

Incorporation of *Min/Max*, *With/Without* and *Pairing* Restrictions in Column Generation Solution Algorithms for Aircrew Rostering

George Kozanidis
0000-0002-4325-2975
Department of Mechanical
Engineering, University of
Thessaly, Volos 38334, Greece
Email: gkoz@uth.gr

Nikos Polychronopoulos
Department of Mechanical
Engineering, University of
Thessaly, Volos 38334, Greece
Email:
n.polychronopoulos@hotmail.com

Andreas Gavranis
0000-0001-8102-516X
Department of Mechanical
Engineering, University of
Thessaly, Volos 38334, Greece
Email: agavra@uth.gr

Abstract—We consider the column generation solution framework handling the crew rostering problem in the context of commercial aviation. Motivated by practical requirements arising in realistic environments, we modify the algorithmic methodology to enable the incorporation of three special restriction types termed *min/max*, *with/without* and *pairing*, respectively. *Min/max* restrictions impose suitable lower/upper bounds on the number of a specific category crew members assigned to certain flight duties. *With* restrictions ensure that crew members of a specific category assigned to certain flight duties are coupled with crew members of another specific category. *Without* restrictions forbid the simultaneous assignment of crew members belonging to two distinct specific categories to certain flight duties. Finally, *pairing* restrictions dictate that the number of crew members of each of two mutually exclusive, but collectively exhaustive, categories assigned to certain flight duties with even total crew complement is also even. After properly adjusting the algorithmic methodology for accommodating these restrictions, we present computational results demonstrating how their incorporation affects algorithmic behavior and computational performance.

Index Terms—Column generation, branch & price, commercial aviation, airline management, crew rostering, *min/max* restrictions, *with/without* restrictions, *pairing* restrictions.

I. INTRODUCTION

COLUMN generation comprises one of the most popular solution frameworks for the efficient treatment of complex decision-making problems arising in the context of airline management. Focusing on the algorithmic variation that handles the crew rostering problem, we incorporate into the column generation methodology three special types of restrictions for accommodating practical user requirements arising in realistic environments. These restrictions are applicable to flight duties with crew complement strictly greater than 1, i.e., flight duties that must be assigned to more than 1 crew members.

The first restriction type, termed *min/max*, imposes a lower/upper bound on the number of crew members of a specific category assigned to some flight duty. The second restriction type, termed *with/without*, is distinguished into

two sub-types. *With*-type restrictions ensure that the assignment of a specific category crew member to a certain flight duty is only allowed if a crew member of another specific category is also assigned to the same flight duty. *Without*-type restrictions, on the other hand, forbid the simultaneous assignment of crew members belonging to two distinct specific categories to the same flight duty.

The last restriction type, termed *pairing*, necessitates the existence of some attribute/characteristic with only two crew categories involved, such that any crew member always belongs to exactly one of the two by default. Such a restriction dictates that the number of crew members of each of these two categories assigned to some specific duty with even total crew complement must also be even, in order to minimize relevant crew handling costs which increase considerably otherwise.

We develop the methodology for incorporating the above restrictions into the column generation algorithmic variation handling the crew rostering problem. This includes the modification of the master problem formulation for ensuring their satisfaction, as well as the suitable adjustment of the column generation subproblem for ensuring they are properly taken into consideration within the pricing algorithmic procedure.

The main contribution of the present work is the extension of the fundamental column generation framework for enabling the accommodation of special restrictions arising in realistic environments. Besides illustrating how these restrictions can be properly handled, the current research provides valuable insight for the treatment of similar restriction types that may arise in different situations.

The remainder of this work is structured as follows. Section 2 presents a literature review on solution algorithms for aircrew rostering, while Section 3 presents the problem definition and develops the model formulation. Section 4 develops the associated algorithmic implementation, while Section 5 presents computational results drawn from realistic experiments. Finally, Section 6 summarizes this work and suggests fruitful directions for future research.

II. LITERATURE REVIEW

A large number of decision-making problems arising in the regular operation of commercial aviation, such as tail assignment [1] and trainee assignment [2], necessitate the employment of sophisticated optimization models and solution algorithms. Aircrew rostering [3] comprises one of the most widely studied relevant problems, due to its high complexity, its tremendous significance, and its wide applicability.

Typically, the crew rostering problem is considered within the context of the crew scheduling framework [4], which, besides the crew rostering problem, also includes the problem of constructing pairings through the integration of flight legs, off days, rests, training duties, etc., termed crew pairing [5].

Three are the most common design schemes for the aircrew rostering problem. The first one, often referred to as *fair assignment* in the related literature, optimizes an objective encompassing the duty non-coverage cost, as well as the aggregate soft quality cost of the crew rosters [6]. The second one, typically termed *preferential bidding system* in the related literature, also minimizes the duty non-coverage cost as primal objective, but instead of secondarily optimizing soft quality, it optimizes the cost related to preference satisfaction with respect to specific crew requests that have been expressed in advance [7]. These two schemes comprise two different versions of the so-called *personalized crew rostering framework*. The third one, termed *bidline rostering*, builds a set of anonymous rosters in advance, with the final group schedule obtained after each crew member expresses specific preference for roster assignment [8]. The special restriction types introduced above are considered in the context of the personalized version of the crew rostering problem; therefore, we focus on this problem variation in what follows.

Column generation [9] comprises one of the major breakthroughs for the solution of the aircrew rostering problem. Ryan [10] formulates a set partition optimization model, which employs a specialized column generation approach to solve the LP relaxation of the master problem to near optimality, and then employs branch & bound to obtain an integer solution using the column variables that have been identified so far. Gamache et al. [11] formulate a set-partition optimization model for the aircrew rostering problem, utilizing a heuristic column generation-based methodology for identifying integer solutions.

Fahle et al. [12] develop a column generation heuristic solution algorithm for the problem, which exploits constraint programming in the shortest path pricing subproblem for handling the computational difficulties raised by the complex rules/regulations each crew roster needs to comply with.

The solutions returned by heuristic algorithms often exhibit moderate quality; this has motivated the development of exact methodologies such as branch & price [13] for the treatment of the problem. Branch & price comprises an innovative extension of branch & bound that ensures exact optimality by employing column generation in each node of the branching tree. It has been exploited for the solution of numerous deci-

sion-making problems arising in the regular operation of commercial aviation, such as logistics planning [14], [15], gate assignment [16] and vacation planning [17].

One of the earliest works on branch & price is that of Desrosiers et al. [18], although the specific term for this methodology was later introduced by Barnhart et al. [19] who illustrate its application on the generalized assignment and the crew scheduling problem. Gamache and Soumis [20] present an exact branch & price solution methodology for a rostering problem variation that includes crew requests, which, instead of assigning in advance a specific subset of these requests, incorporates them into the optimization procedure. More information on column generation methodologies for aircrew rostering can be found in the survey of Kasirzadeh et al. [21].

III. PROBLEM DESCRIPTION & MATHEMATICAL FORMULATION

We consider the crew rostering problem in the context of commercial aviation, which aims to build rosters (schedules) for assignment to a crew member group, while optimizing an aggregate objective function. A crew roster is formed by selecting a proper subset of duties respecting the active legality restrictions imposed by air safety regulations and government/union agreements. The specific duty selection prescribes an associated cost, which reflects the extent to which the roster exhibits certain desirable quality characteristics aimed at increasing crew satisfaction and operational efficiency. The problem objective includes the duty non-coverage cost and the roster quality cost, with the former exhibiting higher priority, due to its indisputable superior significance.

The activities subject to coverage include flight duties called *pairings* and ground duties called *standbys*. Each pairing is formed by connecting a subset of pairwise temporally and locally compatible flight legs, coupled with rest periods. Temporal compatibility necessitates that the arrival time of each flight leg precedes the departure time of the immediately next one, while local compatibility necessitates that the arrival airport of each flight leg coincides with the departure airport of the immediately next one. Each pairing must begin and end at the corresponding crew group airport home base. Each standby, on the other hand, is a stand-alone independent duty, typically characterized by its service location (home, airport), as well as by its service time (morning, evening, night, etc.).

Column generation is a specialized algorithmic procedure for handling linear programming problems with an exponentially large number of decision variables. In the aviation context under consideration, each of these variables corresponds to a crew roster. Instead of explicitly enumerating all of them, a task that would necessitate an extraordinary amount of time, the column generation algorithmic approach formulates a so-called *restricted master problem* including a very small subset of them, which is moderately being enriched with additional beneficial ones until the optimal solution is obtained.

The criterion exploited for identifying these extra variables is their reduced cost with respect to the optimal solution of the master problem LP relaxation. The involved computations are

performed by a shortest path network optimization problem termed *pricing subproblem*. This problem is invoked numerous times along algorithmic execution, each time identifying new beneficial rosters for addition to the master problem.

The pricing subproblem ensures satisfaction of the complex hard rules that each roster needs to respect for the associated crew member in isolation. Typical examples of such restrictions are the imposition of an upper bound on the total flight time, the requirement that a rest period of specific minimum duration is ensured after certain long-lasting duties, and the imposition of a lower bound on the number of off-work days. In addition, each roster must respect the set of pre-assigned activities which may have been appended in advance to the monthly plan of the crew member under consideration.

On the other hand, rules applicable to the entire crew group are handled at the master problem level. An example of such a restriction is the satisfaction of a specific upper bound on the total number of days off from work assigned collectively to the crew group for a given day of the planning horizon. The special restrictions we consider in this research work must also be satisfied combinedly by all crew members.

For the formulation of the master problem, we introduce the following mathematical notation:

Sets:

I : set of crew members, indexed by i .

R_i : set of rosters of crew member i , indexed by j .

D : set of standby/pairing duties, indexed by d .

Parameters:

c_{ij} : quality cost of roster j of crew member i , $i \in I, j \in R_i$,

h : cost for each uncovered unit of duty complement,

b_d : crew complement of duty d , $d \in D$,

a_{ijd} : binary parameter that takes the value 1 if roster j of crew member i includes (covers) duty d , and 0 otherwise, $i \in I, j \in R_i, d \in D$.

Decision variables:

x_{ij} : binary decision variable that takes the value 1 if roster j is assigned to crew member i , and 0 otherwise, $i \in I, j \in R_i$,

y_d : integer slack decision variable denoting the number of complement units of duty d that remain uncovered, $d \in D$.

Utilizing the above notation, the master problem is formulated as follows:

$$\text{Min } \sum_{i \in I} \sum_{j \in R_i} c_{ij} \cdot x_{ij} + \sum_{d \in D} h \cdot y_d \quad (1)$$

$$\text{s.t. } \sum_{j \in R_i} x_{ij} = 1, i \in I \quad (2)$$

$$y_d + \sum_{i \in I} \sum_{j \in R_i} a_{ijd} \cdot x_{ij} = b_d, d \in D \quad (3)$$

$$x_{ij} \in B = \{0,1\}, y_d \in Z^+, i \in I, j \in R_i, d \in D \quad (4)$$

The objective function (1) minimizes the total system cost, which includes the cost of uncovered duties plus the roster quality cost. Cost coefficient h is always much larger than cost coefficients c_{ij} , imposing the lexicographic priority between the two objective terms. Constraint set (2) includes one multiple choice constraint for each crew member, ensuring the assignment of exactly one among all the alternative rosters. Constraint set (3) monitors duty coverage based on the as-

signed rosters; slack variable y_d is equal to the number of uncovered complement units, imposing the corresponding objective cost in case it becomes positive. Finally, constraint set (4) imposes integrality restrictions on the decision variables.

IV. SPECIAL RESTRICTIONS' INCORPORATION

The crew rostering problem is typically solved separately for each crew position. The complement of a duty is equal to 1 when only one crew member holding the position under consideration needs to be assigned to it. This is common for positions such as CP (captain) and FO (first officer). On the other hand, the number of FAs (flight attendants) that need to be assigned to a duty is naturally an integer greater than 1.

In realistic commercial airline environments, crew members typically belong to specific categories, based on their particular attributes/characteristics. Some examples include categories reflecting experience level (junior, senior, etc.), language knowledge (Chinese, French, German, etc.), and origin (Asian, European, etc.). The three restriction types under consideration pertain exclusively to duties with complement strictly greater than 1, i.e., duties $d \in D$ for which $b_d > 1$. In the following subsections, we elucidate in detail the corresponding requirements and setting parameters for each of them. For this analysis, we extend the mathematical notation introduced in the previous section with two additional sets:

Sets:

G : set of crew categories, indexed by g .

G_i : set of crew categories of crew member i , $i \in I$.

A. Min/max Restriction Type

A *min/max* restriction type imposes a lower/upper bound on the number of crew members of a specific category assigned to some particular duty. For example, a *min*-type restriction may impose the requirement that at least two Chinese speaking crew members are assigned to some pairing, while a *max*-type restriction may impose the requirement that at most one junior crew member is assigned to it.

To accommodate a *max*-type restriction, we add to the master problem formulation a less or equal ($\leq$) constraint with right-hand side equal to the corresponding upper bound. In the left-hand side, this constraint is meant to include the rosters of crew members belonging to the specific category for which the restriction is imposed, provided that they include (cover) the duty to which the rule applies. Let $M \in G$ and $p \in D$ denote this crew category and duty, respectively, and u be the specified upper bound. Mathematically, the *max* restriction is expressed with the following constraint:

$$\sum_{i \in I: M \in G_i} \sum_{j \in R_i} a_{ijp} \cdot x_{ij} \leq u \quad (5)$$

A generated roster is appended to the left-hand side of this constraint, if and only if it satisfies the category-duty combination specified by the rule. Clearly, the inclusion of this constraint and its proper population with rosters ensure satisfaction of the *max*-type restriction upon algorithmic termination.

Consider a *min*-type restriction next. Following a similar approach, one could add to the master problem a greater or

equal ($\geq$) constraint with right-hand side equal to the corresponding lower bound. The decision of whether a generated roster should be appended to the left-hand side of this constraint would similarly necessitate the check of whether it satisfies the corresponding category-duty combination. A difficulty that arises in this case, however, is that this modeling choice leaves open the possibility of infeasibilities, since the presence of crew rosters which are necessary for constraint satisfaction in the left-hand side cannot be guaranteed. To give a particular example, the solution that consists of empty rosters (no duties covered), commonly used for initializing the master problem, cannot be utilized under this modeling approach, because it renders the master problem infeasible.

To circumvent this difficulty, we express a *min*-type restriction as a less or equal constraint with right-hand side equal to the difference between the total complement of the associated duty and the corresponding lower bound imposed on the number of crew members of the specific category. Using a similar but reverse logic as the one adopted in the case of a *max*-type restriction, the left-hand side of this constraint is meant to include rosters covering the duty in question which pertain to crew members that do not belong to the category specified by the rule. This adjustment is clearly equivalent to the imposition of the desirable lower bound on the number of crew members of the specific category prescribed by the rule. This modeling approach handles properly potential master problem infeasibilities, since the constraint is still satisfied even when its left-hand side is empty. Let $M \in G$ and $p \in D$ denote the crew category and duty, respectively, for which the restriction applies, and l be the specified lower bound. Mathematically, this restriction is expressed with the following constraint:

$$\sum_{i \in I: M \notin G_i} \sum_{j \in R_i} a_{ijp} \cdot x_{ij} \leq b_p - l \quad (6)$$

B. With/without Restriction Type

Moving on to the next case, a *with*-type restriction imposes the requirement that for each crew member of a specific category assigned to a particular duty, a crew member of another specific category must be assigned to the same duty, too. For example, a *with*-type restriction may dictate that each junior crew member assigned to some duty must be accompanied by a senior crew member. This one-to-one coupling restriction works only in one direction, i.e., the total number of junior crew members assigned to the duty cannot exceed the total number of senior crew members, whereas the opposite is clearly permissible. For simplicity, we call the former crew members *dependent* and the latter ones *independent*. As far as the involved crew categories are concerned, the convention is that any given crew member can be dependent, independent, or neither of the two, but a crew member cannot be both dependent and independent simultaneously.

To incorporate a *with* restriction, we add to the master problem formulation a less or equal ($\leq$) constraint. This constraint is meant to include in its left-hand side the rosters of dependent crew members covering the duty in question, and in its right-hand side the rosters of independent crew members

covering this duty. This ensures that the number of dependent crew members to which the duty will be assigned will not exceed the number of independent ones, without imposing any restriction, whatsoever, on the latter ones. Let $M_1 \in G$ denote the dependent crew category, $M_2 \in G$ denote the independent one, and $p \in D$ denote the duty for which the restriction applies. Mathematically, this restriction is expressed with the following constraint:

$$\sum_{i \in I: M_1 \in G_i} \sum_{j \in R_i} a_{ijp} \cdot x_{ij} \leq \sum_{i \in I: M_2 \in G_i} \sum_{j \in R_i} a_{ijp} \cdot x_{ij} \quad (7)$$

A *without*-type restriction, on the other hand, imposes the requirement that the number of crew members assigned to a given duty must be equal to 0 for at least one of two specified crew categories, the convention being again that any crew member can belong to neither or one of the two, but no crew member can belong to both of them at the same time.

To incorporate this restriction, we add to the master problem formulation two less or equal ($\leq$) constraints, as well as a binary auxiliary decision variable. The right-hand side of the first constraint is set equal to the duty complement multiplied by this variable, while that of the second constraint is set equal to the duty complement multiplied by one minus this variable. Each of the two constraints is meant to include in its left-hand side rosters of crew members belonging to one of the two specified categories covering the duty in question. Let $M_1 \in G$ and $M_2 \in G$ denote the corresponding crew categories, $p \in D$ denote the duty for which the restriction applies, and z be the auxiliary decision variable. Mathematically, this restriction is expressed with the following two constraints:

$$\sum_{i \in I: M_1 \in G_i} \sum_{j \in R_i} a_{ijp} \cdot x_{ij} \leq b_p \cdot z \quad (8)$$

$$\sum_{i \in I: M_2 \in G_i} \sum_{j \in R_i} a_{ijp} \cdot x_{ij} \leq b_p \cdot (1 - z) \quad (9)$$

The binary nature of z ensures that the right-hand side of one of these two constraints is equal to 0, while that of the other one is equal to the corresponding duty complement. This way, the number of crew members assigned to this duty is always equal to 0 for one of the two specified categories, with no restriction, whatsoever, imposed on the other one.

C. Pairing Restriction Type

The final restriction type we consider necessitates the existence of two mutually exclusive but collectively exhaustive crew categories, such that any crew member always belongs to exactly one of the two. An extra prerequisite for the activation of this restriction is that the crew complement of the duty in question is even; when this is the case, the *pairing*-type restriction requires that the number of crew members assigned to this duty for each of the two specified categories is even, too. The aim is to avoid extra costs which arise when the crew members of these two categories are not treated in pairs.

To express a *pairing*-type restriction, we add to the master problem formulation an equality ($=$) constraint whose right-hand side is equal to an integer auxiliary variable multiplied by 2. The left-hand side of this constraint is meant to include the rosters of crew members belonging to one of the two specified categories which cover the duty in question. This way,

the sum of these crew members is ensured to be an even number (0, 2, 4, etc.). Let $M \in G$ denote any of the two corresponding crew categories, $p \in D$ denote the duty to which the restriction applies, and w be the auxiliary decision variable. Mathematically, the *pairing*-type restriction is expressed with the following constraint:

$$\sum_{i \in I: M \in G_i} \sum_{j \in R_i} a_{ijp} \cdot x_{ij} = 2 \cdot w \quad (10)$$

D. Algorithmic Modifications

After incorporating the special restrictions as illustrated above, some proper algorithmic modifications are deemed necessary for the column generation methodology, as explained next. The first one concerns the proper initialization of the master problem formulation. Although the master problem is typically initialized with an empty roster for each crew member, rosters containing a proper subset of duties may alternatively be utilized to this end in some cases ([22], [23]). When this is the case, the special constraints need to be populated with the initial roster variables, based on the associated crew categories and duties covered.

The next algorithmic modification regards the shortest path pricing subproblem. Each time a partial roster is extended through the inclusion of an additional duty, a check of whether the inclusion of this duty necessitates the population of some special constraints with the associated roster variable needs to be performed. In such a case, the dual variables associated with those constraints need to be taken into consideration for the computation of this roster variable's reduced cost. In addition, if this roster is deemed promising upon conclusion of the pricing subproblem and its addition to the master problem is decided, the associated roster variable should be appended to the proper special constraints.

V. COMPUTATIONAL RESULTS

In this section, we present computational results from realistic experiments for assessing how the inclusion of the special constraint types affects the computational performance of the column generation solution framework. For the execution of these experiments, we have exploited our long-term research collaboration with AIMS INTL DWC LLC [24], one of the largest worldwide providers of commercial airlines' operations management software.

Since 1983, AIMS has been engaged in the development of a specialized software package which facilitates the support and coordination of numerous airline operations, such as air-crew management, fleet planning, flight scheduling, etc. Among others, the AIMS software includes sophisticated solution algorithms for the treatment of complex decision-making problems encountered in commercial aviation, such as pairing generation, crew rostering, flight and maintenance scheduling, vacation planning, etc.

The proposed model extension incorporating the special restrictions under consideration has been fully implemented within AIMS software. The execution of the computational experiments for which results are presented next was performed on live databases of AIMS' airline clients to which

AIMS holds access for maintenance and support purposes. The actual code includes several specialized techniques for avoiding excessive computational times, such as limits on the size of the branch & price tree, integrality gap tolerances, execution time limits, etc. These techniques are not fixed but have been properly customized depending on the special characteristics of each particular airline client.

The source code interacts with commercial optimization software IBM ILOG CPLEX [25], which performs various specialized operations necessitated for algorithmic execution. All the computational experiments were executed on an Intel (R) Core (TM) i7-1165G7 @ 2.80GHz Processor with 16 GB system memory. Computer processors employed in realistic environments exhibit superior technical characteristics, leading to significantly higher computational performance.

We tested the proposed methodology on 12 realistic crew rostering problem instances, each time with Flight Attendants (FA) as the crew position subject to assignment. Table 1 presents the data parameters of these instances, i.e., the problem index (i), the number of crew members (N), the number of duties (standbys + pairings) subject to assignment (D) and the total complement (TC), computed as the weighted sum of the duties, with weights the corresponding crew complements.

TABLE I.
DATA PARAMETERS OF THE PROBLEM INSTANCES

i	N	D	TC
1	1061	787	3675
2	1089	801	3918
3	1104	832	4194
4	1073	799	3876
5	1128	867	4233
6	1102	809	4051
7	1239	984	5332
8	1255	991	5498
9	1247	965	5154
10	1118	863	4349
11	1094	796	4042
12	1139	880	4426

The next 5 tables present the results of these experiments. Specifically, Table 2 presents the results when no special restrictions are present, Table 3 only with *min/max*-restrictions activated, Table 4 only with *with*-restrictions activated, Table 5 only with *without*-restrictions activated, and finally, Table 6 only with *pairing*-restrictions activated. Note that the same 12 problem instances were tested in all 5 model variations.

All these tables, besides the first one, include a column labeled “# *SR*”, which presents the actual number of activated special restrictions. The next 3 columns present the obtained results, i.e., the time required for the solution of the master LP relaxation at the root node of the branch & price tree (*LP Time*), the time required for obtaining the final integer solution (*IP Time*) and the number of nodes of the branch & price tree (*Nodes*). The computational time results are presented in the form HH:MM:SS, where HH shows the number of hours, MM the number of minutes and SS the number of seconds. Note that the *IP Time* results are not reported in isolation, i.e., they include the associated *LP Time* counterparts.

The computational times reported in the following tables pertain exclusively to the column generation algorithmic methodology. The actual computational times are practically 10-15% higher, since they also include processing time for database access, parameter information loading, as well as several pre-processing procedures which need to be executed before the column generation solution framework is invoked.

TABLE II.
COMPUTATIONAL RESULTS - NO SPECIAL RESTRICTIONS

<i>i</i>	<i>LpTime</i>	<i>IpTime</i>	<i>Nodes</i>
1	10:25:48	20:36:12	1821
2	11:08:03	21:49:57	1833
3	11:27:13	22:47:36	1871
4	11:03:36	22:55:12	1859
5	12:50:48	25:43:48	1921
6	11:52:31	24:40:49	1863
7	15:26:12	31:00:48	1997
8	16:06:29	33:00:36	2015
9	15:54:19	32:52:31	2005
10	15:12:30	31:23:44	1919
11	12:45:09	25:50:24	1905
12	16:03:14	32:03:18	1957

TABLE III.
COMPUTATIONAL RESULTS - ONLY *MIN/MAX* SPECIAL RESTRICTIONS

<i>i</i>	# <i>SR</i>	<i>LpTime</i>	<i>IpTime</i>	<i>Nodes</i>
1	57	10:38:17	20:49:36	1835
2	65	11:21:42	22:45:25	1867
3	83	11:39:28	23:28:11	1893
4	78	11:21:03	23:38:26	1861
5	69	13:15:33	26:40:49	1939
6	81	12:00:15	25:29:18	1893
7	70	16:07:01	32:34:18	2009
8	97	16:23:08	33:43:23	2027
9	95	16:16:12	33:46:44	2045
10	80	15:44:48	32:11:27	1945
11	76	13:18:12	26:51:07	1925
12	82	16:38:17	33:32:04	1985

TABLE IV.
COMPUTATIONAL RESULTS - ONLY *WITH* SPECIAL RESTRICTIONS

<i>i</i>	# <i>SR</i>	<i>LpTime</i>	<i>IpTime</i>	<i>Nodes</i>
1	73	10:53:09	21:07:16	1849
2	88	11:39:33	23:12:05	1859
3	64	11:54:51	23:18:34	1881
4	71	11:31:10	24:22:56	1877
5	69	13:04:22	26:31:33	1919
6	70	12:14:18	25:38:04	1897
7	63	15:46:29	32:53:10	2019
8	82	16:49:13	33:04:51	2015
9	57	16:31:47	34:37:23	2079
10	62	15:41:56	32:18:19	1927
11	70	13:23:04	27:40:53	1937
12	75	16:31:39	34:14:41	1997

TABLE V.
COMPUTATIONAL RESULTS - ONLY *WITHOUT* SPECIAL RESTRICTIONS

<i>i</i>	# <i>SR</i>	<i>LpTime</i>	<i>IpTime</i>	<i>Nodes</i>
1	26	10:41:11	21:28:32	1859
2	35	11:49:08	25:31:27	2011
3	19	11:35:16	23:50:44	1897
4	38	11:59:23	26:07:16	1977
5	21	13:11:28	26:33:55	1943
6	12	12:01:30	25:36:47	1881
7	15	15:51:39	32:09:36	2011

8	26	16:25:27	35:11:58	2053
9	18	16:18:40	33:52:09	2053
10	33	15:27:12	34:56:19	2033
11	24	13:14:22	28:01:33	1989
12	17	16:24:04	33:17:29	1993

TABLE VI.
COMPUTATIONAL RESULTS - ONLY *PAIRING* SPECIAL RESTRICTIONS

<i>i</i>	# <i>SR</i>	<i>LpTime</i>	<i>IpTime</i>	<i>Nodes</i>
1	12	10:49:41	21:32:37	1853
2	13	11:37:52	22:59:20	1871
3	10	11:47:02	23:37:52	1881
4	13	11:39:14	23:11:33	1871
5	12	13:01:29	26:30:53	1963
6	13	12:08:29	25:51:37	1891
7	12	15:45:18	46:39:33	5305
8	13	16:34:11	33:51:21	2045
9	10	16:24:03	37:17:44	2193
10	10	16:09:57	41:45:56	3627
11	12	13:06:09	27:57:29	1957
12	13	16:23:14	34:43:05	2009

As the above results demonstrate, the *min/max*, and the *with* restriction types do not appear to increase the computational times considerably; the *without* and the *pairing* restriction types, on the other hand, result in substantially higher computational times for some problem instances (highlighted in bold). An interesting relevant observation regards the fact that the size of the branch & price tree also appears to be significantly larger in those problem instances, as depicted by the corresponding number of tree nodes.

An important observation that provides valuable insight into this behavior is the fact that the imposition of the *without* and the *pairing* restriction types necessitates auxiliary decision variables, whereas that of the *min/max* and the *with* restriction types does not. Combined with the fact that a solution in which all roster variables satisfy the integrality constraints may still fail to be feasible simply because the associated auxiliary decision variables do not, this observation helps justify why some computational times are significantly higher.

To provide a specific example for the *without* restriction type, consider the case of a duty with total complement equal to 4. A solution in which all roster variables are integer may assign to this duty two crew members of the first of the mutually exclusive crew categories and two of the second one, with the corresponding auxiliary variable being equal to 0.5, and thus failing to satisfy integrality.

Similarly, a solution in which all roster variables are integer may assign to a duty with an active pairing restriction three crew members of the specified crew category, with the corresponding auxiliary variable being equal to 1.5, and thus failing to satisfy integrality.

Of course, the algorithmic methodology can always backtrack and reverse previously made poor branching decisions that lead to such pitfalls, but this is both time consuming and tedious. The branching strategy (choice of the next decision variable on which branching is performed) is a vital element of the algorithmic methodology that can crucially affect computational performance; consequently, several alternative branching schemes have been proposed in the literature [26].

Based on proper experimentation, in our algorithmic implementation we adopt the strategy of branching on the largest fractional variable. For roster variables, the branch on 1 decision assigns a specific roster to the corresponding crew member, while the branch to 0 decision prohibits this assignment. For a *without* auxiliary decision variable, each of the two branching values determines which of the two mutually exclusive crew categories can be assigned to the duty in question. Finally, for a pairing auxiliary decision variable, each of the two branches imposes an even number as a lower/upper bound on the number of crew members of the associated crew category assigned to this duty.

To circumvent the computational effort challenges that can potentially arise in the presence of *without* and/or *pairing* restrictions, we adopt a slightly different branching strategy. Specifically, we give higher branching priority to the auxiliary decision variables associated with the special restrictions, and we do not choose to branch on a roster variable unless all of them are integer valued.

This is a very crucial algorithmic amendment that improves computational performance significantly, as justified by additional computational results presented next. The rationale motivating its adoption is that it avoids feeding the algorithmic implementation with misleading information that would significantly deteriorate its performance.

More specifically, in the case of a *without* restriction type, it clarifies in advance which of the two mutually exclusive crew categories will be freely assigned to the duty in question and which will be prohibited from assignment, thus assisting the algorithm in deciding earlier which particular roster variables will be used to this end. Similarly, in the case of a *pairing* restriction type, it excludes in advance the assignment of the intermediate odd number of crew members to the duty in question, which would otherwise appear as feasible if the auxiliary variable assumed a fractional value.

Failing to adopt this strategy in case of a *without* restriction type often leads the algorithm to assume that both mutually exclusive categories can be assigned to the duty in question simultaneously, which results in a large number of roster assignment decisions that need to be reversed in order to reach a feasible solution. Similarly, failing to adopt it in case of a *pairing* restriction often leads the algorithm to assume that an odd number of the specified category crew members can be assigned to the duty, which, in a similar way, necessitates the reversion of a large number of roster assignment decisions in order to reach a feasible solution.

The next two tables present computational results for the above problem instances with the same *without* and *pairing* restriction types activated, the only difference being that the strategy of not branching on a roster variable unless all *without/pairing* auxiliary decision variables are integer-valued is adopted (we use the term *without/pairing priority branching* for this strategy). In the presence of many fractionally valued auxiliary decision variables, the algorithm chooses to branch on the one with the largest value.

TABLE VII.

 COMPUTATIONAL RESULTS - ONLY *WITHOUT* SPECIAL RESTRICTIONS - *WITHOUT* PRIORITY BRANCHING

<i>i</i>	<i>LpTime</i>	<i>IpTime</i>	<i>Nodes</i>
1	10:37:11	21:21:17	1841
2	11:44:08	23:53:52	1893
3	11:38:27	24:01:16	1885
4	12:01:37	24:41:28	1879
5	13:14:52	26:24:08	1951
6	12:00:19	25:21:12	1869
7	15:54:42	32:13:29	2017
8	16:24:13	33:51:05	2029
9	16:16:04	33:08:59	2031
10	15:24:43	33:17:33	1951
11	13:07:14	27:08:27	1947
12	16:18:40	32:37:11	1977

TABLE VIII.

 COMPUTATIONAL RESULTS - ONLY *PAIRING* SPECIAL RESTRICTIONS - *PAIRING* PRIORITY BRANCHING

<i>i</i>	<i>LpTime</i>	<i>IpTime</i>	<i>Nodes</i>
1	10:45:04	21:17:02	1837
2	11:35:52	22:26:11	1841
3	11:43:03	23:39:28	1875
4	11:41:35	23:23:57	1885
5	13:00:37	26:16:27	1961
6	12:05:43	24:58:50	1869
7	15:48:33	32:17:29	2005
8	16:37:13	33:58:40	2037
9	16:27:10	34:03:15	2041
10	16:17:26	35:06:22	1989
11	12:59:26	26:44:18	1937
12	16:25:57	34:31:36	2003

The above results demonstrate that the computational challenges posed by the *without/pairing* restriction types are handled quite satisfactorily by the *without/pairing priority branching* strategy. Its adoption avoids the excessively large branch & price trees, and in turn the dramatically increased computational times encountered sometimes otherwise. For this reason, it has been adopted as the standard branching strategy in the actual algorithmic implementation.

Note that the computational time required for the solution of the master LP relaxation at the root of the branch & price tree is independent of the adopted branching strategy. Consequently, the corresponding *Lp Time* values should coincide in Tables 5 and 7, as well as in Tables 6 and 8. The small differences observed are negligible and can be explained by the inevitable variance in the computer processor's task workload.

VI. SUMMARY AND FUTURE RESEARCH

In this research work, we have addressed the problem of incorporating special restrictions into column generation algorithms for aircrew rostering. We considered three types of restrictions typically encountered in commercial aviation, and we illustrated how the algorithmic methodology needs to be modified for accommodating them.

Computational results from realistic problem instances indicate that, in some special cases, the inclusion of some specific restriction types may lead to substantially increased branch & price trees and, consequently, computational times.

Mainly, this behavior appears to be present because these restriction types exploit auxiliary decision variables which may provide misleading information to the optimizer if their exact integer values have not been determined promptly.

To properly handle these difficulties, we devised a specialized branching strategy which gives higher branching priority to the auxiliary decision variables. Additional computational experiments confirm that its adoption addresses successfully the difficulties introduced by the increased computational times; consequently, it has been adopted as standard in the actual algorithmic implementation.

The present research can be extended in many fruitful ways. An interesting direction for future research involves the development of additional special techniques for handling the computational challenges potentially arising by the incorporation of special restrictions that involve auxiliary decision variables. An innovative idea in that direction is to determine the optimal integer values of these decision variables and temporarily enforce them to the master LP relaxation for identifying its optimal dual solution utilized for the solution of the shortest path pricing subproblem.

Another interesting direction for future research is the investigation of how the incorporation of the special restrictions affects the quality of the final solutions. As the present research is currently in progress, we solely investigate the corresponding effect on computational performance; such a future research extension will clearly render it more complete.

ACKNOWLEDGMENT

The research is conducted in the operating framework of the University of Thessaly Innovation, Technology Transfer Unit and Entrepreneurship Center “One Planet Thessaly”, under the “University of Thessaly Grants for Scientific Publication Support” action and is funded by the Special Account of Research Grants of the University of Thessaly. The authors have not used any AI tool, whatsoever, during the preparation of this work and take full responsibility for its content.

REFERENCES

- [1] M. Fuentes, L. Cadarso, V. Vaze, and C. Barnhart, “The tail assignment problem: A case study at Vueling Airlines,” *Transp. Res. Proc.*, 52, pp. 445-452, 2021. <https://doi.org/10.1016/j.trpro.2021.01.052>
- [2] G. Kozanidis, “Optimal assignment of aircrew trainees to simulator and classroom training sessions subject to seniority and preference restrictions,” *J. Air Transp. Manag.*, 59, pp. 143-154, 2017. <https://doi.org/10.1016/j.jairtraman.2016.11.012>
- [3] H. Dawid, J. König, and C. Strauss, “An enhanced rostering model for airline crews,” *Comput. Oper. Res.*, 28(7), pp. 671-688, 2001. [https://doi.org/10.1016/S0305-0548\(00\)00002-2](https://doi.org/10.1016/S0305-0548(00)00002-2)
- [4] B. Gopalakrishnan, and E. L. Johnson, “Airline crew scheduling: State-of-the-art,” *Ann. Oper. Res.*, 140, pp. 305-337, 2005. <https://doi.org/10.1007/s10479-005-3975-3>
- [5] A. Aydemir-Karadag, B. Dengiz, and A. Bolat, “Crew pairing optimization based on hybrid approaches,” *Comput. Ind. Eng.*, 65(1), pp. 87-96, 2013. <https://doi.org/10.1016/j.cie.2011.12.005>
- [6] P. R. Day, and D. M. Ryan, “Flight attendant rostering for short-haul airline operations,” *Oper. Res.*, 45(5), pp. 649-661, 1997. <https://doi.org/10.1287/opre.45.5.649>
- [7] M. Gamache, F. Soumis, D. Villeneuve, J. Desrosiers, and E. Gélinas, “The preferential bidding system at Air Canada,” *Transp. Sci.*, 32(3), pp. 246-255, 1998. <https://doi.org/10.1287/trsc.32.3.246>

- [8] A. I. Z. Jarrah, and J. T. Diamond, “The problem of generating crew bidlines,” *Interfac.*, 27(4), pp. 49-64, 1997. <https://doi.org/10.1287/inte.27.4.49>
- [9] J. Desrosiers, and M. E. Lübbecke, “A primer in column generation,” in *Column Generation*, G. Desaulniers, J. Desrosiers, and M.M. Solomon, Eds. Boston, MA, US: Springer, pp. 1-32, 2005. https://doi.org/10.1007/0-387-25486-2_1
- [10] D. Ryan, “The solution of massive generalized set partitioning problems in air crew rostering,” *J. Oper. Res. Soc.*, 43, pp. 459-467, 1992. <https://doi.org/10.1057/jors.1992.72>
- [11] M. Gamache, F. Soumis, G. Marquis, and J. Desrosiers, “A column generation approach for large-scale aircrew rostering problems,” *Oper. Res.*, 47(2), pp. 247-263, 1999. <https://doi.org/10.1287/opre.47.2.247>
- [12] T. Fahle, U. Junker, S. E. Karisch, N. Kohl, M. Sellmann, and B. Vaaben, “Constraint programming based column generation for crew assignment,” *J. Heuristics*, 8(1), pp. 59-81, 2002. <https://doi.org/10.1023/A:1013613701606>
- [13] J. Desrosiers, and M. E. Lübbecke, “Branch-price-and-cut algorithms,” in *Encyclopedia of Operations Research and Management Science*. Chichester, UK: John Wiley & Sons, pp. 109-131, 2011. <https://doi.org/10.1002/9780470400531.eorms0118>
- [14] G. Kozanidis, “Branch and price for covering shipments in a logistic distribution network with a fleet of aircraft,” *Optim. Methods Softw.*, 33(2), pp. 221-248, 2018. <https://doi.org/10.1080/10556788.2017.1281923>
- [15] G. Kozanidis, “Column generation for optimal shipment delivery in a logistic distribution network,” in *Sustainable Logistics and Transportation*, D. Cinar, K. Gakis, and P.M. Pardalos, Eds. Springer Optimization and Its Applications, vol. 129, pp. 87-112. Berlin: Springer-Verlag, 2018. https://doi.org/10.1007/978-3-319-69215-9_5
- [16] Z. Liu, and Q. Xiang, “A branch-and-price algorithm for the airport gate assignment problem considering the trade-off between robustness and efficiency,” *Transp. Res. C Emerg. Technol.*, 154, 104232, 2023. <https://doi.org/10.1016/j.trc.2023.104232>
- [17] G. Kozanidis, “An integrated column generation solution framework for optimal aircrew vacation planning subject to seniority ranking and priority preference satisfaction,” *Omega*, 135, 103224, 2025. <https://doi.org/10.1016/j.omega.2025.103324>
- [18] J. Desrosiers, F. Soumis, and M. Desrochers, “Routing with time windows by column generation,” *Networks*, 14(4), pp. 545-565, 1984. <https://doi.org/10.1002/net.3230140406>
- [19] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, and P. H. Vance, “Branch-and-price: Column generation for solving huge integer programs,” *Oper. Res.*, 46(3), pp. 316-329, 1998. <https://doi.org/10.1287/opre.46.3.316>
- [20] M. Gamache, and F. Soumis, “A method for optimally solving the rostering problem” in *Operations Research in the Airline Industry*, G. Yu, Ed. Boston, MA, US: Springer, pp. 124-157, 1998. https://doi.org/10.1007/978-1-4615-5501-8_5
- [21] A. Kasirzadeh, M. Saddoune, and F. Soumis, “Airline crew scheduling: Models, algorithms, and data sets,” *EURO J. Transp. Logist.*, 6(2), pp. 111-137, 2017. <https://doi.org/10.1007/s13676-015-0080-x>
- [22] G. Kozanidis, and O. Moschopoulos, “Set-cover master problem formulations for maximum flight coverage in branch & price solution algorithms for optimal aircrew rostering,” *Oper. Res. Int. J.*, 25(3), 69, 2025. <https://doi.org/10.1007/s12351-025-00950-0>
- [23] G. Kozanidis, and O. Moschopoulos, “Set-cover master problem formulations in branch & price solution methodologies for optimal aircrew scheduling,” in *Lecture Notes in Operations Research, Annual International Conference of the German Operations Research Society*, Hamburg, Germany, August 29 - September 1, pp. 249-256, 2023. https://doi.org/10.1007/978-3-031-58405-3_32
- [24] AIMS INTL DWC-LLC (2025). Airline solutions package and aviation software. <https://www.aimsairlinesoftware.com/> Last accessed on April 13, 2026.
- [25] IBM ILOG CPLEX Optimization Studio v. 22.1.1 (2022). <https://www.ibm.com/docs/en/icos/22.1.1> . Last accessed on April 13, 2026.
- [26] F. Vanderbeck, “On Dantzig-Wolfe decomposition in integer programming and ways to perform branching in a branch-and-price algorithm,” *Oper. Res.*, 48(1), pp. 111-128, 2000. <https://doi.org/10.1287/opre.48.1.111.12453>