even larger. Classic Benders decomposition is known to under-perform in such cases, for which reason a two-phase Benders method was introduced [27], and extended to a three–phase for the case where the Benders subproblem is also IP [10,12]. During each phase of the latter method, the Benders algorithm is executed iterating between the master and subproblems, with the difference that: in Phase 1 the IP constraints of both problems are relaxed, in Phase 2 the IP constraints are only introduced for the master problem, and finally, in Phase 3 the IP constraints are introduced in the subproblem. Phase 3 is typically executed only once, unless infeasibilities are encountered, and is therefore incomplete. The interested reader is referred to [32] for a complete method in Phase 3 additionally generating strong cuts. Their method is out of this paper's scope, as in the present experiments Phase 3 had only to be solved once without generating any cuts.

The methods discussed so far can be used to formulate the extension of the three–phase Benders algorithm, which includes Pareto-optimal cuts [12,29]. Based on the fact that the Magnanti–Wong problem (30)–(33) is independent of the subproblem solution, and by using (36)–(37) for the initialization of approximations (34)–(35), one may change the execution order within the Magnanti–Wong iterations. Thus, one can first generate a Pareto-optimal cut, then solve the Benders master problem, and finally the subproblem. In this sense the algorithm is:

Initialization: set $\mathbf{x}^0 \leftarrow \mathbf{1}$ and $\mathbf{s}^0 \leftarrow \mathbf{1}$; LP relax both Benders problems
Phase 1: generate a Pareto-optimal cut solving (30)–(33) for the given $(\mathbf{x}^0, \mathbf{s}^0)$; solve the Benders master problem and then the subproblem
  if (all subproblems are feasible **and** criterion (28), $h = 1/8$, is not met**)**
    or no new cuts can be generated due to violation of criterion (27), $h_f = 0.01/100$
  then introduce IP constraints on Benders master problem; **goto Phase 2**
  else use (34)–(35) to compute $(\mathbf{x}^0, \mathbf{s}^0)$; **goto Phase 1**
Phase 2: generate a Pareto-optimal cut solving (30)–(33) for the given $(\mathbf{x}^0, \mathbf{s}^0)$; solve the IP Benders master problem and then the subproblem
  if (all subproblems are feasible **and** criterion (28), $h = 1/2$, is not met**)**
    or no new cuts can be generated due to violation of criterion (27), $h_f = 0.1/100$
  then introduce IP constraints on Benders subproblem; **goto Phase 3**
  else use (34)–(35) to compute $(\mathbf{x}^0, \mathbf{s}^0)$; **goto Phase 2**
Phase 3: solve the IP Benders subproblem
  if all subproblems are feasible
  then terminate the algorithm with IP solutions for both problems
  else add a Pareto-optimal cut (24), remove IP constraints from Benders subproblem; **goto Phase 2**

It should be noticed that in practice the above computed points might not be core points, and hence the corresponding cut does not necessarily eliminate the current Benders master problem solution. Therefore, in order to guarantee convergence of the algorithm one could periodically add the cuts of the Benders subproblem too, since the corresponding dual values have already been computed.

4. Antagonistic models and methods

This section discusses airline scheduling methods and formulations to be compared with the original integrated approach introduced in Section 2 and 3. More specifically, semi-integrated and sequential methods known from the literature are presented, followed by the introduction of alternative integrated formulations.

4.1. Best known methods from the literature

The best method available from the literature, solving the airline scheduling problem, is a combination of semi-integrated models executed in a sequential manner. This method consists of initially solving the integrated FA with MR problem [5] and feeding the acquired solution of each fleet into the integrated MR with CP problem [12]; this method will be referred to hereafter as *semi-integrated*. Both of the integrated models included in this method are special cases of the integrated airline scheduling model (1)–(9). For this reason instead of the model of [5], the Benders master problem (19)–(26) with empty Π^f_{points} and Π^f_{rays} sets is solved using the algorithm discussed in Section 2.3. This does not have to be implemented anew since it is the solution of the Benders master problem on the first iteration if one starts immediately from Phase 2 of the algorithm presented in Section 3.3. Furthermore, in order to assist the solution of this model towards low CP costs, the crew costs per flying-time for each fleet f and each leg l are temporarily added to the c_{fl} costs of FA. Of course these costs are removed before feeding the solution to the combined MR with CP problem. Finally, the integrated MR with CP problem is nothing more than model (1)–(9) for a single fleet, and the methodology of the previous sections may be used, as it coincides with that of [12], for the special case of a single fleet.

To further compare the integrated method's savings with those savings coming from the semi-integrated method, the next-best method has to be used too. In this instance, the results of the integrated FA with MR problem for each fleet are fed to the CP with short-connections problem (11)–(14), where w_p must be integer; this method is referred to hereafter as *sequential*. This CP model is once more a special case of model (1)–(9), and it is solved by Phase 3 of the algorithm discussed in Section 3.3.

4.2. Introducing alternative integrated formulations

In Sections 2.2 and 2.3 the integrated model (1)–(9) was decomposed having the CP with short-connections as a Benders subproblem. From the work of [12] it is known that for the integrated MR and CP problem, it is more efficient having MR with short-connections in the Benders subproblem. It would therefore be interesting to experiment by decomposing the fully integrated model (1)–(9), so that MR with short-connections is in the Benders subproblem too. For this reason [33] used the following constraint:

$$\sum_{f \in F} \sum_{p \in P^f} a_{lp} w_p = 1, \quad \forall l \in L, \tag{38}$$

which assigns one fleet per leg, and constraint (9), in the Benders master problem. Additionally, in order to keep more information in the Benders master problem, and have an efficient solution, one would need to place the majority of the costs—FA and CP—on the Benders master problem, and the rest—related to through flights of MR—in the subproblem.

The experiments, however, showed that even small instances of about 100 legs were immensely difficult to solve even with the help of the enhanced Magnanti–Wong method. Note that, in this decomposition, on the Benders master problem crew pairings were assigned; these optimal pairings, however, were using more aircraft than available for certain fleets and were rendering the Benders subproblem infeasible. Therefore, the inefficiency of the inverse decomposition may be attributed to the lack of information concerning the available aircraft to be scheduled on the Benders master problem.

4.2.1. Introducing FA constraints

One way to tackle the previously mentioned issue is by strengthening the Benders master problem with additional FA constraints, and therefore extending the original model (1)–(9). A good candidate for this is the ready-time network of [16], created for each fleet separately, and where each network's node is a leg's departure or ready-time for every station. Moreover, the legs are represented by arcs connecting different stations' nodes, and the nodes of each station are connected in their time order with the so called *ground arcs*. The time-line is wrapped around the scheduling horizon. Flow in the leg arcs identifies whether a leg is assigned to that fleet, and flow on ground arcs counts the number of aircraft on ground at the given period. The former arcs are modeled by constants x_{fl}, which with the help of equations (3) and (10), are $x_{fl} = \sum_{p \in P^f} a_{lp} w_p$. As a result one obtains the following integrated model:

$$\min \sum_{f \in F} \sum_{p \in P^f} c_p^+ w_p, \tag{39}$$

$$\text{subject to} \quad \sum_{f \in F} \sum_{p \in P^f} a_{lp} w_p = 1, \quad \forall l \in L, \tag{40}$$

$$\sum_{p \in P^f} b_{stp} w_p + y_{fstt^+} - y_{fst^-t} = 0, \quad \forall fst \in N, \tag{41}$$

$$\sum_{p \in P^f} d_p w_p + \sum_{s \in S} y_{fs\hat{t}^-\hat{t}^+} \leqslant n_f, \quad \forall f \in F, \tag{42}$$

$$\sum_{p \in P^f} a_{lp} w_p - \sum_{r \in R^f} e_{lr} v_r = 0, \quad \forall l \in L, \ \forall f \in F, \tag{43}$$

$$\sum_{p \in P^f} s_p^{ij} w_p - \sum_{r \in R^f} s_r^{ij} v_r \leqslant 0, \quad \forall (i,j) \in S^f, \ \forall f \in F, \tag{44}$$

$$q_m - q_{m^-} + \sum_{r \in R^f} (e_{mr}^+ - e_{mr}^-) v_r = 0, \quad \forall m \in M^f, \ \forall f \in F, \tag{45}$$

$$\sum_{m \in \hat{M}^f} q_m + \sum_{r \in R^f} \hat{e}_r v_r \leqslant n_f, \quad \forall f \in F, \tag{46}$$

$$v_r \in \{0,1\}, \quad \forall r \in R^f, \ \forall f \in F, \tag{47}$$

$$q_m \geqslant 0, \quad \forall m \in M^f, \ \forall f \in F, \tag{48}$$

$$w_p \in \{0,1\}, \quad \forall p \in P^f, \forall f \in F, \tag{49}$$

$$y_{fstt^+} \geqslant 0, \quad \forall fst \in N, \tag{50}$$

where all new symbols used above are explained in Table 2. The following definitions are also used:

$$b_{stp} := \sum_{l \in D^{fst}} a_{lp} - \sum_{l \in A^{fst}} a_{lp}, \quad p \in P^f, \ f \in F, \tag{51}$$

$$d_p := \sum_{l \in \hat{L}^f} a_{lp}, \quad p \in P^f, \ f \in F. \tag{52}$$

The integrated model (39)–(50) is similar to the original model (1)–(9). In the former model the objective is now simplified so that the MR problem is only a feasibility problem, and therefore the objective function (39) does not include the respective costs. Constraint (40) is essentially the same as constraint (2)–through constraint (3)—and constraints (43)–(49) correspond to constraints (3)–(9). Concerning the additional constraints (41) and (42), the former conserves the flow in the ready-time network and the latter restricts the aircraft used in that network to be no more than the available per fleet. The former is achieved with the assistance of formula $\sum_{p \in P^f} b_{stp}$, which gives the number of legs outgoing minus those incoming at node fst—cf. definitions (51) and (10)—and the latter is achieved since

Table 2
Notation of the alternative models

A^{fst}, D^{fst}	set of legs ready-to-depart (after arrival) and departing, respectively, from node fst
$\hat{L}^f$	set of legs crossing the count-time $\hat{t}$
N	set of nodes fst
$y_{fst_1t_2}$	variable counting aircraft on ground arc fst_1t_2
$a_{pst_1t_2}$	= number of times pairing $p \in P^f$ includes a short-connection forcing aircraft on ground arc fst_1t_2
fst	a node indexed by fleet f, station s, and time t
fst_1t_2	ground arc of fleet f, station s between times $[t_1, t_2)$
t^+, t^-	are, respectively, the times succeeding and preceding time t
$\hat{t}$	the *count-time*, an arbitrary time chosen to measure the total number of aircraft
t'	the arrival time corresponding to ready-time t

formula $\sum_{p\in P^f} d_p w_p$ gives the number of aircraft flying during the count-time—cf. definitions (52) and (10)—while formula $\sum_{s\in S} y_{fs\hat{t}^-\hat{t}^+}$ is the number of legs on ground at that time.

Model (39)–(50) is solved using Benders decomposition with constraints (43)–(48) formulating the subproblem. The latter is a feasibility problem, hence only feasibility cuts are generated. The enhanced Magnanti–Wong method can be used here too; it was proven by the experiments, however, that instead of the approximation (37) it would be better setting the relevant short-connection variables ($\check{s}_f^{ij} := \sum_{p\in P^f} s_p^{ij} w_p$) close to null, a result in line with the observations of [12] when MR with short-connections is a Benders subproblem that only generates feasibility cuts.

4.2.2. Further incorporating plane-count constraints

One can further strengthen the alternative model (39)–(50) by enhancing the aircraft routing information with the incorporation of plane-count constraints [3,4]. These constraints work on variables y_{fstt^+}, which give the number of aircraft on ground. The fact that a pairing which includes a short-connection forces the same aircraft to be used is also reflected on the number of aircraft that are on the ground [3,4]. In this paper the following constraint is proposed to be added on model (39)–(50):

$$\sum_{p\in P^f} a_{pstt^+} w_p \leqslant y_{fstt^+}, \quad \forall fst \in N, \tag{53}$$

cf. Table 2. Notice that the above constraint also obeys constraint (50) since both a_{pstt^+} and w_p are non-negative by definition. It is therefore possible to maintain the magnitude of the integrated model by incorporating, instead of adding, the plane-count constraints. In order to achieve this, one can rewrite constraint (53) by defining some new variables

$$\tilde{y}_{fstt^+} := y_{fstt^+} - \sum_{p\in P^f} a_{pstt^+} w_p, \quad fst \in N, \tag{54}$$

$$\tilde{y}_{fstt^+} \geqslant 0, \quad \forall fst \in N. \tag{55}$$

Using definition (54) one can substitute y_{fstt^+} in model (39)–(50). Since as already commented constraint (50) is redundant in the presence of constraint (55), and the former may be removed. Therefore the integrated model which includes plane-count constraints is equivalent to model (39)–(50) where one needs to substitute y_{fstt^+} with $\tilde{y}_{fstt^+}$. Hence, one may use the same decomposition and algorithms used for model (39)–(50) to solve the problem with the plane-count constraints incorporated.

It is important to stress that constraint (53) differs from that introduced by [4]; the latter is

$$\sum_{p\in P^f} a_{pst't'^+} w_p - \sum_{\tilde{t}=t'^+}^{t} \sum_{l\in D^{fs\tilde{t}}} a_{lp} w_p \leqslant y_{fstt^+}, \quad \forall fst \in N, \tag{56}$$

cf. Table 2. The above constraint does not always obey constraint (50) and additionally it is easy to prove that it is weaker than constraint (53). Since constraint (50) is not obeyed, the plane-count constraints of [4] have to be added for every node of the ready-time network. This renders their model 40% larger, a fact that made it much slower in the experiments of the present paper.

Finally, notice that the model of [4] was proposed to be used when the maintenance requirements are easily fulfilled. However, the above model—with the incorporated plane-count constraints—can be applied in such practical applications and ensure that one obtains a definite solution, avoiding the possibility of infeasibilities.

5. Computational experiments

The algorithms presented in the previous sections were implemented in order to evaluate the benefits of the integrated methodology. Thus, the resource-constrained shortest path algorithms were implemented in C++, and the rest of the algorithms in ECL^{i}PS^e v5.8 [34]. ECL^{i}PS^e is a constraint logic programming language of Cisco Technology Inc., and includes a column and cut generation library that can use different linear solvers [35]; in the present paper ILOG CPLEX 9.030 [36] was chosen. The experiments were performed on a single computer having a single-core 64-bit AMD Athlon™ processor at 2.4 GHz, with 2 GB of RAM memory, and running the 64-bit Linux kernel version 2.6.11.

5.1. The data sets

The data sets used for the experiments of this paper were derived from those provided by a major European and a major North American airline. The schedule of the former is medium-haul with a single central hub, while the schedule of the latter is long-haul with a hub-and-spoke structure. *Hub-and-spoke* networks are characterized by high activity in hub stations, which are connected with each other as well as with spoke stations with lower activity. In the European instance, almost all the legs depart from or arrive at that central hub. The European airline is scheduling in total 372 legs per day, and the American over 2100. In both cases six fleets are considered, and although this is accurate in the case of the European airline it is not for the American. In the latter case different fleets were merged into fleet groups having similar cost characteristics and ready-times.

To enable examination of the introduced algorithms' scalability, reduced data sets had to be generated by eliminating some of the present stations. Moreover, the number of aircraft for each fleet was kept in proportion to the number of legs. The various instances considered in this paper are shown in Table 3, where for each *Instance*: *All*, Maintenance, and *Crew* represent, respectively, the number of: all stations, stations with maintenance facilities, and crew bases. *Aircraft* and *Short-connections* give respectively the aggregate number of available aircraft and all possible short-connections for all fleets.

Although the largest instance of Table 3 contains 705 legs, the methodologies of the present paper can still be used by large airlines, which have as many as two thousand legs. Airlines' fleets can be naturally partitioned into short-, medium-, and long-haul categories. This partitioning is due to the fact that different aircraft types have different ranges, and different weight restricting them to fly towards specific destinations. Notice, that if any of the previously mentioned categories has a lot more than 700 flights, then the method will still probably be practicable for the rest fleet categories, and hence could at least be partially applied.

Concerning the FA, ready-times were provided for all legs and fleets. Additionally, there was information supplied on the passenger demand and spilling costs depending on the aircraft used, as well as the rest of the operating and

Table 3
Characteristics of the data sets

Instance	Stations			Aircraft	Short-connections
(network\legs)	All	Maintenance	Crew		
c\214	28	1	1	48	1709
c\280	40	1	1	62	2768
c\372	55	1	1	77	4650
hs\196	25	2	2	49	1583
hs\346	38	3	4	86	3825
hs\506	54	5	4	120	7832
hs\705	62	6	5	167	13,486

Note 1: In all cases six fleets were accounted, *legs* is the total number of scheduled ones, and the *network* structure is either with one central hub (c) or hub-and-spoke (hs).

Table 4
Results of the integrated method

	c214	c280	c372	hs196	hs346	hs506	hs705
Phase 1							
CPU MP	0.18	0.34	0.48	0.20	0.66	2.41	9.8
CPU SP	0.09	0.21	0.50	0.22	0.98	3.89	10.7
CPU	0.27	0.55	0.98	0.42	1.64	6.30	20.5
Iterations	8	9	8	15	16	18	10
Optimality cuts	29	35	34	73	75	91	43
Feasibility cuts	0	0	0	0	0	0	2
Phase 2							
CPU MP	0.07	0.34	1.20	0.17	1.91	3.22	6.04
CPU SP	0.00	0.00	0.05	0.01	0.14	0.22	0.02
CPU	0.07	0.34	1.25	0.18	2.05	3.44	6.06
Iterations	1	1	5	1	3	2	1
Optimality cuts	0	0	5	0	9	4	0
Phase 3							
CPU	0.00	0.00	0.01	0.00	0.02	0.13	1.21
Overall							
CPU	0.35	0.89	2.25	0.60	3.71	9.87	27.8
$_{\max}Gap\%$	1.10	0.79	0.21	1.41	1.06	0.83	0.73

Note 2: All *CPU* times are given in hours.

maintenance costs involved. No through flight's revenue was available, and for this reason routing costs c_r were considered null, while maintenance costs were accounted on average in c_{fl} constants. Aircraft were allowed to fly for 4 days maximum before returning to a maintenance station to spend a minimum of 8 h. The scheduling horizon was daily, which is accurate for the European airline, however, the American's weekly schedule had to be adapted.

Regarding the crew data, no exact information was provided, thus labor regulations were obtained from [37] and adapted for the medium- and long-haul case. The most important deviation from [37] concerns the time away from base, which was set for the European airline to 4 days, and for the American to 5 days. Moreover, the minimum sit-time was considered 30 min. Wages were inferred from the average wages per person and per year of the airline that provided the data. Overnight costs were inferred from typical hotel prices of the cities the airline was flying to. Finally, deadheading was not considered since such information was not available.

5.2. Experimental results

5.2.1. The original integrated method

In Table 4 the experimental results of the original integrated model (1)–(9), for the instances discussed in the previous subsection are demonstrated. This table provides statistics of the *Overall* algorithm execution as well as for each *Phase* individually. *CPU MP*, *CPU SP*, and *CPU* are, respectively: the *CPU* run-times of the Benders master problem, the Benders subproblem and the sum of run-times of both problems. Additionally, *Iterations*, *Optimality cuts*, and *Feasibility cuts* are, respectively, the number of Benders iterations, and the number of optimality and feasibility cuts generated. There were no feasibility cuts on Phase 2, however, the feasibility cuts generated in Phase 1 imply that there is always a possibility of infeasible CP problems that any sequential method cannot handle in cases where dead-heading is not used. Furthermore, Phase 3 was executed once and no cuts had to be generated. Finally, $_{max}Gap\%$ is the maximum optimality gap due to termination criteria (27) and (28) as well as due to the termination of the IP search on the first encountered solution. The maximum optimality gap is computed by $(c^{\mathrm{IP}} - LB^{\mathrm{LP}})/LB^{\mathrm{LP}} \times 100$, where $LB^{\mathrm{LP}} = \max_i(\bar{v}_r^i)$ and c^{IP} are, respectively, the lower bound of the LP relaxation at the end of Phase 1, and the cost upon termination of the algorithm.

Regarding the results of Table 4 it should be commented that the termination parameters of criteria (27) and (28)s, were kept constant across all instances, and the IP algorithm was always terminating when the first solution was encountered. These particular choices were the most efficient for the largest data set. Since, however, schedules with

Table 5
Results of the semi-integrated and sequential methods

	c214	c280	c372	hs196	hs346	hs506	hs705
FA + MR							
CPU LP	0.07	0.10	0.20	0.05	0.18	0.56	1.33
CPU IP	0.10	0.09	0.57	0.06	0.60	1.76	4.18
CPU	0.17	0.19	0.77	0.11	0.42	1.20	2.85
Gap%	2.07	1.05	0.22	0.09	0.54	0.22	0.11
MR + CP							
CPU	0.01	0.06	0.13	0.04	0.32	0.66	3.62
Gap%	−0.25	−0.31	−0.20	−0.35	−0.26	−0.48	−0.30
(*combined with* FA + MR *is semi-integrated*)							
Overall CPU	0.18	0.25	0.90	0.15	0.74	1.86	6.47
CP							
CPU	0.01	0.04	0.10	0.02	0.07	0.38	2.17
Gap%	−0.11	−0.06	−0.21	0.12	0.13	0.16	0.03
(*combined with* FA + MR *is sequential*)							
Overall CPU	0.18	0.23	0.87	0.13	0.49	1.58	5.02

Note 3: All *CPU* times are given in hours.

different structure and number of legs concern different airlines, it generally makes more sense to adapt these parameters or to have a more thorough IP search, leading to smaller ${}_{max}Gap\%$.

5.2.2. Best known methods from the literature

In Table 5 the experimental results of the semi-integrated and sequential methods discussed in Section 4.1 are demonstrated. *Overall CPU* is the total execution-time of these methods, and *Gap%* is computed by $(c^{\mathrm{IP}} - c^{\mathrm{LP}})/c^{\mathrm{LP}} \times 100$, where c^{LP} and c^{IP} are, respectively: the cost of the LP relaxation and the cost of the full IP problem. The negative *Gap%* found when solving the IP CP problem should be attributed to the fact that dominance-relaxation heuristics were used in the resource-constrained shortest path algorithm. Willing to estimate the quality sacrificed due to these heuristics the IP CP problem was also solved without them. It was observed that the gap between the sequential solutions including heuristics and those excluding them was not more than 0.16% [33]. The latter were however a lot slower than the former. Note that if one wishes to cover this gap it is possible exclude heuristics solely in Phase 3 where the final IP CP is solved.

5.2.3. Comparison with best known methods

Provided with the results of all these methods, one can compare them by the cost due to each and estimate the cost savings in every case. The first such comparison is between the integrated method and the best one available from the literature: the semi-integrated. The second comparison is between the semi-integrated and the next-best: the sequential. Finally, it is tempting to contrast the cost savings due to the first comparison with those savings due to the second. This contrast is illustrated by the *Savings ratio* which is computed by $Savings_{\mathrm{Integrated}}/Savings_{\mathrm{Semi\text{-}integrated}}$, where $Savings_{\mathrm{Integrated}}$ and $Savings_{\mathrm{Semi\text{-}integrated}}$ are the savings due to the first and the second comparison respectively. All these are presented in Table 6, where *Savings%* and *CPU ratio* are, respectively, computed by $(c_2 - c_1)/c_2 \times 100$ and CPU_1/CPU_2, when comparing Method 1 vs. Method 2. In this case c_i and CPU_i are the costs and *CPU* time of Method i. Since a daily schedule was solved, *Savings/Year* is the projection of the savings in a year's time. $Savings/_{max}Gap$ is given by $(c^{IP}_{\mathrm{Integrated}} - c^{IP}_{\mathrm{Semi\text{-}integrated}})/(c^{\mathrm{IP}}_{\mathrm{Integrated}} - LB^{\mathrm{LP}}_{\mathrm{Integrated}})$.

From the results of Table 6, it is clear that significant savings can be achieved if one utilizes integrated airline scheduling. These savings are up to 24 million US dollars per year for the largest instance of 700 legs and 6 fleets. Although optimality was not reached, the ratio of savings to the maximum optimality gap is high enough to justify the heuristics and the algorithm termination strategies used. This could be further improved for the smaller instances, by changing the parameters of criteria (27) and (28) and employing a more thorough IP search.

The significance of these savings can also be seen in contrast to those savings due to the semi-integrated method. This contrast is illustrated by the *Savings ratio* of the former to the latter. Obviously, the integrated method is far more

Table 6
Comparison of the integrated, semi-integrated and sequential methods

	c214	c280	c372	hs196	hs346	hs506	hs705
Integrated vs. semi-integrated							
Savings%	1.82	1.14	0.50	2.57	1.68	1.51	1.91
Savings/Year	7.08	6.02	3.53	9.10	10.8	14.6	24.2
CPU ratio	1.94	3.56	2.50	4.00	5.01	5.31	4.30
$Savings/_{max}Gap$	1.66	1.44	2.45	1.82	1.59	1.80	2.63
Semi-integrated vs. sequential							
Savings%	0.03	0.10	0.08	0.31	0.26	0.47	0.22
Savings/Year	0.13	0.50	0.59	1.10	1.66	4.58	2.80
CPU ratio	1.00	1.09	1.03	1.15	1.51	1.18	1.29
(Integrated vs. semi-integrated)/(semi-integrated vs. sequential)							
Savings ratio	56.8	12.0	5.95	8.30	6.53	3.19	8.66

Note 4: All *Savings/Year* are given in millions of US dollars.

successful in cutting-down airlines' costs. The advantage of the integrated methodology comes with the disadvantage of slower run-times. Finally, the first impression that *CPU ratios* give, is that the present method follows the scalability of the semi-integrated model, which in its turn follows that of the sequential.

5.2.4. Comparison with alternative integrated formulations introduced

Experiments showed that, for the alternative integrated formulations, that it is more efficient to directly start from Phase 2 of the three–phase algorithm. Additionally, since initializations (36)–(37) are used with approximation (34)–(35), in the experiments that were performed the option to use these initializations was either turned on, or off. Apart from these modifications the algorithm was identical to that of Section 3.3. Finally, the Benders master problem was solved using an algorithm similar to that presented in Section 2.3.

In Table 7 the results of the best performing alternative for each instance, are presented, and compared with the results of the original integrated model (1)–(9). *PCC* refers to the usage of plane-count constraints, *Initialization* is *on* when the approximation $(\mathbf{x}^0, \breve{\mathbf{s}}^0) \leftarrow (\mathbf{1}, \mathbf{0})$ is used initially or *off* if a solution of the Benders master problem is used instead. In the latter case the normal execution order of the Magnanti–Wong iterations is followed, namely Benders master problem, followed by the subproblem, and finally the Magnanti–Wong problem. The *CPU ratio* is obtained by $CPU_{\text{alternative}}/CPU_{\text{integrated}}$, and the *Savings ratio* by $Savings_{\text{alternative}}/Savings_{\text{integrated}}$. As before, Phase 3 was executed only once without generating any cuts, in all cases.s

The results presented in Table 7 demonstrate that even the best alternative formulation—for each instance—is rather inefficient in comparison to the original one. Moreover, the savings of the alternative are typically smaller than those of the original, as illustrated by the *Savings ratio*. This can can be explained by the fact that the maximum optimality gap of the alternative formulation is large. The later is illustrated by the low $Savings/_{max}Gap$ ratios. These ratios could be improved by a more thorough IP search, but this would make the already poor solution-times even worse. Finally, no method was able to find a solution for instances *hs506* and *hs705* within a timeout period set to 24 and 36 h, respectively, however, more than 19 and 11 cuts were accordingly generated.

Notice additionally, that for all instances more than one Benders iteration and numerous cuts had to be generated before obtaining a feasible MR solution. This implies that the integrated models of the Benders master problem cannot generally give MR feasible solutions. These models were the simple FA time-line and its enhancement with plane-count constraints. Thus one way or another, it seems that the MR problem has to be solved, and the alternative formulations attempted offer at least such an opportunity, contrary to other models in the literature [4].

5.3. Speed-up estimate of a straightforward parallelization

It is interesting to notice that a straightforward parallelization can speed-up the overall solution-time. This parallelization is based on the observation that some problems are decomposed per fleet, and instead of solving them sequentially one could use a number of single-core *CPU* computers equal to the number of fleets and solve these subproblems

Table 7
Results of the most efficient alternative integrated formulation for each data set, and comparison with the original integrated model

	c214	c280	c372	hs196	hs346	hs506	hs705
Best alternative							
PCC	no	no	yes	yes	no	n/a	n/a
Initialization	on	on	off	off	on	n/a	n/a
Phase 2							
CPU MP	0.37	0.81	15.9	2.44	3.65	>21	>28
CPU SP	0.03	0.02	0.37	0.13	0.59	>1	>4
CPU	0.40	0.83	16.3	2.57	4.24	>24	>36
Iterations	5	2	9	22	7	>9	>4
Cuts	5	1	11	56	9	>19	>11
Phase 3							
CPU	0.01	0.01	0.05	0.02	0.07	n/a	n/a
Overall							
CPU	0.41	0.84	16.3	2.72	4.31	>24	>36
${}_{max}Gap\%$	1.05	0.95	1.33	2.72	2.41	n/a	n/a
Savings%	1.86	0.99	−0.61	1.31	0.37	n/a	n/a
$Savings/{}_{max}Gap$	1.82	1.07	n/a	0.50	0.16	n/a	n/a
Savings/Year	7.25	5.22	−4.28	4.64	2.36	n/a	n/a
vs. Original							
CPU ratio	1.17	0.94	7.25	4.53	1.16	>2.4	>1.3
Savings ratio	1.02	0.87	n/a	0.51	0.47	n/a	n/a

Note 5: All *CPU* times are given in hours and all *Savings/Year* are given in millions of US dollars.

Table 8
Estimate of solution-times for the straightforward parallelization of the integrated method

	c214	c280	c372	hs196	hs346	hs506	hs705
CPU MP	0.19	0.51	1.23	0.29	1.97	4.23	12.5
CPU SP	0.03	0.07	0.24	0.06	0.39	1.23	3.9
CPU	0.22	0.58	1.47	0.35	2.36	5.46	16.5
Speed-up%	59	53	53	71	57	81	68

Note 6: All *CPU* times are given in hours.

in parallel. The problems that are fleet decomposed are: the column generation subproblem of the Benders master problem, the Benders subproblem, the Magnanti–Wong problem, and the second stage of the IP algorithm presented in Section 2.3.

To have an estimate of these parallel run-times, each time a problem needs to be solved for more than one fleet, the fleet problem that took most time has to be accounted, as this would only affect the parallel's algorithm run-time. Such an estimate is presented in Table 8, where the *Speed-up%* is computed by $(CPU_{\text{sequential}} - CPU_{\text{parallel}})/CPU_{\text{parallel}} \times 100$. The above estimates are reasonably accurate as the information that needs to be exchanged between the computers is minimal, and hence the network speed will not significantly influence the run-times.

6. Conclusions and future work

In this paper, the previously thought intractable integrated airline scheduling problem was solved for realistic instances of European and North American airlines. These airlines had different network structures, allowing a wide evaluation. The largest solved instance scheduled 700 legs for 6 fleets, and succeeded in reducing overall operating costs by 24 million US dollars in comparison to the best known method from the literature. Even the largest airlines have short-, medium- or long-haul fleets that operate as many legs, and the present method can be either completely or partially applied. The solution-time of this instance can be 16.5 h when using a straightforward parallelization method with six single-core-CPU computers. Alternative integrated models and decompositions were also attempted but did not prove to be as efficient as the original. The successful solution of these models is due to an enhanced Benders decomposition method combined with accelerated column generation.

One could possibly try to extend the present work by introducing time-windows to departure-time of the legs. Such a flexibility within the airline scheduling model may well generate even greater cost savings. Finally information provided by the Benders cuts could be used for efficient re-optimization of the airline scheduling problem.

Acknowledgements

The author is indebted to Dr. Jonathan Lever, Prof. Mark Wallace, Dr. Andy Eremin, Dr. Wilhelm Cronholm, Dr. Olli Kamarainen, and Imogen Rivers for their support in different stages of the project. The author would also like to thank the anonymous referees for their helpful comments.

References

[1] Barnhart C, Belobaba P, Odoni A. Applications of operations research in the airline industry. Transportation Science 2003;37(4):368–91.
[2] Barnhart C, Lu F, Shenoi R. Integrated airline schedule planning. In: Yu G, editor. Operations Research in the Airline Industry. Boston: Kluwer; 1998. p. 384–403.
[3] Klabjan D, Johnson E, Nemhauser G, Gelman E, Ramaswamy S. Airline crew scheduling with time windows and plane-count constraints. Transportation Science 2002;36(3):337–48.
[4] Sandhu R, Klabjan R. Integrated airline planning, in: Proceedings 44th AGIFORS Annual Symposium, Singapore, 2004.
[5] Barnhart C, Boland N, Clarke L, Johnson E, Nemhauser G, Shenoi R. Flight string models for aircraft fleeting and routing. Transportation Science 1998;32(3):208–20.
[6] Clarke L, Hane C, Johnson E, Nemhauser G. Maintenance and crew considerations in fleet assignment. Transportation Science 1996;30(3): 249–60.
[7] Desaulniers G, Desrosiers J, Dumas Y, Solomon M, Soumis F. Daily aircraft routing and scheduling. Management Science 1997;43(6): 841–55.
[8] Rexing B, Barnhart C, Kniker T, Jarrah A, Krishnamurthy N. Airline fleet assignment with time windows. Transportation Science 2000;34(1): 1–20.
[9] Mercier A, Soumis F. An integrated aircraft routing, crew scheduling and flight retiming model. Computers & Operations Research 2006;34(8):2251–65.
[10] Cordeau J-F, Stojković G, Soumis F, Desrosiers J. Benders decomposition for simultaneous aircraft routing and crew scheduling. Transportation Science 2001;35(4):375–88.
[11] Cohn A, Barnhart C. Improving crew scheduling by incorporating key maintenance routing decisions. Operations Research 2003;51(3): 387–96.
[12] Mercier A, Cordeau J-F, Soumis F. A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. Computers & Operations Research 2005;32(6):1451–76.
[13] Mercier A. A theoretical comparison of feasibility cuts for the integrated aircraft routing and crew pairing problem. Transportation Science [to appear].
[14] Papadakos N. Accelerating column generation by heuristic deepest-cut pricing. Working Paper, 2007.
[15] Papadakos N. Practical enhancements to the Magnanti–Wong method. Working Paper, 2007.
[16] Hane C, Barnhart C, Johnson E, Marsten R, Nemhauser GL, Sigismondi G. The fleet assignment problem: solving a large-scale integer program. Mathematical Programming 1995;70:211–32.
[17] Kniker T. Itinerary-based airline fleet assignment. Ph.D. thesis, Massachusetts Institute of Technology, 1998.
[18] Desaulniers G, Desrosiers J, Dumas Y, Marc S, Rioux B, Solomon M, Soumis F. Crew pairing at Air France. European Journal of Operational Research 1997;97:245–59.
[19] Desaulniers G, Desrosiers J, Gamache M, Soumis F. Crew scheduling in air transportation. In: Crainic TG, Laporte G, editors. Fleet Management and Logistics. Boston: Kluwer; 1998. p. 169–85 [Chapter 8].
[20] Barnhart C, Cohn A, Johnson E, Klabjan D, Nemhauser G, Vance P. Airline crew scheduling. In: Hall R, editor. Handbook of Transportation Science. Boston: Kluwer Scientific Publishers; 2003. p. 57–93 [Chapter 3].
[21] Benders J. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 1962;4(3):238–52.
[22] Desaulniers G, Desrosiers J, Ioachim I, Solomon M, Soumis F. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In: Crainic TG, Laporte G, editors. Fleet Management and Logistics. Boston: Kluwer; 1998. p. 57–93 [Chapter 3].
[23] Desrochers M. La fabrication d'horaires de travail pour les conducteurs d'autobus par une méthode de génération de colonnes, Ph.D. thesis, Université de Montréal, Centre de recherche sur les Transports, CP 6128, Succursale "Centreville", Montréal, Québec, H3C 3J7, publication # 470; 1986.
[24] Desrosiers J, Dumas Y, Solomon M, Soumis F. Time constrained routing and scheduling. In: Ball M, Magnanti TL, Monma C, Nemhauser G, editors. Handbooks in Operations Research and Management Science, vol. 8. Amsterdam: Network Routing, North-Holland; 1995. p. 35–139 [Chapter 2].

[25] Vanderbeck F. Decomposition and column generation for integer programs, Ph.D. thesis, Université catholique de Louvain; 1994.
[26] Lübbecke M. Engine scheduling by column generation. Ph.D. thesis, Technischen Universität Braunschweig; 2001.
[27] McDaniel D, Devine M. A modified Benders' partitioning algorithm for mixed integer programming. Management Science 1977;24(2):312–9.
[28] Magnanti T, Wong R. Accelerating Benders decomposition: algorithmic enhancement and model selection criteria. Operations Research 1981;29(3):464–84.
[29] Cordeau J-F, Soumis F, Desrosiers J. Simultaneous assignment of locomotives and cars to passenger trains. Operations Research 2001;49(4): 531–48.
[30] Lasdon L. Optimization Theory for Large Systems. London: Macmillan; 1970.
[31] Santoso T, Ahmed S, Goetschalckx M, Shapiro A. A stochastic programming approach for supply chain network design under uncertainty. European Journal of Operational Research 2005;167:96–115.
[32] Chu Y, Xia Q. A hybrid algorithm for a class of resource constrained scheduling problems. In: Barták R, Milano M, editors. Integration of AI and OR techniques in constraint programming for combinatorial optimization problems: second international conference, CPAIOR 2005, Prague Czech Republic. Berlin, Heidelberg: Springer; 2005. p. 110–24.
[33] Papadakos N. Integrated airline scheduling: models and decomposition techniques. Ph.D. thesis, Imperial College London; 2006.
[34] Cheadle A, Harvey W, Sadler A, Schimpf J, Shen K, Wallace M. $\mathrm{ECL}^i\mathrm{PS}^e$: An introduction, Tech. Rep. IC-Parc-03-1, IC-Parc, Imperial College London; 2003.
[35] Eremin A. Using dual values to integrate row and column generation into constraint logic programming. Ph.D. thesis, Imperial College London; 2003.
[36] ILOG S.A., ILOG Cplex 9: User's Manual and Reference Manual; 2004.
[37] Ho R, Johnson E, Shaw T. Modeling tools for airline crew scheduling and fleet assignment problems. In: Ciriani T, Gliozzi S, Johnson E, Tadei R, editors. Operational Research in Industry. London: MacMillan Press Ltd; 1999. p. 1–24.