Column Generation Approaches to a Robust Airline Crew Pairing Model For Managing Extra Flights*

Elvin Çoban[1], İbrahim Muter[1], Duygu Taş[1], Ş. İlker Birbil[1], Kerem Bülbül[1], Güvenç Şahin[1], Y. İlker Topçu[2], Dilek Tüzün[3], and Hüsnü Yenigün[1]

[1] Sabancı University, Orhanlı-Tuzla, 34956 Istanbul, Turkey
(elvinc, imuter, duygutas)@su.sabanciuniv.edu,
(sibirbil, bulbul, guvencs, yenigun)@sabanciuniv.edu
[2] Istanbul Technical University, Maçka-Beşiktaş, 34367 Istanbul, Turkey
ilker.topcu@itu.edu.tr
[3] Yeditepe University, Kayışdağı-Kadıköy, 34755 Istanbul, Turkey
dtuzun@yeditepe.edu.tr

1 Introduction

The airline crew pairing problem (CPP) is one of the classical problems in airline operations research due to its crucial impact on the cost structure of an airline. Moreover, the complex crew regulations and the large scale of the resulting mathematical programming models have rendered it an academically interesting problem over decades. The CPP is a tactical problem, typically solved over a monthly planning horizon, with the objective of creating a set of crew pairings so that every flight in the schedule is covered, where a crew pairing refers to a sequence of flights operated by a single crew starting and ending at the same crew base.

This paper discusses how an airline may hedge against a certain type of operational disruption by incorporating robustness into the pairings generated at the planning level. In particular, we address how a set of extra flights may be added into the flight schedule at the time of operation by modifying the pairings at hand and without delaying or canceling the existing flights in the schedule. We assume that the set of potential extra flights and their associated departure time windows are

* This research has been supported by The Scientific and Technological Research Council of Turkey under grant 106M472.

known at the planning stage. We note that this study was partially motivated during our interactions with the smaller local airlines in Turkey which sometimes have to add extra flights to their schedule at short notice, e.g., charter flights. These airlines can typically estimate the potential time windows of the extra flights based on their past experiences, but prefer to ignore this information during planning since these flights may not need to be actually operated. Typically, these extra flights are then handled by recovery procedures at the time of operation which may lead to substantial deviations from the planned crew pairings and costs. The reader is referred to [3] for an in-depth discussion of the conceptual framework of this problem which we refer to as the Robust Crew Pairing for Managing Extra Flights (RCPEF). In [3], the authors introduce how an extra flight may be accommodated by modifying the existing pairings and introduce a set of integer programming models that provide natural recovery options without disrupting the existing flights. These recovery options are available at the planning stage and render operational recovery procedures that pertain to crew pairing unnecessary.

The main contribution of this work is introducing a column generation algorithm that can handle the robust model proposed in the next section. This model poses an interesting theoretical challenge and is not amenable to a traditional column generation algorithm designed for the conventional CPP. We point out that in [3] the authors explicitly generate all possible crew pairings and solve the proposed integer programs by a commercial solver. This approach is clearly not computationally feasible for large crew pairing instances, and in the current work we present our preliminary algorithms and results for large instances of RCPEF. We demonstrate the proposed solution approaches on a set of actual data acquired from a local airline [2].

2 Robust Airline Crew Pairing Problem

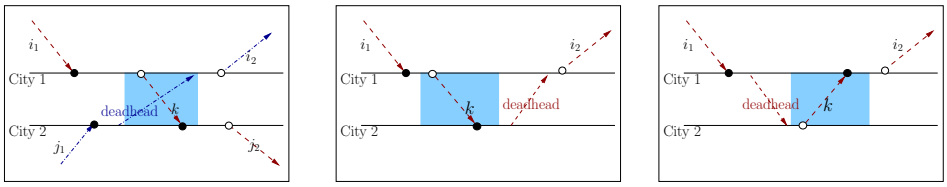
In this section, we first introduce the proposed robust model and then discuss the difficulties that arise while solving this model by conventional methods. This leads us to the two solution approaches presented in this paper.

In [3], the authors examine several recovery options for managing the extra flights at the planning level. They classify the possible solutions into two types:

- **Type A.** Two pairings are selected and (partially) swapped to cover an extra flight.

- **Type B.** One pairing with sufficient connection time between two consecutive legs is modified to cover an extra flight.

In this work, we incorporate one Type A and two Type B solutions as illustrated in Figure 1 where the estimated time window of the extra flight k is depicted by the blue (shaded) rectangles. In Figure 1(a), the original pairings p and q, covering the flight legs i_1, i_2 and j_1, j_2, respectively, are partially swapped so that the extra flight k is inserted into the flight schedule (Type A). The resulting pairings after swapping are illustrated in the figure where the term deadhead refers to a repositioning of crew members to another airport as passengers on a flight, train, etc. In Figures 1(b) and 1(c), the original pairing p is modified to accommodate the extra flight k (Type B). The feasibility rules that define both Type A and B solutions are explained in detail in [3].



(a) Pairings p,q are swapped. (b) Pairing p is modified. (c) Pairing p is modified.

Fig. 1. Recovery options for covering the extra flight k.

The proposed robust mathematical model is given below:

$$\min \sum_{p\in\mathcal{P}} c_p y_p + \sum_{k\in\mathcal{K}} d_k z_k + \sum_{k\in\mathcal{K}} d_k \left[\sum_{p\in\mathcal{P}} (1-y_p)\bar{a}_{kp} + \sum_{p,q\in\mathcal{P}} (1-x^k_{(p,q)})\bar{a}_{pqk} \right] \tag{1}$$

s.t

$$\sum_{p\in\mathcal{P}} a_{ip} y_p \geq 1, \qquad \forall i \in \mathcal{F}, \tag{2}$$

$$\sum_{p\in\mathcal{P}} \bar{a}_{kp} y_p + \sum_{p,q\in\mathcal{P}} \bar{a}_{pqk} x^k_{(p,q)} \geq 1 - z_k, \qquad \forall k \in \mathcal{K}, \tag{3}$$

$$2\bar{a}_{pqk} x^k_{(p,q)} \leq y_p + y_q, \qquad \forall p,q \in \mathcal{P}, \forall k \in \mathcal{K} \tag{4}$$

$$y_p \in \{0,1\}, \qquad p \in \mathcal{P}, \tag{5}$$

$$z_k \in \{0,1\}, \qquad k \in \mathcal{K}, \tag{6}$$

$$x^k_{(p,q)} \in \{0,1\}, \qquad p,q \in \mathcal{P}, k \in \mathcal{K}, \tag{7}$$

where $\mathcal{F}$ is the set of all flights, $\mathcal{K}$ is the set of all extra flights, and $\mathcal{P}$ is the set of all feasible pairings. Here, c_p is the cost of pairing p, and d_k is the opportunity cost of failing to cover extra flight k. Furthermore, we define the parameters $a_{ip} = 1$ if flight i is included in pairing p, and 0 otherwise; $\bar{a}_{kp} = 1$ if extra flight k can be inserted into pairing p as a Type B solution, and 0 otherwise; and $\bar{a}_{pqk} = 1$ if pairings p and q can form a Type A solution to cover extra flight k, and 0 otherwise. The decision variable y_p is set to 1 if pairing p is selected, and 0 otherwise. Also, let $x^k_{(p,q)}$ be an auxiliary binary variable that takes the value 1 if two pairings p and q forming a Type A solution for extra flight k are both included in the solution, and 0 otherwise. Finally, we define the binary variable z_k equal to 1 if no Type A or B solution is present in the solution for extra flight k, and 0 otherwise.

The objective (1) minimizes the sum of the pairing costs and the opportunity costs of not accommodating the extra flights. Constraints (2) and (3) are the coverage constraints for the regular and extra flights, respectively. Observe that the model may opt for not covering an extra flight k if this is too expensive, setting z_k to 1. Constraints (4) prescribe that a Type A solution for extra flight k formed by pairings p and q is only possible if both of these pairings are selected.

The formulation (1)-(7) has both exponentially many variables, one for each pairing, and exponentially many constraints of type (4) which makes it both practically and theoretically challenging. Typical crew pairing models incorporate exponentially many variables, but have a fixed number of constraints and are solved by traditional column generation approaches where the pricing subproblem is a multi-label shortest path problem solved over an appropriate flight/duty network. (See [1] for a review of these concepts.) In our proposed robust model, the number of constraints (4) is not known a priori and depends on the pairings present in the model. Thus, ideally this formulation requires simultaneous row and column generation. In the next section, we present our preliminary algorithms developed for the problem RCPEF.

3 Solution Approaches

In both approaches presented here, the primary goal is to fix the number constraints in the model before applying column generation.

The Static Approach

In the "static" approach, all pairings that construct Type A solutions are generated a priori before column generation is applied to the linear

programming (LP) relaxation of (1)-(7). To this end, we identify all possible flights and connections that may appear in a pairing before covering an extra flight or its associated deadhead by a breadth-first-search and then construct pairings over this reduced network. We next run pairwise feasibility checks on these generated pairings to determine Type A solutions. Thus, all constraints (4) are identified and added to the model along with the associated auxiliary variables $x^k_{(p,q)}$ before the column generation procedure is invoked to identify Type B solutions and new pairings that may lower the objective function value. Upon termination of the column generation procedure, a primal heuristic is applied if the LP optimal solution is not integral.

We point out that the static approach is an exact method for solving the LP relaxation of (1)-(7) because all constraints (4) are explicitly included in the model. Clearly, the computational effort for this algorithm will be excessive for large instances of RCPEF.

The Dynamic Approach

In the static approach, all constraints (4) are incorporated into the formulation prior to column generation. In the "dynamic" approach, we opt for the complete opposite for speed. We exclude all variables $x^k_{(p,q)}$ and constraints (4) from the formulation and dynamically generate pairings that reduce the objective and yield Type B solutions. After the column generation terminates, we check whether the available pairings yield any Type A solutions and add the associated constraints and variables to the model. Next, we solve the LP relaxation of (1)-(7) with the available constraints and variables and invoke a primal heuristic, if necessary, in order to obtain an integer feasible solution to RCPEF.

The proposed dynamic approach does not necessarily provide an optimal solution to the LP relaxation of (1)-(7) because pairings leading to Type A solutions may be missed during the column generation. In order to reach a compromise between speed and solution quality, we promote that at least N (partial) pairings that may potentially form Type A solutions are kept on each node during the pricing subproblem. At the end of the pricing subproblem, such pairings are added to a special pool. This pool is examined for Type A solutions after the column generation terminates.

4 Computational Results

In this section, we present our preliminary results on the proposed static and dynamic approaches. Our primary goal is to illustrate the

trade-off between robustness (as indicated by the number of Type A and B solutions obtained) and computational effort. We conducted a numerical study on two sets of actual data. The results are presented in Table 1 where in each cell the number of Type A solutions identified is followed by the solution time in parentheses. No more than 2 Type B solutions per extra flight were obtained in all cases.

		Dynamic					**Static**
$\mid \mathcal{F} \mid$	$\mid \mathcal{K} \mid$	$N = \mathbf{0}$	$N = \mathbf{10}$	$N = \mathbf{50}$	$N = \mathbf{100}$	$N = \mathbf{500}$	
	1	0(0.17)	10(0.31)	14(0.62)	18(0.71)	18(0.96)	18(0.50)
42	2	0(0.17)	10(0.32)	86(0.67)	98(0.75)	118(1.03)	118(0.70)
	3	0(0.20)	10(0.42)	93(0.71)	106(0.81)	128(1.31)	128(1.34)
	1	10(0.71)	40(1.07)	64(1.39)	64(1.70)	64(2.57)	64(1.83)
96	2	20(0.86)	80(1.15)	128(1.64)	128(1.98)	128(2.87)	128(5.86)
	3	38(0.86)	60(1.25)	136(1.76)	141(2.06)	141(3.42)	141(9.70)

Table 1. Comparison of the number of Type A solutions and CPU times for the dynamic and static approaches.

Two trends are clear from Table 1. First, the performance of the dynamic approach depends critically on the value of N. There is a threshold value for N above which extra solution time is spent with no additional benefit. Second, the dynamic approach outperforms the static approach for large problem instances.

5 Future Research

The results in Section 4 point to a clear need for simultaneous row and column generation for solving the proposed model. We are going to pursue this interesting direction in the future.

References

1. Klabjan D (2005) Large-scale models in the airline industry. In: Desaulniers G, Desrosiers J, Solomon MM (eds) Column Generation. Springer
2. Çoban E (2008) Column generation approaches to a robust airline crew pairing model for managing extra flights. MS Thesis, Sabancı U, Istanbul
3. Tekiner H, Birbil Şİ, Bülbül K (2008) Robust crew pairing for managing extra flights. To appear in Computers and Operations Research, DOI: 10.1016/j.cor.2008.07.005