ROBUST CREW PAIRING FOR MANAGING EXTRA FLIGHTS

by

HATİCE TEKİNER

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabancı University

August 2006

ROBUST CREW PAIRING FOR MANAGING EXTRA FLIGHTS

APPROVED BY

Assoc. Prof. Dr. Ş. İlker Birbil ....
(Thesis Supervisor)

Assist. Prof. Dr. Kerem Bülbül ....

Assist. Prof. Dr. Cem Güneri ....

Assist. Prof. Dr. Gürdal Ertek ....

Assist. Prof. Dr. Tonguç Ünlüyurt ....

DATE OF APPROVAL: ....

to my family

&

Hulusi

Acknowledgments

It is a pleasure to express my gratitude to all who made this thesis possible. I would like to thank my thesis advisor Ş. İlker Birbil for his inspiration, guidance, patience, enthusiasm and motivation. Without his support, it would be impossible to complete this thesis. I would like to thank İlker Topçu, from Istanbul Technical University, for providing us the regulations used by the Turkish airline companies. I am indebted to Kerem Bülbül for helping me with ILOG optimization suite. I am grateful to my family for the concern, caring, love and support they provided throughout my life. Lastly, I would like to express my gratitude to Hulusi Moğulkoç for his endless love and support, continuous understanding and caring which helped me throughout this research.

ROBUST CREW PAIRING FOR MANAGING EXTRA FLIGHTS

Hatice Tekiner

Industrial Engineering, MS Thesis, 2006

Thesis Supervisor: Assoc. Prof. Dr. Ş. İlker Birbil

Keywords: crew pairing, robustness, extra flight, crew swapping, deadhead, computational results.

Abstract

The airline industry encounters many optimization problems such as scheduling flights, assigning the fleet, scheduling the crew. Among them, the crew scheduling problem is the most studied one. The main reason is that the crew cost is one of the largest components of the operational cost for an airline company. Therefore, there are many models proposed in the literature to find a cost efficient crew schedule. Most of those models divide the crew scheduling problem into two separate problems, namely the crew pairing and the crew assignment problems. The crew pairing problem that we study here aims at finding the least costly subset of pairings, which cover the scheduled flights.

Although there are many approaches to solve the crew pairing problem, most of them assume no disruptions during the operation. However disruptions due to weather conditions, maintenance problems, and so on are common problems leading to higher operational crew cost in practice. These kinds of disruptions result in delaying or canceling some scheduled flights. Another disruption that local airline companies face is *adding an extra flight to predetermined (regular) flight schedule*. In this study, we propose a model that provides robust crew pairing schedule in the case of adding an extra flight to the regular flight schedule. Two solution approaches are along with the mathematical model are proposed. The objective of the proposed model is to maximize the total number of solutions, while maintaining the increase in the crew cost at an acceptable level. A crew pairing problem is then solved by both the proposed model and the conventional model. Finally, computational experiments are conducted to demonstrate the benefits of the proposed model.

EK UÇUŞLARIN YÖNETİMİ İÇİN DAYANIKLI EKİP EŞLEME

Hatice Tekiner

Endüstri Mühendisliği, Yüksek Lisans Tezi, 2006

Tez Danışmanı: Doç. Dr. Ş. İlker Birbil

Anahtar kelimeler: ekip eşleme, dayanıklılık, yeni ucuş, ekip değiştirme, ölü uçuş, sayısal analiz sonuçları

Özet

Hava yolları endüstrisi uçuş planlama, filo atama, ekip planlama gibi bir çok optimizasyon problemi ile karsılaşmaktadır. Ekip maliyeti hava yolu şirketlerinin operasyonel maliyetleri içinde çok büyük bir paya sahip olduğu için, ekip planlama problemi belirtilen problemler içinde en çok çalışılan problemdir. Literatürde, ekip maliyetini en aza indirmeyi amaçlayan bir çok model bulmak mümkündür. Bu modellerin çoğunda ekip planlama problemi iki kısımda çözülmektedir, ekip eşleme problemi ve ekip atama problemi. Burada çalıştığımız ekip eşleme problemi bütün uçuşları bir kez kapsayan en az maliyete sahip eşlemeleri seçmeyi amaçlamaktadır.

Ekip eşleme problemi için bir çok yaklaşım olmasına rağmen, çoğu yaklaşım operasyon sürecinde herhangi bir aksaklığın yaşanmadığını varsaymaktadır. Ancak, hava yolu operasyonlarında hava durumu, bakım problemleri gibi nedenlerden dolayı kaynaklanan aksaklıkların yaşanması yaygın bir problemdir ve pratikteki ekip maliyetinin planlanandan daha yüksek olmasına neden olmaktadır. Bu aksaklıklar mevcut uçuş planında yer alan bazı uçuşların gecikmesine veya iptal edilmesine neden olabilmektedir. Ancak, yerel hava yolları şirketlerinin karşılaştığı başka bir aksaklık daha bulunmaktadır. Bu aksaklık kısaca *var olan ucuş programına yeni bir uçuşun eklenmesi* olarak tanımlanabilir. Bu çalışmada, bu tip bir aksamaya dayanıklı olabilecek bir ekip eşleme modeli önerilmektedir. Eklenebilecek bütün uçuşların bilindiği veya tahmin edilebildiği varsayılmaktadır. İki çeşit çözüm yaklaşımı matematiksel modelleri ile birlikte önerilmiştir. Sunulan modelin amacı ekip eşleme maliyetini kabuledilebilir bir seviyede tutarak, toplam çözüm sayısını maksimum yapmaktır. Çalışmamızda, ekip eşleme problemi önerilen model ve geleneksel model kullanılarak çözülmüş ve sunduğumuz modelin faydalarını göstermek için sayısal sonuçları verilmiştir.

Table of Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

The airline industry encounters many scheduling problems. One of the largest component of the operational cost of an airline is the crew cost, therefore, building a cost efficient crew schedule is widely studied in the literature. Due to the complex governmental and international regulations, and the size of the problems, the construction of an efficient schedule is a challenging task. Crew scheduling problems are usually solved in two phases: first the crew pairing problem, and then, the crew assignment problem are solved. The crew pairing problem involves the construction of the sequence of flights (pairings). In the crew assignment phase, the pairings are allocated to the individual crew members. In this study, the first part i.e., the crew pairing problem is considered.

The crew pairing problem aims to select a subset of all feasible pairings to cover each flight exactly once with minimum crew cost. This problem can be represented as a set partitioning problem,

$$
\begin{aligned}
\min \quad & \sum_{j \in P} c_j y_j \\
\text{s.t} \quad & \sum_{j \in P} a_{ij} y_j = 1, \qquad i \in F, \\
& y_j \in \{0, 1\},
\end{aligned}
\tag{1.1}
$$

where F is the set of all flights, P is the set of all pairings, c_j is the cost of the pairing j, $a_{ij} = 1$ if the flight leg i is covered by pairing j, 0 otherwise. The objective is to minimize the total pairing cost, and the constraints guarantee that each flight is covered only once. The model also includes constraints, which ensure that work distribution over crew base is matched with the crew base resource. These constraints are called balancing constraints.

The majority of the literature is concentrated on large size problems for big airline companies. However, there are many local companies that face different planning and operational stage problems. We consider an operational problem, which is solved with some recovery models, at the planning stage. Many local airline companies need to consider the demand for extra (irregular) flights besides their regular flight schedule. For example, during the summer season, the demand for flights to popular holiday destinations increases. Similarly, some customers (businessmen, government officers, and so on) may ask to hire a plane leading to a congestion in the regular schedule. However, the local companies do not know in advance the demand for these extra flights but they usually predict these possible extra flights from their past experiences. Therefore, most of the local airline companies do not consider such flights in the planning phase. Instead, they try to handle the extra flights at the operational level. Moreover, when there is high demand for a certain flight, the airline companies also consider adding an extra flight to the schedule at a time close to the highly demanded regular flight. Since there is no certain information about the time high demand occurs, the companies do not treat this extra flight as a regular flight. Instead, they add this extra flight if all necessary resources are available during the operation by revising their schedule. If covering the extra flight is crucial for the company, the company may even cancel or delay some regular flights to squeeze in the irregular flight. Hence, the operational crew cost becomes much higher than the originally estimated one.

Another reason for higher operational crew cost is disruptions or irregular operations. Weather conditions, congestions and employee sicknesses are typical causes of such disruptions. Therefore, robust airline crew pairing has become an important research area within airline scheduling. There are some studies aimed at constructing robust crew scheduling in the case of possibility of canceling or delaying a flight. In our study, the disruption is due to extra flights added to the flight schedule during the operation.

1.1 Contributions of This Research

Airline companies with a regular flight schedule occasionally need to add an extra flight to their regular schedule. These extra flights may be added by the company to meet an unforeseen demand for a specific route. There might also be a request from certain customers; such as tourism agencies, big firms, and so on, to add an extra flight. These requests might be very important, especially for relatively small local companies, which try to survive under heavy competition. They may even cancel or delay their regular flights to cover these extra flights.

The model proposed in this thesis provides a solution approach to manage the extra flights. Although the solutions may yield slightly higher cost than the cost found by solving the conventional crew pairing problem, our model builds a crew schedule by considering the possible extra flights. Hence, the recovery procedures to change the schedule after adding the extra flights may not be used . This leads to big savings from the recovery cost. Besides, there might not be any cancellation or delay of regular flights when the schedule produced by our model is used. So, although the company pays a slightly higher cost, its customers will receive a better service.

The company may even profit from managing the potential extra flights. For instance, if the company predicts or observes that demand for a regular flight becomes higher than usual at certain periods, then the company may consider adding an extra flight at a time close to its regular flight's time. Since the time of high demand is not known exactly in advance, adding this extra flight as a regular flight for the entire period may be unnecessary. On the other hand, if the company solves its crew pairing problem without considering these possible extra flights, when high demand occurs it may not be possible to add the extra flights without canceling or delaying regular flights.

We observe that our model and its extensions distribute the flights over pairings more evenly than the conventional crew pairing problem. Since the work-load resulting from the pairings will also be similar, the crews might be indifferent to their assignments. Therefore, having even distribution may also help solving the crew assignment problem. Our model may also encourage longer connection times. This may help the airline companies to handle the delays and the cancellations more efficiently. Therefore, our model may lead to a robust schedule against such disruptions.

1.2 Outline

A literature survey of crew pairing is presented in Chapter 2, in which we also discuss several solution methodologies, commonly used in the literature. In Chapter 3, we describe our problem and define a set of solutions and their feasibility conditions. We continue by proposing a model to solve the problem of adding an extra flight. We present in Chapter 4 our computational results to demonstrate the benefit of the proposed model. Finally, the conclusions and the future research are given in Chapter 5.

CHAPTER 2

LITERATURE REVIEW

Airline crew scheduling is composed of four phases. Firstly, an airline company determines the flights to be flown in a given time period. This phase is called *flight scheduling*. Then, fleet assignment problem is solved to allocate available aircrafts to flight legs according to the estimated demand for each flight leg. Third step is aircraft routing problem, which guarantees that each aircraft receives adequate maintenance checks. Finally, the crew scheduling problem is solved. We refer the reader interested in the schedule planning, fleet assignment and aircraft routing problems to Klabjan [20]. Hoffman *et al.* also explain optimization methods for fleet assignment problem [14].

The crew scheduling problem differs according to domestic and international operations, and according to cockpit and cabin crews. Generally, the international operations are sparse, and hence, finding a feasible solution may require using deadhead flights, i.e. crew members fly as passengers. Moreover, international operations are usually scheduled weekly. Therefore, solving the international airline crew scheduling problem requires specific attention. There are several studies in the literature to solve this kind of scheduling problems such as Barnhart *et al.* [5], and Barnhart and Shenoi [6]. On the other hand, domestic flights are denser than international ones, and hence, deadheading is not common. In addition, domestic flights operate on daily schedule. The distinction between scheduling cockpit crews and cabin crews is due to the fact that the fleet family types require different kind of qualifications for cockpit crews. Therefore, the cockpit crew scheduling problem is solved for each fleet family type separately. Hence we focus on the cockpit scheduling problem for domestic operations.

2.1 Definitions, Feasibility Rules and Cost Structures

The following terms that we shall define below have become standardized in the literature. We mainly use the works of Barnhart *et al.* [7], and Johnson and Gopalakrishnan

[16].

A flight leg is a nonstop flight which is also called *segment*. The sequence of the flight legs, where the arrival station of a flight is the same as the departure station of a next flight over a working day of a crew, 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 the duty period is called *a sit time*. *A crew base* is the airline station, where the crews are stationed. *A pairing* is a sequence of duty periods with an overnight rest in between. The overnight rest is generally called *a rest* or *a layover*. *A schedule* is defined as a sequence of pairings with the time off periods between two pairings. Unlike the duty period and pairings, schedules are formed for individual crew members. It is also common that a crew flies as passengers to reposition the crew to cover a flight, to return the crew to the crew base and so on. This kind of flight is called a *deadhead flight*. The network representation of some of the terms is given in Figure 2.1. The arcs shows the flight legs, the boxes with cross-lines represent the sit time between the flight legs and the box with horizontal lines represents the rest time. This network shows a two-day pairing.

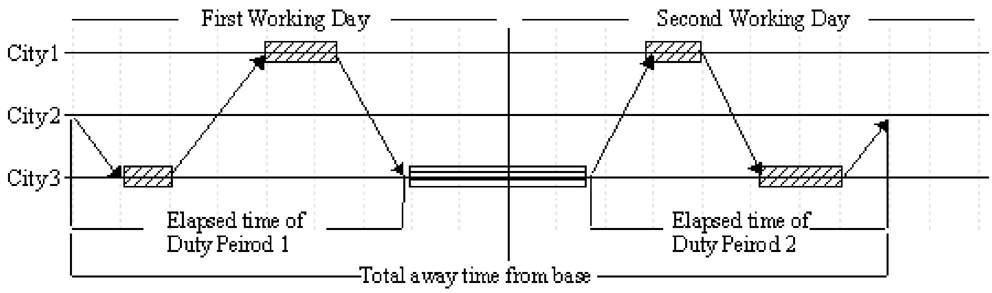


Figure 2.1: Some definitions of the crew pairing problem

There are many constraints on feasible duty periods and pairings. The international and governing agencies such as the Federal Aviation Administration (FAA), labor organizations, and the airlines themselves restrict crew scheduling due to the safety regulations, work conditions, and so on. The feasibility rules for the duty periods are as follows:

- The flights in a duty period should be sequential in time and space.
- There is a restriction on minimum and maximum connection/sit time, which are denoted by $minSit$ and $maxSit$ respectively.

- There is a restriction on the maximum elapsed time of a duty, which is denoted by $maxElapse$.

- There is a restriction on total number of flying hours, which is denoted by $maxFly$.

On the other hand, the feasibility rules for the pairings are as follows:

- A pairing should begin and end at the same crew base.

- There is a restriction on maximum number of duty periods in a pairing, which is denoted by $maxDuties$.

- There is a restriction on minimum and maximum rest time, which are denoted by $minRest$ and $maxRest.$ respectively.

- There is a restriction on the elapsed time of a pairing, which is denoted by $maxTAFB$.

- There is a rule called *8-in-24* forced by FAA, which states that crew can fly more than 8 hours in a 24 hour period if the rest between two duty periods and the rest after the second duty period are sufficient.

The crews are paid for the total flying time, and they receive some compensation for the excess time spent in connection and for the rest time period. Since the rate for each fleet family is the same, the cost of a pairing is expressed in terms of time. Therefore, the total flying time of all flights provides a lower bound for the crew scheduling problem.

The cost of a duty period is the maximum of the following three quantities:

- The total flying time in a duty period, which is generally denoted by $Ftime$.

- The fraction of the total elapsed time of a duty. The fraction is denoted by f_d, and the total elapsed time is denoted by $Etime$.

- The number of hours which the company guarantees its crew to be paid, which is denoted by $minguar$.

Formally, the cost of a duty period d is given by:

$$c_d = \max\{Ftime, f_d * Etime, minguar\}.$$

The cost of a pairing is the maximum of the following quantities:

- Total cost of duties in a pairing.
- The fraction of total elapsed time of a pairing. The fraction is denoted by f_p and total elapsed time is denoted by $TAFB$.
- The total number of guaranteed hours for each duty period.

Formally, the cost of a pairing p is as follows:

$$c_p = \max\{\sum_{d \in P} c_d, f_p * TAFB, nd * minguar\},$$

where nd is the number of duties in a pairing.

2.2 Crew Pairing vs. Assignment Problems

As stated above, the crew scheduling problem is solved in two stages. An airline company first solves the crew pairing problem to determine a subset of pairings that has the minimum cost, and covers each flight exactly once. Then, the crew assignment problem is solved to assign each crew to the subset of tasks. The crew assignment problem is solved in two stages. Firstly, a set of schedules are generated such that each pairing is included in as many schedules as possible to satisfy the necessary staff for each pairing. Then, these schedules are assigned to the individual crews. The reasons why the pairing problem and assignment problem are solved separately can be summarized as follows:

- In the crew pairing problem, we assign whole crews to the flights, but in the crew assignment problem, crews are assigned individually, where each pairing is covered by multiple crews.
- The objective function of the crew pairing problem is to minimize the labor costs, while the crew assignment problem aims at increasing the overall crew satisfaction and balancing the work distribution.

Although the crew scheduling problem is partitioned into two, solving individual problems is still a challenge due to the large number of pairings, complex work rules, and nonlinear crew costs. Since formulating a legal pairing mathematically is almost impossible, the pairings should be enumerated for legality check and cost calculation. Therefore, there are two major components of the crew pairing problem, the generation of feasible pairings, and the optimization of the resulting problem. The crew pairing problem is generally formulated as a set partitioning problem (1.1). In this formulation, the rows of the problem corresponds to the flights and the columns represent feasible pairings. If deadheading is allowed in the problems, one can use the set covering problem instead of the set partitioning problem. In this case, a flight leg that is actually flown (a flying leg) is represented by one row, and deadhead is represented by another row. However, the cost of a flying leg and deadheading are different, this adds an extra difficulty for evaluating the cost of the pairings including the deadheads. This yields an important drawback , namely the optimal solution of set covering problem can be a suboptimal solution. Therefore, in the literature there are several studies on handling deadheads. We refer the interested reader to [16].

The crew pairing problem of domestic flights is generally solved in three stages: daily, weekly exceptions and transition. In daily crew pairing problem, it is assumed that each flight is repeated every day. Therefore, the flights that are repeated more than four days per a week are generally selected, and those flights are assumed to operate daily. Since all flights are not repeated every day, the solution of daily problem is not feasible in practice. Therefore, the pairings which include at least one flight that is not repeated every day, are broken. The flights of broken pairings, and the flights, which operate for less than four days, are then solved together. This step is called a weekly transition problem. In this step, the flights should be fixed to specific day of the week. The airline gets its weekly schedule after solving both Daily and weekly exceptions problems. Another problem occurs when the airlines change their flight schedule. In this case, some of the flights may not be in the next period's schedule, so some of the pairings become infeasible. To cover the flights during this changeover period, the transition problem is solved.

In this thesis we focus only on the crew pairing problem. We mention only three approaches used to solve the crew assignment problem: rostering, bidline and preferential bidding. In the rostering process, the crew members give their preferences for individual pairings and pattern, then the problem with the objective of meeting as many preferences as possible and minimizing the potential cost is solved. In the bidline process, bidlines (generic monthly assignments) are firstly generated and, according to seniority, crew members bid for bidlines. As an alternative process, individual rosters, based on individual preferences, can be formed sequentially. This process is called preference bidding. In Barnhart *et al.* the details about the model and the solution approach to the crew assignment problem can be found [7].

2.3 Crew Pairing Problem: Solution Approaches

There are three main steps for solving the crew pairing problem: generating the pairings, solving the LP relaxation, and finding promising solutions of the problem. In the subsequent part of this section, we review different methodologies that are proposed to deal with these steps.

2.3.1 Pairing Generation

There are two main network structures used in a pairing generation. The first one is the flight network, the second one is the duty network. The nodes in the flight network represent the arrival and departure of each flight leg, and the arcs represent the flight legs. Two additional nodes, called source and sink, are added to the network. Both of these nodes represent the same crewbase. In the flight network, there are also special arcs called *connection arcs*, which represent the possible connections. An illustrative flight network example is given in Figure 2.2. The solid lines represent flight legs and the dashed arcs represent the possible connections. The connection arcs are added between the arrival node and departure node if the time between them is either within the legal sit time or the rest time. Each flight arc is replicated as many times as the maximum number of duties in a pairing. However, the pairings start only from the first day flights. Therefore, the source node is connected to the departure node of the flights which depart on the first day from the specified crew base. Since the pairings should end at the same crew, all arrival nodes, which arrive at the specified crew base, are connected to the sink. The path from the source to the sink is a legal pairing. This network structure helps to construct the legal pairings easily with respect to connection times, but other feasibility rules are not embedded into the network. Therefore, labels are used for checking feasibility rules.

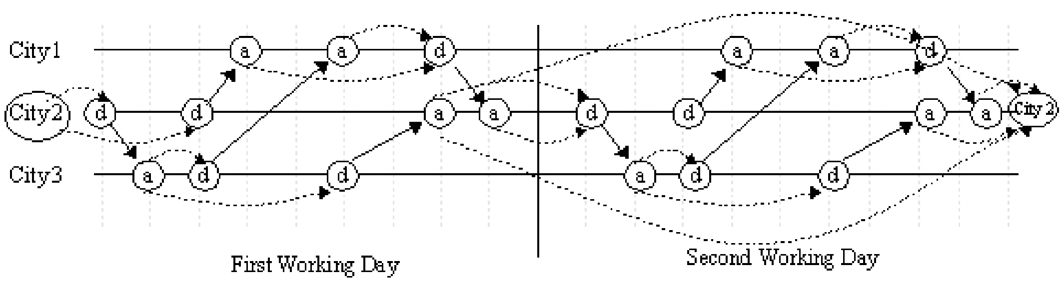


Figure 2.2: Network representation of the flight network

The nodes in the duty period network represent the departure and arrival of each duty period, as well as the special nodes, source and sink. Arcs represents the legal duty periods. There are also arcs for legal connections between duties. Each duty period is replicated as many times as the maximum number of duties in a pairing. In a duty period network, if the flight is covered once, the connection between duties, which share the same flight, should be eliminated.

The construction of both flight and duty network requires a careful study of data structure and algorithms. To find pairings, depth-first search algorithm is commonly used on a duty network or flight network.

2.3.2 Partial Enumeration of Pairings

In the literature, some approaches use partial enumeration of the pairings. One of well-known approaches is called the *TRIP* approach. This approach begins with an initial solution. The initial solution is generally found by modifying the previous period solution. If finding an initial solution is difficult, then using a crew for each flight leg can be an initial solution. This approach randomly selects a small number of pairings among the pairings in the incumbent solution. Then, all possible pairings that can cover the flights, which are already covered by the selection, are generated. Next, for those flights the crew pairing problem is solved to arrive at IP solution. If there is a subset of pairings which has smaller crew cost, then selected pairings are replaced with the recently generated pairings. This procedure continues until a predetermined number of iterations elapses, or until there is no change in the objective function value for several iterations. The drawback of this approach is that it may end up at a local minimum. Anbil *et al.* use the *TRIP* approach to solve the crew pairing problem at American Airlines [1]. They propose some enhancements to decrease the impact of a local minimum. The suggestions are simultaneously considering alternative optimal pairings and solution paths, solving larger subproblems, and allowing TRIP to make moves in the direction that initially do not improve the objective function. Moreover, Klabjan *et al.* generate the pairings randomly [18]. They use a probability for selecting a connection, while generating a pairing. These probabilities are estimated by using connection times. The idea is based on the assumption that the smaller the connection time, the lower the cost will be. Therefore, the probability of selecting a shorter connection is higher than selecting a longer connection. Another approach is developed by Housos and Elmroth [15]. They develop a methodology for solving the subproblems that evaluate all possible connections in one day while fixing the connections all other days. Their solution to the problem tries to find the best subset of connections among all the connections of free days while keeping all the connections of other days fixed. They use some preprocessing heuristics to find and lock some connections to reduce the solution space. They firstly lock the connections at airports which have only one arrival and departure. Also, they find locally optimal connections for airports with low traffic by matching algorithm.

2.3.3 Solving LP Relaxation

In the literature, different methods are used to solve LP relaxation of the crew pairing problem. Anbil *et al.* use the *SPRINT* method associated with the *TRIP* approach to improve the quality of the solution [2]. The *SPRINT* method requires the selecting of a subset of columns, and solving the LP over those columns. Then, dual variables of this subproblem are used to price out the columns of the original problems. If there are columns with almost zero reduced cost, then a small subset of the columns that have negative reduced cost is added to the subproblem. Anbil *et al.* selects new columns based on best bucket first, and uses the follow-on branching rule to find an integer solution [2]. They solve a problem with 5.5 million columns with only 25 subproblems.

Another widely used method in the literature is column generation. Problem (1.1) is the master problem in the column generation method. It contains all the possible pairings. A problem with a subset of pairings is called the restricted master problem. Overall, the column generation method follows the following steps:

- Solve the restricted master problem to find an optimal solution.
- Solve the pricing subproblem [17] to determine the column, which may improve the incumbent solution.
- Update the restricted master problem by adding the generated pairings.

If no columns are found at the second step, it means that the current optimal solution of the restricted master problem is also optimal for the master problem. This procedure is repeated until the optimal solution is found.

Crainic and Rousseau use a column generation method to solve airline crew pairing problem [9]. They formulate crew pairing problems as set covering problem. They use an algorithm which firstly generates one day pairings and solves the crew pairing problem over those pairings to find a pairing that improves the solution. Then, they increase the number of duty periods in a pairing by one at each iteration and try to find a pairing with negative reduced cost. This process continues until they find no pairing with negative cost. Another study using column generation is Anbil *et al.* [3]. Column generation approach is also used to solve other scheduling problems. Desrochers and Soumis use the column generation to solve urban transit crew scheduling problem [10]. The problem is modeled as set covering model and the shortest path algorithm with resource constraints is used to find new feasible workdays which can improve the objective function.

In the literature, different pricing criteria and several methods for finding the pairings that meet these criteria are used. It is common to use the reduced cost as the pricing criteria. Anbil *et al.* use the reduced cost criteria and shortest path column generation scheme [3]. They use the duty tree, which is composed of duty arcs and connection arcs. They also use the dual variables of the restricted master problems to reduce the arc costs, component wise. Then, depth-first search algorithm is applied in the tree to find a pairing with negative reduced cost. Bixby *et al.* introduce a new pricing criteria using a score obtained by dividing the pairing cost with the sum of the dual values of the legs in the pairing [8]. This criterion reduces the number of iterations. Klabjan and Makri also use the same criterion [19]. They develop some pruning rules to fathom the column enumeration. They use shortest path algorithms to find the pairings meeting the selection criteria. They add columns with a low ratio of cost over sum of the dual prices in the column, instead of adding columns with the low reduced costs. This ratio is called *a score*. The contribution of this article is embedding this rule in a column generation algorithm, where columns are enumerated by depth-first search algorithm during pricing.

Klabjan and Makri define a new network called *the mixed segment/duty timeline network*. This network has two nodes for each leg (arrival and departure nodes) and two types of arcs. For each duty there is a duty arc that connects the departure node of the first flight to the arrival node of the last flight in the duty. There is a connection arc between the arrival node and departure node if they are at the same station and the time between them is within the interval [minRest, maxRest]. They define the score for a pairing p, where c_p is the pairing cost and P is the set of all possible pairings, as $s_p = c_p / \sum_{i \in p} y_i$. Since the pairing cost is nonnegative, pairing p has negative reduced cost if and only if $\sum_{i \in p} y_i > 0$ and $s_p < 1$. Therefore, the goal during the pricing is to solve

$$\min_p\{s_p | \sum_{i \in p} y_i\}.$$

If this minimum is greater or equal to 1 then the current solution is optimal. To enumerate all of pairings in every iteration of a column generation algorithm is too time consuming. Therefore, some pruning rules are developed. A partial pairing is defined as a sequence of duties that start at a crew base and meets all feasibility rules except ending at the same station. Pairing is generated using the depth-first search on the proposed network. Partial pairings are expanded with suitable duties and when a pairing is found, backtracking is applied. Pruning is a procedure that fathoms depth-first search algorithm of a partial pairing before a pairing is actually obtained. They develop two types of pruning rules, approximate and exact. Approximate pruning rules cut partial pairings that can produce full pairings with scores greater than or equal to 1. Exact pruning rules prune only partial pairings that would always result in pairings with score greater than or equal to 1. At first, they apply approximate pruning rules, and when the pairings with score less than 1 is not found by approximate pruning, then they apply exact pruning rules.

Along the same line, Vance *et al.* divide the decision process into two parts [25]. First, they select the duty periods that partition the flight legs, they then select the pairings, which further partition these duties. They use dynamic column generation. The formulation they presented is shown to provide a tighter LP bound.

2.3.4 Finding Integer Solutions

There are three approaches to find a good integer solution to a crew pairing problem in the literature. The first approach uses the *off line* column generation. That is, a small subset of the pairings are enumerated, and then the problem is solved optimally for this subset. An application of such model is given in [13]. The second approach uses dynamic column generation to solve LP relaxation of the set partitioning problem, it then uses branch-and-bound to find integer solutions over this subset of columns generated. Klabjan *et al.* propose an algorithm that solves an LP relaxation of set partitioning problem and finds an integer solution by using several million pairings which has low reduced cost [18]. The last approach is called *branch-and-price.* The dynamic column generation is applied throughout the *branch-and-bound* tree. Vance *et al.* propose an approximation that solves column generation subproblems approximately, and it does not necessarily consider all of the unexplored nodes in the search

[26]. They use labels to represent the state of each path in the network with respect to the pairing cost and rules. A multi-label shortest path algorithm is used to find the attractive pairings. They find the dominated paths (the paths which have a higher cost than at least one non dominated path), and then they only use non-dominated paths to generate the pairings. As an approximation of the cost, they place an upper bound on the number of calls to the pricing subproblem instead of generating pairings until no pairings with negative reduced cost are found at each node. They also compute a target value for the LP relaxation at any node in the tree. The target LP value forced column generation when the value of the LP relaxation increases above its target value after fixing a follow-on, and this value also prevented column generation when there has been little or no change in the LP bound. Vance *et al.* also limit the number of non-dominated paths that is saved to any node in the network to speed up the shortest path algorithm. When it is relatively easy to find columns with negative reduced costs, a small number of non-dominated paths is used, otherwise, a larger set is used. When column generation has ceased at a node, and the value of IP solution is less than or equal to the target IP value, algorithm stops; otherwise backtracking is performed. Detailed information about branch and price approaches for large integer programming problems is given by Barnhart *et al.*, [4].

There are different branching rules used in *branch-and-bound* approach. In the literature, the *follow-on* branching rule is most frequently used one (see Vance *et al.* [26], and Anbil *et al.* [3]). They are motivated from a general rule used for set partitioning problems developed by Ryan and Foster [22]. For the crew pairing problem, Vance *et al.* define the *follow-on* branching as two flights should be covered consecutively on one branch and not covered consecutively on the other branch [26].

Another branching strategy is proposed by Klabjan *et al.*, which is called *timeline branching* [18]. They first select a flight and generate all the pairings which contain this flight. Then, they divide those pairings according to the connection time between the selected flight and its following flight. That is on one branch there are only pairings, which have the connection times less than some threshold value, and on the other branch there are only pairings, which have the connection times greater than the threshold value. They show that it is possible to select such pairing and to find a threshold time value. Klabjan *et al.* has also combined timeline branching and follow-on branching and proposed a new branching strategy called strong *branching* [18]. Change in each node is calculated by number of dual simplex iterations, and accordingly branching rules select the branching.

2.4 Crew Recovery Problem

The airline schedule rarely operates as planned due to some disruptions such as maintenance problems, weather conditions, and so on. Therefore, at the operational level, the crew recovery problem (CRP) has to be solved. The objective of this problem is to modify the disrupted schedule as quickly as possible with minimum costly reassignment. Barnhart *et al.* define main differences between the planning problem and the recovery problem as follows [7]:

- Time horizon to solve the problem.
- The recent flying history is taken into account in the recovery model.
- Reserve crews can be used to solve the recovery problem.
- Feasibility rules for pairings.
- The objective of the problem.

The solution time for recovery problem is very important. Therefore, it is not necessary to reassign all the crews; only those crews, whose pairings were disrupted, and a small number of additional crews that are enough for good swapping, should be considered. This reduces the size of the problem.

Lettovský *et al.* solve the CRP with the objective function of minimum cancellations and minimum additional cost [21]. They consider the monthly flown hours, partially flown pairings and future assignments. The CRP differs from the crew pairing problem in terms of its dynamic environment and its requirement to provide solutions with limited impact to the crew's original schedule. Lettovský *et al.* introduce a heuristic to reduce the size of the problem and provide computationally inexpensive deadhead selection heuristic. Then, they formulate CRP as an integer problem and solve this problem using primal-dual subproblem simplex method.

A preprocessing technique is used to extract a subset of the schedule for rescheduling. They limit the crew involved search by a combination of a predefined time window and a maximum number of crew per mis-connection. After the crews are determined, the corresponding pairings are divided into two parts, flown and unrealized. Then, they develop a crew-pairing generator, which enumerate feasible continuation for the partially flown pairing and to find integer solutions. All partially flown pairings are pairwise connected to generate a full pairing. For deadhead selection, they consider four types of deadheads; multiple assignment, out of flight time, catching up and misconnection. In addition, for reducing the storage requirement, they use tree-based data structure.

Desroisers *et al.* also work on the operational crew recovery problem [11]. In their paper, they call this problem the operational crew pairing problem. Their primary goal is to solve this problem as quickly as possible. According to the size of the disturbances, they chose the operational period and the amount of time to solve the problem. Then, they determine the involved crew, i.e. the crew whose monthly schedule is affected by the disturbances. They model the problem as multi-commodity network flow problem with resource variables, and solve the problem by using Dantzig-Wolfe decomposition in conjunction with the branch and bound method.

2.5 Robust Airline Crew Pairing Models

Recall that there can be some disruptions in the scheduling due to the weather conditions, maintenance problems, and so on. The crew schedule cannot operate as planned due to the such disruptions. Above, we explained how the airline company can solve this problem at operational level. However, there are several studies which try to build a robust model against those disruptions. In this section, we give a short summary of such studies.

Schaefer *et al.* consider the crew pairing problem with an objective of minimizing the expected crew cost [23]. They approximate the objective function coefficients by simulation. They assume that the cost of a pairing is independent of the cost of other pairings. They show that this assumption is true when the recovery is made according to the push-back recovery principle, which states each flight should be delayed until all resources are available. Thus, they use a push-back recovery procedure.

The crew pairing problem under uncertainty is also considered in Yen and Birge [27]. They formulate the crew pairing problem as a two-stage stochastic problem and consider the effect of sit time maximization on the robustness. They try to identify the pairings that are less sensitive to the delay. Ehrgott and Ryan penalize the connections, which do not have enough time to cover the expected delay [12]. They then solve a bicriteria optimization problem, where the objective is to minimize the cost as well as the total penalty.

Shebalov and Klabjan propose another approach to solve robust crew pairing problems in [24]. They define move-up crews, and try to maximize the total number of move-up crews, while controlling the crew cost. The move-up crew is defined as crews that can be swapped during the operation. In this study, we are also motivated by a similar idea. Consider the case that a flight is delayed and its crew should fly another flight leg. However, due to the delay this crew cannot fly that leg. Then, another crew covers the flight of the disrupted crew. This crew is called move-up crew.

Shebalov and Klabjan present a crew pairing model that considers move-up crews. In addition to capturing the crew cost, this model also obtains the crew schedules that have many opportunities for crew swapping. The article introduced a new objective function that captures the number of move-up crews. For each flight, the number of move-up crews are counted. A move-up crew is ready to fly before the departure time of the considered flight and has the same remaining duty periods as the pairing covering the considered flight. Obviously, there is a trade-off between maximizing the number of move-up crews and minimizing the crew cost. Therefore, it is assumed that traditional crew pairing problem is initially solved, that is, planned crew cost is given, and to prevent increasing the cost too much, an upper bound on the crew cost is added to model. They solved the model by a combination of Lagrangian relaxation and delayed column generation. The model that we propose in this study is also similar to the model given by Shebalov and Klabjan [24].

CHAPTER 3

ROBUST CREW PAIRING FOR MANAGING EXTRA FLIGHTS

The crew pairing problem is at the fourth step of airline scheduling. Hence, we assume that the flight scheduling and the fleet assignment problems are solved. Since the qualifications of the cockpit crews are different for each equipment type, the crew pairing problem should be solved for each equipment type. We assume that the flight schedule for a selected equipment type is also given. Then, our model tries to find the crew pairings, which can cover the potential extra flights that might be added to the current schedule. The drawback of such pairings might be a slight increase in the crew cost.

The extra flights are the flights that are not in regular flight schedule, but they may be added to the schedule during the operation. There are two reasons for requiring the additional extra flight. A customer can demand an extra flight from the company or the company may want to add an extra flight due to the high demand for a scheduled flight. We assume that at the planning phase, the company knows or predicts possible extra flights at the planning stage.

There are two types of solutions that can be included into the robust crew schedule:

Type A: Select two pairings such that swapping the crews covers the extra flight.

Type B: Select a pairing such that there is enough time between two consecutive legs of these pairings to cover the extra flight.

We adopt the notation given in Table 3.1.

As stated in Table 3.1, the term *deadhead* is slightly different. Here, a deadhead represents a flight without passengers. Let TI_{ij} represents the time between departure of flight j and arrival of flight i, then the limitations on legal pairings of our model are defined and estimated as follows.

$\underline{SD}_{ij}$ is the minimum required time to cover flight j after flight i and $\overline{SD}_{ij}$ is the upper bound on the connection time between flights i and j if the crew should deadhead

Notation	Definition
F	set of all flights
K	set of all possible extra flights
P	set of all pairings
P_i	set of all pairings covering flight i
d_k	departure station of extra flight k
a_k	arrival station of extra flight k
dt_k	departure time of extra flight k
at_k	arrival time of extra flight k
F_i	fly time of flight i
D_{ij}	deadhead from a_i to d_j without passenger
DT_{ij}	deadhead time from a_i to d_j
RT_{ij}	required time to cover flight j after flight i
TFT	total fly time of a pairing
$FT(i)$	total fly time until arrival of flight$i$
$RFT(i)$	total remaining fly time after flight$i$
$ET(i)$	total elapsed time until arrival of flight i
$RET(i)$	total remaining elapsed time after first consecutive departure

Table 3.1: Notation and definitions

from the arrival station of flight i to the departure station of the flight j and if all the connections between departures and arrivals are sits;

$$\begin{aligned}\underline{SD}_{ij} &= minSit + DT_{ij} + minSit,\\ \overline{SD}_{ij} &= maxSit + DT_{ij} + maxSit.\end{aligned}$$

$\underline{RD}_{ij}$ is the minimum required time to cover flight j after flight i and $\overline{RD}_{ij}$ is the upper bound on the connection time between flights i and j if the crew should deadhead from the arrival station of flight i to the departure station of flight j and one of the connection is rest period;

$$\begin{aligned}\underline{RD}_{ij} &= minSit + DT_{ij} + minRest,\\ \overline{RD}_{ij} &= maxSit + DT_{ij} + maxRest.\end{aligned}$$

There are nine possible cases with different feasibility conditions, six of which are *Type A* and the remaining three are of *Type B*. In the following sections, all possible cases are throughly explained .

3.1 Type A Solutions

Two pairings can form a *Type A* solution only if the pairings are also feasible after swapping. In addition, we should also consider the remaining work schedule of the involved crews.

Let $p^{'}$ be the pairing which covers flight i^1 and flight i^2 consecutively and can be used to built a *Type A* solution for extra flight k and let $p^{''}$ be the pairing, which covers flight j^1 and flight j^2 consecutively, and which, can be used to built a *Type A* solution. We say that $p^{'}$ and $p^{''}$ compose *Type A* solution for flight k if the crews of both pairings are from the same base, finish on the same day and satisfy the appropriate *feasibility conditions*. These feasibility conditions are related to the relative arrival and departure time and station of the corresponding flights. We cover these feasibility conditions explicitly when we discuss different solutions later. The set of pairing tuples $(p^{'}, p^{''})$, which form *Type A* solution for extra flight k, is denoted by $\mathbf{P}_k$. Formally,

$$\mathbf{P}_k = \{(p^{'}, p^{''}) : p^{'}, p^{''} \in P \text{ and they form a Type A solution}\}$$

There are six different *feasibility conditions* for *Type A* solution. Three of them cover the extra flight by one deadhead and three of them cover by two deadheads. We assume that more than two deadheads are not acceptable.

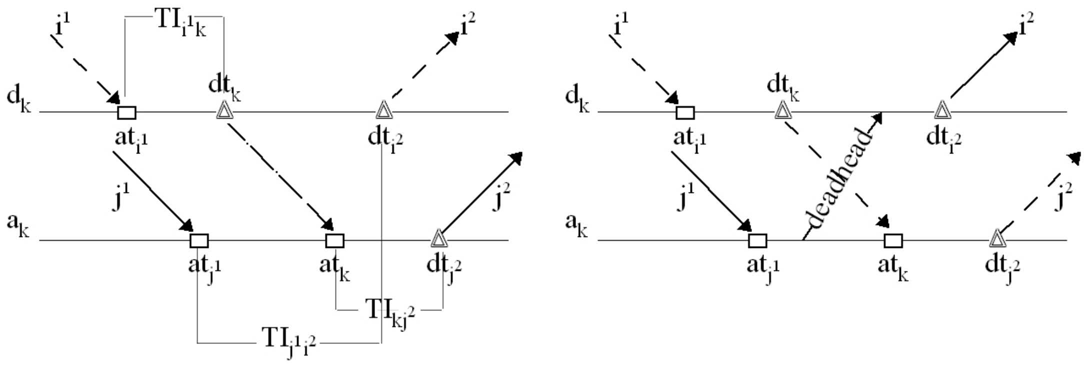
In the following sections, we define two pairings, $p^{'}$ and $p^{''}$. In each solution type, the flights in these pairing should satisfy different feasibility conditions so that involved pairings compose a swapping, that is a *Type A* solution.

In the following figures, the dashed lines show the flights covered by pairing $p^{'}$, solid lines show the flights covered by $p^{''}$, and dot-dashed line shows the extra flight. The left subfigures show the feasibility conditions for the pairings to compose a solution and the right subfigures show the solution for the extra flight.

Solution A.1 : One Deadhead

If there is a pairing $p^{'}$ covering a flight arriving at d_k before dt_k, like i^1, which is the latest arrival before the extra flight k and there is a pairing $p^{''}$ covering a flight departing from a_k after at_k, like j^2, which is the first nearest departure after arrival of flight k as in Figure 3.1, we can say that these two pairings is a *A.1 solution*, only if they satisfy feasibility conditions presented in Table 3.2.

As it can be seen in Figure 3.1(a), if there are two such pairings satisfying feasibility



(a) Feasibility conditions of the flights in pairings $p^{'}$ and $p^{''}$ for Solution A.1

(b) Solution A.1 for extra flight k

Figure 3.1: Illustrative example of Solution A.1 on the flight network

Pairing	Feasibility Conditions		
$p^{'}$	$minSit \leq TI_{i^1k} \leq maxSit$	or	$minRest \leq TI_{i^1k} \leq maxRest$
	$minSit \leq TI_{kj^2} \leq maxSit$	or	$minRest \leq TI_{kj^2} \leq maxRest$
	$FT(i^1) + F_k + RFT(j^2) \leq maxFT$	and	$ET(i^1) + TI_{i^1j^2} + RET(j^2) \leq MaxTAFB$
$p^{''}$	$\underline{SD}_{j^1i^2} \leq TI_{j^1i^2} \leq \overline{SD}_{j^1i^2}$	or	$\underline{RD}_{j^1i^2} \leq TI_{j^1i^2} \leq \overline{RD}_{j^1i^2}$
	$FT(j^1) + F_k + RFT(i^2) \leq maxFT$	or	$ET(j^1) + TI_{j^1i^2} + RFT(i^2) \leq maxFT$

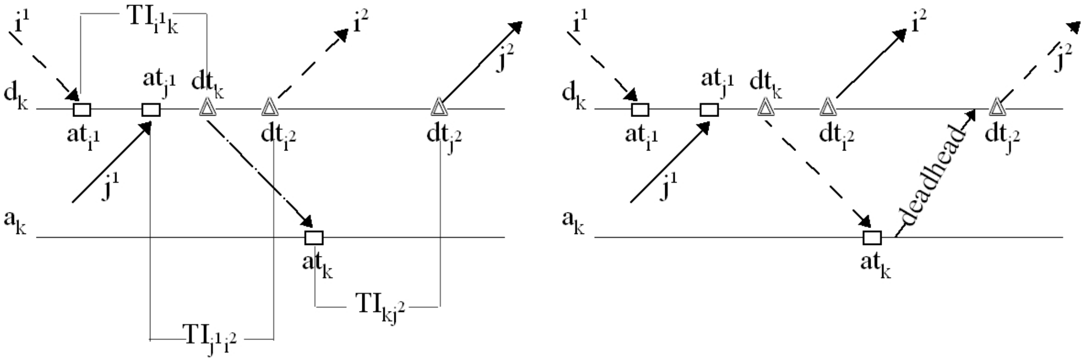
Table 3.2: Feasibility conditions for Solution A.1

conditions for the solution A.1, we can assign the crew of pairing $p^{'}$ to extra flight k and j^2 and all the following flights after j^2. And, we can use the crew of pairing $p^{''}$ to cover flight i^2 and all the following flights after i^2.

Solution A.2 : One Deadhead

If there is a pairing $p^{'}$ covering a flight arriving at d_k before dt_k, like i^1, which is the latest arrival before the extra flight k and there is a pairing $p^{''}$ covering a flight departing from a_k after at_k, like j^2, which is the first nearest departure after arrival of flight k as in Figure 3.2, we can say that these two pairings compose a *A.2 solution*, only if they satisfy feasibility conditions presented in Table 3.3.

As it can be seen in Figure 3.2(a), if there is such two pairings satisfying solution A.2, we can assign the crew of pairing $p^{'}$ to extra flight k and j^2 and all the following flights after j^2. And, we can use the crew of pairing $p^{''}$ to cover flight i^2 and all the following flights after i^2.



(a) Feasibility conditions of the flights in pairings $p^{'}$ and $p^{''}$ for Solution A.2

(b) Solution A.2 for extra flight k

Figure 3.2: Illustrative example of Solution A.2 on the flight network

Pairing	Feasibility Conditions		
$p^{'}$	$minSit \leq TI_{i^1k} \leq maxSit$	or	$minRest \leq TI_{i^1k} \leq maxRest$
	$\underline{SD}_{kj^2} \leq TI_{kj^2} \leq \overline{SD}_{kj^2}$	or	$underlineRD_{kj^2} \leq TI_{kj^2} \leq \overline{RD}_{kj^2}$
	$FT(i^1) + 2 * F_k + RFT(j^2) \leq maxFT$	and	$ET(i^1) + TI_{i^1j^2} + RET(j^2) \leq MaxTAFB$
$p^{''}$	$minSit \leq TI_{j^1i^2} \leq maxSit$	or	$minRest \leq TI_{j^1i^2} \leq maxRest$
	$FT(j^1) + RFT(i^2) \leq maxFT$	or	$ET(j^1) + TI_{j^1i^2} + RFT(i^2) \leq maxFT$

Table 3.3: Feasibility conditions for Solution A.2

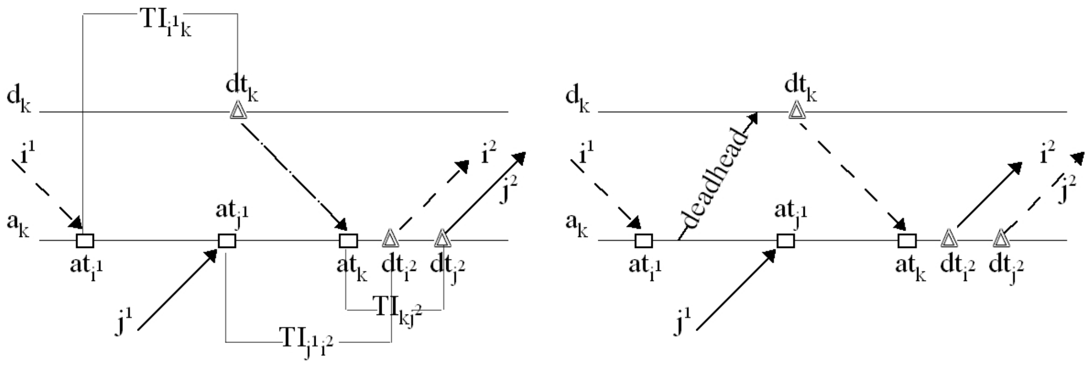
Solution A.3 : One Deadhead

If there is a pairing $p^{'}$ covering a flight arriving at d_k before dt_k, like i^1, which is the latest arrival before extra flight k and there is a pairing $p^{''}$ covering a flight departing from a_k after at_k like j^2, which is the first nearest departure after arrival of flight k as in Figure 3.3(a), we can say that these two pairings is a *A.3 solution* only if they satisfy feasibility conditions presented in Table 3.4.

Pairing	Feasibility Conditions		
$p^{'}$	$\underline{SD}_{i^1k} \leq TI_{i^1k} \leq \overline{RD}_{i^1k}$	or	$\underline{RD}_{i^1k} \leq TI_{i^1k} \leq \overline{RD}_{i^1k}$
	$minSit \leq TI_{kj^2} \leq maxSit$	or	$minRest \leq TI_{kj^2} \leq maxRest$
	$FT(i^1) + 2 * F_k + RFT(j^2) \leq maxFT$	and	$ET(i^1) + TI_{i^1j^2} + RET(j^2) \leq MaxTAFB$
$p^{''}$	$minSit \leq TI_{j^1i^2} \leq maxSit$	or	$minRest \leq TI_{j^1i^2} \leq maxRest$
	$FT(j^1) + RFT(i^2) \leq maxFT$	or	$ET(j^1) + TI_{j^1i^2} + RFT(i^2) \leq maxFT$

Table 3.4: Feasibility conditions for Solution A.3

As it can be seen in Figure 3.3(b), if there are two such pairings satisfying feasibility



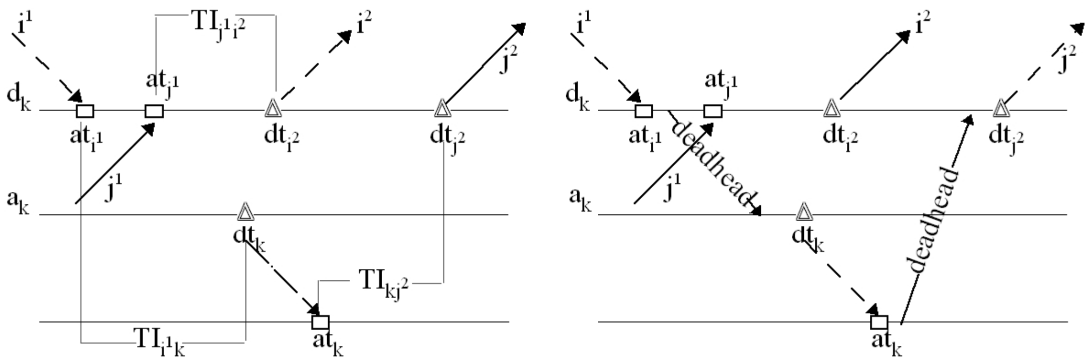
(a) Feasibility conditions of the flights in pairings $p^{'}$ and $p^{''}$ for Solution A.3

(b) Solution A.3 for the extra flight k

Figure 3.3: Illustrative example of Solution A.3 on the flight network

conditions for A.3, we can assign the crew of pairing $p^{'}$ to extra flight k and j^2, and all the following flights after j^2. And, we can use the crew of pairing $p^{''}$ to cover flight i^2, and all the following flights after i^2.

Solution A.4 : Two Deadheads



(a) Feasibility conditions of the flights in pairings $p^{'}$ and $p^{''}$ for Solution A.4

(b) Solution A.4 for extra flight k

Figure 3.4: Illustrative example of Solution A.4 on the flight network

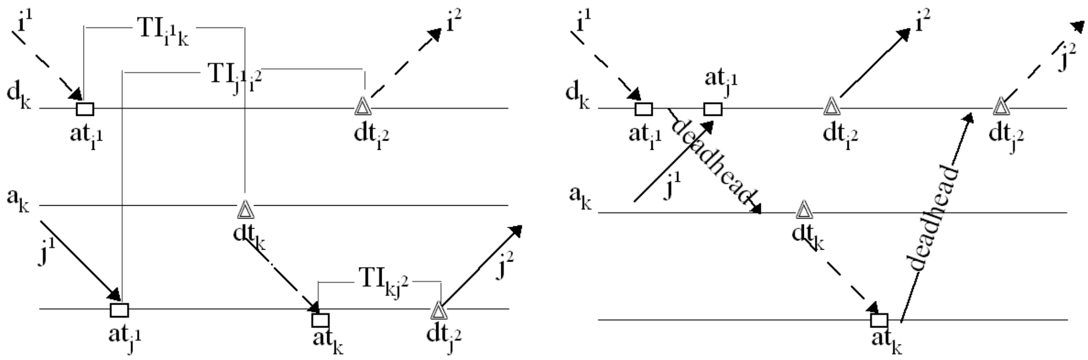
If there is a pairing $p^{'}$ covering a flight arriving at any station h other than d_k or a_k before dt_k, like i^1 which is the latest arrival before extra flight k and there is a pairing $p^{''}$ covering a flight departing from any station other than a_k or d_k after at_k, like j^2, which is the first nearest departure after arrival of flight k as in Figure 3.4(a), we can say that these two pairings is a *A.4 solution*, only if they satisfy feasibility conditions presented in Table 3.5.

Pairing	Feasibility Conditions		
$p^{'}$	$\underline{SD}_{i^1k} \leq TI_{i^1k} \leq \overline{RD}_{i^1k}$	or	$\underline{RD}_{i^1k} \leq TI_{i^1k} \leq \overline{RD}_{i^1k}$
	$\underline{SD}_{kj^2} \leq TI_{kj^2} \leq \overline{SD}_{kj^2}$	or	$\underline{RD}_{kj^2} \leq TI_{kj^2} \leq \overline{RD}_{kj^2}$
	$FT(i^1) + DT_{i^1k} + F_k + DT_{kj^3} + RFT(j^2) \leq maxFT$	and	$ET(i^1) + TI_{i^1j^2} + RET(j^2) \leq MaxTAFB$
$p^{''}$	$minSit \leq TI_{j^1i^2} \leq maxSit$	or	$minRest \leq TI_{j^1i^2} \leq maxRest$
	$FT(j^1) + RFT(i^2) \leq maxFT$	or	$ET(j^1) + TI_{j^1i^2} + RFT(i^2) \leq maxFT$

Table 3.5: Feasibility conditions for Solution A.4

As it can be seen in Figure 3.4(b), if there are two such pairings satisfying feasibility conditions for A.4, we can assign the crew of pairing $p^{'}$ to extra flight k and j^2 and all the following flights after j^2. And, we can use the crew of pairing $p^{''}$ to cover flight i^2 and all the following flights after i^2.

Solution A.5 : Two Deadheads



(a) Feasibility conditions of the flights in pairings $p^{'}$ and $p^{''}$ for Solution A.5

(b) Solution A.5 for extra flight k

Figure 3.5: Illustrative example of Solution A.5 on the flight network

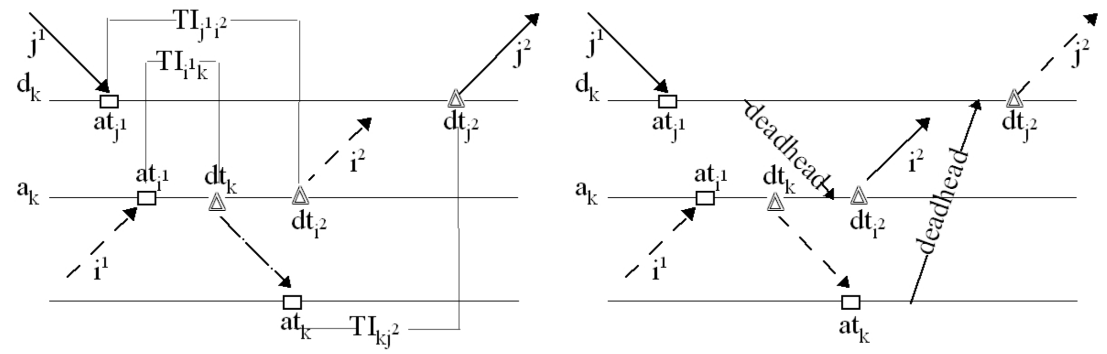
If there is a pairing $p^{'}$ covering a flight arriving at any station h other than d_k or a_k before dt_k, like i^1 which is the latest arrival before extra flight k and there is a pairing $p^{''}$ covering a flight departing from a_k after at_k, which is the first nearest departure after arrival of flight k as in Figure 3.5(a), we can say that these two pairings is a *A.5 solution*, only if they satisfy feasibility conditions presented in Table 3.6.

As it can be seen in Figure 3.5(b), if there are two such pairings satisfying solution A.5, we can assign the crew of the pairing $p^{'}$ to extra flight k and j^2 and all the following flights after j^2. And, we can use the crew of pairing $p^{''}$ to cover flight i^2 and all the following flights after i^2.

Pairing	Feasibility Conditions		
$p^{'}$	$\underline{SD}_{i^1k} \leq TI_{i^1k} \leq \overline{SD}_{i^1k}$	or	$\underline{RD}_{i^1k} \leq TI_{i^1k} \leq \overline{RD}_{i^1k}$
	$minSit \leq TI_{kj^2} \leq maxSit$	or	$minRest \leq TI_{kj^2} \leq maxRest$
	$FT(i^1) + DT_{i^1k} + F_k + RFT(j^2) \leq maxFT$	and	$ET(i^1) + TI_{i^1j^2} + RET(j^2) \leq MaxTAFB$
$p^{''}$	$\underline{SD}_{j^1i^2} \leq TI_{j^1i^2} \leq \overline{SD}_{j^1i^2}$	or	$\underline{RD}_{j^1i^2} \leq TI_{j^1i^2} \leq \overline{RD}_{j^1i^2}$
	$FT(j^1) + DT_{j^1i^2} + RFT(i^2) \leq maxFT$	or	$ET(j^1) + TI_{j^1i^2} + RFT(i^2) \leq maxFT$

Table 3.6: Feasibility conditions for Solution A.5

Solution A.6 : Two Deadheads



(a) Feasibility conditions of the flights in pairings $p^{'}$ and $p^{''}$ for Solution A.6

(b) Solution A.6 for extra flight k

Figure 3.6: Illustrative example of Solution A.6 on the flight network

If there is a pairing $p^{''}$ covering a flight arriving at any station h other than d_k or a_k before dt_k, like j^1, which is the latest arrival before extra flight k and there is a pairing $p^{'}$ covering a flight departing from a_k after at_k,which is the first nearest departure after arrival of flight k, like i_2, as in Figure 3.6(a), we can say that these two pairings is a *A.6 solution*, only if they satisfy feasibility conditions presented in Table 3.7.

Pairing	Feasibility Conditions		
$p^{'}$	$minSit \leq TI_{i^1k} \leq maxSit$	or	$minRest \leq TI_{i^1k} \leq maxRest$
	$\underline{SD}_{kj^2} \leq TI_{kj^2} \leq \overline{SD}_{kj^2}$	or	$\underline{RD}_{kj^2} \leq TI_{kj^2} \leq \overline{RD}_{kj^2}$
	$FT(i^1) + F_k + DT_{kj^2} + RFT(j^2) \leq maxFT$	and	$ET(i^1) + TI_{i^1j^2} + RFT(j^2) \leq maxFT$
$p^{''}$	$\underline{SD}_{j^1i^2} \leq TI_{j^1i^2} \leq \overline{SD}_{j^1i^2}$	or	$\underline{RD}_{j^1i^2} \leq TI_{j^1i^2} \leq \overline{RD}_{j^1i^2}$
	$FT(j^1) + DT_{j^1i^2} + RFT(i^2) \leq maxFT$	or	$ET(j^1) + TI_{j^1i^2} + RET(i^2) \leq MaxTAFB$

Table 3.7: Feasibility conditions for Solution A.6

As it can be seen in Figure 3.6(b), if there are two such pairings satisfying solution

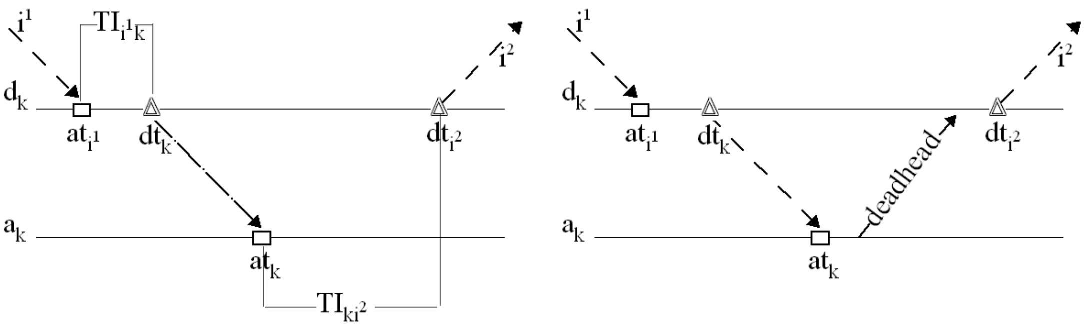
A.6, we can assign the crew of pairing p' to extra flight k and j^2 and all the following flights after j^2, we can use the crew of pairing p'' to cover flight i^2 and all the following flights after i^2.

3.2 Type B Solutions

Type B solutions require only one pairing. If the selected pairing has sufficient connection time to cover extra flight, then this pairing forms a *Type B* solution. Since the crew of the pairing continues its schedule, the remaining flights of the pairing after handling extra flight, are still covered ensuring that the maximum fly time is not exceeded.

There are three different *feasibility rules*. A pairing should satisfy at least one of these rules to compose a *Type B* solution. Therefore, there are three *Type B* solutions. One deadhead is required for the first two solutions and two deadheads are required for the last solution. In the following figures, the dashed lines show the pairing that forms a *Type B* solution and the dot-dashed lines show the extra flights. Moreover, the left subfigures indicate the feasibility conditions for a pairing to compose a solution, and the right subfigures show the solution for the extra flight.

Solution B.1 : One Deadhead



(a) Feasibility conditions of the flights in pairing p' for Solution B.1

(b) Solution B.1 for extra flight k

Figure 3.7: Illustrative example of Solution B.1 on the flight network

Suppose that p' covers two consecutive flights, flight i^1 and flight i^2 as in Figure 3.7(a). Pairing p' composes a *Type B* solution by satisfying the feasibility conditions stated in Table 3.8.

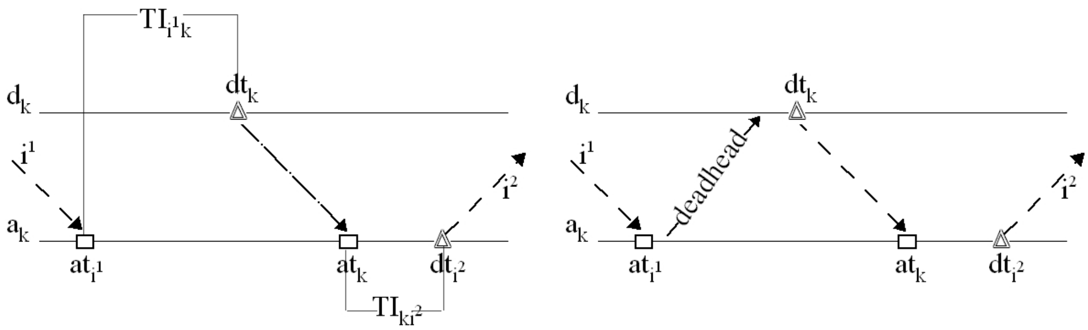
If there is a pairing $p^{'}$ covering a flight arriving at d_k before dt_k, like i^1, and a flight departing from d_k after at_k, like i^2, this pairing is a B.1 solution for extra flight k, if it satisfies the feasibility conditions presented in Table 3.8.

Pairing	Feasibility Conditions
$p^{'}$	$minSit \leq TI_{i^1k} \leq maxSit$ or $minRest \leq TI_{i^1k} \leq maxRest$
	$\underline{SD}_{ki^2} \leq TI_{ki^2} \leq \overline{SD}_{ki^2}$ or $\underline{RD}_{ki^2} \leq TI_{ki^2} \leq \overline{RD}_{ki^2}$
	$FT(i^1) + 2 * F_k + RFT(i^2) \leq maxFT$

Table 3.8: Feasibility conditions for Solution B.1

As it can be seen in Figure 3.7(b), if there are two such pairings satisfying solution B.1, we can assign the crew of pairing $p^{'}$ to extra flight k and cover the remaining flights.

Solution B.2 : One Deadhead



(a) Feasibility conditions of the flights in pairing $p^{'}$ for Solution B.2

(b) Solution B.2 for extra flight k

Figure 3.8: Illustrative example of Solution B.2 on the flight network

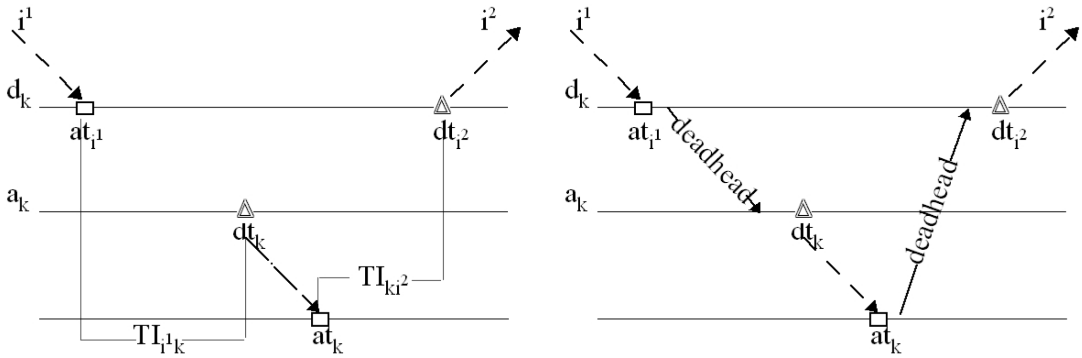
If there is a pairing $p^{'}$ covering a flight arriving at a_k before dt_k, like i^1, and a flight departing from a_k after at_k, like i^2, as in Figure 3.8(a), this pairing composes a B.2 solution for extra flight k, if it satisfies feasibility conditions presented in Table 3.9.

Pairing	Feasibility Conditions
$p^{'}$	$\underline{SD}_{i^1k} \leq TI_{i^1k} \leq \overline{SD}_{i^1k}$ or $\underline{RD}_{i^1k} \leq TI_{i^1k} \leq \overline{RD}_{i^1k}$
	$minSit \leq TI_{ki^2} \leq maxSit$ or $minRest \leq TI_{ki^2} \leq maxRest$
	$FT(i^1) + 2 * F_k + RFT(i^2) \leq maxFT$

Table 3.9: Feasibility conditions for Solution B.2

As it can be seen in Figure 3.8(b), if there are two such pairings satisfying solution B.2, we can assign the crew of pairing $p^{'}$ to extra flight k and cover the remaining flights.

Solution B.3 : Two Deadheads



(a) Feasibility conditions of the flights in pairing $p^{'}$ for Solution B.3

(b) Solution B.3 for extra flight k

Figure 3.9: Illustrative example of Solution B.3 on the flight network

If there is a pairing $p^{'}$ covering a flight arriving at a station h other than a_k and d_k before dt_k, like i^1, and covering a flight departing from a station h other than a_k and a_k after at_k, like i^2, as in Figure 3.9(a), then this pairing compose a B.3 solution, if it satisfies the feasibility conditions presented in Table 3.10.

Pairing	Feasibility Conditions
$p^{'}$	$\underline{SD}_{i^1k} \leq TI_{i^1k} \leq \overline{SD}_{i^1k}$ or $\underline{RD}_{i^1k} \leq TI_{i^1k} \leq \overline{RD}_{i^1k}$
	$\underline{SD}_{ki^2} \leq TI_{ki^2} \leq \overline{SD}_{ki^2}$ or $\underline{RD}_{ki^2} \leq TI_{ki^2} \leq \overline{RD}_{ki^2}$
	$FT(i^1) + DT_{i^1k} + F_k + DT_{ki^2} + RFT(i^2) \leq maxFT$

Table 3.10: Feasibility conditions for Solution B.3

As it can be seen in Figure 3.9(b), if there are two such pairings satisfying solution B.2, we can assign the crew of pairing $p^{'}$ to extra flight k and cover the remaining flights.

3.3 Mathematical Model

In the previous section, we explained the *feasibility conditions* for the pairings, which yield either *Type A* solution or *Type B* solution for extra flight k. Now, we introduce a new variable $t^k_{(p',p'')}$ which counts the number of *Type A* solutions for extra flight k. When a pairing, which satisfies the feasibility conditions for *Type B* solution, is selected, it composes a solution itself. Thus, we do not define a new variable to count *Type B* solutions. Instead we define a set $\tilde{P}_k$, which consist of the pairings satisfying one of the feasibility conditions of *Type B* solutions for extra flight k. Our objective is to maximize the total number of solutions to cover the extra flights. That is,

$$\max \sum_{k\in K} \sum_{(p',p'')\in \mathbf{P}_k} t^k_{(p',p'')} + \sum_{k\in K} \sum_{p\in \tilde{P}_k} y_p.$$

We assume that the crew schedule, which contains large number of solutions for each extra flight, yields a more robust schedule. However, there is a trade-off between maximizing the number of solutions for the extra flights and the cost of the schedule. Hence, we first solve the traditional crew pairing problem and find the least costly optimal solution. Then, we add a constraint such that it ensures that the increase in the cost is within an acceptable level.

Before presenting the model, we first introduce the parameters and then decision variables. The model parameters are as follows:

K	:	Set of all possible extra flights,
$\mathbf{P}_k$	:	Set of all pairings that yield a Type A solution for extra flight $k \in K$,
$\tilde{P}_k$	:	Set of all pairings that yield a Type B solution for extra flight $k \in K$,
P	:	Set of all pairings,
F	:	Set of all scheduled flights,
P_i	:	Set of all pairing covering flight $i \in F$.
r	:	Robustness factor which represents how much extra cost can be absorbed.

The decision variables, on the other hand, are given as below:

$t^k_{(p',p'')}$	:	1 if both p' and p'' are selected where $(p',p'') \in \mathbf{P}_k$; 0 otherwise,
y_p	:	1 if pairing $p \in P$ is selected; 0 otherwise.

The robust crew pairing model that we consider then becomes

$$\begin{array}{lll}
\max & \sum_{k\in K} \sum_{(p',p'')\in \mathbf{P}_k} t^k_{(p',p'')} + \sum_{k\in K} \sum_{p\in \tilde{P}_k} y_p & \\
\text{s.t} & \sum_{p\in P_i} y_p = 1, & i \in F, \\
& 2t^k_{(p',p'')} \leq y_{p'} + y_{p''}, & (p',p'') \in \mathbf{P}_k, p \in P, k \in K, \\
& \sum_{p\in P} c_p y_p \leq (1+r)C_{opt}, & \\
& t_{(p',p'')} \in \{0,1\}, & (p',p'') \in \mathbf{P}_k, k \in K, \\
& y_p \in \{0,1\}, & p \in P.
\end{array} \tag{3.1}$$

The objective function aims at maximizing the total number of solutions for all extra flights. The first set of constraints are the standard set partitioning constraints which ensure to cover all scheduled flights once. The second constraints count the number of *Type A* solutions for extra flight k. Recall that the pairings which satisfy the feasibility rules should be selected together. The last constraint is used to control the cost of the schedule. Here, C_{opt} is the optimal solution of the conventional crew pairing problem (See problem (1.1)).

3.4 Extensions

Our model does not have any constraint to prevent a pairing to be a part of several solutions within *Type A* or *Type B*. If two solution types for the same flight has common pairings, this issue is not important. In practice, the decision makers can choose one of the solutions suitable for recovery. However, if a pairing is a part of a solution to more than one extra flight, this issue becomes important especially if both extra flights occur on the same day. This issue is called the double counting problem. Figure 3.10 shows an example of double counting for a pairing. In the figure, there are three pairings, A, B and C that cover flights a_1-a_2, b_1-b_2 and c_1-c_2, respectively. There are two extra flights, namely k_1 and k_2. Pairing A forms a solution for extra flight k_1 with pairing C. On the other hand, pairing B forms a solution for extra pairing k_2 with pairing C. If two extra flights are added on the same day, we can only use pairing C to cover only one extra flight. Therefore, although the number of solutions seem to be two, in practice it is only one.

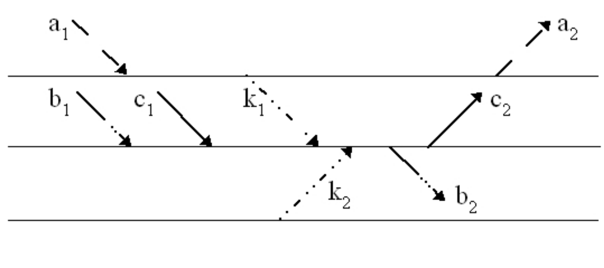


Figure 3.10: Illustration of double counting problem

If one requires that a paring can be only in one extra flight solution, then a new binary variable ${z_p}^k$ should be defined such that ${z_p}^k$ is equal to 1, if the pairing p yields a solution for extra flight k, 0 otherwise. Consequently, the following constraints should be added to the proposed model

$$\begin{aligned} &\sum_{k\in K} z_p^k \leq 1, && p\in P, \\ &z_p^k \leq \sum_{(p',p'')\in \mathbf{P}} (t^k_{(p,p'')} + t^k_{(p',p)}) + \sum_{p\in \tilde{P}_k} y_p, && p\in P, (p,p'')\in \mathbf{P}, (p',p)\in \mathbf{P}, k\in K, \end{aligned} \tag{3.2}$$

and the sum of z_p^k over each extra flight k should be added to the objective function. The first set of constraints in (3.2) guarantees that each pairing can be a solution for at most one extra flight. The second set of constraints in (3.2) ensures that if the pairing p is not used in any solution type for extra flight k, then the corresponding variable z_p^k becomes zero. Since we add the sum of the variables z_p^k to the objective function, this forces z_p^k to be equal to one if it is used at least in one solution type for extra flight k.

The overall problem becomes,

$$\begin{aligned} \max \quad & \sum_{k\in K} \sum_{(p',p'')\in \mathbf{P}_k} t^k_{(p',p'')} + \sum_{k\in K} \sum_{p\in \tilde{P}_k} y_p + z_p^k && \\ \text{s.t} \quad & \sum_{p\in P_i} y_p = 1, && i\in F, \\ & 2t^k_{(p,p'')} \leq y_{p'} + y_{p''}, && (p,p'')\in \mathbf{P}_k, p\in P, k\in K, \\ & \sum_{p\in P} c_p y_p \leq (1+r)C_{opt}, && \\ & \sum_{k\in K} z_p^k \leq 1, && p\in P, \\ & z_p^k \leq \sum_{(p',p'')\in \mathbf{P}} (t^k_{(p,p'')} + t^k_{(p',p)}) + \sum_{p\in \tilde{P}_k} y_p, && p\in P, (p,p'')\in \mathbf{P}, (p',p)\in \mathbf{P}, k\in K, \\ & t_{(p,p'')} \in \{0,1\}, && (p,p'')\in \mathbf{P}_k, k\in K, \\ & y_p \in \{0,1\}, && p\in P. \end{aligned} \tag{3.3}$$

We can also use weights for the extra flights. These weights can be calculated by taking into account the possibility of adding that flight and the importance of covering that flight. Then, the objective of our problem becomes maximization of the weighted sum of possible coverings.

If one wants to make the solutions evenly distributed among the extra flights, then another variable should be added to the model. Let z be the maximum number of solutions for all extra flights, and S_k be the total number of solutions for each extra flight k. Then, $z = \min\{S_k : \quad k\in K\}$. To calculate the z value, the following constraints should be added

$$\begin{aligned} &\sum_{(p',p'')\in \mathbf{P}} t^k_{(p',p'')} + \sum_{y_p\in \tilde{P}_k} y_p = S_k, && k\in K, \\ &z \leq S_k, && k\in K, \end{aligned} \tag{3.4}$$

and a positive multiple of z value should be added to the objective function. Then problem (3.1) becomes

$$
\begin{aligned}
\max \quad & \sum_{k\in K} \sum_{(p',p'')\in \mathbf{P}_k} t^k_{(p',p'')} + \sum_{k\in K} \sum_{p\in \tilde{P}_k} y_p + z \\
\text{s.t} \quad & \sum_{p\in P_i} y_p = 1, && i \in F, \\
& 2t^k_{(p,p'')} \le y_{p'} + y_{p''}, && (p,p^{''}) \in \mathbf{P}_k, p\in P, k\in K, \\
& \sum_{p\in P} c_p y_p \le (1+r)C_{opt}, && \\
& \sum_{(p',p'')\in \mathbf{P}} t^k_{(p',p'')} + \sum_{y_p\in \tilde{P}_k} y_p = S_k, && k\in K, \\
& z \le S_k, && k \in K, \\
& t_{(p,p'')} \in \{0,1\}, && (p,p^{''}) \in \mathbf{P}_k, k\in K, \\
& y_p \in \{0,1\}, && p \in P.
\end{aligned}
\tag{3.5}
$$

CHAPTER 4

COMPUTATIONAL RESULTS

In this study, we solve two illustrative problems. Usually, the local companies have a small sized flight network. Therefore, in our examples it is possible to generate all feasible pairings. To generate the pairings, we wrote a C++ code. We used the depth-first search algorithm on the flight network. On the flight network, the arrival and the departure of each flight are denoted by nodes, and a flight is denoted by an arc from the corresponding departure node to the matching arrival node. The possible connections are also represented as arcs. All the departure nodes from the crew base are connected to the source node, s and all the arrival nodes to the crew base are connected to the sink node, t. For each crew base, all feasible pairings are enumerated. OPL 4.1 ([28]) is used to solve the conventional problem 1.1 and the proposed crew pairing problem 3.1.

Then, we wrote another code to preprocess all feasible pairings to find all *pairing tuples* which together yield a Type A solution for each possible extra flight, and all the pairings which yield Type B solution for each possible extra flight. In after wards, we generate the sets $\mathbf{P}_k$ and $\tilde{P}_k$ for each $k \in K$.

To compare the proposed model and the conventional model with respect to the robustness, we solved two representative problems. Then, we used the recovery methods generally used in local companies, and compared both the solutions in terms of flight delays and cancellation caused by the corresponding solution.

For our computational experiments, two representative problems have formed with actual data. For the first problem, the case of extra flight demanded from the customer was considered. For the second problem, the case of extra flight added by the company was considered. For each problem, the corresponding model was solved for two extra flights, and the results were compared with the conventional crew pairing problem solution. Since all the flights are flown every day, solving weekly exception problem is not required in our computational experiments.

4.1 Extra Flights Demanded by the Customers

In this problem, the company has 38 flights and six airplanes. The company uses two crew bases, which have four and two airplanes, respectively. The company flies from and to four cities; Istanbul, Ankara, Izmir and Antalya. The flight data can be found at the appendix A, Table A.1. The company uses one-day pairings to cover its flights. Due to their past experiences, the company predicts that an extra flight from Ankara to Istanbul at 15:00 can be demanded from the company. We denote this extra flight by k_1 in the subsequent part of this section. Moreover, during the summer season, an extra flight from Istanbul to Izmir at 16:30 can also be demanded. This extra flight is denoted by k_2. Covering these extra flights has a priority for its reputation. Since it is not certain when those flights are demanded, the company wants to construct its pairings without considering those flights. And, when these extra flights are demanded, the company allow breaking a pairing that can cover the extra flight. If there is no airplane covering the extra flight without causing a delay of scheduled flights, some scheduled flights can be delayed. If this delay is above a certain value, the company cancels these flights. If covering the extra flight results in exceeding maximum fly time, then some scheduled flights should be canceled. For this problem, the company cancels a flight when this flight is delayed for at least two hours.

We first generated all the one-day pairings. The total number of pairings from Istanbul and Ankara bases were 309 and 209, respectively. By using OPL, we solved the conventional crew pairing problem and found $C_{opt} = 4722$. The pairings in the optimal solution are given in Table 4.1. The network representations of the pairings found by the conventional crew pairing problem is given in appendix B.

Pairing ID	Flights ID						
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 4.1: Pairings found by the conventional CPP for the first problem

Then, we applied recovery procedure described above to find a solution for those extra flights. None of the pairings can cover the extra flights without delaying or canceling other flights. The solutions can be as follows:

- Break Pairing 7: If this pairing is used to cover extra flight k_1, then the flights 15 and 19 should be canceled and the crew should deadhead from Istanbul to Ankara. If this pairing is used for the second extra flight, the flight 21 should be canceled since it is delayed 150 minutes. If both extra flights are covered by this solution, again the flights 15 and 19 should be canceled.

- Break Pairing 64: The crew in this pairing flies eight hours, therefore if it is desired to cover the extra flights, then at least one flight should be canceled. For extra flight k_1, the flights 16 and 20 should be canceled and the crew should deadhead from Izmir to Istanbul, and for extra flight k_2, the flights 34 and 26 should be canceled and the crew should deadhead from Istanbul to Ankara. If both extra flights are covered by this pairing, the flights 16, 20 and 34 should be canceled and a deadhead from Izmir to Ankara is required.

- Break Pairing 160: To cover extra flight k_1, the flights 8 and 11 should be canceled, in the case of extra flight k_2, the flights 8 and 11 should be canceled and the crew should deadhead from Izmir to Istanbul. To cover both extra flights requires cancellation of the flights 8 and 11 and deadheading from Izmir to Istanbul.

- Break Pairing 186: When k_1 is covered with this pairing, flight 30 is delayed 120 minutes, however, since the maximum total flying time is exceed when extra flights are then flied by the crew of this pairing, flight 30 should be canceled. To cover k_2, flight 30 should be canceled and a deadhead from Ankara to Istanbul is required. Covering both flights results in exceeding the maximum total flying hour, and hence, flight 30 should canceled.

- Break Pairing 334: The pairing can cover extra flight k_1 if flight 35 is canceled, and the pairing can cover extra flight k_2 if flight 23 is canceled. Although flight 23 is delayed for 120 hours, which is in the acceptable level, total flying hour exceeds the maximum amount. Therefore, flight 23 should be canceled. Both extra flights are covered only if flights 35 and 23, are canceled.

- Break Pairing 412: Extra flight k_1 can be covered if flight 33 is canceled, and flight k_2 is covered if flight 14 is canceled. The reason for cancellation is again exceeding the maximum flying time. To cover both extra flights, flights 33 and 14 should be canceled.

We then solved our proposed model for the same problem data. We allowed the cost of the solution to be at most 25% above the optimal cost found by the conventional CPP. Our model provides one solution for each extra flight. The solution of our model is given in Table 4.2. The network representations of the pairings found by the conventional crew pairing problem is given in appendix C.

Pairing ID	Flights ID					
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 4.2: Pairings found by the proposed model for the first problem

For extra flight k_1, the solution generated by our model satisfies the first feasibility condition of Type A solution. The involved pairings are 192 and 67. The original pairings and flight network representation of the solution is given in appendix D, Figure D.1. Pairing 192 covers its first three flights and then covers extra flight k_1. Then, it covers flights 21, 5 and 38, which were originally flown by pairing 67. Pairing 67 deadheads from Istanbul to Ankara after flying its first three flights, and then continues with flights 34, 26 and 36, which were originally flown by pairing 192.

For the second extra flight, the solution is again a Type A solution but in this case, the pairings satisfy the second feasibility condition of Type A solutions. The involved pairings are 5 and 160. The network representations of the original pairings and the solution are given in appendix D, Figure D.2. The pairing 5 covers its first four flights, and then flies the extra flight k_2. Then it deadheads from Izmir to Istanbul to cover flights 25 and 24, which were originally flown by pairing 160. Pairing 160 covers flights 4,7,8,11,14 and 18, last two of which were originally flown by pairing 5.

For this problem, if the company uses the conventional model to construct its pairings, the managers should cancel two flights to cover the extra flights. On the other hand, the solution of our model makes it possible to cover those extra flights without canceling or delaying any other flight. The cost of the solution of our model is 4837 minutes, which is approximately 2.5% above the optimal solution found by the conventional model.

4.2 Extra Flights Added by the Company

The second problem illustrates another issue faced by the local companies. In this case, the company wants to add an extra flights due to for instance high demand. A problem instance with 58 flights is formed from the actual data. The flight data of this problem is given in the appendix E, Table 4.3. In this representative problem, there are two flights with high demand. When the demand is high, the company consider adding an extra flight at a time close to the original departure time. The company, however, does not want to delay or cancel any other flight. That is, the company adds the extra flights if all the resources are available for the flight. The company has one crew base and it has ten airplanes. The company from and to Istanbul, Ankara, Izmir, Antalya, Adana, Dalaman, and Bodrum. The company uses one-day pairings. The high demanded flights are from Istanbul to Ankara at 15:00, and from Istanbul to Antalya at 11:25. For those flights, the company considers adding an extra flight at 15:10 to Ankara and 11:50 to Antalya. We denote the first extra flight by k_1, and the second extra flight by k_2.

Firstly, we generated all the pairings for this problem. In total, there are 2213 pairings. Then, the conventional crew pairing model is solved. The optimal pairing cost is 7089 minutes. The minimum cost pairings and the flights covered by these pairings are given in Table 4.3. The network representations of the pairings found by the conventional crew pairing problem is given in appendix F. For the solution found by the conventional CPP, there is no way of adding the extra flights without delaying or canceling the other flights. Therefore, the company can not schedule those extra flights.

Pairing ID	Flights ID							
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 4.3: Pairings found by the conventional CPP for the second problem

We then solved our proposed model (3.1) for the same problem data. We allowed the cost of the solution to be at most 20 % above the optimal cost found by the conventional CPP. The pairings which maximize the total solutions is given in Table 4.4. The network representations of the pairings found by the conventional crew pairing problem is given in appendix G. Our model provides one Type A solution for the first extra flight and four type A solutions for the second extra flight. The total cost of solution found by our model is 7209 minutes. Again, it is slightly higher than the optimum cost (approximately 1.7%).

Pairing ID	Flights ID							
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		
909	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 4.4: Pairings found by the proposed model for the second problem

For extra flight k_1, there is only one solution. This solution is represented in appendix H, Figure H.1. In this solution, the pairings form a *Type A* solution while satisfying the first feasibility condition, Solution A.1. Pairing 2108 flies its first two scheduled flights, and then covers the extra flight. Then, it flies flights 21, 15 and 25, which were originally flown by pairing 1039. Pairing 1039 covers its first three flights and deadheads from Ankara to Istanbul to fly flights 14 and 24, which were originally flown by pairing 2108.

For extra flight k_2, there are four solutions. Since we do not restrict our model to assign a pairing to multiple solutions, the same pairing appears in more than one solution. Since there is at least one solution, which is composed of different pairings, double counting of the same pairing is not a issue for our solution. However, it is also possible to model the same problem by adding some constraints as in (3.5) to prevent double counting.

There are two *Type A* solutions which satisfy the first feasibility condition, Solution A.1. This solution is given in appendix H, Figure H.2. In this solution, pairing 91 firstly covers its first two flights, and then flies extra flight k_2. It then continues flights 31, 47, and 56, which were originally flown by pairing 970. Pairing 970 flies its first flight, and then deadheads from Antalya to Istanbul to cover flights 46, 55, 48 and 57, which were originally flown by pairing 91. The second Solution A.1 for extra flight k_1 is formed by pairings 1039 and 970, as given in appendix H, Figure H.3. Pairing 1039 flies its first two flights, and then covers extra flight k_2. After it flies flights 31, 47, and 56, which were originally flown by 970 before swapping. Pairing 970 firstly covers its first flights and then deadheads from Antalya to Istanbul and flies flights 11,21,15 and 25, which were originally flown by pairing 1039 before swapping.

For extra flight k_2, there are also two solutions (both Solution A.2). The first one is given in appendix H, Figure H.4. The solution is composed of pairings 91 and 2108. The pairing 91 flies its first two flights, and then covers extra flight k_2. After that, it deadheads from Antalya to Istanbul to cover flights 14 and 24, which were originally flown by pairing 2108. Pairing 2108 flies flights 44, 53, 46, 55, 48 and 57, last four of which were originally flown by pairing 91 before swapping. The second Solution A.2 is composed of pairings 1039 and 2108, which are shown in appendix H, Figure H.5. Pairing 1039 flies its first two flights, and then covers extra flight k_2. After that, it deadheads from Antalya to Istanbul to cover flights 14 and 24, which were originally flown by pairing 2108. Pairing 2108 flies flights 44, 53, 11, 21, 15 and 25, last four of which were originally flown by pairing 91 before swapping.

CHAPTER 5

CONCLUSION AND FUTURE RESEARCH

In this study, we show that a robust crew schedule can be built at the planning stage to manage the potential extra flights that may be added to the flight schedule irregularly. We conclude that when an extra flight is added, the solution of the robust model provides less operational cost than the schedule found by solving the conventional crew pairing problem.

The proposed robust model adds new constraints to the conventional model. Therefore, powerful techniques, like delayed column generation, cannot be used easily. One possible remedy in this case is to use Lagrangean relaxation methods. In this thesis, the well-known column generation methods are not considered. As a future research, we intend to study column generation along with Lagrangean relaxation methods.

We conducted experiments on relatively small networks and reported our results for one-day pairing problems. In case of two-days pairings and large networks, the number of variables in the resulting problem increases exponentially. Due to the structure of the proposed model, not only we have huge number of variables but also huge number of constraints. An efficient way is certainly required to handle the large amount of data. We reserve this issue for our future research as well.

Bibliography

[1] Anbil, R., Gelman, E., Patty, B. and Tanga, R., Recent advances in crew pairing optimization at American Airlines, Interfaces, 21, 62-74, 1991.

[2] Anbil, R., Tanga, R., and Johnson, E.L., A Global approach to crew-pairing optimization, IBM Systems Journal, 31(1), 71-78, 1992.

[3] Anbil, R., Forrest, J.J. and Pulleyblank, W.R., Column generation and the airline crew pairing problem, Documenta Mathematica, Extra Volume ICM (3), 677-686, 1998.

[4] Barnhart, C., Johnson, E.L., Nemhauser, G., Savelsbergh, M. and Vance, P.H., Branch-and-Price: Column generation for solving huge integer programs, Operations Research, 46(3), 316-329, 1998.

[5] Barnhart, C., Hatay, L. and Jonson, E.L., Deadhead selection for the long-haul crew pairing problem, Operations Research, 43(3), 491-499, 1995.

[6] Barnhart, C. and Shenoi R.G., An approximate model and solution approach for the long-haul crew pairing problem, Transportation Science, 32(3), 221-231, 1998.

[7] Barnhart, C., Cohn, A.M., Johnson, E.L., Klabjan, D., Nemhauser, G.L. and Vance, P.H., Airline crew scheduling, In R. W. Hall, editor, Handbook of Transportation Science, Kluwer Scientific Publishers, 2003.

[8] Bixby, R., Gregory, J.W., Lustig, I.J., Marsten, R. and Shanno, D., Very large scale linear programming: A case study in combining interior point and simplex methods, Operations Research, 40, 885-897, 1992.

[9] Crainic, G. and Rousseau, J., The column generation principle and the airline crew scheduling problem, Informs, 25(2), 136-151, 1987.

[10] Desrochers, M. and Soumis, F., A column generation approach to the urban transit crew scheduling problem, Transportation Science, 23(1), 1989.

[11] Desrosiers, J., Stojkovic, M. and Soumis, F., The operational airline crew scheduling problem, Transportation Science, 32(3), 232-245, 1998.

[12] Ehrgott, M. and Ryan, D., Constructing robust crew schedules with bicriteria optimization, Journal of Multi-Criteria Decision Analysis, 11, 139-150, 2002.

[13] Hoffman, K.L. and Padberg, M., Solving airline crew scheduling problems by branch-and-cut, Management Science, 39(6),657-682, 1993.

[14] Hoffman, K.L., Padberg, M. and Rushmeier, R.A., Recent advances in exact optimization of airline scheduling problems, 1995.

[15] Housos, E. and Elmroth, T., Automatic optimization of subproblems in scheduling airline crews, Interfaces, 27, 68-77, 1997.

[16] Johnson, E.L. and Gopalakrishnan, B., Airline crew scheduling: State-of-the-Art, Annals of Operations Research, 140, 305-337, 2005.

[17] Martin, Richard Kipp, Large scale linear and integer optimization : a unified approach, Kluwer Academic Publishers, 1999.

[18] Klabjan, D., Johnson, E.L., Nemhauser, G., Gelman, E. and Ramaswamy, S., Solving large airline crew scheduling problems: Random pairing generation and strong branching, Computational Optimization and Applications, 20, 73-91, 2001.

[19] Klabjan, D., and Makri, A., A new pricing scheme for airline crew scheduling, Informs Journal on Computing, 16(1), 56-67, 2004.

[20] Klabjan, Diego, Large-scale models in the airline industry, In G. Desaulniers, J. Desrosiers, M.M. Solomon, editors, Column Generation, Kluwer Academic Publishers, 2003.

[21] Lettovský, L., Johnson, E.L., and Nemhauser, G., Airline crew recovery, Transportation Science, 34(4), 337-348, 2000.

[22] Ryan, D.M. and Foster, B., An integer programming approach to scheduling, In A. Wren, editor, Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling, North-Holland Publishing Company, 269-280, 1981.

[23] Schaefer, A., Johnson, E.L., Kleywegt, A., and Nemhauser, G., Airline crew scheduling under uncertainty, Transportation Science, 39(3), 340-348, 2005.

[24] Shebalov, S. and Klabjan, D., Robust airline crew pairing: Move-up crews, To appear in Transportation Science, 2004, `https://netfiles.uiuc.edu/klabjan/www/articles/robust_cs.pdf`.

[25] Vance, P.H., Barnhart, C., Johnson, E.L. and Nemhauser G.L., Airline crew scheduling: A new formulation and decomposition algorithm, Operations Research, 45(2), 188-200, 1997.

[26] Vance, P.H., Atamtürk, A., Barnhart, C., Gelman, E., Johnson, E., Krishna, A., Mahidhara, D., Nemhauser, G. and Rebello, R., A heuristic branch-and-price approach for the airline crew pairing problem, 1997, `http://citeseer.ist.psu.edu/vance97heuristic.html`.

[27] Yen, J. and Birge, J., A stochastic programming approach to the airline crew scheduling problem, Technical Report, University of Washington, 2000, `http://users.iems.northwestern.edu/~jrbirge/html/misc_papers/OR_crew_scheduling3.pdf`.

[28] ILOG Software, `http://www.ilog.com`, July 2006.

Appendix A

First Problem Flight Data

Flight ID	Or-Des	DT-AT	Flight ID	DT-AT	DT-AT
1	Ist-Ank	07:00-08:00	**20**	Ist-Ank	15:00-16:00
2	Ist-Iz	06:00-07:00	**21**	Ist-Iz	17:00-18:00
3	Iz-Ank	10:05-11:20	**22**	Iz-Ist	19:20-20:20
4	Ist-Ant	08:25-09:40	**23**	Ist-Ank	17:00-18:00
5	Iz-Ank	19:20-20:40	**24**	Iz-Ist	22:00-23:00
6	Iz-Ist	09:00-10:00	**25**	Ist-Iz	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-Iz	09:00-10:00	**28**	Ist-Ank	23:45-00:45
10	Ist-Iz	11:00-12:00	**29**	Ank-Iz	07:45-09:05
11	Ant-Ist	16:50-18:05	**30**	Ank-Iz	17:00-18:20
12	Iz-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	19:00-20:00	**33**	Ank-Ist	14:00-15:00
15	Ist-Iz	13:00-14:00	**34**	Ank-Ist	17:00-18:00
16	Iz-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	Iz-Ist	15:00-16:00	**38**	Ank-Ist	22:00-23:00

Table A.1: Flight data for the first problem

Appendix B

First Problem Pairings - Conventional Model

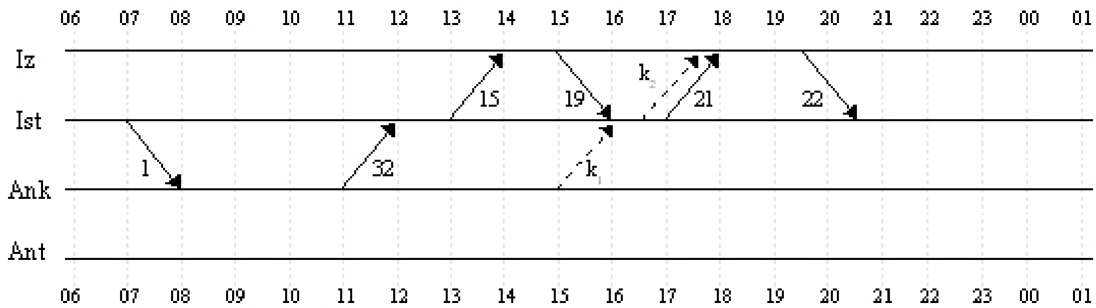


Figure B.1: Flights covered by pairing 7

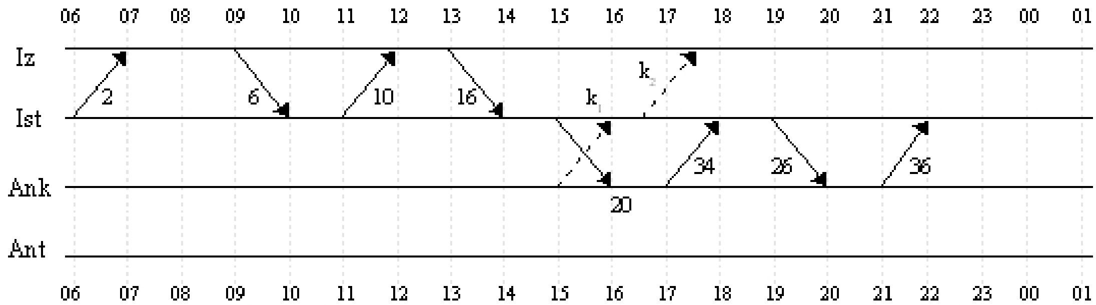


Figure B.2: Flights covered by pairing 64

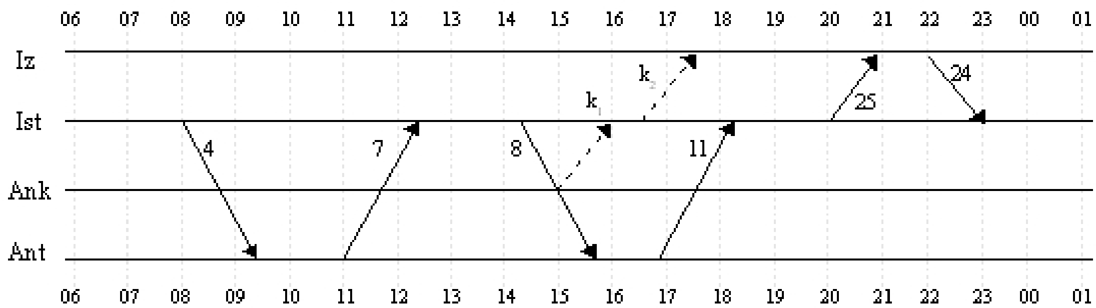


Figure B.3: Flights covered by pairing 160

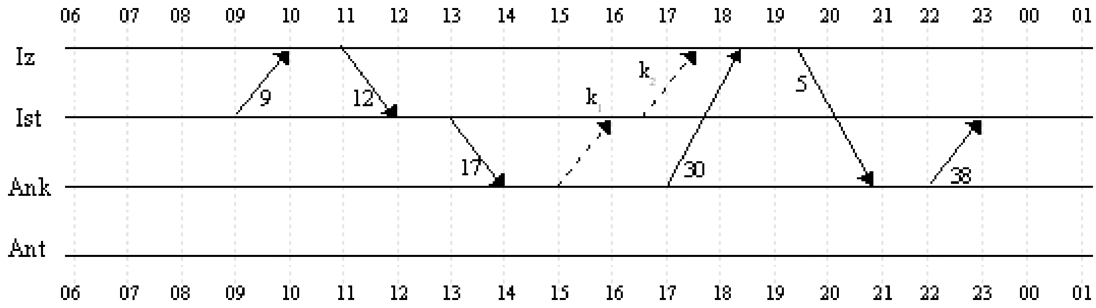


Figure B.4: Flights covered by pairing 186

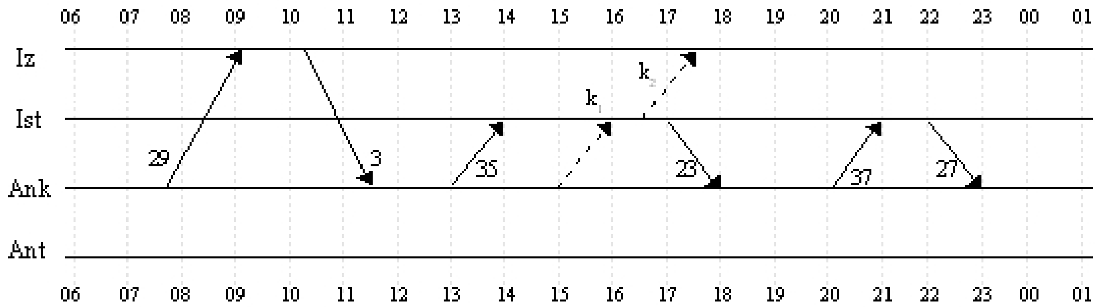


Figure B.5: Flights covered by pairing 334

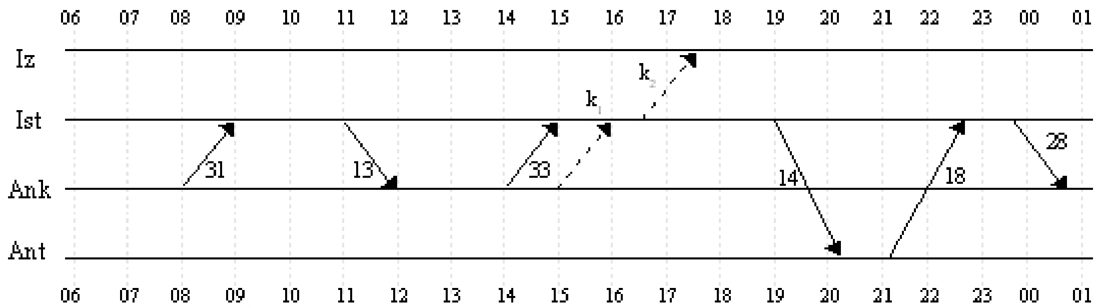


Figure B.6: Flights covered by pairing 412

Appendix C

First Problem Pairings - Proposed Model

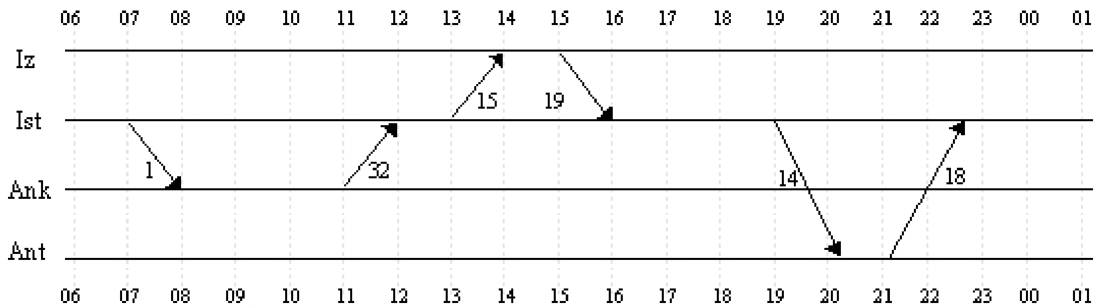


Figure C.1: Flights covered by pairing 5

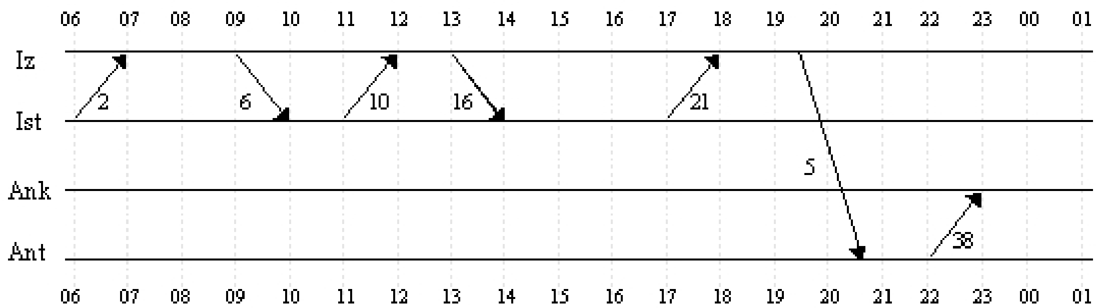


Figure C.2: Flights covered by pairing 67

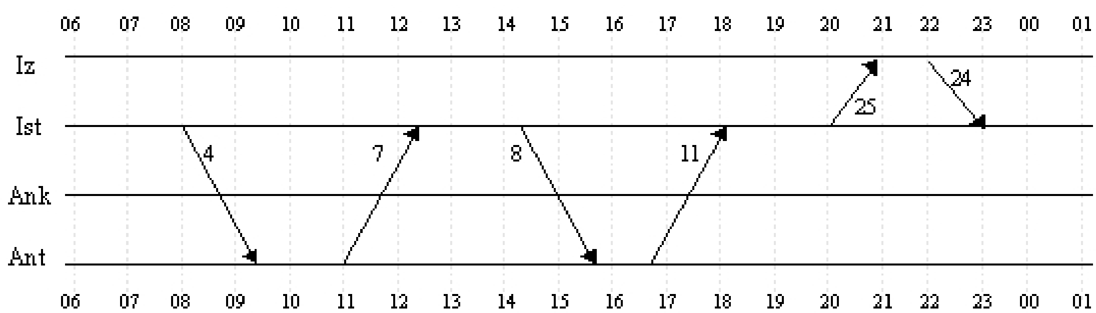


Figure C.3: Flights covered by pairing 160

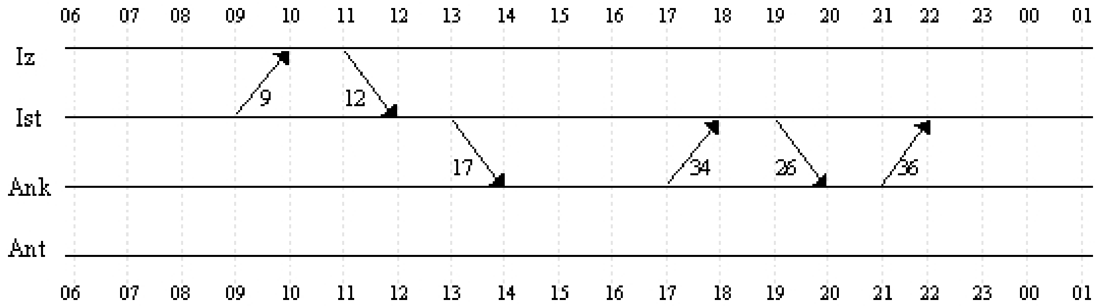


Figure C.4: Flights covered by pairing 192

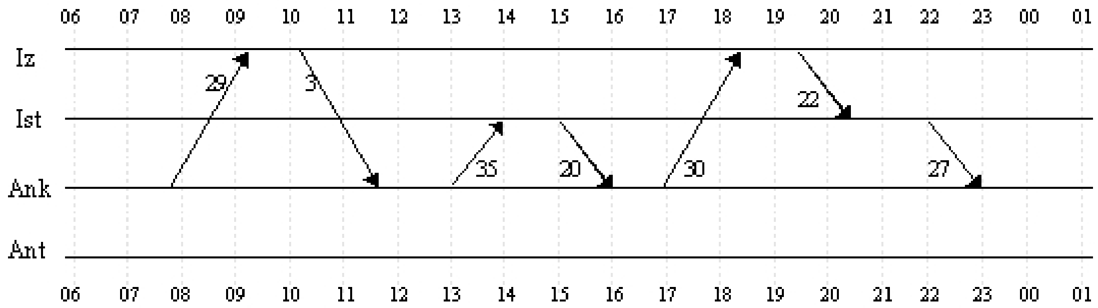


Figure C.5: Flights covered by pairing 323

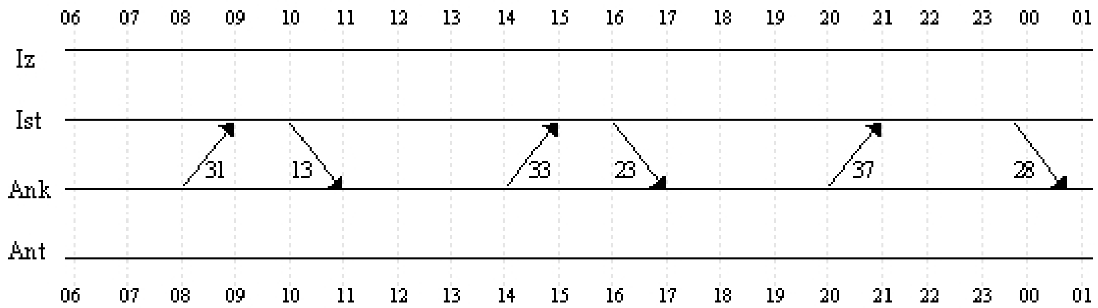


Figure C.6: Flights covered by pairing 419

Appendix D

First Problem Solutions

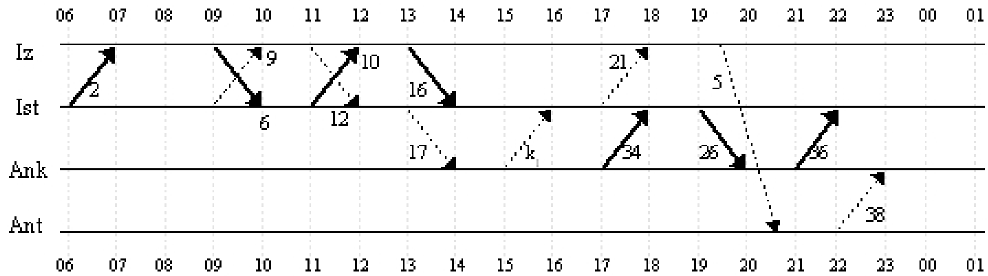


Figure D.1: Solution for extra flight k_1 by swapping pairings 67 and 192

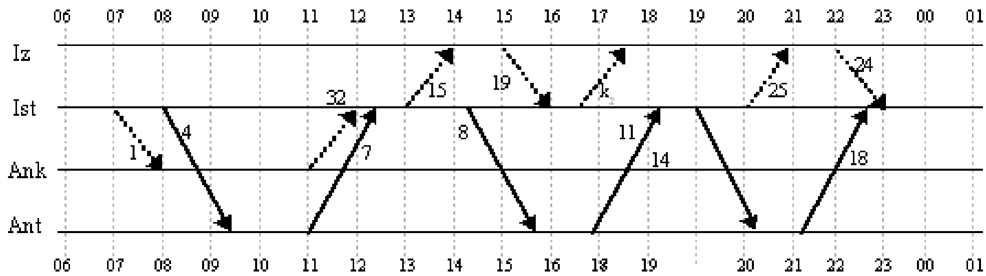


Figure D.2: Solution for extra flight k_2 by swapping pairings 5 and 160

Appendix E

Second Problem Flight Data

Flight ID	Or-Des	DT-AT	Flight ID	DT-AT	DT-AT
1	Ist-Ad	07:00-08:35	**30**	Ant-Ist	10:40-11:55
2	Ist-Ad	11:15-12:50	**31**	Ant-Ist	13:40-14:55
3	Ist-Ad	14:15-15:50	**32**	Ant-Ist	16:45-18:00
4	Ad-Ist	09:35-11:10	**33**	Ist-Bod	14:30-15:40
5	Ad-Ist	13:50-15:25	**34**	Ist-Bod	19:30-20:40
6	Ad-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-Iz	07:00-08:00
13	Ist-Ank	17:30-18:30	**42**	Ist-Iz	08:00-09:00
14	Ist-Ank	18:00-19:00	**43**	Ist-Iz	08:30-09:30
15	Ist-Ank	19:00-20:00	**44**	Ist-Iz	11:00-12:00
16	Ist-Ank	20:00-21:00	**45**	Ist-Iz	13:00-14:00
17	Ank-Ist	09:00-10:00	**46**	Ist-Iz	15:00-16:00
18	Ank-Ist	11:00-12:00	**47**	Ist-Iz	16:00-17:00
19	Ank-Ist	12:00-13:00	**48**	Ist-Iz	20:00-21:00
20	Ank-Ist	15:00-16:00	**49**	Ist-Iz	23:45-00:45
21	Ank-Ist	17:00-18:00	**50**	Iz-Ist	10:00-11:00
22	Ank-Ist	19:00-20:00	**51**	Iz-Ist	10:30-11:30
23	Ank-Ist	19:30-20:30	**52**	Iz-Ist	11:00-12:00
24	Ank-Ist	20:00-21:00	**53**	Iz-Ist	13:00-14:00
25	Ank-Ist	21:00-22:00	**54**	Iz-Ist	15:00-16:00
26	Ank-Ist	22:00-23:00	**55**	Iz-Ist	17:00-18:00
27	Ist-Ant	07:20-08:35	**56**	Iz-Ist	19:20-20:20
28	Ist-Ant	11:25-12:40	**57**	Iz-Ist	22:00-23:00
29	Ist-Ant	14:25-15:40	**58**	Iz-Ist	01:45-02:45

Table E.1: Flight data for the second problem

Appendix F

Second Problem Pairings - Conventional Model

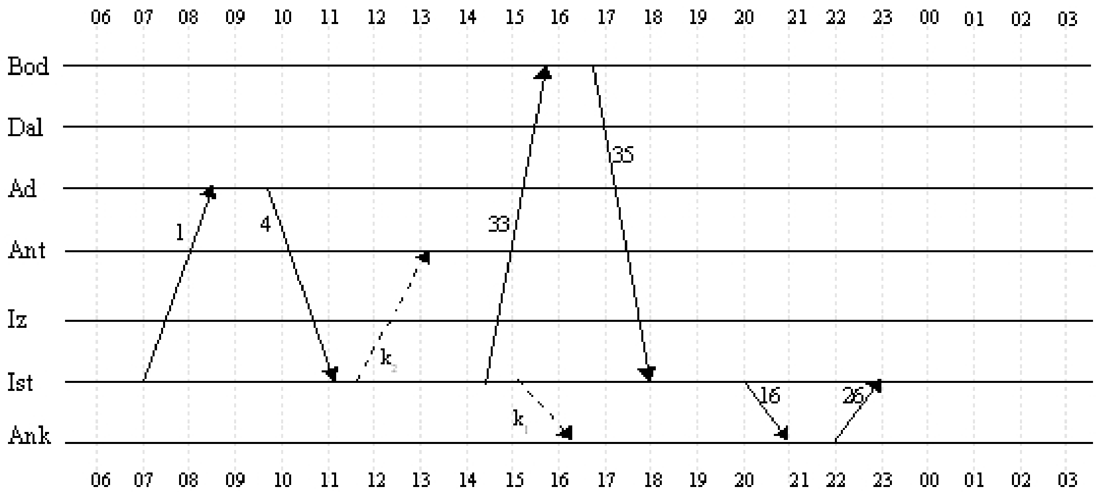


Figure F.1: Flights covered by pairing 53

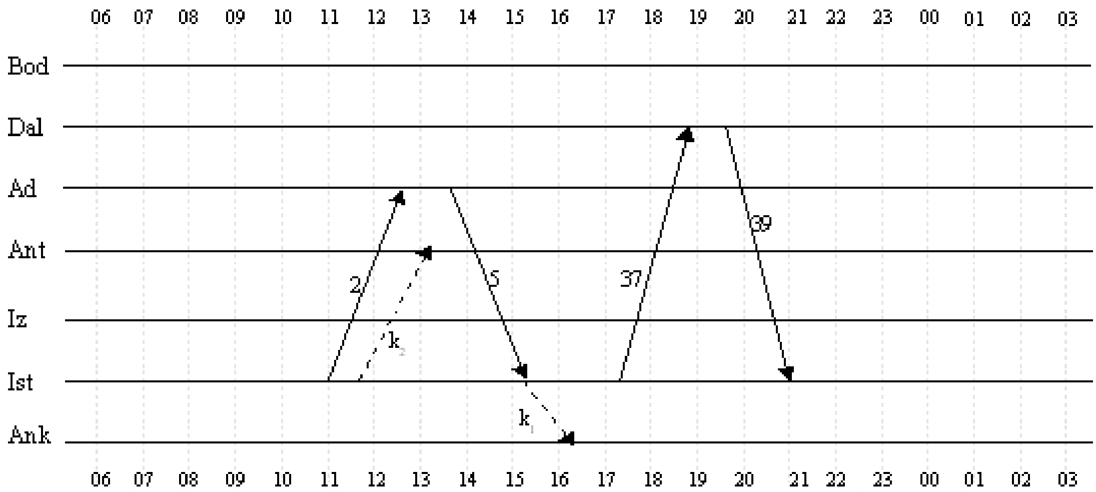


Figure F.2: Flights covered by pairing 119

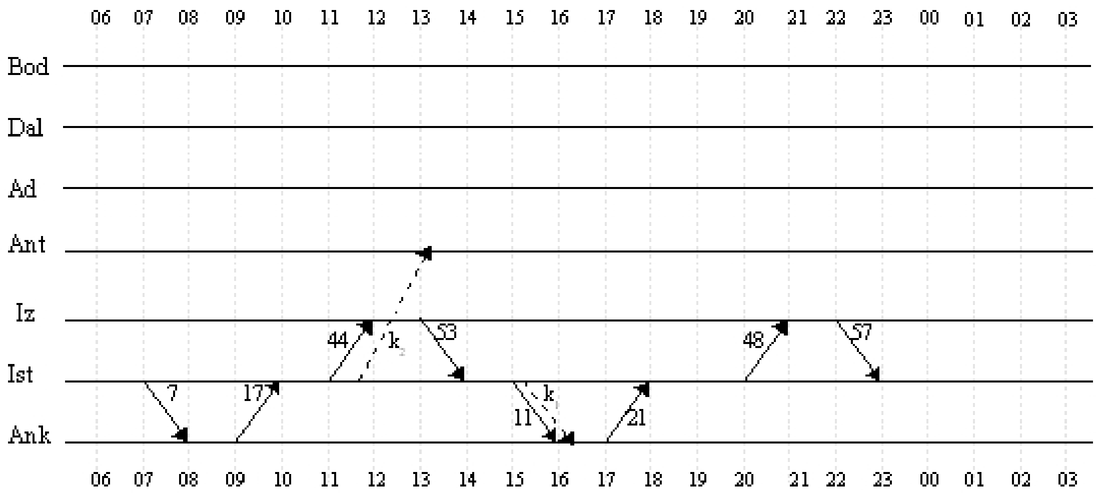


Figure F.3: Flights covered by pairing 215

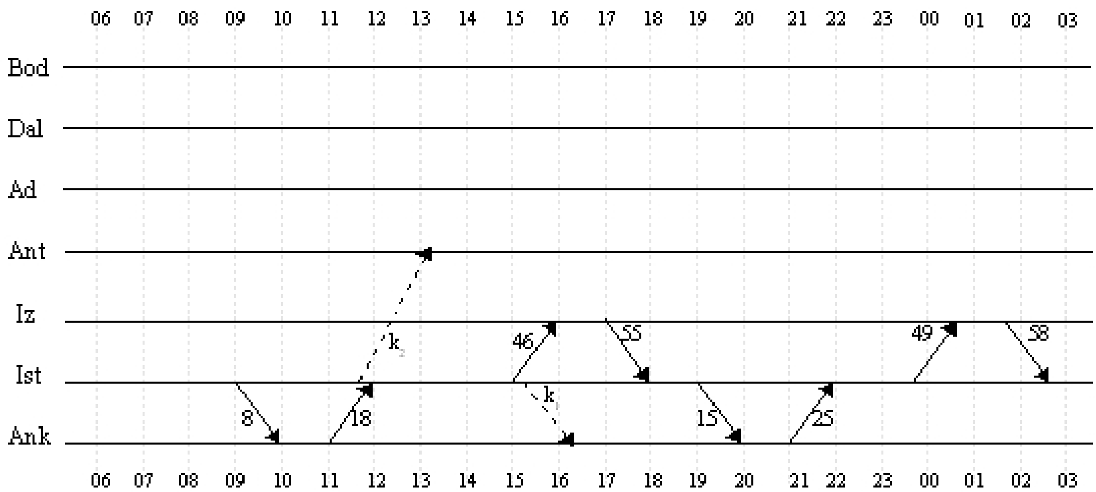


Figure F.4: Flights covered by pairing 617

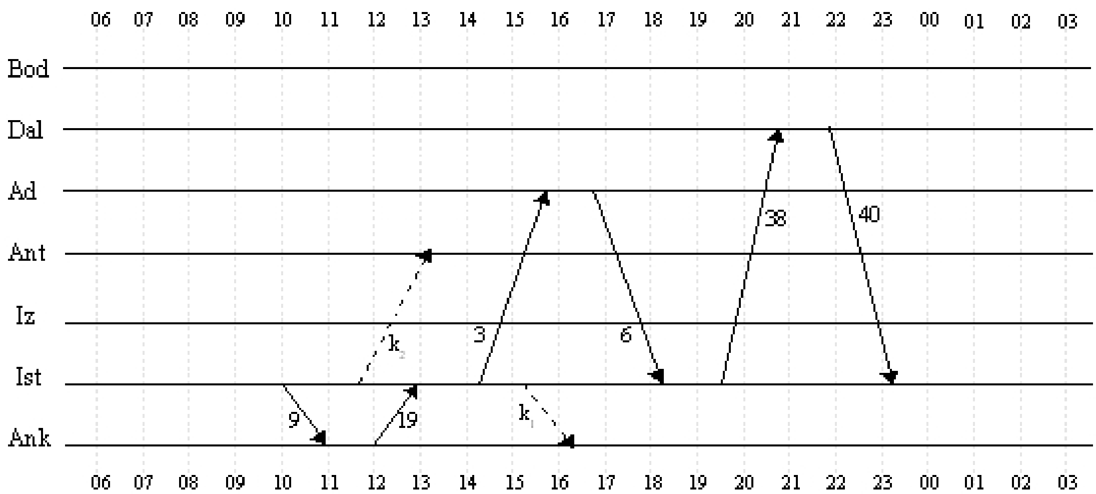


Figure F.5: Flights covered by pairing 686

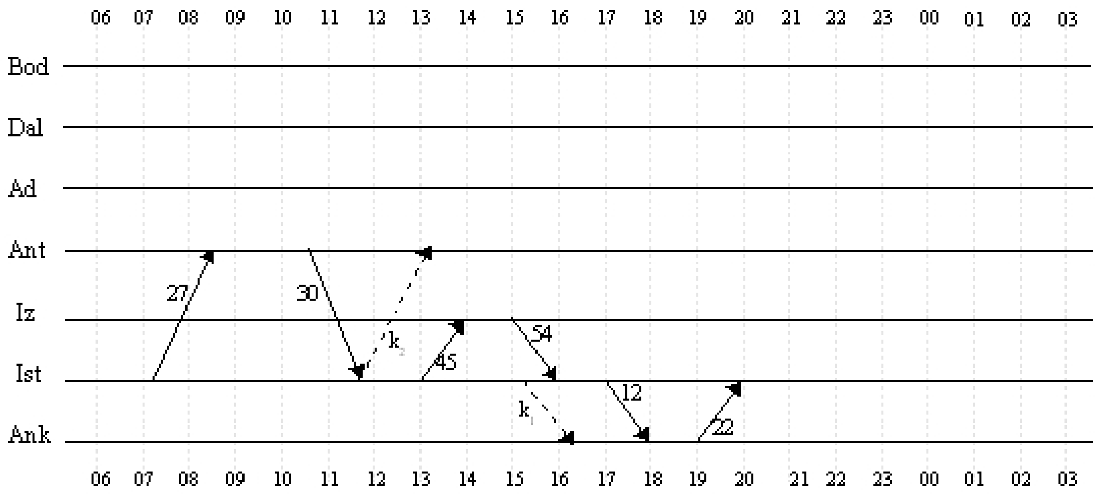


Figure F.6: Flights covered by pairing 909

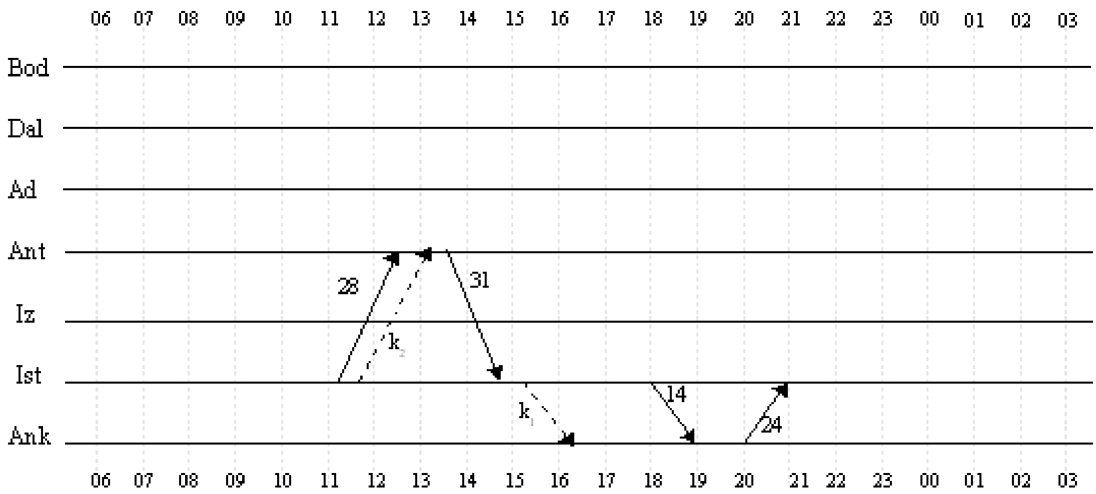


Figure F.7: Flights covered by pairing 962

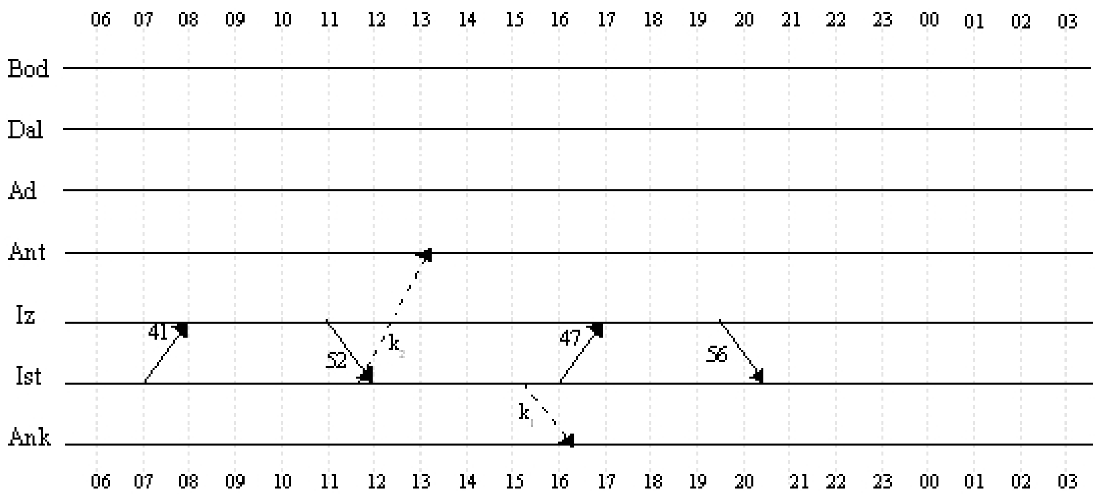


Figure F.8: Flights covered by pairing 1359

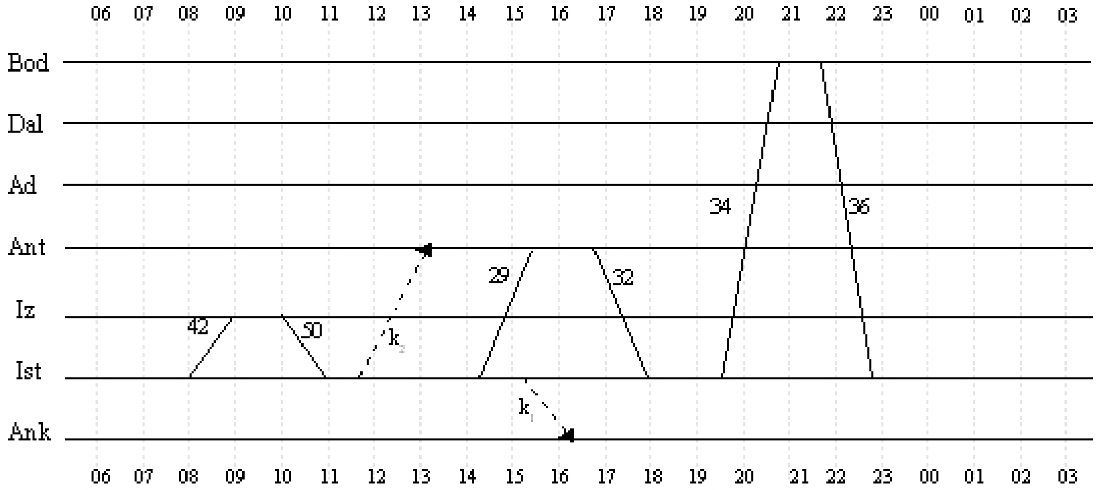


Figure F.9: Flights covered by pairing 1424

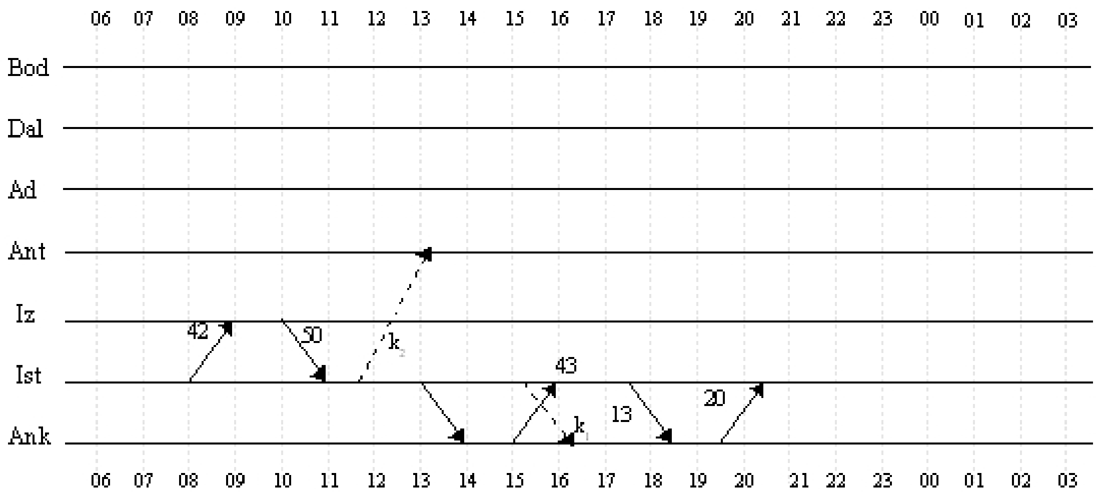


Figure F.10: Flights covered by pairing 1796

Appendix G

Second Problem Pairings - Proposed Model

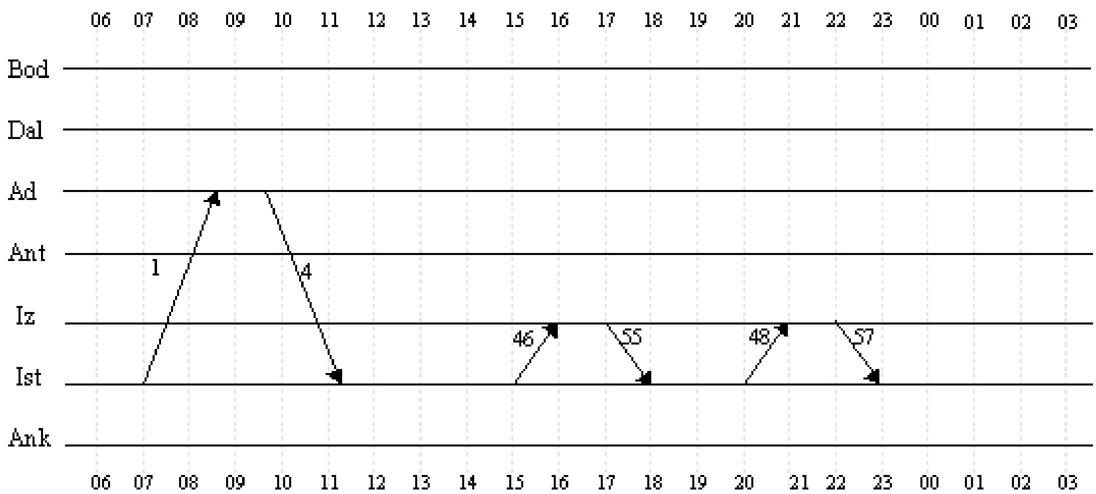


Figure G.1: Flights covered by pairing 91

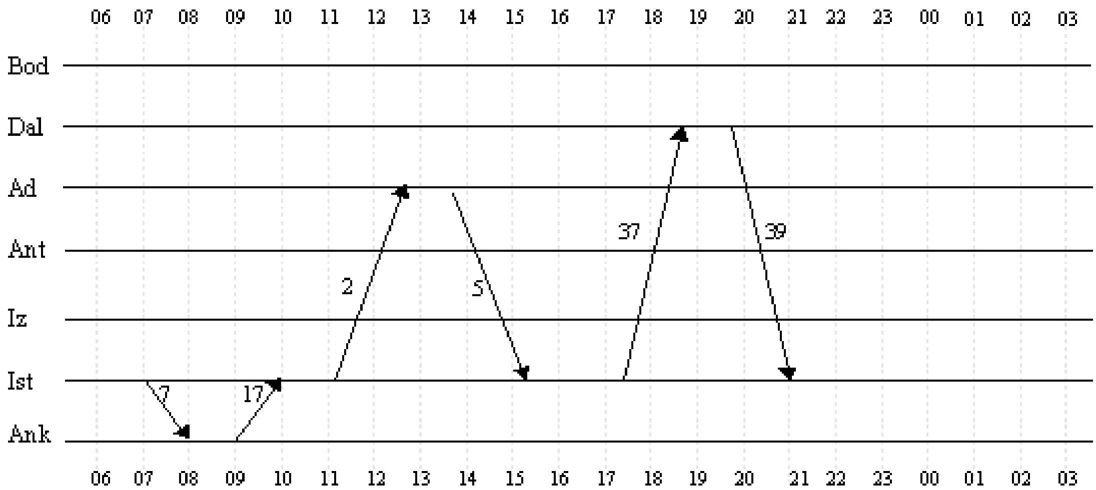


Figure G.2: Flights covered by pairing 148

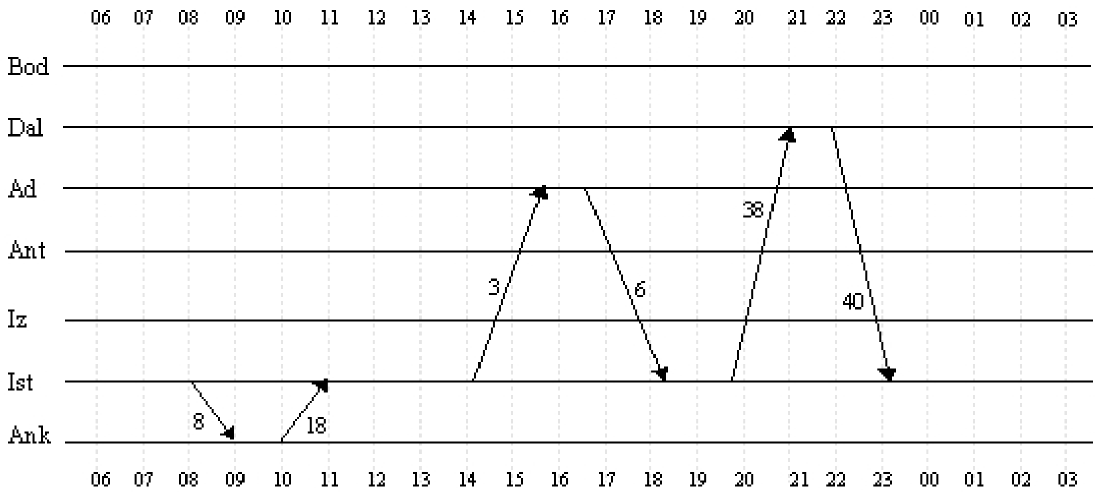


Figure G.3: Flights covered by pairing 507

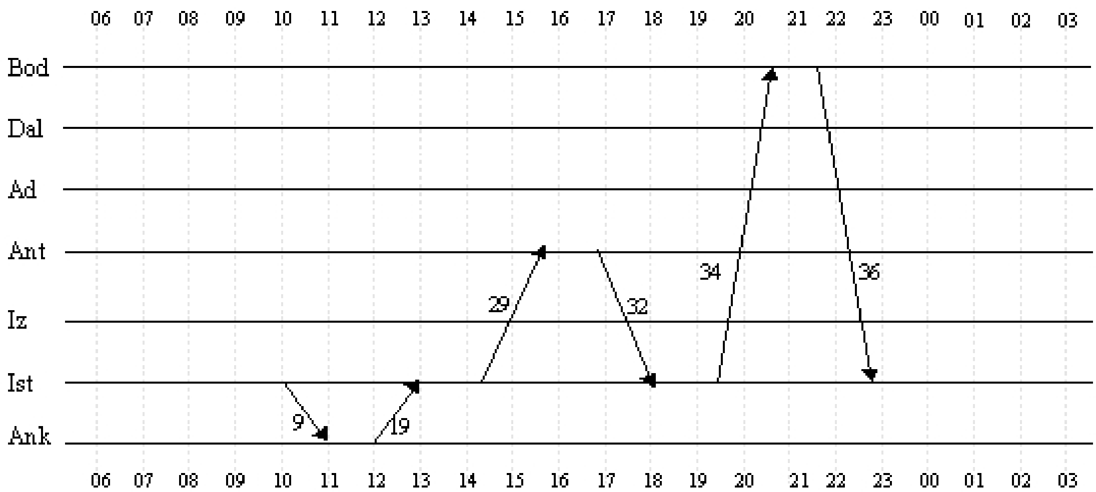


Figure G.4: Flights covered by pairing 715

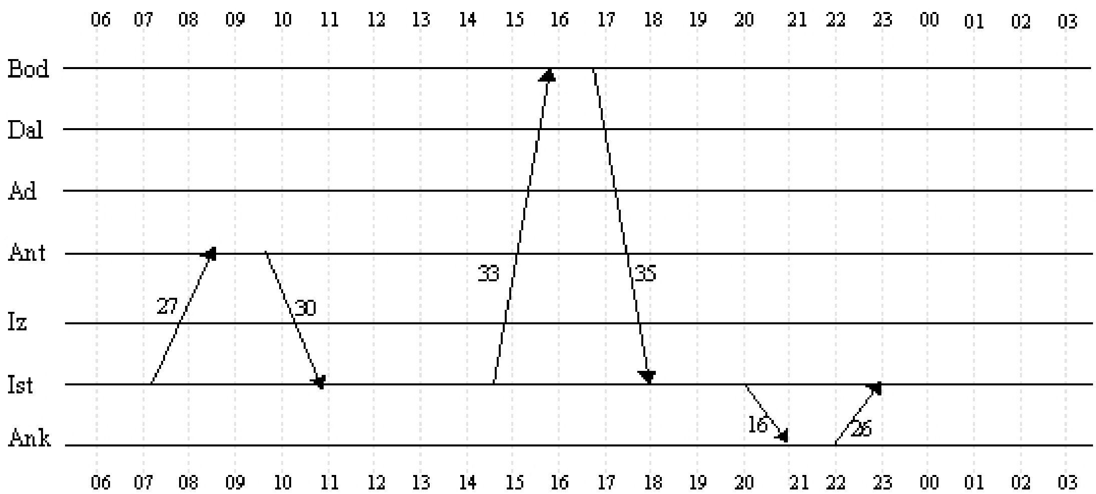


Figure G.5: Flights covered by pairing 904

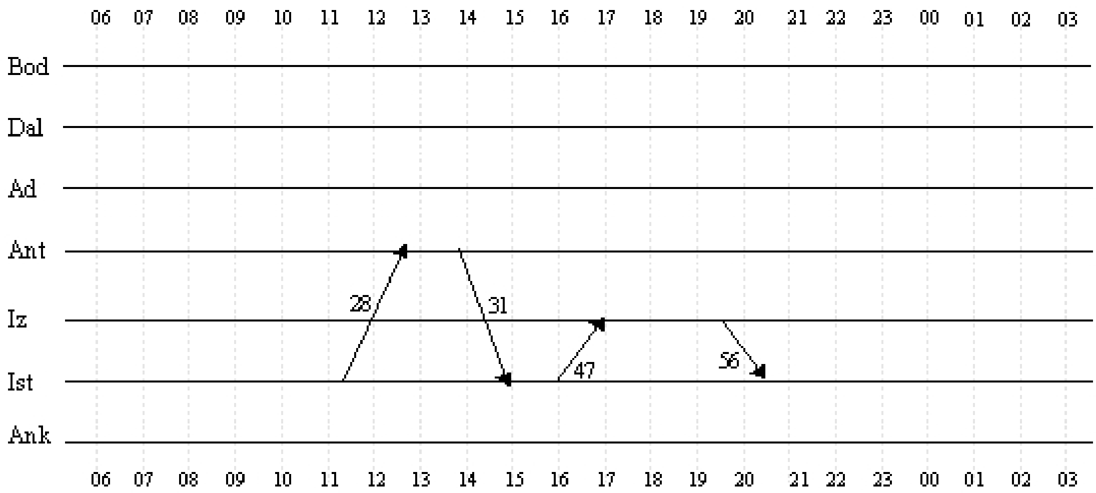


Figure G.6: Flights covered by pairing 970

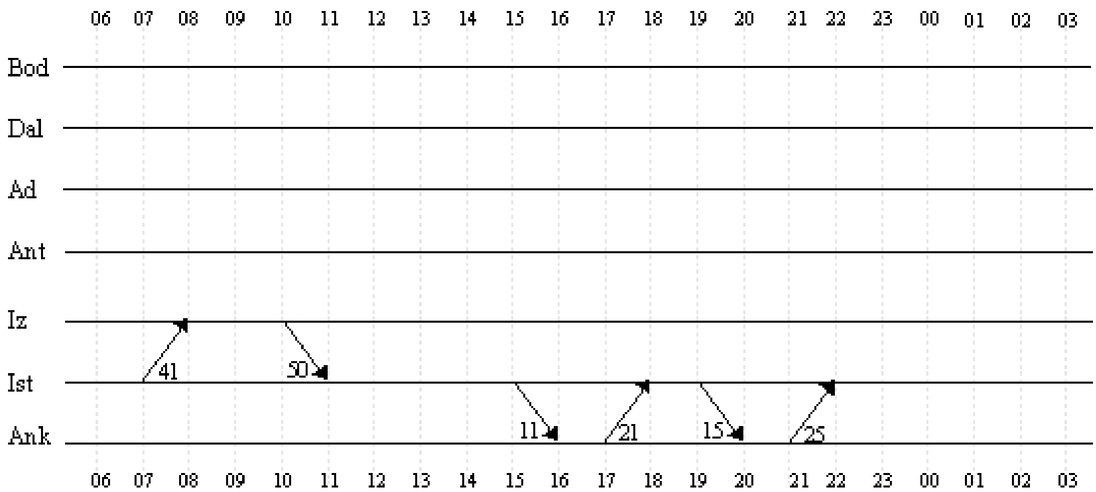


Figure G.7: Flights covered by pairing 1039

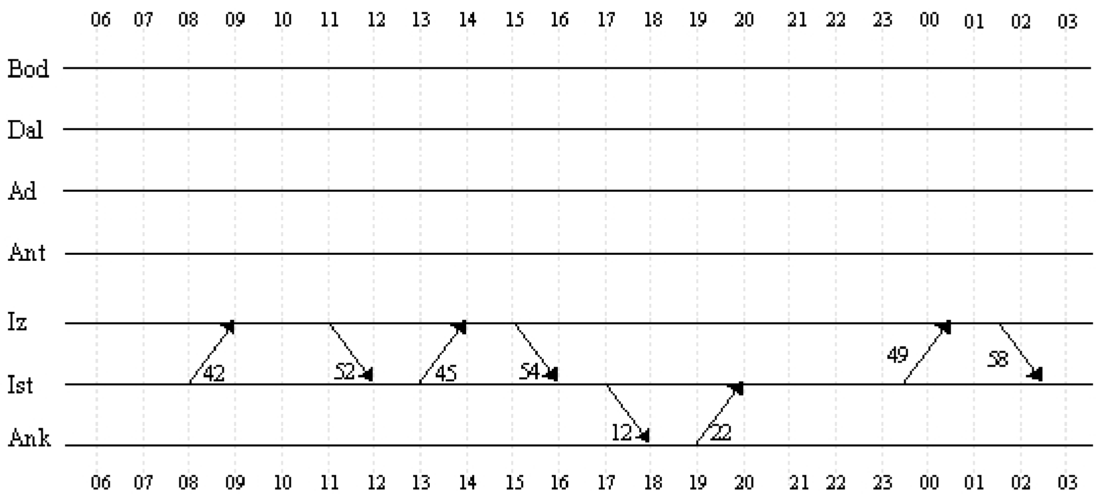


Figure G.8: Flights covered by pairing 1680

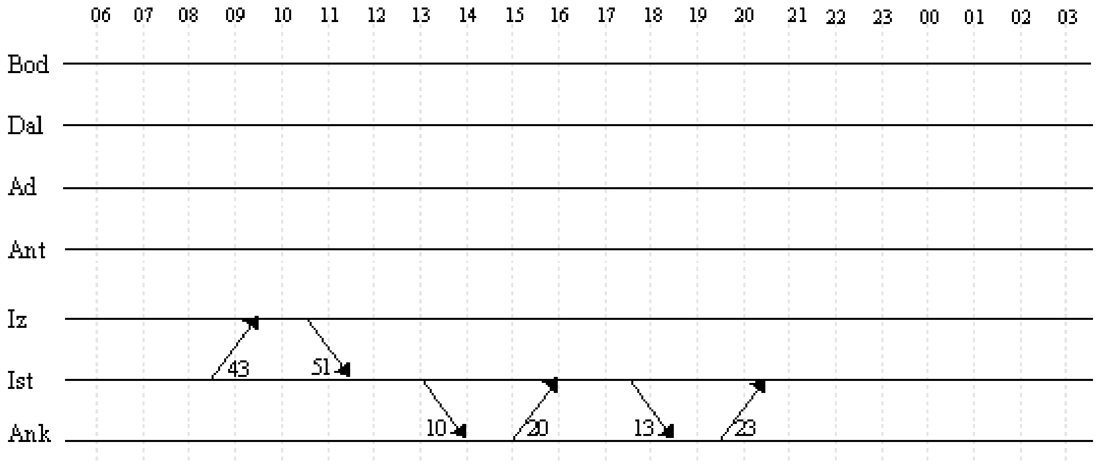


Figure G.9: Flights covered by pairing 1796

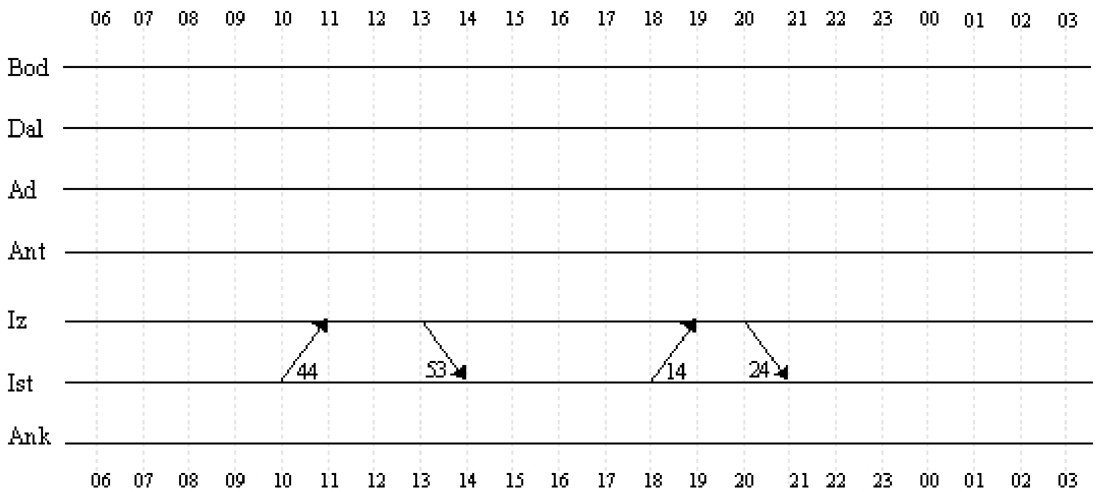


Figure G.10: Flights covered by pairing 2108

Appendix H

Second Problem Solutions

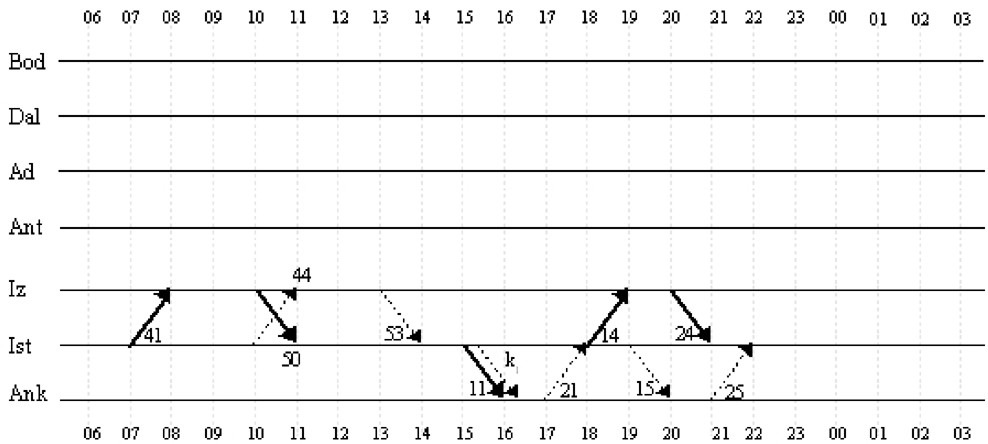


Figure H.1: Solution for extra flight k_1 by swapping pairings 1039 and 2108

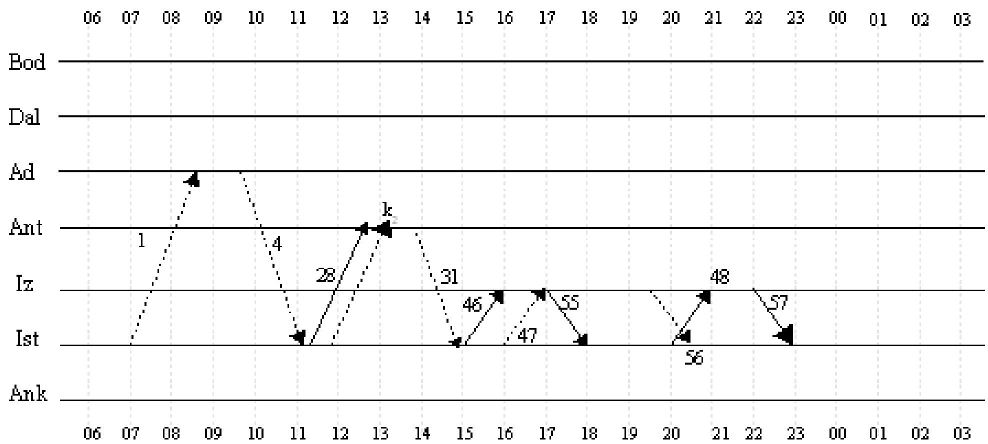


Figure H.2: Solution for extra flight k_2 by swapping pairings 91 and 970

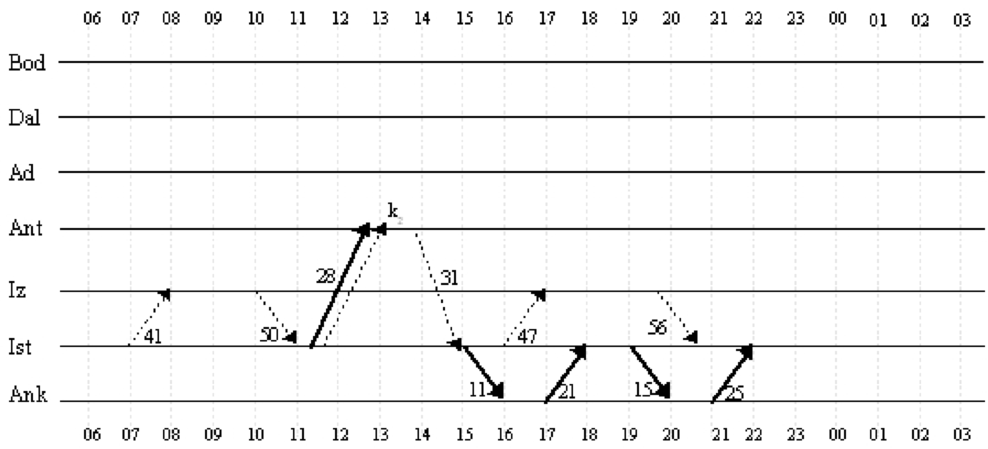


Figure H.3: Solution for extra flight k_2 by swapping pairings 970 and 1039

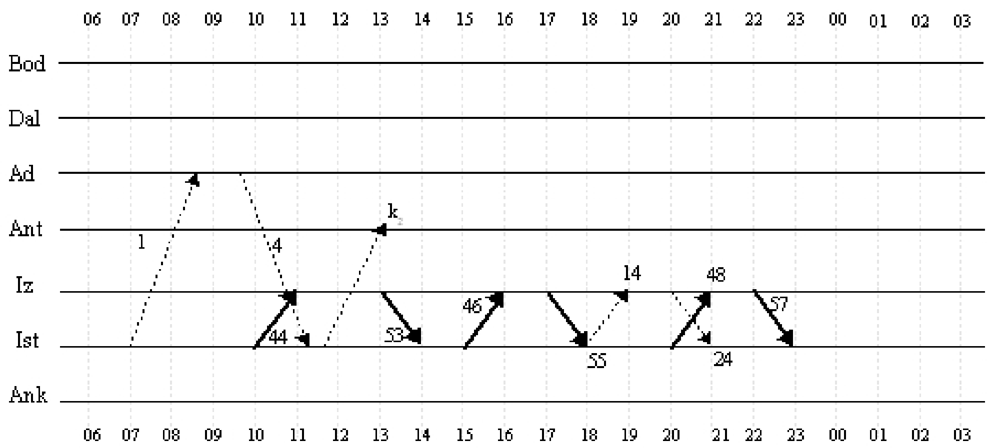


Figure H.4: Solution for extra flight k_2 by swapping pairings 91 and 2108

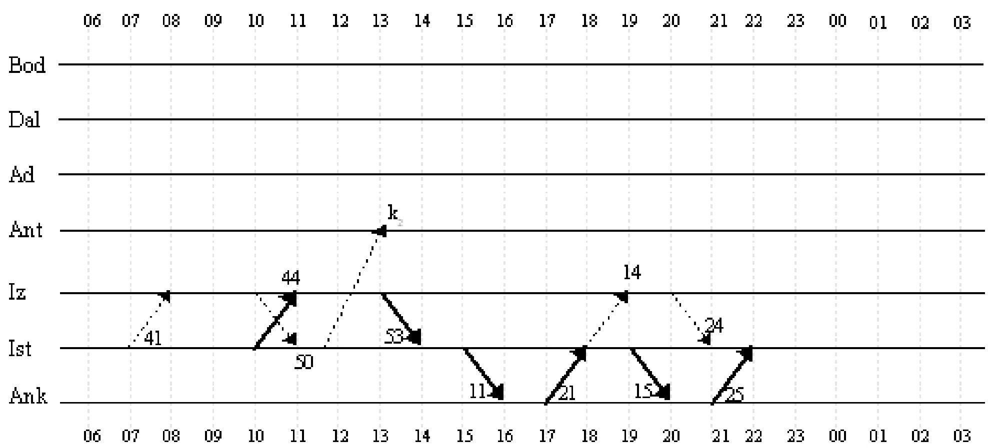


Figure H.5: Solution for extra flight k_2 by swapping pairings 1039 and 2108