

Robust Crew Pairing for Managing Extra Flights

Hatice Tekiner

Manufacturing Systems and Industrial Engineering,
Faculty of Engineering and Natural Sciences, Sabancı University, 34956 Istanbul, Turkey.
htekiner@su.sabanciuniv.edu

Ş. İlker Birbil

Manufacturing Systems and Industrial Engineering,
Faculty of Engineering and Natural Sciences, Sabancı University, 34956 Istanbul, Turkey.
sibirbil@sabanciuniv.edu

Kerem Bülbül

Manufacturing Systems and Industrial Engineering,
Faculty of Engineering and Natural Sciences, Sabancı University, 34956 Istanbul, Turkey.
bulbul@sabanciuniv.edu

ABSTRACT: This paper discusses a modeling approach to robust crew pairing when a set of extra flights is likely to be added to the regular flight schedule. The set of these possible extra flights is known at the planning stage. We demonstrate that these extra flights may be incorporated into the schedule if necessary by modifying the planned crew pairings appropriately and without delaying or canceling existing flights. To this end, we either identify a pair of crews whose schedules may be (partially) swapped while adding an extra flight into the schedule or show that an extra flight may be inserted into the schedule of a crew without affecting others. We note that deadheading may be necessary in either case. For these two types of solutions, we define the appropriate feasibility rules with respect to the common airline regulations. We then propose two robust mathematical programming models that consider incorporating such solutions into the set of selected pairings while keeping the increase in the crew cost at an acceptable level. The baseline solution for comparison is found by a conventional crew pairing model in the literature which ignores robustness at the planning stage and relies on recovery procedures at the time of operation. We also propose the variations of the two models, where the double counting of the possible solutions across extra flights is prevented. Finally, we conduct computational experiments on a set of data generated from the actual data of an airline company. We solve the crew pairing problem both with the proposed robust models and the conventional model. Our results demonstrate the benefits of the proposed modeling approach and indicate that the proposed robust models provide natural options to recovery without disrupting the existing flights at a relatively small incremental cost, which is visible at the planning stage.

Keywords: Robustness; crew pairing; extra flights; charters; mathematical models; computational study

1. Introduction. The airlines have been applying operations research techniques to their planning problems at strategic, tactical and operational levels since the 1950's. In particular, the American airline companies have been using mathematical techniques heavily in order to increase their profitability and market shares after the Airline Deregulation Act of 1978 intensified the competition. The demand for mathematical approaches from the airlines prompted a rapid development of many existing and new areas in operations research; such as, revenue and yield management, network planning and design, and so on. One of the major areas of focus has been the airline crew scheduling problem as the crew costs come only second to the fuel costs for a typical airline and is widely studied in the literature [4, 10]. Due to the complex regulations and the large size of the problems, the construction of an effective schedule is a very challenging task. Therefore, the crew scheduling problem is generally divided into *crew pairing* and *crew assignment* problems. In the crew pairing problem, the objective is to determine a crew for each flight leg in the schedule, and in this problem a crew is considered as an entity, i.e., the composition of the crew and the identities of its crew members are not taken into account. In the crew assignment problem, a schedule for each crew member is determined by assigning the individual crew members to the constructed pairings. The crew rosters are typically created for an entire month and published at the end of the previous month. Therefore, the crew scheduling problem is considered as a tactical level problem, and crew schedules are subject to change depending on operational contingencies. In this

study, we concentrate on the crew pairing problem, and in particular we focus on the robustness of the generated pairings to certain types of changes at the time of operation.

Advances in mathematical programming techniques and the speed of computers allowed the airlines to remove the slacks in their crew schedules and decrease their planned costs over the last two decades. However, this at the same time implies that the crew schedules became more sensitive to irregularities or disruptions in airline operations leading to a potentially substantial gap between the actual and the planned costs. Weather conditions, airport congestion and employee sicknesses are typical causes of such disruptions, and some studies in the literature construct robust schedules hedging against flight cancellations and delays [8, 15]. In our study, the source of disruptions is a set of extra flights added to the flight schedule during operation, and our objective is to relieve the adverse effects of these new flights by considering them while generating crew pairings.

This study was partly motivated by our interactions with the smaller local airlines in Turkey. The majority of the literature is concentrated on large size problems for major airline companies [1, 16]. However, our conversations with the officials of local airlines revealed a gap between the current state of the literature and the types of planning and operational problems faced by these airlines. One of these issues is the management of flights added to the schedule at short notice, e.g., charter flights. Currently, such disruptions are typically handled by employing recovery procedures during operation [11]. In this study, we focus on the effects of extra flights on the crew costs and demonstrate the benefits of accounting for potential changes during planning.

There may be several reasons for adding new flights into the schedule during operation. For instance, some local airline companies observe demand to popular holiday destinations during the summer and add new flights as needed in order to ensure full flights. For domestic destinations, such flights would typically be added on Friday afternoons and early Monday mornings in selected weeks. In addition, certain types of customers, such as businessmen, government officials, sports teams, and so on, occasionally demand charter flights outside the regular schedule at short notice. Moreover, the need for scheduling an extra flight may also arise due to the way the airline industry is regulated. Crew members are typically licensed to fly either on a single aircraft type (cockpit crew) or on two or three aircraft types (cabin crew). Therefore, the crew pairing problem decomposes by fleet type and is solved separately for each fleet type. However, the aircraft type for a given flight may later be changed due to maintenance (breakdown) or demand (upgrade/downgrade to a larger/smaller aircraft) concerns. In essence, this implies that a flight is deleted from the schedule of one fleet and is added to another one. (Note that this is an issue for any type of airline.) In any case, the visibility of the airlines into these events is generally low, and most of the local airline companies disregard the possibility of adding extra flights while constructing monthly crew schedules and postpone this issue to the operational level. In this paper, we provide crew pairing models that can account for this uncertainty given a *robustness budget*. We first solve the crew pairing problem by ignoring the set of potential extra flights, which provides us with a baseline crew cost. Given the solution to this model, inserting an extra flight into the schedule requires recovery actions during operation. This typically leads to delays and/or cancellations of existing flights depending on the relative importance of the extra flight and may cause substantial cost hikes. Then, we set up and solve robust crew pairing models which provide us with recovery options for each extra flight if possible. These recovery options do not affect the existing flights in the schedule. In the proposed models, the total crew pairing cost is constrained to be no more than a user-specified percentage above the baseline cost provided by the conventional model. We assume here that each potential extra flight is associated with a time window of possible departure times. In other words, the airline may add an extra flight at any time in its respective time window without disrupting other flights. Therefore, we only require that the airline determines certain time intervals in which it would ever consider adding extra flights based on past experiences, etc.

The outline of the paper is as follows. In Section 2, we introduce some common definitions used in airline crew pairing. We then provide a literature survey about the airline crew pairing problem and the concept of robustness. This is followed, in Section 3, by the description of our problem. In this section, we also explain the possible solutions to cover an extra flight and then formally introduce our models in Section 4. Subsequently, we give our computational results in Section 5. Finally, we state our conclusions and discuss some future research directions in Section 6.

2. The Crew Pairing Problem. A crew pairing is a sequence of flights separated by connection times at the airports and rest periods at the end of each work day. A pairing may span several work days and is flown by a single crew. In the crew pairing problem, the objective is to select a subset of all feasible

pairings in order to cover each flight leg in the flight schedule exactly once with the minimum total crew cost. This problem may be stated as a set partitioning problem

$$\begin{aligned} \min \quad & \sum_{p \in \mathcal{P}} c_p y_p \\ \text{s.t.} \quad & \sum_{p \in \mathcal{P}} a_{ip} y_p = 1, \quad i \in \mathcal{F}, \\ & y_p \in \{0, 1\}, \quad p \in \mathcal{P}, \end{aligned} \quad (1)$$

where \mathcal{F} is the set of all flights, \mathcal{P} is the set of all possible pairings, c_p is the cost of pairing $p \in \mathcal{P}$, $a_{ip} = 1$ if flight leg i is covered by pairing p , and 0 otherwise, and y_p is a binary variable assuming a value of 1 if pairing p is selected, and 0 otherwise. The objective is to minimize the total cost of selected pairings. The set of equality constraints guarantees that each flight is covered only once. Additional side constraints, such as staff capacity constraints limiting the number of pairings originating at a particular airport, may also be included in this model.

There is a large body of literature available on airline crew pairing. (See [3, 10] for surveys.) The main difficulty in solving the crew pairing problem is the huge number of pairings (variables/columns) present in (1). Even problems with a few hundred flights may have billions of possible pairings [3], and it becomes computationally prohibitive to explicitly enumerate all possible pairings and then solve (1). Therefore, most methods developed for the crew pairing problem focus on either local optimization methods or column generation based methods in which new pairings are introduced only as needed. Anbil *et.al.* use local improvement heuristics [1]. These heuristics begin with an initial solution and then select a small subset of the pairings in the current solution. All pairings are generated for the flights that are covered by the selected subset, and then, the crew pairing problem is solved for these flight legs to find a set of pairings with a lower cost covering these flights. If a better solution is found, then the original pairings are replaced by the newly identified set of pairings. In a follow-up study, Anbil *et.al.* introduce a method which employs the optimum dual variables in the linear programming (LP) relaxation of (1), which in this case includes only a small subset of all possible pairings, in order to identify promising pairings which may lower the total cost [2]. In early works, like Crainic and Rousseau [6], the column generation idea is used to solve the LP relaxation of the crew pairing problem. Subsequently, dynamic column generation techniques are implemented within branch-and-bound schemes to find integer solutions. For instance, Vance *et.al.* generate columns dynamically throughout the branch-and-bound tree, and at each node of the search tree, they apply column generation with the follow-on branching rule [16]. The follow-on branching rule is based on a branching rule developed earlier by Ryan and Foster [13] and states that given a fractional solution and a pair of corresponding flight legs, these two flights must be covered consecutively on one branch, and cannot follow each other on the other branch. A different modeling approach to the crew pairing problem is provided by Vance *et.al.* [17] who first construct work days (duty periods) composed of several flights legs and then string these work days together in order to form pairings. They solve their formulation by dynamic column generation and demonstrate that it leads to tighter LP bounds than the conventional set partitioning formulation (1).

In all of these studies, the reduced cost of a pairing with respect to the current LP relaxation (restricted master problem) is the pricing criterion in the column generation. In other words, the reduced cost determines whether a pairing is introduced as a new variable into the problem or not. There are also other pricing criteria employed in the literature. For example, Bixby *et.al.* introduce a new pricing criterion obtained by dividing the pairing cost by the sum of the dual values of the constraints corresponding to the flights covered by the pairing [5]. They observe a significant decline in the number of iterations required to solve the LP relaxations. Makri and Klabjan also use the same criterion [12] and develop pruning rules to fathom the column enumeration.

Recently, robust approaches to crew pairing started to emerge. Schaefer *et.al.* solve the crew pairing problem with the objective of minimizing the expected total cost [14]. These authors approximate the objective function coefficients in the model by simulation where they delay the flights if necessary until all required resources become available (push-back recovery). The crew pairing problem under uncertainty is also considered in Yen and Birge [18]. They formulate the crew pairing problem as a two-stage stochastic program and explore the effect of sit time between successive flights on the robustness of the solution. Ehrgott and Ryan penalize the connections without enough time to absorb the expected delay [8]. Then, they solve a bi-criteria optimization problem where the objective is to minimize the total cost along with the total penalty. Another approach to the robust crew pairing problem is presented in Shebalov and Klabjan [15]. These authors define the concept of move-up crews, i.e., crews whose schedules may be swapped if necessary during operation, as a means of achieving robustness. In their

model, they maximize the total number of move-up crews while controlling the crew cost.

An issue closely related to the robust crew pairing problem is crew recovery. In this problem, the objective is to restore the disrupted schedules as quickly as possible during operation. A model for this problem is presented by Lettovský *et.al.* [11]. By using preprocessing techniques, these authors first determine the crews whose schedules are to be re-constructed for recovery. Then, the possible pairings for these crews are generated, and the crew recovery model is solved.

All of the papers cited above strive to build more robust schedules with respect to disruptions. The disruptions taken into account in these papers, however, are due to delays or cancellations of flights. None of these studies considers disruptions due to extra flights added to the flight schedule. In this paper, we follow an idea similar to that proposed by Shebalov and Klabjan [15] and also define move-up crews as a resolution against disruptions. Thus, we propose an underlying recovery procedure that involves swapping crew schedules when extra flights are introduced into the schedule. We propose two robust models that aim at incorporating recovery opportunities as well as the variations of these two models that prevent double counting of the possible solutions across extra flights. The proposed models in this paper differ from that of Shebalov and Klabjan. The foremost difference is that we explicitly consider disruptions caused by the extra flights added to the flight schedule. To resolve these disruptions, we develop several possible solutions that involve a set of *pairing pairs* as well as repositioning flights (deadheads). Shebalov and Klabjan, on the other hand, focus on a set of pairings that yield swapping solutions at the hub airports without considering a specific type of disruption. In addition, the resulting mathematical model here becomes more involved than that in their paper, because the set of pairing pairs requires adding constraints to and deleting constraints from the model depending on the selected pairings. We further discuss this issue in Section 4. In Shebalov and Klabjan [15], the number of constraints in the model is constant regardless of the number of pairings included in the model. We observe that our model may lead to longer connection times; a concept that is promoted in [14, 8, 18]. We also give an in-depth study of the double counting issue in Section 5.

We next summarize some frequently used crew scheduling terminology that we also adopt in this paper. A flight leg (segment) is a nonstop flight. A sequence of flight legs over a working day of a crew, where the arrival station of a flight is equal to the departure station of the next flight, is called a duty period. The flight legs in a duty period are separated with short rest periods. The time between two flight legs in a duty period is referred to as sit time. The sit time is restricted to be between a minimum and a maximum sit time. The validity of a duty period is determined according to regulations imposed by national and international organizations (e.g. Federal Aviation Administration, International Airport Transportation Association). The cost of a duty period is determined by the maximum of three quantities: a fraction of the elapsed time of the duty, the minimum guaranteed pay per duty (expressed as time), and the total flight time in the duty period. A thorough account of the feasibility rules and the cost structure is given by Barnhart *et.al.* [3] and Johnson *et.al.*[10]. A pairing is a sequence of duty periods with an overnight rest between two successive duty periods. The overnight rest is generally called a rest or a layover. There are also lower and upper bounds for the rest time. Each pairing begins and ends at the same crew base. A crew base is an airline station where the crews are stationed. There are some additional feasibility rules for a pairing such as; the total time on duty in a pairing cannot exceed some threshold value, a crew can be away from its base at most for a certain amount of time (total away from base), and so on. The cost of a pairing is also given by the maximum of three quantities: the total cost of the duties in the pairing, a fraction of the total elapsed time of the pairing, and the minimum guaranteed pay per duty (expressed as time) multiplied by the number of duties in the pairing. We again refer the interested reader to Barnhart *et.al.* [3] and Johnson *et.al.* [10] for the details about the constraints and the cost structure. It is also common that crew members fly as passengers for repositioning purposes or in order to return to base after the last flight in the pairing. This is frequently referred to as deadheading.

3. Problem Statement and Solution Approaches. The problem studied in this paper is about hindering the adverse affects of a set of extra flights potentially added to the regular flight schedule. To follow the remaining part of this section, we summarize our notation in Table 1.

Although the exact times for possible extra flights are not known in advance, it can be assumed that the time interval (time window) in which an extra flight might be added is given. This assumption is not too restrictive, since the information about the time windows can usually be gathered using past experiences or mining the historical data. Figure 1 shows the time window for an extra flight k . In the figure, the horizontal lines denote the departure and arrival cities, and the movement from left to right shows the advance of time. To denote the earliest departure and the latest arrival times of flight k , we

\mathcal{K}	: set of all possible extra flights
d_k	: departure station of extra flight k
a_k	: arrival station of extra flight k
$t_d(k)$: departure time of extra flight k
$t_a(k)$: arrival time of extra flight k
$t_f(i)$: flight time of flight i
$t(i, j)$: time between arrival of flight i and departure of flight j
$t_h(i, j)$: deadhead time from a_i to d_j
$\overrightarrow{T}_f(i)$: total flight time <i>until</i> arrival of flight i , including flight time of i
$\overleftarrow{T}_f(i)$: total remaining flight time <i>after</i> flight i , excluding flight time of i
$\overrightarrow{T}_e(i)$: total elapsed time <i>until</i> arrival of flight i
$\overleftarrow{T}_e(i)$: total remaining elapsed time <i>from</i> the departure of the flight immediately following i
t_s^{\min}	: minimum sit time
t_s^{\max}	: maximum sit time
t_r^{\min}	: minimum rest time
t_r^{\max}	: maximum rest time
T_b^{\max}	: upper bound on the duration of a pairing (total away from base)
T_f^{\max}	: upper bound on the total flight time in a pairing

Table 1: The notation used in the paper.

use two fictitious flights k' and k'' , respectively.

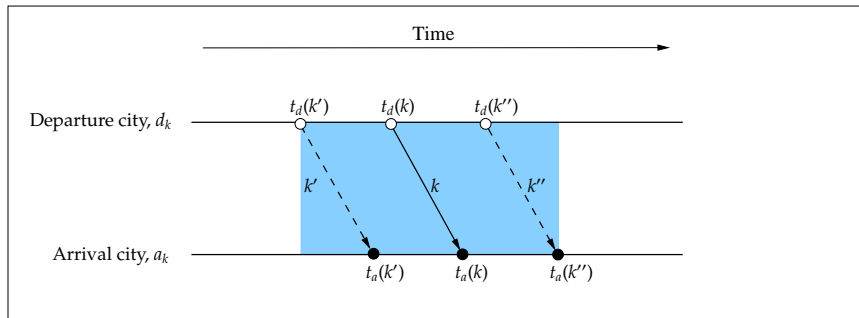


Figure 1: The time window for the extra flight k .

To cover the extra flights, the airline companies mostly use deadheading. However, using more than two deadheads to cover an extra flight is quite rare and undesirable. Therefore, we assume that no more than two deadheads are allowed in a solution. As discussed in Section 2, we use the term deadhead if a crew flies as passengers for repositioning purposes. Depending on the regulations, the time spent during repositioning may be partially counted as flight time in the duty period. Therefore, we allow the deadhead time $t_h(i, j)$ to be different than the corresponding actual flight time. Moreover, the minimum sit and rest times for a deadhead flight can be different than those for a regular flight. The value $t_{hs}^{\min}(i, j)$ is the minimum required time to cover flight j after flight i , whereas $t_{hs}^{\max}(i, j)$ is the upper bound on the connection time between flights i and j under two conditions; first, the crew deadheads from the arrival station of flight i to the departure station of flight j and second, all connections between departures and arrivals are sits:

$$t_{hs}^{\min}(i, j) = t_s^{\min} + t_h(i, j) + t_s^{\min},$$

$$t_{hs}^{\max}(i, j) = t_s^{\max} + t_h(i, j) + t_s^{\max}.$$

The value $t_{hr}^{\min}(i, j)$ is the minimum required time to cover flight j after flight i , whereas $t_{hr}^{\max}(i, j)$ is the upper bound on the connection time between flights i and j under two conditions; first the crew

deadheads from the arrival station of flight i to the departure station of flight j and second, one of the connections is a rest period:

$$\begin{aligned} t_{hr}^{\min}(i, j) &= t_s^{\min} + t_h(i, j) + t_r^{\min}, \\ t_{hr}^{\max}(i, j) &= t_s^{\max} + t_h(i, j) + t_r^{\max}. \end{aligned}$$

To deal with the proposed problem of managing extra flights, we examined all possible recovery options with at most two deadheads. This led to the classification of the possible solutions into the following two types:

Type A. Two pairings are selected such that swapping the crews of these pairings covers an extra flight.

Type B. One pairing is selected such that there is enough time between two consecutive legs of this pairing to cover the extra flight.

There are nine possible solutions; six of these solutions are of type A, (A.1, \dots , A.6), the remaining three are of type B, (B.1, B.2, B.3). A solution can be implemented, only if it satisfies a set of feasibility conditions. These conditions are related to the connection times, the total flight time and the total elapsed time. When a crew pairing model secures a large number of these feasible solutions, we call such a model a *robust* mathematical model. The main ideas behind the solutions as well as the corresponding feasibility conditions are quite similar. Moreover, we extensively use figures to illustrate different solutions. Therefore, we list in this section only two type A solutions (A.1 and A.4) and one type B solution (B.1). The remaining type A and type B solutions are given in Appendices A and B, respectively.

Two pairings can form a type A solution only if the pairings are also feasible after swapping. Let $p \in \mathcal{P}$ be the pairing that covers flights i_1 and i_2 , consecutively. Similarly, let $q \in \mathcal{P}$ be the pairing that covers flights j_1 and j_2 in the given order (see Figure 2). We say that p and q compose a type A solution for the extra flight k , if the crews of these two pairings are from the same base, and they also satisfy certain feasibility conditions that are mainly concerned with the relative arrival-departure times and the connection times between the swapped flights. After swapping, the first pairing is composed of flights up to and including flight i_1 as well as the flight j_2 and its subsequent flights, where the extra flight k is inserted between flights i_1 and j_2 . Similarly, the second pairing is composed of flights up to and including flight j_1 as well as the flight i_2 and its subsequent flights. When necessary, deadheads are inserted between i_1 and j_2 as well as between j_1 and i_2 . The set of pairing pairs (p, q) , which form (feasible) type A solutions for the extra flight k is denoted by $\mathcal{P}_A(k)$:

$$\mathcal{P}_A(k) = \{(p, q) : p, q \in \mathcal{P} \text{ and they form a type A solution for the extra flight } k\}.$$

One of the type A solutions with one deadhead is illustrated in Figure 2. The proposed solution A.1 can be implemented only if the pairing pair (p, q) satisfies the following feasibility conditions:

$$\begin{aligned} & \left((t_s^{\min} \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_s^{\max}) \text{ or } (t_r^{\min} \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_r^{\max}) \right), \\ & \left((t_s^{\min} \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_s^{\max}) \text{ or } (t_r^{\min} \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_r^{\max}) \right), \\ & \vec{T}_f(i_1) + t_f(k) + \vec{T}_f(j_2) \leq T_f^{\max}, \\ & \vec{T}_e(i_1) + t(i_1, j_2) + \vec{T}_e(j_2) \leq T_b^{\max}, \\ & \left(t_{hs}^{\min}(j_1, i_2) \leq t(j_1, i_2) \leq t_{hs}^{\max}(j_1, i_2) \text{ or } t_{hr}^{\min}(j_1, i_2) \leq t(j_1, i_2) \leq t_{hr}^{\max}(j_1, i_2) \right), \\ & \left(\vec{T}_f(j_1) + t_h(j_1, i_2) + \vec{T}_f(i_2) \leq T_f^{\max} \text{ and } \vec{T}_e(j_1) + t(j_1, i_2) + \vec{T}_e(i_2) \leq T_b^{\max} \right). \end{aligned} \tag{A.1}$$

The first two conditions in (A.1) ensure that the connection times between flight i_1 and the extra flight k and between flight k and the subsequent flight j_2 are within the allowable limits. The third and fourth conditions impose the limits on the total flight time and the total elapsed time for the original pairing p , respectively. The fifth condition requires that there is sufficient time to deadhead from a_k to d_k in order to cover flight i_2 after flight j_1 , and the last condition is the total flight and the elapsed time constraints for the original pairing q .

Similarly, one of the type A solutions with two deadheads is illustrated in Figure 3. This solution A.4

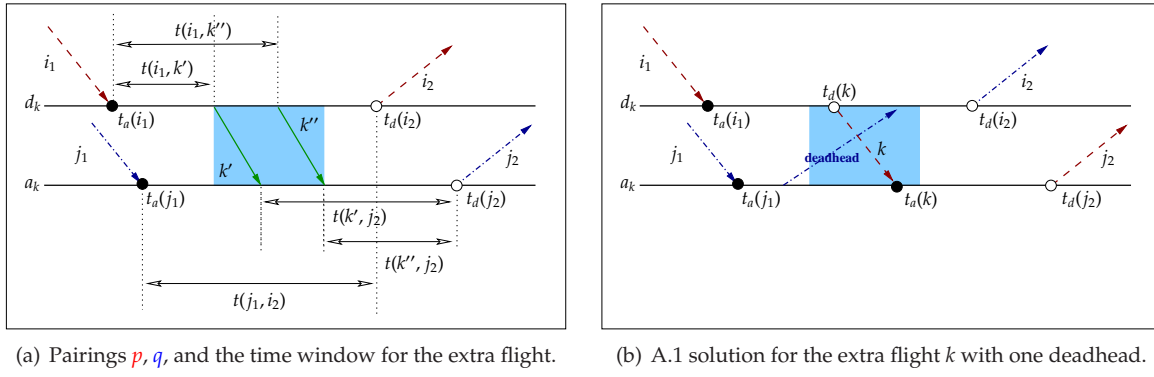


Figure 2: A sample A.1 solution on the flight network.

is valid only if the pairing pair \$(p, q)\$ satisfies the following feasibility conditions:

$$\begin{aligned}
 & \left((t_{hs}^{\min}(i_1, k') \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_{hs}^{\max}(i_1, k'')) \text{ or } (t_{hr}^{\min}(i_1, k') \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_{hr}^{\max}(i_1, k'')) \right), \\
 & \left((t_{hs}^{\min}(k'', j_2) \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_{hs}^{\max}(k', j_2)) \text{ or } (t_{hr}^{\min}(k'', j_2) \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_{hr}^{\max}(k', j_2)) \right), \\
 & \bar{T}_f(i_1) + t_h(i_1, k) + t_f(k) + t_h(k, j_2) + \bar{T}_f(j_2) \leq T_f^{\max}, \\
 & \bar{T}_e(i_1) + t(i_1, j_2) + \bar{T}_e(j_2) \leq T_b^{\max}, \\
 & (t_s^{\min} \leq t(j_1, i_2) \leq t_s^{\max} \text{ or } t_r^{\min} \leq t(j_1, i_2) \leq t_r^{\max}), \\
 & (\bar{T}_f(j_1) + \bar{T}_f(i_2) \leq T_f^{\max} \text{ and } \bar{T}_e(j_1) + t(j_1, i_2) + \bar{T}_e(i_2) \leq T_b^{\max}).
 \end{aligned} \tag{A.4}$$

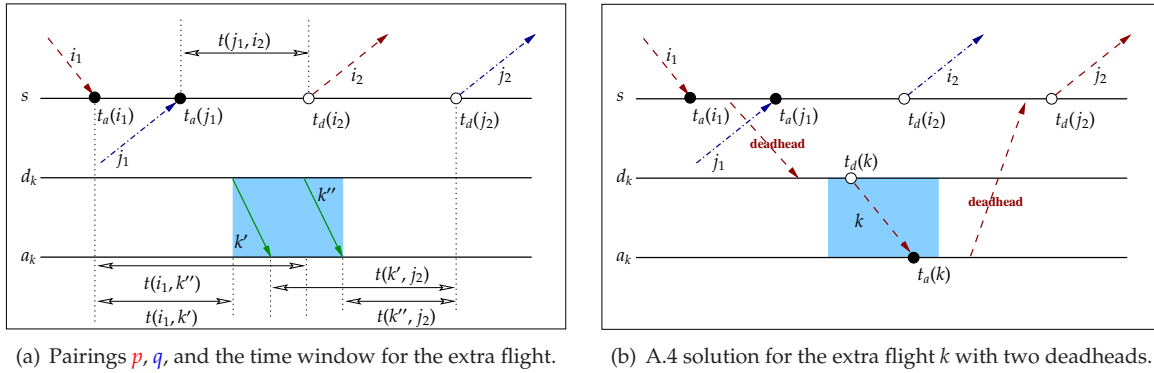


Figure 3: A sample A.4 solution on the flight network.

Type B solutions require only one pairing. A pairing forms a type B solution only if the selected pairing has available connection times to cover the extra flight, and the remaining work schedule for the involved crew does not violate the constraints. As before, we ensure the validity of a solution by checking different feasibility rules. The set of pairings that form type B solutions for the extra flight \$k\$ is given as

$$\mathcal{P}_B(k) = \{p \in \mathcal{P} : p \text{ forms a type B solution for the extra flight } k\}.$$

Figure 4 illustrates one of the type B solutions. This solution B.1 can be implemented only if the

The objective function maximizes the total number of type A and B solutions for all extra flights. The first set of constraints are the standard set partitioning constraints. The second and third set of constraints ensure that a type A solution $(p, q) \in \mathcal{P}_A(k)$ is counted only if both pairings p and q are selected, and $p \in \mathcal{P}_B(k)$ is not accounted for as a type B solution unless p is selected as a pairing, respectively. To keep the cost of the selected pairings at an acceptable level, the fourth constraint is included in the model. The constant ϵ in this constraint represents how much extra cost we can tolerate. In practice, covering some of the extra flights may be more important than others. This can be achieved by assigning greater weights to the solutions corresponding to these extra flights in the objective of (2).

The model proposed above makes no attempt to provide a solution for each extra flight, and while the optimal solution for (2) may present several ways of covering an extra flight k_1 , it may fail to provide any solution to another extra flight k_2 even if such an alternate optimal solution exists. In this case, the following constraint may be included in (2), where lb_s is a lower bound on the number of solutions required for each extra flight. Constraints of this form may also be incorporated if it is absolutely crucial to cover a subset of the extra flights.

$$\sum_{(p,q) \in \mathcal{P}_A(k)} x_{(p,q)}^k + \sum_{p \in \mathcal{P}_B(k)} w_p^k \geq lb_s, \quad k \in \mathcal{K}.$$

Alternatively, one may solve the model (3) below which would provide the largest value of lb_s as its optimal objective function value z . This type of objective may be used to distribute the solutions among the extra flights as equitably as possible.

$$\begin{aligned} & \max \quad z \\ & \text{s.t.} \quad \sum_{p \in \mathcal{P}} a_{ip} y_p = 1 && i \in \mathcal{F}, \\ & \quad 2x_{(p,q)}^k \leq y_p + y_q, && (p, q) \in \mathcal{P}_A(k), k \in \mathcal{K}, \\ & \quad w_p^k \leq y_p, && p \in \mathcal{P}_B(k), k \in \mathcal{K}, \\ & \quad \sum_{(p,q) \in \mathcal{P}_A(k)} x_{(p,q)}^k + \sum_{p \in \mathcal{P}_B(k)} w_p^k \geq z, && k \in \mathcal{K}, \\ & \quad \sum_{p \in \mathcal{P}} c_p y_p \leq (1 + \epsilon) C_{opt}, \\ & \quad x_{(p,q)}^k \in \{0, 1\}, && (p, q) \in \mathcal{P}_A(k), k \in \mathcal{K}, \\ & \quad w_p^k \in \{0, 1\}, && p \in \mathcal{P}_B(k), k \in \mathcal{K}, \\ & \quad y_p \in \{0, 1\}, && p \in \mathcal{P}. \end{aligned} \tag{3}$$

As we mentioned in Section 1, it is quite common in practice to use a column generation method for solving the crew pairing problem. It is important to note here that the second set of constraints in (2) and (3) depend on the selected pairing pairs. Therefore, when a column (pairing) is generated and if another pairing yielding a type A solution together with this pairing is already present in the column pool, then one constraint should be added to the model. Similarly, when a column is discarded, the constraints formed by the pairs involving this column should be deleted from the model. Thus, both the number of columns and rows in the restricted master problem may change with every iteration. This type of approach is both theoretically and practically more challenging and is not commonly considered in the literature [7].

Our model does not incorporate any constraints in order to prevent a pairing from participating in several type A or B solutions. When two solutions for the same extra flight share pairings, no conflict arises, and the decision maker may implement any of these solutions, depending on other criteria. However, if a pairing is included in different solutions for more than one extra flight, then we may face a problem if both extra flights need to be inserted into the schedule on the same day. This issue is referred to as *double counting* [15]. Figure 5 illustrates an instance of double counting, where there are three pairings A, B and C, covering flights a_1 - a_2 , b_1 - b_2 and c_1 - c_2 , respectively, and the extra flights are denoted by k_1 and k_2 . In this example, pairing C forms a type A.1 solution for extra flight k_1 with pairing A, and also a type A.3 solution for extra flight k_2 with pairing B. Both of these solutions would be counted separately by the model (2). However, if both k_1 and k_2 have to be flown on the same day, then we can use pairing C to cover only one of these extra flights. In the following, we present two additional models that tackle this double counting issue, where the parameter ub_s controls the number of solutions a pairing p may appear in for a given extra flight, and a new set of binary variables u_p^k keep track of whether pairing p participates in any type A or B solution for extra flight k .

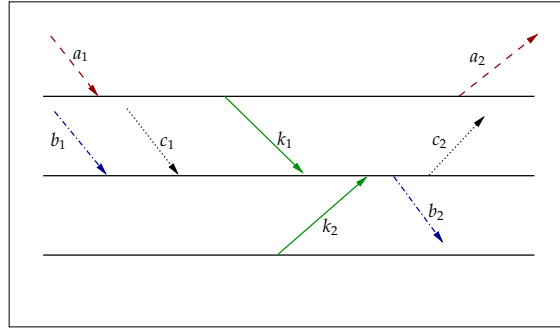


Figure 5: An illustration of the double counting problem.

$$\begin{aligned}
\max \quad & \sum_{k \in \mathcal{K}} \left(\sum_{(p,q) \in \mathcal{P}_A(k)} x_{(p,q)}^k + \sum_{p \in \mathcal{P}_B(k)} w_p^k \right) \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}} a_{ip} y_p = 1, & i \in \mathcal{F}, \\
& 2x_{(p,q)}^k \leq y_p + y_q, & (p,q) \in \mathcal{P}_A(k), k \in \mathcal{K}, \\
& w_p^k \leq y_p, & p \in \mathcal{P}_B(k), k \in \mathcal{K}, \\
& \sum_{(p,q) \in \mathcal{P}_A(k)} x_{(p,q)}^k + \sum_{p \in \mathcal{P}_B(k)} w_p^k \leq ub_s u_p^k, & p \in \mathcal{P}, k \in \mathcal{K}, \\
& \sum_{k \in \mathcal{K}} u_p^k \leq y_p, & p \in \mathcal{P}, \\
& \sum_{p \in \mathcal{P}} c_p y_p \leq (1 + \epsilon) C_{opt}, \\
& x_{(p,q)}^k \in \{0, 1\}, & (p,q) \in \mathcal{P}_A(k), k \in \mathcal{K}, \\
& w_p^k \in \{0, 1\}, & p \in \mathcal{P}_B(k), k \in \mathcal{K}, \\
& u_p^k \in \{0, 1\}, & p \in \mathcal{P}, k \in \mathcal{K}, \\
& y_p \in \{0, 1\}, & p \in \mathcal{P}.
\end{aligned} \tag{4}$$

The model (4) maximizes the total number of type A and B solutions for all extra flights like (2), but restricts how a given pairing may be used for forming solutions for a given extra flight and across multiple extra flights. The fourth set of constraints enforces that a given pairing appears in at most ub_s solutions for a given extra flight, and the fifth set of constraints mandates that a pairing cannot be shared in solutions across extra flights. The corresponding model for (3) that bans double counting is presented below.

$$\begin{aligned}
\max \quad & z \\
\text{s.t.} \quad & \sum_{p \in \mathcal{P}} a_{ip} y_p = 1, & i \in \mathcal{F}, \\
& 2x_{(p,q)}^k \leq y_p + y_q, & (p,q) \in \mathcal{P}_A(k), k \in \mathcal{K}, \\
& w_p^k \leq y_p, & p \in \mathcal{P}_B(k), k \in \mathcal{K}, \\
& \sum_{(p,q) \in \mathcal{P}_A(k)} x_{(p,q)}^k + \sum_{p \in \mathcal{P}_B(k)} w_p^k \geq z, & k \in \mathcal{K}, \\
& \sum_{(p,q) \in \mathcal{P}_A(k)} x_{(p,q)}^k + \sum_{p \in \mathcal{P}_B(k)} w_p^k \leq ub_s u_p^k, & p \in \mathcal{P}, k \in \mathcal{K}, \\
& \sum_{k \in \mathcal{K}} u_p^k \leq y_p, & p \in \mathcal{P}, \\
& \sum_{p \in \mathcal{P}} c_p y_p \leq (1 + \epsilon) C_{opt}, \\
& x_{(p,q)}^k \in \{0, 1\}, & (p,q) \in \mathcal{P}_A(k), k \in \mathcal{K}, \\
& w_p^k \in \{0, 1\}, & p \in \mathcal{P}_B(k), k \in \mathcal{K}, \\
& u_p^k \in \{0, 1\}, & p \in \mathcal{P}, k \in \mathcal{K}, \\
& y_p \in \{0, 1\}, & p \in \mathcal{P}.
\end{aligned} \tag{5}$$

Double counting is promoted in the original models (2) and (3) as it may increase the objective function without increasing the pairing cost. This may be desirable as it restricts the number of crews whose schedules will potentially be subject to change, facilitating the recovery process in the face of disruptions. However, allowing double counting is a risk prone approach, in case multiple extra flights are added simultaneously to the schedule. On the other hand, if we take a completely risk averse stance and enforce that each pairing can contribute to the solutions of at most one extra flight, then we must accept a higher pairing cost upfront and also run the danger of failing to identify potential solutions if only a subset of the potential extra flights have to be handled. These concepts will be illustrated by the computational results detailed in the next section, and one of our main conclusions in this research is that relying on just a specific type of model may lead to costly errors in the crew pairings generated. Sound decision making for the robust crew pairing problem for managing extra flights presented in this paper requires that all of the models above are employed in an integrated fashion.

5. Computational Results. To illustrate the proposed solution approach, we conduct numerical experiments on three problems adapted from the actual data of a local airline company. We solve these problems by both the conventional model (1) and the proposed robust models (2-5). We then compare the pairing solutions obtained on the basis of the swapping (type A solutions) and insertion (type B solutions) opportunities they include. The results indicate on one hand that the proposed robust models provide natural options to recovery without disrupting existing flights at a relatively small incremental cost, which is visible at the planning stage. On the other hand, the conventional model yields crew pairing solutions with little slack, and restoring crew schedules after disruptions in this case relies on recovery procedures at the time of operation. Thus, the resulting additional cost is unpredictable during the planning stage (see Section 1 for an in-depth discussion).

Our personal communications with two airline companies in Turkey indicate that deadheading is a common practice in the Turkish airline industry. Following this observation, we implicitly assume in our analysis that the cost of deadheading is small relative to the cost of delaying or canceling flights. We also note that adhering to the scheduled flight times bears a substantial importance for preserving the reputation of an airline and preventing loss of customer goodwill.

Our focus in this paper is on demonstrating the value of a robust model in managing extra flights rather than solving large scale problems. Therefore, we make no attempt to solve our models by large scale mathematical programming techniques like branch-and-price. The problem instances we consider are small enough that allows us to construct all feasible pairings. We only consider one-day pairings. The cost and feasibility parameters used in this study are given in Table 2. We allow for shorter sit connections when inserting an extra flight into the schedule. In all robust models, we allow the total cost to exceed that of the conventional model by at most 3%. When we deal with the robust models that avoid double counting, we assume that a pairing may appear in at most 2 solutions for a given extra flight. We generate the flight network, all feasible pairings for each crew base in the network, and the sets $\mathcal{P}_A(k)$ and $\mathcal{P}_B(k)$ for all $k \in \mathcal{K}$ by a C++ code. Then, both the conventional model (1) and the robust models (2-5) are implemented and solved using ILOG OPL Studio 5.5/CPLEX 11.0 [9] running on an HP workstation with 2 Intel Xeon Quadcore 1.6 GHz processors and 4 GB memory.

f_p, f_d	=	0.875
mg	=	200 min.
t_s^{\min}	=	60 min. for regular flights
t_s^{\min}	=	30 min. for extra flights
t_s^{\max}	=	240 min.
T_f^{\max}	=	480 min.
T_b^{\max}	=	1440 min.
ϵ	=	0.03
ub_s	=	2

Table 2: The parameters used in generating pairings and finding type A and B solutions.

In the first problem, there are 38 flights between the cities Istanbul (IST), Ankara (ANK), Izmir (IZM), and Antalya (ANT). Four and two aircraft are assigned to the crew bases Istanbul and Ankara, respectively. The complete flight data is given in Table 18 of Appendix C. The company anticipates that it may add an extra flight k_1 from Ankara to Istanbul some time between 14:30 (earliest arrival time;

see Figure 1) and 16:30 (latest arrival time; see Figure 1) on certain weekdays due to a recent increase in business travel. In addition, to capture some of the demand by summer travelers, an extra flight k_2 from Istanbul to Izmir during the time window 16:30-18:00 is also considered. The length of both of these extra flights is one hour. The company is reluctant to offer these flights on a regular basis due to the volatility of demand caused by high competition. Therefore, pairings are constructed by excluding these flights. However, if sufficient demand is realized, then the extra flights are incorporated into the flight schedule by delaying and/or canceling some regular flights if necessary.

In this problem, 309 and 209 pairings originate from IST and ANK, respectively. The optimal cost of the conventional crew pairing model (1) is $C_{opt} = 4722$ minutes. The pairings selected in the optimal solution are presented in Table 3. We also solve the problem by the robust models with double counting (2) and (3), and by the robust models with no double counting (4) and (5). For this particular problem instance, all robust models find the same optimal pairings as shown in Table 4.

Pairing ID	Flight IDs						
7	1	32	15	19	21	22	
64	2	6	10	16	20	34	26 36
160	4	7	8	11	25	24	
186	9	12	17	30	5	38	
334	29	3	35	23	37	27	
412	31	13	33	14	18	28	

Table 3: Optimal pairings from the conventional model (1) for the 38-flight instance.

Pairing ID	Flight IDs						
5	1	32	15	19	14	18	
67	2	6	10	16	21	5	38
160	4	7	8	11	25	24	
192	9	12	17	34	26	36	
323	29	3	35	20	30	22	27
419	31	13	33	23	37	28	

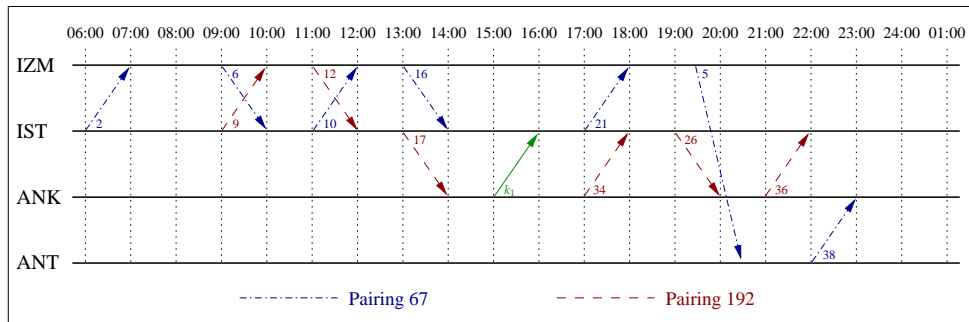
Table 4: Optimal pairings from the robust models (2 - 5) for the 38-flight instance.

In Table 5, the optimal solutions of the conventional model (1) and the proposed robust models (2) to (5) are compared on the basis of the type A and B solutions they include. If the company uses the conventional model to construct its pairings, adding the extra flights k_1 and k_2 during their respective required time windows would entail substantial modifications to the existing pairings, possibly delaying and canceling some of the other flights in the schedule. On the other hand, the optimal solution of our robust models offers a type A.1 solution for covering k_1 comprised by pairings 192 and 67 and a type A.2 solution for covering k_2 made up by pairings 5 and 160. The total cost of this set of robust optimal pairings is 4837 minutes, which is less than 2.5% above the total cost of the conventional model. These recovery options for incorporating the extra flights k_1 and k_2 are illustrated in Figures 6 and 7, respectively. Note that both extra flights are covered by only one solution, and hence the robust models (2) and (3) yield the same optimal pairings. Similarly, the pairings to be swapped for the extra flights k_1 and k_2 are distinct; hence, the double counting problem discussed in Section 4 is not present here. This also explains the identical optimal solutions obtained from (4) and (5) that prevent double counting across extra flights.

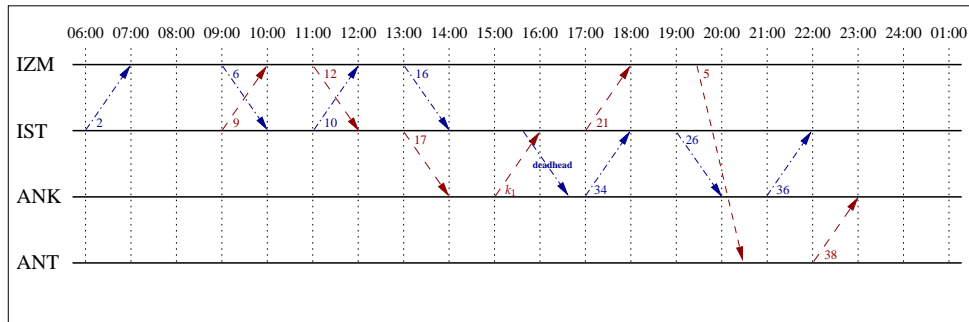
	Cost	k_1 (ANK-IST)	k_2 (IST-IZM)	CPU Time (sec.)
CM (1)	4722	-	-	< 1
RM (2-5)	4837	A.1; (192,67)	A.2; (5,160)	< 1

CM: Conventional Model
RM: Robust Model

Table 5: Recovery options provided by the conventional and proposed models for the 38-flight instance.

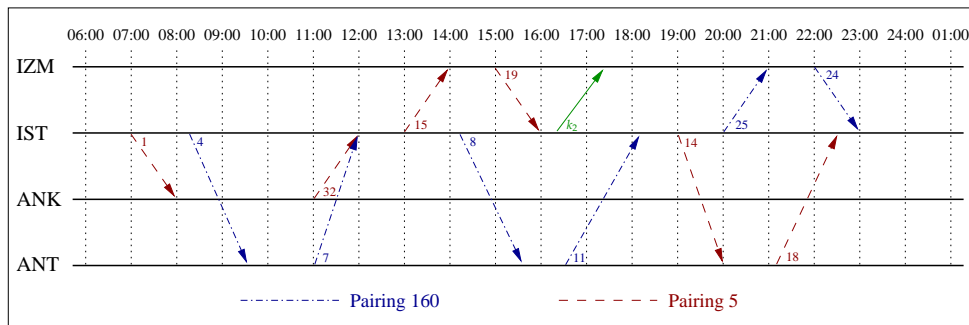


(a) The original flights covered by pairings 192 and 67.

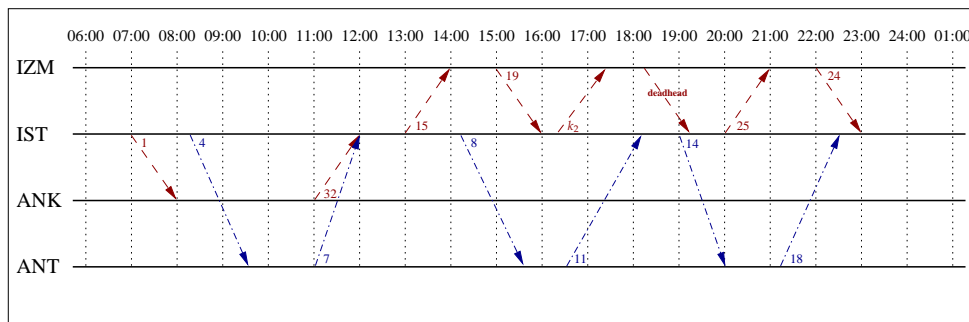


(b) The solution after swapping (Type A.1).

Figure 6: Adding the extra flight k_1 from ANK to IST at 15:00 in the 38-flight instance.



(a) The original flights covered by pairings 5 and 160.



(b) The solution after swapping (Type A.2).

Figure 7: Adding the extra flight k_2 from IST to IZM at 16:30 in the 38-flight instance.

In the second sample problem, we have 58 flights between the cities Istanbul (IST), Ankara (ANK), Izmir (IZM), Antalya (ANT), Adana (ADA), Dalaman (DAL), and Bodrum (BOD). Istanbul is the only crew base with 10 aircraft. See Table 19 of Appendix C for the complete flight data. In this problem, the flights from Istanbul to Ankara at 15:00 and Istanbul to Antalya at 11:25 are in high demand. In order to cope with this demand, the company schedules charter flights by hiring aircraft from other airline companies whenever possible. The time windows considered for the additional flights IST-ANK (extra flight k_1) and IST-ANT (extra flight k_2) are 15:00-16:30 and 11:40-13:10, respectively. The flight duration from Istanbul to Ankara is one hour, while it takes 75 minutes to fly from Istanbul to Antalya.

For this sample problem, the conventional crew pairing model (1) yields a total cost of 7089 minutes based on a total number of 2213 pairings. Table 6 presents the optimal pairings for the conventional model, whereas tables 7 to 10 present the optimal pairings for the proposed robust models. As in the previous example, our robust models achieve slightly higher pairing costs as shown in Table 11.

Pairing ID	Flight IDs						
53	1	4	33	35	16	26	
119	2	5	37	39			
215	7	17	44	53	11	21	48 57
617	8	18	46	55	15	25	49 58
686	9	19	3	6	38	40	
909	27	30	45	54	12	22	
962	28	31	14	24			
1359	41	52	47	56			
1424	42	50	29	32	34	36	
1796	43	51	10	20	13	23	

Table 6: Optimal pairings from the conventional model (1) for the 58-flight instance.

Pairing ID	Flight IDs							
91	1	4	46	55	48	57		
148	7	17	2	5	37	39		
507	8	18	3	6	38	40		
715	9	19	29	32	34	36		
904	27	30	33	35	16	26		
970	28	31	47	56				
1039	41	50	11	21	15	25		
1680	42	52	45	54	12	22	49 58	
1796	43	51	10	20	13	23		
2108	44	53	14	24				

Table 7: Optimal pairings from the robust model (2) for the 58-flight instance.

We next compare the conventional and proposed robust models for the second sample problem. From Table 11, we observe that only k_2 can be incorporated into the flight schedule by swapping crew schedules if the pairings are formed by solving the conventional model (1). As the third row, labeled as RM (2), shows the first robust model (2) allows both extra flights to be covered if an increase of 1.7% in total cost can be tolerated. Possible ways of restoring crew pairings suggested by (2) after inserting flights k_1 and k_2 are displayed in Figure 8. Note that if both extra flights k_1 and k_2 are to be accommodated, then only the type A.1 solution (91,970) may be selected for k_2 since pairings 1039 and 2108 are required for covering k_1 . This is an expected result, since the first robust model does not prevent double counting but tries to increase the number of possible recovery options.

As shown in row RM (3) of Table 11, the total cost of the second robust model is slightly less than that of the first model. However, the solution of the second robust model seems to suffer from double counting. In our experiments, we observed that the pairing problems usually have multiple optimal solutions. Since the second model (3) does not prevent double counting, the solver returns one of those optimal solutions. Having obtained the solutions for the first robust model, this observation becomes

Pairing ID	Flight IDs					
51	1	4	33	35	15	25
139	7	17	2	5	13	23
508	8	18	3	6	48	57
692	9	19	11	21	16	26
854	27	30	10	20	12	22
971	28	31	47	56	49	58
1179	41	51	29	32	38	40
1478	42	50	46	55	34	36
2002	43	52	45	54	37	39
2108	44	53	14	24		

Table 8: Optimal pairings from the robust model (3) for the 58-flight instance.

Pairing ID	Flight IDs							
34	1	4	11	21	16	26		
148	7	17	2	5	37	39		
507	8	18	3	6	38	40		
724	9	19	33	35	48	57		
898	27	30	29	32	34	36		
946	28	31	12	22				
1106	41	50	46	55	15	25		
1622	42	52	10	20	13	23	49	58
1870	43	51	45	54	14	24		
2126	44	53	47	56				

Table 9: Optimal pairings from the robust model (4) for the 58-flight instance.

clear; an alternate solution that does not suffer from double counting could have been obtained by selecting the pairing pairs (2108, 692) and (1039,970).

The last two rows in Table 11 give the solutions obtained by the robust models that prevent double counting. The objective of the third model (4) is to maximize the total number of type A and type B solutions. Therefore, as the solutions in the fifth row show, if the sole objective is to increase the number of solutions, then the highest number is attained when no recovery option is provided for extra flight k_1 . As the last row that are obtained with the fourth model (5) shows, there are in fact solutions for both extra flights when the objective is to distribute the solutions equitably.

The results obtained for all proposed robust models clearly show that one needs to analyze all the models to properly consider alternate recovery options. On one hand, the decision maker may prefer to ignore double counting and aim at obtaining as many recovery options as possible. This choice becomes important especially when the chance of adding all the extra flights to the flight schedule at the same time is rather slim. On the other hand, crew pairings generated by (4) and (5) should be given more attention, when covering all extra flights simultaneously is crucial for the airline. This may be the case when the total revenue generated from the extra flights constitutes a significant portion of the profit or the reputation of the airline (especially for the one that is new in the business) bears the utmost importance for the management.

In the final column of Table 11, we observe that the computational effort for solving both the classical model and the robust models is still within a few minutes. The robust models are clearly harder to solve than the classical model, and maximizing the total number of type A and B solutions by solving (2) or (4) is computationally less expensive than distributing the solutions equitably among the extra flights by solving (3) or (5). In addition, it appears that the robust models (4) and (5) with no double counting across extra flights lead to shorter solution times than their counterpart models (2) and (3) that allow double counting. We conjecture that this is related to the decreased number of alternate optimal solutions and symmetry in the B&B tree due to the additional constraints in (4) and (5).

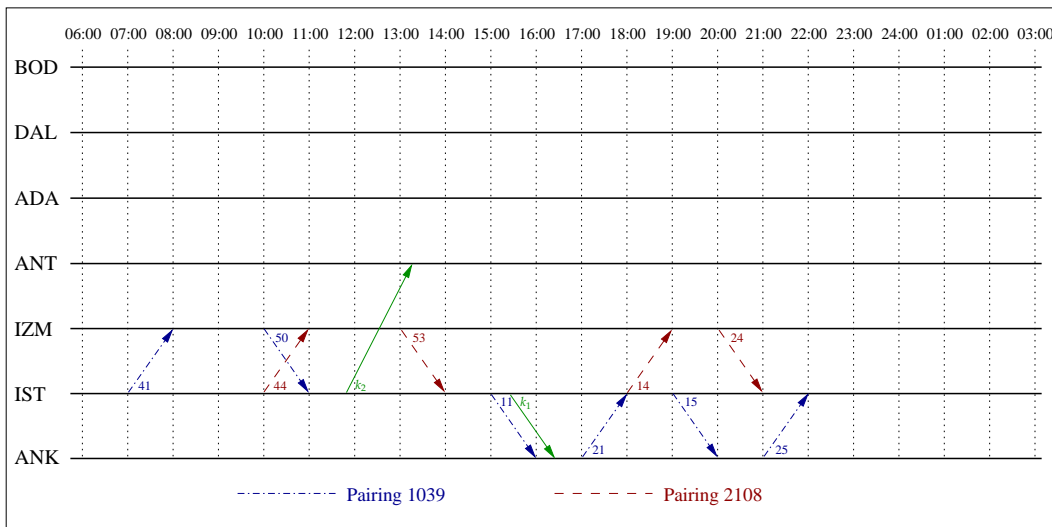
Pairing ID	Flight IDs							
45	1	4	29	32	15	25		
95	2	5	12	22				
239	7	17	44	53	14	24	49	58
603	8	18	45	54	37	39		
724	9	19	33	35	48	57		
851	27	30	3	6	34	36		
970	28	31	47	56				
1111	41	50	46	55	38	40		
1621	42	52	10	20	13	23		
1829	43	51	11	21	16	26		

Table 10: Optimal pairings from the robust model (5) for the 58-flight instance.

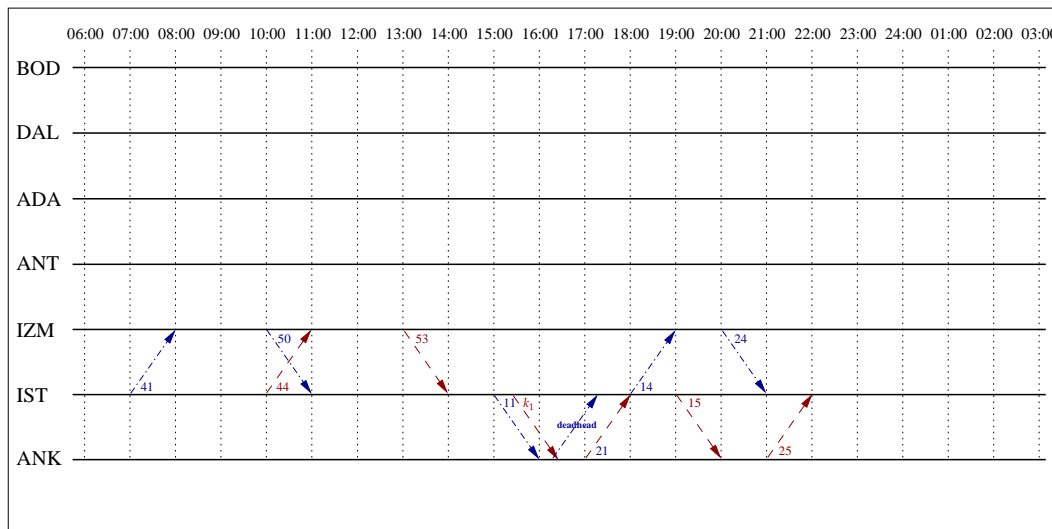
	Cost	k_1 (IST-ANK)	k_2 (IST-ANT)	CPU Time (sec.)
CM (1)	7089	-	A.2; (53,1359),(1424,1359)	< 1
RM (2)	7209	A.1; (2108,1039)	A.1; (91,970),(1039,970) A.2; (91,2108),(1039,2108)	201
RM (3)	7191	A.1; (2108,692)	A.2; (51,2108)	270
RM (4)	7183	-	A.1; (34,946),(1106,946) A.2; (34,2126),(1106,2126)	18
RM (5)	7145	A.1; (239,1829)	A.1; (1111,970)	105

CM: Conventional Model
RM: Robust Model

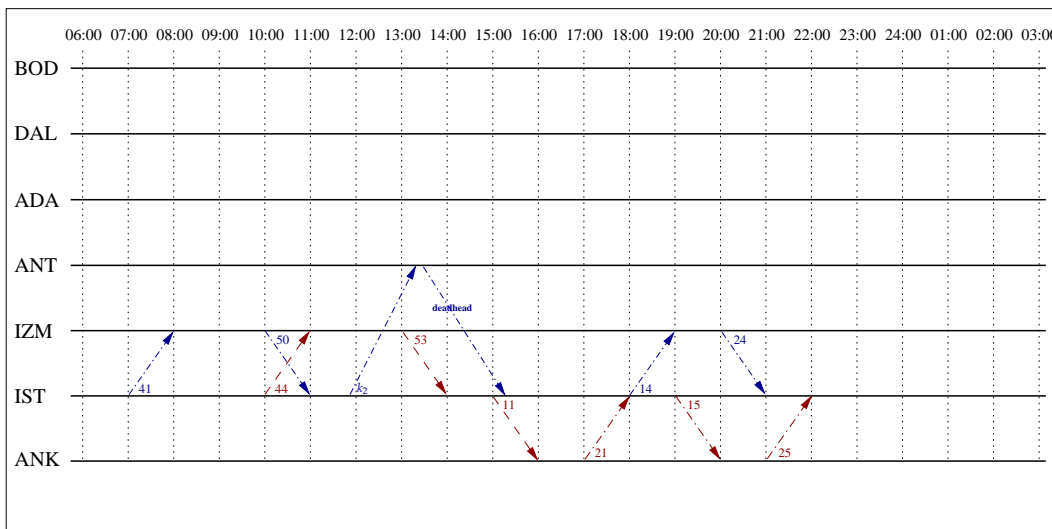
Table 11: Recovery options provided by the conventional and proposed models for the 58-flight instance.



(a) The original flights covered by pairings 1039 and 2108.



(b) The solution after swapping for the extra flight k_1 from IST to ANK at 15:30. (Type A.1)



(c) The solution after swapping for the extra flight k_2 from IST to ANT at 11:55. (Type A.2)

Figure 8: Adding the extra flights in the 58-flight instance.

In our last sample problem, we have 96 flights between the cities Istanbul (IST), Ankara (ANK), Izmir (IZM), Antalya (ANT), and Adana (ADA). Twelve and six aircraft are assigned to the crew bases Istanbul and Ankara, respectively. See Table 20 of Appendix C for the complete flight data. In this sample problem, there are three extra flights: IST-ANK (k_1), ANK-IST (k_2), and IST-IZM (k_3). The flight duration for all three extra flights is 65 minutes. The time windows for the extra flights are 07:35-08:55, 09:10-10:30 and 07:40-09:45, respectively.

Table 12 presents the optimal pairings for the conventional model, whereas tables 13 to 16 present the optimal pairings for the proposed robust models. As shown in Table 17, the total cost for the conventional model (1) is 12730 minutes based on 2977 pairings originating from IST and on 4389 pairings originating from ANK. Notice that no type B solution is identified for the first two sample problems. In those problems, we observe that the pairings include flight patterns such as base-destination-base with short connection times. Therefore, forming a type B solution by squeezing an extra flight in between becomes impossible. However, in the last sample problem not only we obtain type B solutions but also one A.6 solution. These solutions for the extra flights are illustrated in Figure 9.

Pairing ID	Flight IDs					
109	1	7	67	68	47	48
451	2	8	12	21		
769	5	15	18	26	89	90
1223	25	30	53	54		
1367	33	34	91	92	71	72
1788	35	36	45	46		
2081	37	38	14	57	58	31
2340	39	40	69	70	83	84
2624	51	52	93	94		
2664	65	66	43	44	49	50
2811	73	74	77	78		
2818	79	80	16	23	75	76
3068	3	9	17	81	82	32
3468	4	41	42	22	95	96
3566	6	11	20	27		
3958	10	85	86	28	59	60
4253	55	56	19	24	63	64
4301	61	62	13	87	88	29

Table 12: Optimal pairings from the conventional model (1) for the 96-flight instance.

Pairing ID	Flight IDs					
109	1	7	67	68	47	48
461	2	8	12	57	58	31
706	5	13	18	26	89	90
929	9	17	71	72		
1047	11	20	25	30	93	94
1364	33	34	91	92	49	50
1805	35	36	69	70	83	84
2068	37	38	14	21		
2622	51	52	53	54		
2679	65	66	45	46		
2811	73	74	77	78		
2818	79	80	16	23	75	76
3243	3	39	40	87	88	29
3468	4	41	42	22	95	96
3660	6	43	44	27		
3958	10	85	86	28	59	60
4253	55	56	19	24	63	64
4319	61	62	15	81	82	32

Table 13: Optimal pairings from the robust model (2) for the 96-flight instance.

Pairing ID	Flight IDs					
115	1	7	67	68	71	72
461	2	8	12	57	58	31
769	5	15	18	26	89	90
1045	11	20	25	30	53	54
1364	33	34	91	92	49	50
1752	35	36	16	23		
2068	37	38	14	21		
2340	39	40	69	70	83	84
2623	51	52	77	78		
2679	65	66	45	46		
2812	73	74	93	94		
2842	79	80	47	48	75	76
3058	3	9	17	22	95	96
3476	4	41	42	81	82	32
3660	6	43	44	27		
3958	10	85	86	28	59	60
4253	55	56	19	24	63	64
4301	61	62	13	87	88	29

Table 14: Optimal pairings from the robust model (3) for the 96-flight instance.

Pairing ID	Flight IDs					
109	1	7	67	68	47	48
451	2	8	12	21		
706	5	13	18	26	89	90
917	9	17	49	50		
1046	11	20	25	30	77	78
1367	33	34	91	92	71	72
1805	35	36	69	70	83	84
2118	37	38	45	46		
2622	51	52	53	54		
2647	65	66	14	57	58	31
2812	73	74	93	94		
2818	79	80	16	23	75	76
3243	3	39	40	87	88	29
3468	4	41	42	22	95	96
3660	6	43	44	27		
3958	10	85	86	28	59	60
4253	55	56	19	24	63	64
4319	61	62	15	81	82	32

Table 15: Optimal pairings from the robust model (4) for the 96-flight instance.

Pairing ID	Flight IDs					
137	1	7	91	92	71	72
461	2	8	12	57	58	31
706	5	13	18	26	89	90
1224	25	30	77	78		
1339	33	34	67	68	47	48
1772	35	36	43	44	49	50
2145	37	38	85	86		
2313	39	40	45	46	83	84
2624	51	52	93	94		
2649	65	66	16	23	75	76
2810	73	74	53	54		
2851	79	80	69	70		
3058	3	9	17	22	95	96
3479	4	41	42	87	88	29
3567	6	11	20	27	59	60
3909	10	14	21	28		
4253	55	56	19	24	63	64
4319	61	62	15	81	82	32

Table 16: Optimal pairings from the robust model (5) for the 96-flight instance.

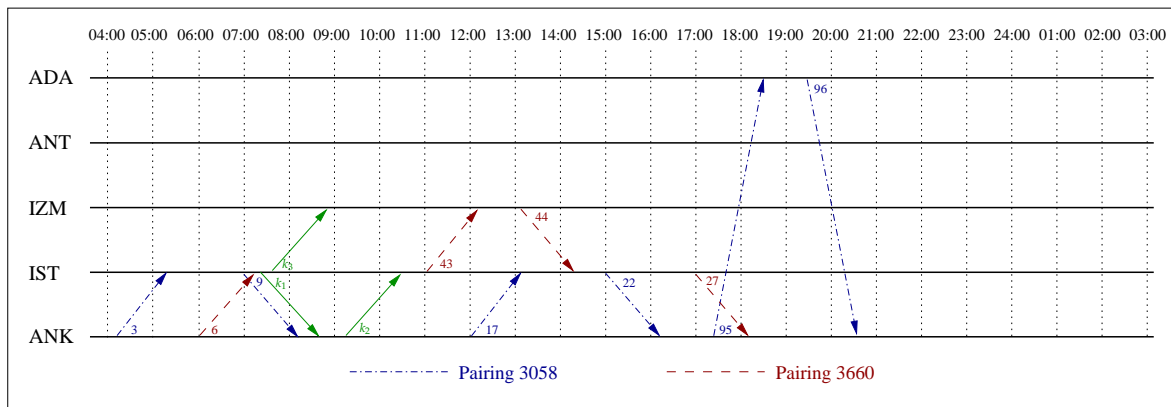
The recovery options for the conventional and proposed robust models are given in Table 17. The second row shows that only the second extra flight k_2 can be recovered by the conventional model. The results for the first robust model (2) in the third row yield nine type A and type B solutions with an increase of 3% in the total cost. When we distribute the solutions equitably with the proposed model (3), we are able to obtain two solutions for extra flights k_1 and k_2 , and one for extra flight k_3 with a slightly less increase in the total cost (2.9%). However, among these solutions we are not able to obtain recovery options without double counting. This is also clear from rows four and five, where the solutions are given for the robust models (4) and (5), respectively. Both of these models prevent double counting, and hence, the results show that it is not possible to cover all the extra flights without double counting. In fact the pairing 3660 plays a crucial role, since it is involved in all the recovery options for both extra flights k_1 and k_3 . In case we solve only (4) and (5), we would not be aware of the alternate solutions suggested by (2) and (3). Therefore, insisting on models with no double counting reflects a *risk averse* behavior in some situations. Overall, we remark that none of the models is superior to the others and it is beneficial for the decision makers to solve all the proposed models.

The solution times in Table 17 confirm the pattern that started to emerge in Table 11. The times for solving the robust models increase steeply with the number of flights in the problem and indicate that there is a clear need to develop elaborate simultaneous row and column generation approaches for obtaining the optimal solutions to our proposed robust models.

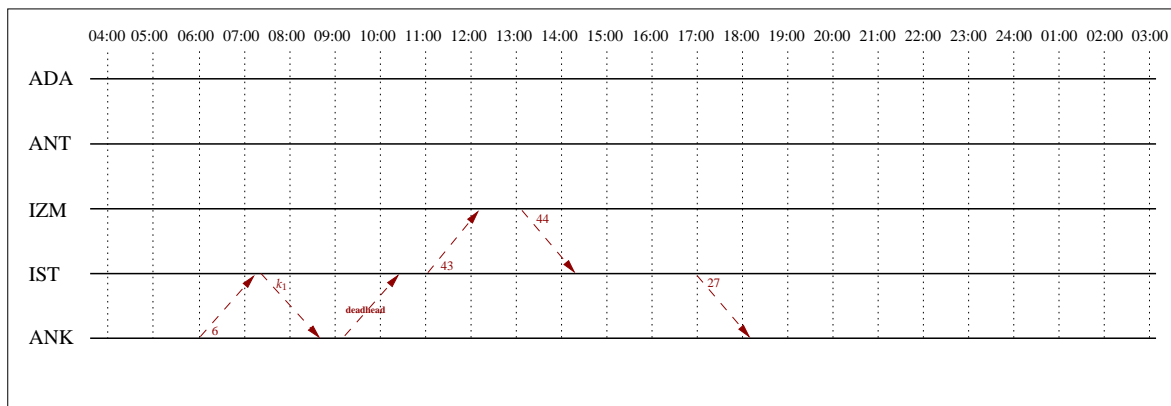
	Cost	k_1 (IST-ANK)	k_2 (ANK-IST)	k_3 (IST-IZM)	CPU Time (sec.)
CM (1)	12730	-	A.2; (3068,3566)	-	< 1
RM (2)	13111	A.1; (3660,4319) A.2; (3468,3660) B.1; 3660	A.1; (929,1805),(929,2068), (929,2679),(4319,3660),(929,461) B.2; 3660		5,424
RM (3)	13103	A.1; (3660,3058) B.1; 3660	A.1; (3660,3058) B.2; 3660	A.6; (3660,3058)	71,837
RM (4)	13111	A.2; (3468,3660) B.1; 3660	A.1; (917,1805),(917,2118)	-	562
RM (5)	12805	-	-	-	16,275

CM: Conventional Model
 RM: Robust Model

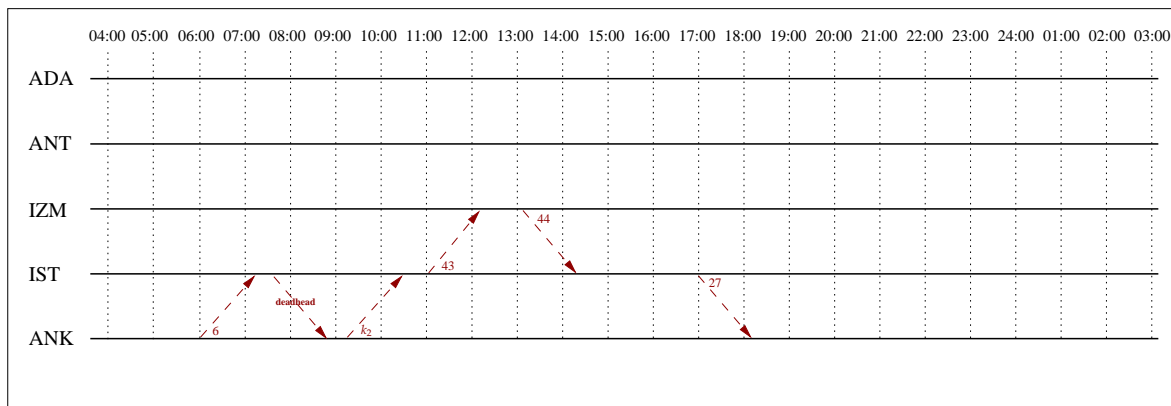
Table 17: Recovery options provided by the conventional and proposed models for the 96-flight instance.



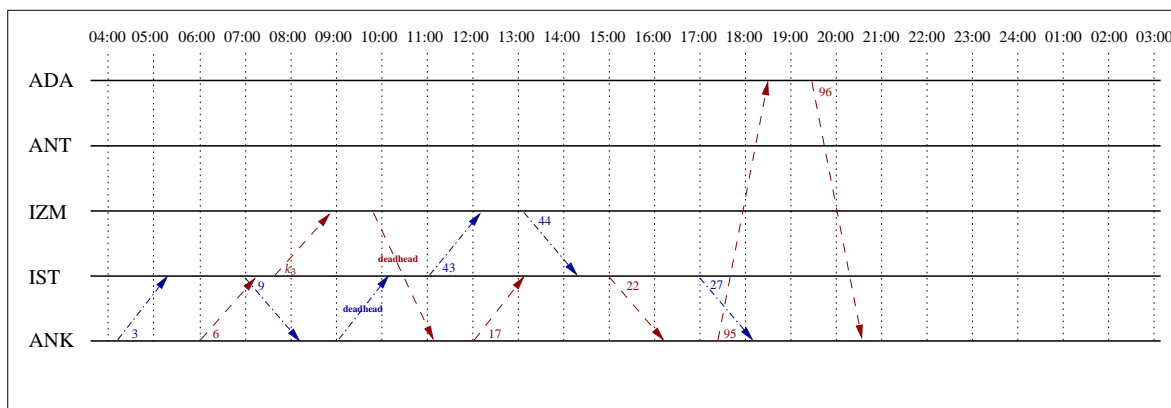
(a) The original flights covered by pairings 3058 and 3660.



(b) The solution for the extra flight k_1 from IST to ANK at 07:35 (Type B.1)



(c) The solution for the extra flight k_2 from ANK to IST at 09:10 (Type B.2)



(d) The solution after swapping for the extra flight k_3 from IST to IZM at 07:40 (Type A.6)

Figure 9: Adding the extra flights in the 96-flight instance.

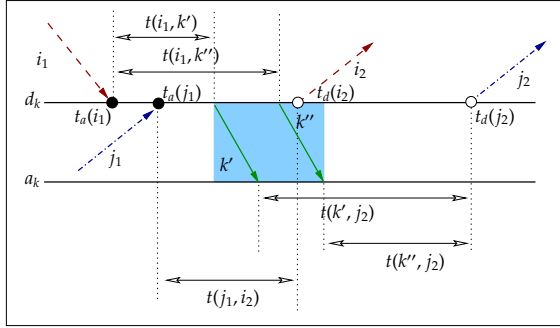
6. Conclusions and Future Research. In this study, we addressed the effects of a set of extra flights added to the flight schedule during operation in the context of the crew pairing problem. We argued that the adverse consequences of operational disruptions can be alleviated more effectively by incorporating robustness measures into the planning process. In particular, we introduced various recovery plans for managing extra flights and demonstrated that these plans can readily be incorporated into the crew pairing models at the planning stage. Our computational results suggest that substantially more robust crew schedules can be achieved at a relatively small incremental cost.

In this paper, we captured the cost of robustness as a budget constraint. In the future, we plan to explore this problem as a multi-objective optimization problem in order to identify the trade-off between the planned crew costs and robustness more precisely. Another interesting research direction is solving the proposed robust model (2) by column generation techniques for large instances. An interesting facet of this problem is that the cardinality of the second set of constraints in (2-5) depends on the pairings included in the model. Clearly, a constraint $2x_{(p,q)}^k \leq y_p + y_q$, where the pairings p and q form a type A solution for an extra flight k , would only be included in the formulation if both p and q are generated. Therefore, an algorithm that solves large instances of (2-5) may need to dynamically generate and delete both columns and constraints.

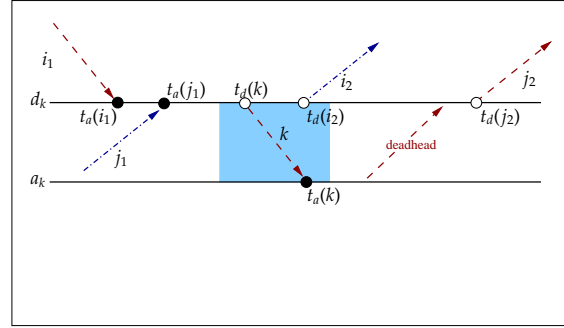
Acknowledgments. This research has been supported by The Scientific and Technological Research Council of Turkey under grant 106M472. We thank the anonymous referee for the invaluable suggestions and comments that helped us improve the paper considerably.

Appendix A. The Remaining Type A Solutions

$$\begin{aligned}
& \left((t_s^{\min} \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_s^{\max}) \text{ or } (t_r^{\min} \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_r^{\max}) \right), \\
& \left((t_{hs}^{\min}(k'', j_2) \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_{hs}^{\max}(k', j_2)) \text{ or } (t_{hr}^{\min}(k'', j_2) \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_{hr}^{\max}(k', j_2)) \right), \\
& \bar{T}_f(i_1) + t_f(k) + t_h(k, j_2) + \bar{T}_f(j_2) \leq T_f^{\max}, \\
& \bar{T}_e(i_1) + t(i_1, j_2) + \bar{T}_e(j_2) \leq T_b^{\max}, \\
& \left(t_s^{\min} \leq t(j_1, i_2) \leq t_s^{\max} \text{ or } t_r^{\min} \leq t(j_1, i_2) \leq t_r^{\max} \right) \\
& \left(\bar{T}_f(j_1) + \bar{T}_f(i_2) \leq T_f^{\max} \text{ and } \bar{T}_e(j_1) + t(j_1, i_2) + \bar{T}_e(i_2) \leq T_b^{\max} \right).
\end{aligned} \tag{A.2}$$



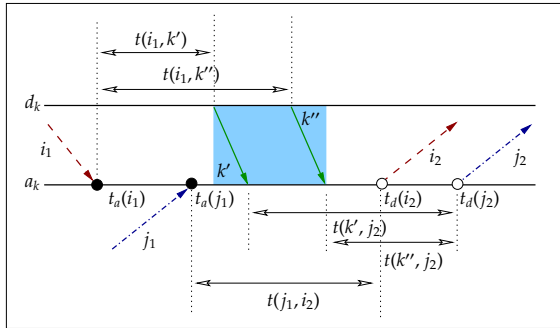
(a) Pairings p, q , and the time window for the extra flight.



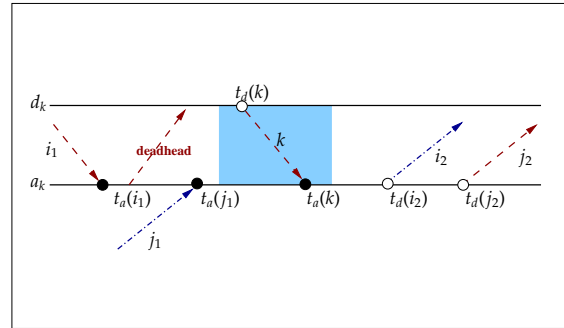
(b) A.2 solution for the extra flight k with one deadhead.

Figure 10: A sample A.2 solution on the flight network.

$$\begin{aligned}
& \left((t_{hs}^{\min}(i_1, k') \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_{hs}^{\max}(i_1, k'')) \text{ or } (t_{hr}^{\min}(i_1, k') \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_{hr}^{\max}(i_1, k'')) \right), \\
& \left((t_s^{\min} \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_s^{\max}) \text{ or } (t_r^{\min} \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_r^{\max}) \right), \\
& \bar{T}_f(i_1) + t_h(i_1, k) + t_f(k) + \bar{T}_f(j_2) \leq T_f^{\max}, \\
& \bar{T}_e(i_1) + t(i_1, j_2) + \bar{T}_e(j_2) \leq T_b^{\max}, \\
& \left(t_s^{\min} \leq t(j_1, i_2) \leq t_s^{\max} \text{ or } t_r^{\min} \leq t(j_1, i_2) \leq t_r^{\max} \right), \\
& \left(\bar{T}_f(j_1) + \bar{T}_f(i_2) \leq T_f^{\max} \text{ and } \bar{T}_e(j_1) + t(j_1, i_2) + \bar{T}_e(i_2) \leq T_b^{\max} \right).
\end{aligned} \tag{A.3}$$



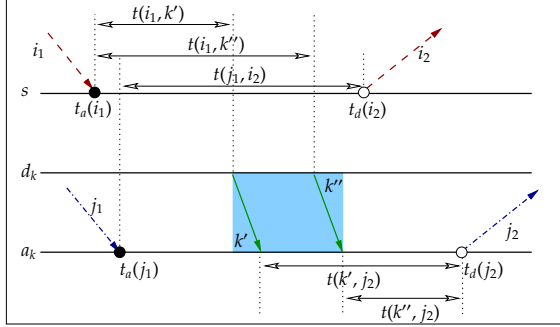
(a) Pairings p, q , and the time window for the extra flight.



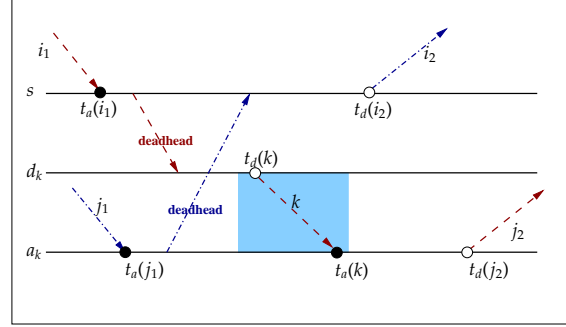
(b) A.3 solution for the extra flight k with one deadhead.

Figure 11: A sample A.3 solution on the flight network.

$$\begin{aligned}
 & \left((t_{hs}^{\min}(i_1, k') \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_{hs}^{\max}(i_1, k'')) \text{ or } (t_{hr}^{\min}(i_1, k') \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_{hr}^{\max}(i_1, k'')) \right), \\
 & \left((t_s^{\min} \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_s^{\max}) \text{ or } (t_r^{\min} \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_r^{\max}) \right), \\
 & \quad \vec{T}_f(i_1) + t_h(i_1, k) + t_f(k) + \vec{T}_f(j_2) \leq T_f^{\max}, \\
 & \quad \vec{T}_e(i_1) + t(i_1, j_2) + \vec{T}_e(j_2) \leq T_b^{\max}, \\
 & \left(t_{hs}^{\min}(j_1, i_2) \leq t(j_1, i_2) \leq t_{hs}^{\max}(j_1, i_2) \text{ or } t_{hr}^{\min}(j_1, i_2) \leq t(j_1, i_2) \leq t_{hr}^{\max}(j_1, i_2) \right), \\
 & \left(\vec{T}_f(j_1) + t_h(j_1, i_2) + \vec{T}_f(i_2) \leq T_f^{\max} \text{ and } \vec{T}_e(j_1) + t(j_1, i_2) + \vec{T}_e(i_2) \leq T_b^{\max} \right).
 \end{aligned} \tag{A.5}$$



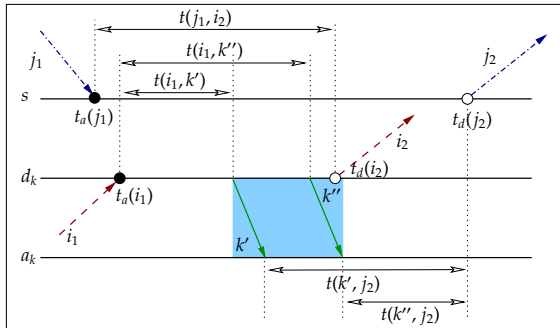
(a) Pairings p, q , and the time window for the extra flight.



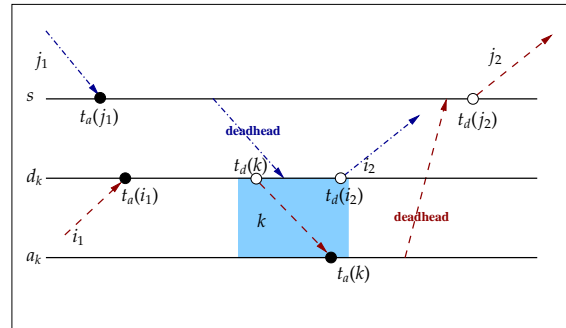
(b) A.5 solution for the extra flight k with two deadheads.

Figure 12: A sample A.5 solution on the flight network.

$$\begin{aligned}
 & \left((t_s^{\min} \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_s^{\max}) \text{ or } (t_r^{\min} \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_r^{\max}) \right), \\
 & \left((t_{hs}^{\min}(k'', j_2) \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_{hs}^{\max}(k', j_2)) \text{ or } (t_{hr}^{\min}(k'', j_2) \leq t(k'', j_2) \text{ and } t(k', j_2) \leq t_{hr}^{\max}(k', j_2)) \right), \\
 & \quad \vec{T}_f(i_1) + t_f(k) + t_h(k, j_2) + \vec{T}_f(j_2) \leq T_f^{\max}, \\
 & \quad \vec{T}_e(i_1) + t(i_1, j_2) + \vec{T}_e(j_2) \leq T_b^{\max}, \\
 & \left(t_{hs}^{\min}(j_1, i_2) \leq t(j_1, i_2) \leq t_{hs}^{\max}(j_1, i_2) \text{ or } t_{hr}^{\min}(j_1, i_2) \leq t(j_1, i_2) \leq t_{hr}^{\max}(j_1, i_2) \right), \\
 & \left(\vec{T}_f(j_1) + t_h(j_1, i_2) + \vec{T}_f(i_2) \leq T_f^{\max} \text{ and } \vec{T}_e(j_1) + t(j_1, i_2) + \vec{T}_e(i_2) \leq T_b^{\max} \right).
 \end{aligned} \tag{A.6}$$



(a) Pairings p, q , and the time window for the extra flight.



(b) A.6 solution for the extra flight k with two deadheads.

Figure 13: A sample A.6 solution on the flight network.

Appendix B. The Remaining Type B Solutions

$$\begin{aligned} & \left((t_{hs}^{\min}(i_1, k') \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_{hs}^{\max}(i_1, k'')) \text{ or } (t_{hr}^{\min}(i_1, k') \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_{hr}^{\max}(i_1, k'')) \right), \\ & \left(t_s^{\min} \leq t(k'', i_2) \text{ or } t_r^{\min} \leq t(k'', i_2) \right), \\ & \bar{T}_f(i_1) + t_h(i_1, k) + t_f(k) + \bar{T}_f(i_2) \leq T_f^{\max}. \end{aligned} \quad (\text{B.2})$$

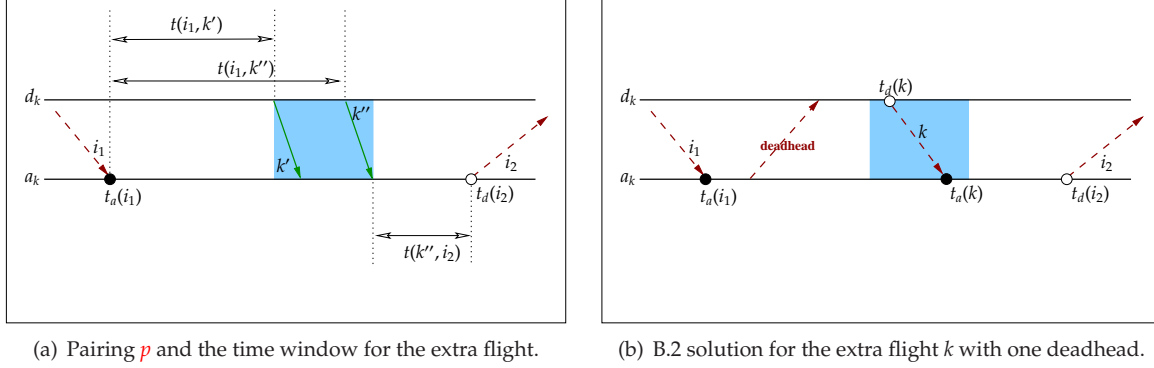


Figure 14: A sample B.2 solution on the flight network.

$$\begin{aligned} & \left((t_{hs}^{\min}(i_1, k') \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_{hs}^{\max}(i_1, k'')) \text{ or } (t_{hr}^{\min}(i_1, k') \leq t(i_1, k') \text{ and } t(i_1, k'') \leq t_{hr}^{\max}(i_1, k'')) \right), \\ & \left((t_{hs}^{\min}(k'', i_2) \leq t(k'', i_2) \text{ and } t(k', i_2) \leq t_{hs}^{\max}(k', i_2)) \text{ or } (t_{hr}^{\min}(k'', i_2) \leq t(k'', i_2) \text{ and } t(k', i_2) \leq t_{hr}^{\max}(k', i_2)) \right), \\ & \bar{T}_f(i_1) + t_h(i_1, k) + t_f(k) + t_h(k, i_2) + \bar{T}_f(i_2) \leq T_f^{\max}. \end{aligned} \quad (\text{B.3})$$

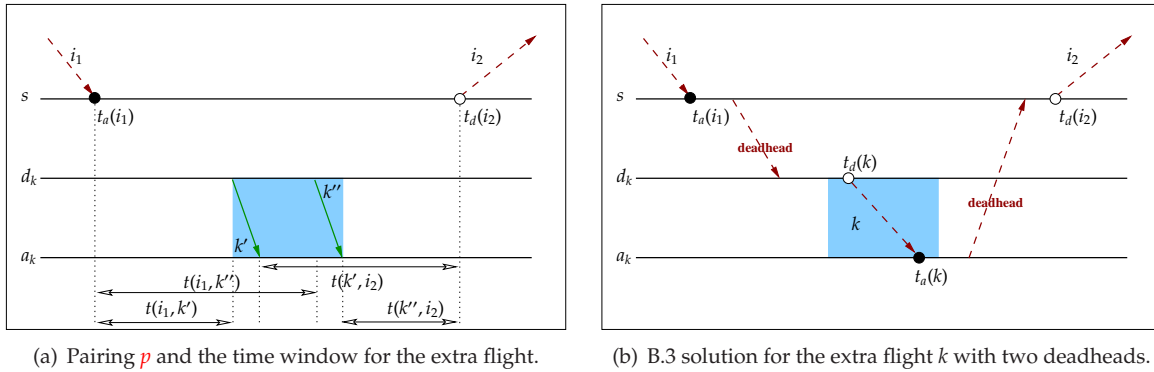


Figure 15: A sample B.3 solution on the flight network.

Appendix C. The Flight Data for the Computational Study. The flight data for all three problem instances used in the computational study are given in Tables 18-20, respectively. In these tables, the second and fifth columns denote the origin and destination pairs, and the third and sixth columns display the associated arrival and departure times.

Flight ID	Org. - Dest.	Dept. T. - Arr. T.	Flight ID	Org. - Dest.	Dept. T. - Arr. T.
1	IST-ANK	07:00-08:00	20	IST-ANK	15:00-16:00
2	IST-IZM	06:00-07:00	21	IST-IZM	17:00-18:00
3	IZM-ANK	10:05-11:20	22	IZM-IST	19:20-20:20
4	IST-ANT	08:25-09:40	23	IST-ANK	17:00-18:00
5	IZM-ANK	19:20-20:40	24	IZM-IST	22:00-23:00
6	IZM-IST	09:00-10:00	25	IST-IZM	20:00-21:00
7	ANT-IST	11:00-12:10	26	IST-ANK	19:00-20:00
8	IST-ANT	14:25-15:50	27	IST-ANK	22:00-23:00
9	IST-IZM	09:00-10:00	28	IST-ANK	23:45-00:45
10	IST-IZM	11:00-12:00	29	ANK-IZM	07:45-09:05
11	ANT-IST	16:50-18:05	30	ANK-IZM	17:00-18:20
12	IZM-IST	11:00-12:00	31	ANK-IST	08:00-09:00
13	IST-ANK	11:00-12:00	32	ANK-IST	11:00-12:00
14	IST-ANT	19:00-20:15	33	ANK-IST	14:00-15:00
15	IST-IZM	13:00-14:00	34	ANK-IST	17:00-18:00
16	IZM-IST	13:00-14:00	35	ANK-IST	13:00-14:00
17	IST-ANK	13:00-14:00	36	ANK-IST	21:00-22:00
18	ANT-IST	21:15-22:30	37	ANK-IST	20:00-21:00
19	IZM-IST	15:00-16:00	38	ANK-IST	22:00-23:00

Table 18: Flight data for the 38-flight instance.

Flight ID	Org. - Dest.	Dept. T. - Arr. T.	Flight ID	Org. - Dest.	Dept. T. - Arr. T.
1	IST-ADA	07:00-08:35	30	ANT-IST	10:40-11:55
2	IST-ADA	11:15-12:50	31	ANT-IST	13:40-14:55
3	IST-ADA	14:15-15:50	32	ANT-IST	16:45-18:00
4	ADA-IST	09:35-11:10	33	IST-BOD	14:30-15:40
5	ADA-IST	13:50-15:25	34	IST-BOD	19:30-20:40
6	ADA-IST	16:50-18:25	35	BOD-IST	16:45-17:55
7	IST-ANK	07:00-08:00	36	BOD-IST	21:40-22:50
8	IST-ANK	09:00-10:00	37	IST-DAL	17:20-18:40
9	IST-ANK	10:00-11:00	38	IST-DAL	19:35-20:50
10	IST-ANK	13:00-14:00	39	DAL-IST	19:40-21:00
11	IST-ANK	15:00-16:00	40	DAL-IST	21:50-23:05
12	IST-ANK	17:00-18:00	41	IST-IZM	07:00-08:00
13	IST-ANK	17:30-18:30	42	IST-IZM	08:00-09:00
14	IST-ANK	18:00-19:00	43	IST-IZM	08:30-09:30
15	IST-ANK	19:00-20:00	44	IST-IZM	11:00-12:00
16	IST-ANK	20:00-21:00	45	IST-IZM	13:00-14:00
17	ANK-IST	09:00-10:00	46	IST-IZM	15:00-16:00
18	ANK-IST	11:00-12:00	47	IST-IZM	16:00-17:00
19	ANK-IST	12:00-13:00	48	IST-IZM	20:00-21:00
20	ANK-IST	15:00-16:00	49	IST-IZM	23:45-00:45
21	ANK-IST	17:00-18:00	50	IZM-IST	10:00-11:00
22	ANK-IST	19:00-20:00	51	IZM-IST	10:30-11:30
23	ANK-IST	19:30-20:30	52	IZM-IST	11:00-12:00
24	ANK-IST	20:00-21:00	53	IZM-IST	13:00-14:00
25	ANK-IST	21:00-22:00	54	IZM-IST	15:00-16:00
26	ANK-IST	22:00-23:00	55	IZM-IST	17:00-18:00
27	IST-ANT	07:20-08:35	56	IZM-IST	19:20-20:20
28	IST-ANT	11:25-12:40	57	IZM-IST	22:00-23:00
29	IST-ANT	14:25-15:40	58	IZM-IST	01:45-02:45

Table 19: Flight data for the 58-flight instance.

Flight ID	Org. - Dest.	Dept. T. - Arr. T.	Flight ID	Org. - Dest.	Dept. T. - Arr. T.
1	IST-ANK	04:00-05:05	49	IST-IZM	16:00-17:05
2	IST-ANK	05:10-6:15	50	IZM-IST	18:05-19:10
3	ANK-IST	04:15-05:20	51	IST-IZM	17:00-18:05
4	ANK-IST	05:30-06:35	52	IZM-IST	19:05-20:10
5	IST-ANK	06:40-07:45	53	IST-IZM	21:45-22:50
6	ANK-IST	06:00-07:05	54	IZM-IST	23:50-00:55
7	ANK-IST	07:00-08:05	55	ANK-IZM	05:45-07:00
8	ANK-IST	07:30-08:35	56	IZM-ANK	08:00-09:15
9	IST-ANK	07:00-08:05	57	ANK-IZM	15:00-16:15
10	ANK-IST	08:00-09:05	58	IZM-ANK	17:15-18:30
11	IST-ANK	09:00-10:05	59	ANK-IZM	20:50-22:05
12	IST-ANK	10:00-11:05	60	IZM-ANK	23:10-00:25
13	ANK-IST	09:00-10:05	61	ANK-ANT	04:15-05:15
14	IST-ANK	11:00-12:05	62	ANT-ANK	06:15-07:15
15	ANK-IST	11:00-12:05	63	ANK-ANT	19:00-20:00
16	IST-ANK	13:00-14:05	64	ANT-ANK	21:00-22:00
17	ANK-IST	12:00-13:05	65	IST-ANT	06:25-07:40
18	IST-ANK	14:00-15:05	66	ANT-IST	08:40-09:55
19	ANK-IST	13:00-14:05	67	IST-ANT	09:30-10:45
20	ANK-IST	14:00-15:05	68	ANT-IST	11:45-13:00
21	ANK-IST	15:00-16:05	69	IST-ANT	12:45-14:00
22	IST-ANK	15:00-16:05	70	ANT-IST	15:00-16:15
23	ANK-IST	16:00-17:05	71	IST-ANT	15:30-16:45
24	IST-ANK	16:00-17:05	72	ANT-IST	17:55-19:10
25	IST-ANK	16:15-17:20	73	IST-ANT	17:00-18:15
26	ANK-IST	17:00-18:05	74	ANT-IST	19:15-20:30
27	IST-ANK	17:00-18:05	75	IST-ANT	18:30-19:45
28	IST-ANK	18:00-19:05	76	ANT-IST	20:45-22:00
29	IST-ANK	19:00-20:05	77	IST-ANT	21:55-23:10
30	ANK-IST	19:00-20:05	78	ANT-IST	00:15-01:30
31	ANK-IST	20:00-21:05	79	IST-ADA	06:20-07:50
32	IST-ANK	20:00-21:05	80	ADA-IST	08:50-10:20
33	IST-IZM	05:00-06:05	81	IST-ADA	15:00-16:30
34	IZM-IST	07:05-08:10	82	ADA-IST	17:30-19:00
35	IST-IZM	06:00-07:05	83	IST-ADA	17:20-18:50
36	IZM-IST	08:05-09:10	84	ADA-IST	19:55-21:35
37	IST-IZM	06:40-07:45	85	IST-ADA	12:15-13:45
38	IZM-IST	08:45-09:50	86	ADA-IST	14:45-16:25
39	IST-IZM	07:00-08:05	87	IST-ADA	14:00-15:30
40	IZM-IST	09:05-10:10	88	ADA-IST	16:30-18:00
41	IST-IZM	09:00-10:05	89	IST-ADA	19:30-21:00
42	IZM-IST	11:05-12:10	90	ADA-IST	22:00-23:30
43	IST-IZM	11:00-12:05	91	IST-ADA	09:15-10:45
44	IZM-IST	13:10-14:15	92	ADA-IST	11:45-13:15
45	IST-IZM	13:00-14:05	93	IST-ADA	21:35-23:05
46	IZM-IST	15:05-16:10	94	ADA-IST	01:30-02:00
47	IST-IZM	14:00-15:05	95	ANK-ADA	17:30-18:30
48	IZM-IST	16:10-17:15	96	ADA-ANK	19:30-20:30

Table 20: Flight data for the 96-flight instance.

References

- [1] R. Anbil, E. Gelman, B. Patty, and R. Tanga, *Recent advances in crew-pairing optimization at American Airlines*, *Interfaces* **21** (1991), 62–74.
- [2] R. Anbil, E.L. Johnson, and R. Tanga, *A global approach to crew-pairing optimization*, *IBM Systems Journal* **31** (1992), no. 1, 71–78.
- [3] C. Barnhart, A.M. Cohn, E.L. Johnson, D. Klabjan, G.L. Nemhauser, and P.H. Vance, *Airline crew scheduling*, *Handbook of Transportation Science* (R.W. Hall, ed.), Springer, 2003, http://www-personal.umich.edu/~amycohn/PAPERS/cs_survey.pdf, pp. 517–561.
- [4] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance, *Branch and price: Column generation for solving huge integer programs*, *Operations Research* **46** (1998), no. 3, 316–329.
- [5] R. Bixby, J.W. Gregory, I.J. Lustig, R. Marsten, and D. Shanno, *Very large scale linear programming: A case study in combining interior point and simplex methods*, *Operations Research* **40** (1992), 885–897.
- [6] T.G. Crainic and J.M. Rousseau, *The column generation principle and the airline crew scheduling problem*, *Infor* **25** (1987), no. 2, 136–151.
- [7] G. Desaulniers, J. Desrosiers, and M.M. Solomon (eds.), *Column generation*, GERAD 25th Anniversary Series, Springer, 2005.
- [8] M. Ehrgott and D.M. Ryan, *Constructing robust crew schedules with bicriteria optimization*, *Journal of Multi-Criteria Decision Analysis* **11** (2002), 139–150.
- [9] ILOG, June 2008, <http://www.ilog.com/products/#optimization>.
- [10] E.L. Johnson and B. Gopalakrishnan, *Airline crew scheduling: State-of-the-art*, *Annals of Operations Research* **140** (2005), 305–337.
- [11] L. Lettovský, E.L. Johnson, and G.L. Nemhauser, *Airline crew recovery*, *Transportation Science* **34** (2000), no. 4, 337–348.
- [12] A. Makri and D. Klabjan, *A new pricing scheme for airline crew scheduling*, *INFORMS Journal on Computing* **16** (2004), no. 1, 56–67.
- [13] D.M. Ryan and B. Foster, *An integer programming approach to scheduling*, *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling* (A. Wren, ed.), North-Holland Publishing Company, 1981, pp. 269–280.
- [14] A.J. Schaefer, E.L. Johnson, A.J. Kleywegt, and G.L. Nemhauser, *Airline crew scheduling under uncertainty*, *Transportation Science* **39** (2005), no. 3, 340–348.
- [15] S. Shebalov and D. Klabjan, *Robust airline crew pairing: Move-up crews*, *Transportation Science* **40** (2006), no. 3, 300–312.
- [16] P.H. Vance, A. Atamtürk, C. Barnhart, E. Gelman, E.L. Johnson, A. Krishna, D. Mahidhara, G.L. Nemhauser, and R. Rebello, *A heuristic branch-and-price approach for the airline crew pairing problem*, <http://www.crewingsolutions.com/docs/branch&price%20ACPO%20Vance%20lec9706.pdf>, June 1997.
- [17] P.H. Vance, C. Barnhart, E.L. Johnson, and G.L. Nemhauser, *Airline crew scheduling: A new formulation and decomposition algorithm*, *Operations Research* **45** (1997), no. 2, 188–200.
- [18] J.W. Yen and J.R. Birge, *A stochastic programming approach to the airline crew scheduling problem*, *Transportation Science* **40** (2006), no. 1, 3–14.