A BRANCH-AND-BOUND ALGORITHM FOR AIRPORT GATE ASSIGNMENT PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

KEREM ALANLI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

SEPTEMBER 2019

Approval of the thesis:

A BRANCH-AND-BOUND ALGORITHM FOR AIRPORT GATE ASSIGNMENT PROBLEM

submitted by **KEREM ALANLI** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ________________

Prof. Dr. Yasemin Serin
Head of Department, **Industrial Engineering** ________________

Prof. Dr. Meral Azizoğlu
Supervisor, **Industrial Engineering, METU** ________________

Assist. Prof. Dr. Özlem Karsu
Co-Supervisor, **Industrial Engineering, Bilkent University** ________________

Examining Committee Members:

Assoc. Prof. Dr. Ayşe Canan Şerbetcioğlu
Industrial Engineering, METU ________________

Prof. Dr. Meral Azizoğlu
Industrial Engineering, METU ________________

Assoc. Prof. Dr. Seçil Savaşaneril
Industrial Engineering, METU ________________

Assoc. Prof. Dr. Mehmet Rüştü Taner
Industrial Engineering, TED University ________________

Assist. Prof. Dr. Özlem Karsu
Industrial Engineering, Bilkent University ________________

Date: 09.09.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Kerem Alanlı

Signature:

ABSTRACT

A BRANCH-AND-BOUND ALGORITHM FOR AIRPORT GATE ASSIGNMENT PROBLEM

Alanlı, Kerem
Master of Science, Industrial Engineering
Supervisor: Prof. Dr. Meral Azizoğlu
Co-Supervisor: Assist. Prof. Dr. Özlem Karsu

September 2019, 63 pages

In this study, we consider an airport gate assignment problem that assigns a number of aircraft to a set of gates so as to minimize total walking distance travelled by all passengers. The aircraft that cannot be assigned to any gate due to overlaps are directed to an apron. The problem is formulated as a mixed-integer nonlinear programming model and then it is linearized. A branch-and-bound algorithm that employs powerful bounding mechanisms is developed. The results of the computational experiment have shown that the mathematical model can handle small sized problem instances, while the branch-and-bound solves relatively larger instances in reasonable time.

Keywords: Airport Gate Assignment Problem, Mixed Integer Linear Programming, Branch-and-Bound Algorithm

ÖZ

HAVALİMANI KAPI ATAMA PROBLEMİ İÇİN BİR DAL-SINIR ALGORİTMASI

Alanlı, Kerem
Yüksek Lisans, Endüstri Mühendisliği
Tez Danışmanı: Prof. Dr. Meral Azizoğlu
Ortak Tez Danışmanı: Dr. Öğr. Üyesi Özlem Karsu

Eylül 2019, 63 sayfa

Bu tezde, belirli sayıda uçağın bir dizi kapıya atandığı bir havalimanı kapı atama problemi ele alınmıştır. Zaman örtüşmelerinden dolayı herhangi bir kapıya atanamayan uçaklar aprona yönlendirilmektedir. Problemin amacı, tüm yolcular tarafından kat edilen toplam yürüyüş mesafesinin minimizasyonudur. Problem, öncelikle bir karışık tam sayılı programlama modeli ile formüle edilmiş, daha sonra ise doğrusal duruma getirilmiştir. Güçlü sınırlama mekanizmaları kullanan bir dal-sınır algoritması geliştirilmiştir. Deneysel çalışmaların sonuçları; matematiksel modelin küçük boyutlu problemlerin üstesinden gelebildiğini, ancak dal-sınır algoritmasının daha büyük boyutlu problemleri kabul edilebilir süre içerisinde çözebildiğini göstermiştir.

Anahtar Kelimeler: Havalimanı Kapı Atama Problemi, Karma Tam Sayılı Doğrusal Programlama, Dal-Sınır Algoritması

To the memory of my beloved mother who continues to guide me with her light

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisors, Prof. Dr. Meral Azizoğlu and Asst. Prof. Dr. Özlem Karsu, for their valuable guidance. Without their creativity, insight and effort; this study would have never been finished. I am especially thankful for their positiveness and patience during my 2-year long thesis journey.

Department of Industrial Engineering at METU was a wonderful place to study. I am honored to be part of this department and I will definitely miss my days there.

I would like to thank my father, Mehmet Alanlı, for his continuous support through the hard times we have faced together.

I am grateful to my aunties Neriman Atik and Meral Benli, and my all other relatives, for never making me feel alone in Ankara and taking care of me like nobody else would do.

Finally, I am more than thankful to my beloved girlfriend, Seray Öztürk. She was always with me during the ups and downs and her beautiful smile was my biggest inspiration to overcome the difficulties.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Assignment problem deals with finding an allocation of a particular set of tasks, activities or people to a set of resources so as to maximize the utility of allocation or minimize its cost. Allocation of tasks to people at workplaces, allocation of vehicles to service areas, allocation of financial resources to government agencies are some examples of assignment problem encountered in real-life.

Airline operational planning is a very important as much complex area, which is widely worked on by operational researchers. To find efficient and effective ways to handle airline operations, the managers and operation planners rely on some assignment problems like crew assignment problem, fleet assignment problem and gate assignment problem.

In this thesis, we study an airport gate assignment problem (AGAP) that considers the allocation of the aircraft to the gates in airports. In this problem, the characteristic behavior of an aircraft is as follows:

1) Aircraft arrives at the airport at specified arrival time.
2) Aircraft occupies a place (gate) in the airport for a specified amount of time.
3) Aircraft departs from the airport at specified departure time.

The AGAP aims to reach a satisfactory solution relative to some pre-defined objectives that are dependent on the airline operational planner’s point of view. To the best of our knowledge, the earliest study on the field of airport gate assignment problems is due to Steuart (1974), where a stochastic model is proposed to find a schedule with minimum number of gate positions and estimate the required number of gate positions.

There are different constraints and preferences that the owner of the AGAP should consider depending on the characteristics of the specific setting. Airports are complex buildings and there could be significant differences between a major airport and a minor airport. One of the main differences is the presence of fixed gates. Major airports have fixed gates which allow passengers to board into the aircraft using a jet bridge. However, some minor airports have only remote gates so the passengers should be transported to the aircraft by a bus or sometimes they should simply walk to the aircraft on the apron. Remote gates are also used in major airports, because the fixed gates may not be sufficient to handle all aircraft at all times. Utilization of fixed gates is more comfortable for the passengers because it is much easier to board into the aircraft by just walking through a tunnel in comparison to taking a 5-to-10 minutes trip in a very crowded bus and climbing to the aircraft using steep stairs.

There are two types of passengers in general:

- Transit passengers,
- Non-transit passengers.

Transit passengers are those who arrive at the airport by a flight and depart on another flight. Non-transit passengers are those who either arrive at the airport from the city and depart with a flight or arrive at the airport by a flight and leave the airport afterwards.

The airports can also be categorized based on the type of flights (and passengers) that they serve. Some airports are too small and unbusy so that there are no transit passengers. In Turkey, some airports are only available for domestic flights, whereas airports in larger cities and touristic areas are "international airports", hence they can serve international flights as well as domestic flights. On the other hand; in some smaller countries, there are no domestic flights and the airports are designed only for international flights. In airports with both domestic and international flights, the international terminal and the domestic terminal are separated because of the difference in identity and passport checking procedures before letting passengers into the terminal area. Some metropolitan airports allow towing, which can be simply defined as transportation of the aircraft from one place to another in the airport using specialized ground vehicles. In such airports, the aircraft can change from its pre-assigned gate to another if required. As one can see, there could be many different dimensions and possibilities in the process of assigning the aircraft to gates and it is the planner's responsibility to determine the best allocation with respect to the availabilities of the facilities and restrictions faced in the airport.

The air transportation traffic has been roughly doubled from the early 1980s to 2006 (Dorndorf, Drexl, Nikulin & Pesch, 2007) and according to International Air Transport Association (2019), average yearly increase in the number of airline passengers is 5.85% between the years 2006 and 2017. As the air traffic grows and the operational environment changes throughout the years, needs of airport operators and airline companies have evolved. Thus, many different objectives have been considered in the AGAP most commonly used of which are stated by Dorndorf et al. (2007) as follows:

- Minimization of the number of ungated (unassigned) aircraft,
- Maximization of preferences/utility of allocation of certain aircraft to specific gates,
- Minimization of total passenger walking distance through the airport,
- Minimization of the deviation from the original or a reference schedule,
- Minimization of the towing costs.

Minimization of the number of ungated aircraft is one of the most commonly discussed objectives in the AGAP studies. It aims to maximize the number of aircraft that are assigned to fixed gates in order to provide a comfortable access for the passengers. In some cases, this objective can be interpreted as maximization of the number of aircraft hosted in an airport if the airport lacks space to handle all scheduled flights.

Maximization of the preferences is discussed more in the recent literature. Maximization of the expected number of visitors to the shops and maximization of the profitability of the shops are two examples of this objective. Such objectives may not be relevant for smaller airports that lack non-critical facilities; they are mostly used for larger airports and hubs. A hub can be briefly described as a very busy airport that is mostly used for connection purposes. Hubs host many passengers from all around the world. As a result, they are also home to shopping facilities of a large variety. In such airports, some specific aircraft may be preferred to be assigned to specific gates due to the profitability concerns. For instance, the aircraft of the countries where alcoholic drinks are forbidden, are not assigned to the gates that are close to the alcohol drink shops.

Minimization of total passenger walking distance is the mostly considered objective in the field. (Aktel, Yagmahan, Özcan, Yenisey & Sansarcı, 2017). It is also an important objective for the passengers' comfort level. Walking distances are directly related to the distances between gates and the distances from the gates to the common areas such as luggage claim area, airport entrance and airport exit. Metropolitan airports are usually quite large buildings, so it may take a significant amount of time and effort for a passenger to travel from the aircraft to luggage claim area or another aircraft.

Minimization of the deviation from the original schedule is considered in cases where unexpected delays occur and the original schedule becomes infeasible. Delays may cause discomfort among passengers and increase in assignment costs because towing operations and increased usage of remote gates might be required. Hence, sticking to the original schedule as much as possible favors both the airline operators and the passengers.

Minimization of the towing cost may be an important concern due to the high costs of towing services.

There are some other objectives considered in the AGAP as stated by Aktel et al. (2017) as follows:

- Minimization of gate idle time,
- Minimization of waiting time,
- Minimization of total connection time,
- Minimization of baggage transport distance,
- Minimization of the total duration of ungated flights,
- Minimization of the number of conflicts,
- Minimization of buffer times.

Our AGAP takes minimization of total passenger walking distance as the objective. We first formulate the problem as a nonlinear mathematical model and then provide its linearized version. We then design a branch-and-bound algorithm involving powerful bounding schemes.

CHAPTER 2

LITERATURE REVIEW

The airport gate assignment problem (AGAP) is a widely studied problem in the operational research literature thanks to its practical importance. A detailed survey on the topic is presented by Dorndorf et al. (2007). AGAP is proven to be NP-hard (Obata, 1979).

Cheng, Ho & Kwan (2012) classified the AGAP variants into two categories with respect to formulation: static AGAP and stochastic & robust AGAP. In the static AGAP, a deterministic model is formulated, typically with the objectives of minimization of waiting time, ungated flights or total walking distance. Stochastic and robust AGAP are formulated taking into account stochastic aspects such as flight delays or disruptions. Commonly used objectives in the stochastic AGAP are minimization of idle time, gate conflicts and flight delays.

Another classification of the studies in the literature can be made with respect to the solution methodology used (Cheng et al. 2012). The AGAP solution methodologies can be categorized into three: expert system approaches, exact solution approaches and heuristic approaches.

Expert systems can be defined as software systems that aim to simulate the decision-making process of human experts. A database which contains rules obtained from human knowledge is given to the software for suggesting solutions. Some studies that propose expert systems for solving the airport gate assignment problem are conducted by Brazile & Swigger's (1988) work, Gosling's (1990) work, Srihari & Muthukrishnan's (1991) work and Su & Srihari's (1993) work.

Attributing to the complexity of the problem, the heuristics are more commonly used than exact algorithms. Yet, there are also several exact algorithms presented in the literature. We first review the studies with exact algorithms, then we continue with studies that propose heuristics.

2.1. Studies with Exact Algorithms

One of the earliest studies on the airport gate assignment problem was Babić, Teodorović & Tošić's work (1984), where the average walking distance covered by the arriving and departing passengers is minimized. They assumed that there are no transit passengers and the flight schedule is such that one airplane could always be assigned to an unoccupied gate. They proposed a depth-first Branch-and-Bound (B&B) algorithm along with a lower bound (LB) that underestimates walking distance of upcoming passengers. The computational experiments on instances with 9 aircraft and 5 gates have shown that using the lower bound significantly improves the performance of the algorithm. The optimum solution is found to be significantly better than random allocation of the airplanes to the gates.

Bihr (1990) considered a special case of the AGAP where the departure gates are fixed and formulated a classical assignment model to assign arriving flights to gates so as to minimize the walking distance of the passengers.

Mangoubi & Mathaisel (1985) proposed a greedy heuristic and an LP relaxation for solving the total walking distance minimizing AGAP. They assumed that a passenger arriving at a gate would be equally likely to board his next flight at any gate; hence used expected walking distances through a uniform distribution for transit passengers, which simplifies the problem. The LP relaxation of the resulting integer programming problem provided an integer feasible solution in their case study; however, the authors note that the matrix is not totally unimodular and a branch and bound approach would be needed if the relaxation fails to return an integer feasible solution. They used real time data acquired from Toronto International Airport (with 138 aircraft and 20 gates) and showed that the actual assignment is significantly worse than the optimum (obtained by the integer solution of the LP relaxation). The proposed heuristic was successful, reaching a feasible solution with an acceptable optimality gap in a very short time, whereas the LP solution took significantly longer time. The most successful method was the LP relaxation with the heuristic solution being used as an initial feasible solution for the LP problem. This method reached optimal solution in nearly the same computation time as that of the greedy heuristic.

Bolat (1999) proposed a branch-and-bound algorithm and a heuristic called branch-and-trim for solving the robust gate assignment problem. The objective is to minimize the difference between the maximum and minimum slack times. They performed computational experiments to compare two different branching rules for the B&B and reported that the choice of the strategy affects the performance. They also observed that the solution time of the B&B algorithm decreases as the gate utilization levels increase. The branch-and-trim heuristic is reported to be quite effective, reaching good solutions in very short time. Computational studies performed using data from Riyadh International Airport have shown that the heuristic significantly outperforms the current procedure in terms of the number of ungated aircraft and number of towing operations required.

Yu, Zhang & Lau (2016) focused on the robustness issue in AGAP. They proposed three different mathematical models and four different heuristics to solve the robust AGAP. The objectives considered in the study were minimization of the expected conflict time between schedules (for robustness), minimization of towing costs and minimization of distance covered by transit passengers. They developed three mathematical models: a network flow model with a quadratic objective function and two mixed integer programming (MIP) models with linearized objective functions. They proposed four different heuristics and compared their performance relative to the CPLEX branch-and-cut scheme. Their experimental results on problem instances with up to 30 flights and 5 gates have shown that one MIP model is far superior to the network flow model with quadratic objective in terms of efficiency. They also have shown that exact quadratic expressions are clearly better than approximate ones that use the average distance assuming a uniform distribution for the experience of transit passengers.

2.2. Heuristics

Haghani & Chen (1998) formulated the AGAP as a binary integer quadratic assignment problem. They proposed a heuristic algorithm for solving the AGAP to minimize total walking distance. Their computational results indicate that the algorithm provides high quality solutions.

Yan & Huo (2001) formulated the problem as a bi-objective mixed 0-1 integer model where the objectives are minimization of total passenger walking distance and minimization of total passenger waiting time. They converted the problem into a single objective problem that minimizes a weighted sum of the two objectives. They used column generation, B&B and the simplex method to solve the resulting problem efficiently. Computational results of a case study in a Taiwanese airport show that the method is useful and provides significant improvement in airport decision-making.

Yan, Shieh & Chen (2002) proposed a simulation framework with flexible buffer times so as to absorb the stochastic delays in real-time assignments. They formulated the problem as in Mangoubi & Mathaisel (1985), taking minimization of total passenger walking distance as the objective. They proposed two greedy heuristics that sort the flights with respect to some rule and then assign the flights sequentially to the nearest gate. The first variant sorts the flights with respect to the number of passengers while the second one sorts them based on the arrival times. Their computational results using the real data from Chiang Kai-Shek Airport have shown that the proposed simulation framework could be useful for airport authorities.

Xu & Bailey (2001) formulated the problem as a mixed 0-1 quadratic integer model and then reformulated it as a mixed 0-1 integer linear programming program. Their objective is the minimization of total passenger connection time. They proposed a tabu search algorithm for solution. They generated test data for five consecutive days with different passenger origin-destination parameters. They showed the benefits of making daily assignments based on the new passengers' origin and destination data over using static assignment, which does not consider passenger connection times.

Their computational results indicated that the presented metaheuristic has clear advantage over the static assignment with the average saving of 24.70%. They also compared the proposed metaheuristic with mathematical model solutions in small instances with 12, 15 aircraft - 3 gates and 20 aircraft - 5 gates. For the test problems, the heuristic could find optimal solutions in less than 0.1 seconds whereas CPLEX could not reach optimal solutions in reasonable time for the larger test instances with 20 aircraft and 5 gates.

Ding, Lim, Rodrigues & Zhu (2005) proposed 4 heuristic algorithms for solving the AGAP. The first one is a greedy algorithm which finds the minimum number of aircraft assigned to remote gates. The second algorithm proposed in the study is a simulated annealing heuristic. The third algorithm is an interval exchange tabu search (TS) algorithm where the moves introduced in simulated annealing are implemented in a TS based algorithm. The final algorithm presented is a hybrid algorithm, which is a combination of their simulated annealing (SA) and interval exchange TS algorithms. The experimental results indicated that SA is more successful than TS in smaller sized instances, whereas the interval exchange TS is more successful in larger sized instances. Even though SA outperforms TS in terms of the CPU time, no clear superiority of one method over the other is found. However, the proposed hybrid algorithm finds optimal solutions in nearly all of the small size instances (15-25 aircraft and 3-6 gates) and reaches better solutions than SA and TS algorithms in large size instances (100-640 aircraft and 16-52 gates). Therefore, the study concludes with the superiority of the hybrid algorithm.

Cheng et al. (2012) compared four metaheuristics that are previously proposed in other studies. They used the genetic algorithm from Bolat (2001), tabu search algorithm from Xu & Bailey (2001), simulated annealing and hybrid tabu search + simulated annealing algorithm from Ding et al. (2005). They modeled the AGAP to minimize total walking distance and modified the algorithms. Their computational results have shown that hybrid algorithm is the best among all metaheuristics, whereas the tabu search algorithm is the best performer among the classical metaheuristics.

Genç, Erol, Eksin, Berber & Güleryüz (2012) combined a stochastic approach with a Big Bang – Big Crunch algorithm to solve the AGAP that maximizes the utilization times of the fixed gates. They used a heuristic called ground time maximization algorithm to find a reasonable initial solution and then used the stochastic approach to improve their solution. Their experiments based on the real data from İstanbul Atatürk Airport have shown the effectiveness of the proposed algorithm.

Şeker & Noyan (2012) proposed new stochastic optimization models for the AGAP and compared the performances of the models. For practical concerns, they implemented a tabu search heuristic since they found that CPLEX could not even construct a branch-and-bound tree.

Marinelli, Dell'Orco & Sassanelli (2015) proposed a Bee Colony Optimization algorithm to solve the AGAP. Their aims were to minimize total walking distance and minimize number of aircraft assigned to apron. They used the real data obtained from Milano Malpensa Airport. Their computational results and a multi-criteria analysis have shown that the assignment obtained through the proposed algorithm is better than the actual assignment.

A recent study considering minimization of total walking distance and number of ungated flights as objectives is done by Aktel et al. (2017). They proposed and compared two metaheuristic algorithms. The first algorithm is a tabu search algorithm where the minimization of the total walking distance is used as a fitness function to determine the best neighbor. Another variant of the tabu search heuristic is also proposed, in which the fitness function is used to or not to accept a tabu move. The second algorithm proposed is simulated annealing, which is quite similar to the one presented in Ding et al. (2005). The authors compared their three proposed algorithms to the greedy algorithm proposed by Ding et al. (2005). The experiments are conducted using 6 randomly generated problem instances of different sizes (100-520 aircraft and 16-44 gates). The experimental results have shown that all proposed algorithms are able to provide acceptable solutions for large-scale problems in reasonable time. All three algorithms proposed in this study have performed significantly better than the greedy approach by Ding et al. (2005). Simulated annealing heuristic was the best performer for three problems, tabu search with probabilistic approach was the best performer for one problem and tabu search with fitness approach was the best performer for the remaining two problems. Therefore, no clear superiority is found.

Kim, Feron & Clarke (2013) proposed a new AGAP model where the objectives are minimization of passenger transit time and minimization of weighted aircraft taxi time. Firstly, they formulated the problem as a quadratic assignment problem then reformulated it as a linear mixed 0-1 integer problem. A good but not exact solution could be reached using B&B, which is terminated after 30 minutes due to practical reasons, and a tabu search algorithm.

Hu & Di Paolo (2009) proposed a genetic algorithm with uniform crossover to solve the multi-objective AGAP. Their objectives were minimization of total walking distance, minimization of total baggage distance and minimization of total passenger waiting time. Weights are used to combine three objectives into one objective. Contrary to the most proposed genetic algorithms in the literature, the authors used relative positions of the aircraft instead of using the absolute positions while building chromosomes. Simulation and computational experiments have proven the effectiveness of the proposed algorithm.

Mokhtarimousavi, Talebi & Asgari (2018) proposed a sorting genetic algorithm for the multi-objective AGAP. They considered three objectives: minimization of total walking distance, minimization of taxiway conflicts and minimization of costs related to towing operations and undesired assignments. The proposed algorithm was compared to some other previously proposed algorithms and it is found competitive in terms of number of solutions, elapsed time and diversity.

Deng, Sun, Zhao, Li & Wang (2018) considered a multi-objective AGAP and proposed an improved ant colony algorithm. Their objectives were minimization of idle time variance, minimization of total walking distance and minimization of aircraft assigned to apron. They used real time data obtained from Guangzhou Baiyun Airport. They compared their proposed algorithm to two different ant colony algorithms and the results have shown that the proposed algorithm could find better solutions at an expense of higher computation time.

Daş (2017) considered the AGAP with the objective of maximizing the number of passengers whose gates are close to shopping facilities and minimizing the total walking distance. Three mathematical models are proposed and a two-phase local search and Pareto local search integration algorithm are suggested.

Yu, Zhang & Lau (2017) proposed an adaptive large neighborhood search algorithm to solve the AGAP. Their objectives were the same as their previous study: robustness, towing and comfort of transfer passengers. (Yu et al. 2016) They compared their algorithm to the tabu search heuristic proposed in Ding et al. (2005), as both algorithms use local search. Their computational experiments have shown the superiority of the adaptive large neighborhood search algorithm over the tabu search.

Jiang, Zeng & Luo (2013) considered airline fairness as one of their objectives. They separately calculated average walking distances of the passengers of each airline and proposed a multi-objective model for the AGAP with two objectives: minimization of total passenger walking distance and minimization of the maximum ratio of average walking distance of passengers of an airline to average passenger walking distance over all airlines. They used Lingo software to find a feasible solution and optimal solutions. Their computational results have shown that their solutions are better than the random allocation.

In this study, we consider the AGAP, in which the total passenger walking distance is minimized. To the best of our knowledge, the closest work in the literature is by

Mangoubi & Mathaisel (1985), who also minimize passenger walking distance. However, they assume that a transit passenger arriving at a gate would be equally likely to board his next flight at any gate; hence used expected walking distances through a uniform distribution for transit passengers, which simplifies the problem. We relax this assumption and consider the actual distances covered by transit passengers. We formulate the problem as a quadratic assignment problem with overlap constraint and then linearize the quadratic term and propose a linear programming formulation (See Pentico (2007) and Loiola, de Abreu, Boaventura-Netto, Hahn & Querido (2007) for reviews on assignment problems and quadratic assignment problems, respectively. See also Bouras, Ghaleb, Suryahatmaja & Salem (2014), and the references therein, for a list of studies that consider quadratic programming formulations for the AGAP).

Other studies that formulate the AGAP as a quadratic assignment problem are Drexl & Nikulin (2008) and Haghani & Chen (1998). Drexl & Nikulin (2008) consider a multi-objective formulation, which minimizes total passenger walking distances, maximizes the gate preferences and minimizes the number of ungated flights. Their formulation involves a quadratic objective and quadratic constraints. They, however, do not consider an exact solution approach and suggest Pareto simulated annealing for finding approximate Pareto solutions. Haghani & Chen (1998) also minimize total passenger walking distance and hence formulate a quadratic assignment problem. They then propose an integer linear programming reformulation and develop a heuristic algorithm for solving the problem while we propose an exact solution approach.

CHAPTER 3

PROBLEM DEFINITION

We study an airport gate assignment problem where *n* aircraft have to be assigned to *m* fixed gates and a remote gate (apron). The gates are already laid out with specified distances in between.

The aircraft are either associated to the domestic flights or international ones. Each aircraft has a schedule defined by its arrival time and departure time specified by the known flight schedule and has a specified set of passengers.

The passengers are either transit passengers departing from other aircraft or non-transit passengers that enter to and exit from the airport using the same point.

We make the following additional assumptions about the airport facilities and properties of passengers:

- No towing is allowed.
- Aircraft behavior is as follows:
    - Firstly, aircraft arrives at the airport at the arrival time and immediately starts occupying its pre-assigned gate.
    - Aircraft continues to occupy the gate until the departure time.
    - Finally, at the departure time, aircraft immediately stops occupying the gate and leaves the airport.
- There are fixed gates in domestic and international terminals.
- There is only one remote gate in the airport that serves both terminals.
- The remote gate is extremely far from all other gates and the airport exit.
- Fixed gates can only handle one aircraft at a time, whereas the remote gate has unlimited capacity. Since the remote gate serves both terminals, there are no unassigned aircraft at any time.

- Transit passengers who will depart the airport from the same gate as they arrived cover zero distance.

Our aim is to minimize the total distance travelled by all passengers which can be expressed as sum of distance covered by non-transit passengers and transit passengers (entrance-to- gate + gate-to-exit + gate-to-gate) as illustrated by the following figure:

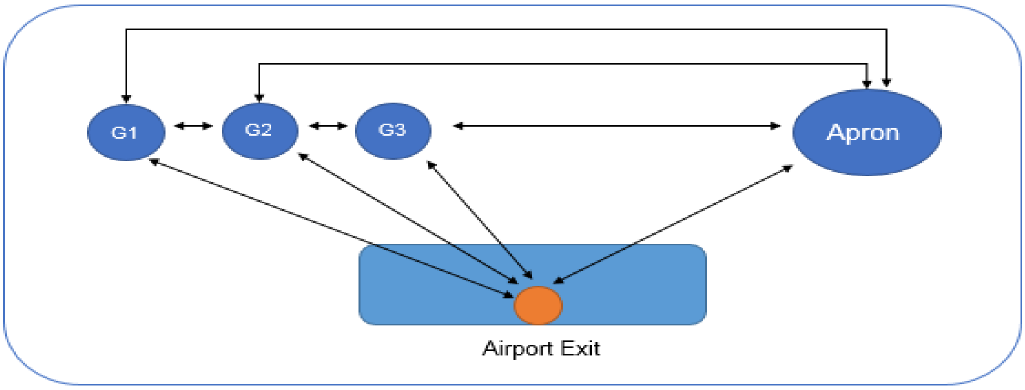

Figure 3.1. Illustration of Objective Function

In Section 3.1 we introduce our notation and give the nonlinear mathematical model and in Section 3.2 we discuss its linearization. We introduce an illustrative example in Section 3.3.

3.1. Nonlinear Mathematical Model

We assume that all parameters are known and not subject to any change, i.e., the system is deterministic and static.

The sets and parameters related to the aircraft are as follows:

I: set of all aircraft

n: total number of the aircraft (i.e., the cardinality of the set I, $|I|$)

I_D: set of all aircraft for domestic flights

I_I: set of all aircraft for international flights

p_{ij}: number of transit passengers between aircraft i and aircraft j ($\forall i, \forall j \in I$)

en_i: number of passengers coming from the entrance for aircraft i ($\forall i \in I$)

ex_i: number of passengers leaving the airport after aircraft i ($\forall i \in I$)

a_i: arrival time of aircraft i ($\forall i \in I$)

d_i: departure time of aircraft i ($\forall i \in I$)

Assume a chronological ordering of all arrival and departure times. Let $\{ad_1, ad_2, \ldots, ad_R\}$ be the associated sorted sequence with duplications removed and let

$$comp_{ir} = \begin{cases} 1, if\ aircraft\ i\ is\ in\ the\ airport\ at\ interval\ [ad_r, ad_{r+1}) \\ 0, otherwise \end{cases}$$

($\forall i \in I, \forall r \in \{1, R\text{-}1\}$)

The sets and parameters related to gates are as follows:

J: set of all gates

J_D: set of all gates at the domestic terminal (i.e., domestic fixed gates and remote gate)

J_I: set of all gates at the international terminal (i.e. international fixed gates + remote)

m: total number of fixed gates (i.e., $|J|$ - 1, $(m+1)^{th}$ gate is the remote gate)

d_{kl}: distance between gates k and l ($\forall k, \forall l \in J$)

ed_k: distance between airport entrance/exit point and gate k ($\forall k \in J$)

The decision variable is as follows:

$$x_{ik} = \begin{cases} 1, if\ aircraft\ i\ is\ assigned\ to\ gate\ k \\ 0, otherwise \end{cases} \qquad (\forall i \in I, \forall k \in J)$$

We consider minimization of total passenger walking distance in the airport. To discourage assignments to apron we use very long distances between any gate and apron. We also assume that the distance between entrance/exit point and the apron is too long.

The nonlinear mathematical formulation of the objective function is stated below:

$$min\ Z = \sum_{i=1}^{n-1} \sum_{k=1}^{m+1} \sum_{j=i+1}^{n} \sum_{l=1}^{m+1} p_{ij} d_{kl} x_{ik} x_{jl} + \sum_{i=1}^{n} \sum_{k=1}^{m+1} (en_i + ex_i)\ ed_k\ x_{ik}$$

The summation function is basically made of two terms. The first term corresponds to the total distance covered by the transit passengers. The distance covered by the transit passengers relies on two assignment decisions, making its function nonlinear. The second term corresponds to the total distance covered by the non-transit passengers.

The constraint sets are as follows:

$$\sum_{k \in J_D} x_{ik} = 1 \qquad (\forall i \in I_D) \qquad (1a)$$

$$\sum_{k \in J_I} x_{ik} = 1 \qquad (\forall i \in I_I) \qquad (1b)$$

$$\sum_{i=1}^{n} comp_{ir} x_{ik} \leq 1 \quad (\forall k \in (J \setminus \{(m+1)\}),\ r = 1, \dots, R\text{-}1) \qquad (2)$$

$$x_{ik} \in \{0, 1\} \qquad (\forall i \in I, \forall k \in J) \qquad (3)$$

Constraint sets (1a) and (1b) ensure the assignment of every aircraft to exactly one gate in its respective terminal, either remote or fixed. Constraint set (2) guarantees that there are no overlapping assignments in fixed gates. An overlapping assignment can be simply defined as an infeasible assignment of two different aircraft whose time intervals spent in the airport overlap to the same gate. Constraint set (3) ensures that the decision variable is binary for all $i \in I$ and $k \in J$.

3.2. Linear Mathematical Model

We now present a linear programming formulation of the airport gate assignment problem. We introduce a new decision variable as:

$$y_{ijkl} = \begin{cases} 1, if\ X_{ik}\ and\ X_{jl}\ are\ both\ equal\ to\ 1 \\ 0, otherwise \end{cases} \quad (\forall i, \forall j \in I; \forall k, \forall l \in J)$$

The linearized objective function becomes:

$$min\ Z = \sum_{i=1}^{n-1} \sum_{k=1}^{m+1} \sum_{j=i+1}^{n} \sum_{l=1}^{m+1} p_{ij}\, d_{kl}\, y_{ijkl} + \sum_{i=1}^{n} \sum_{k=1}^{m+1} (en_i +\ ex_i)\, ed_k\, x_{ik}$$

We introduce the following two new constraint sets:

$y_{ijkl} \geq\ x_{ik} + x_{jl} - 1$ $\quad (\forall i, \forall j \in I; \forall k, \forall l \in J$ and $i \neq j)$

$y_{ijkl} \geq 0$ $\quad (\forall i, \forall j \in I; \forall k, \forall l \in J)$

Note that y_{ijkl} would take value 1 only when both x_{ik} and x_{jl} are assigned to 1, as there are penalized with positive coefficients in the following linearized objective function:

$$min\ Z = \sum_{i=1}^{n-1} \sum_{k=1}^{m+1} \sum_{j=i+1}^{n} \sum_{l=1}^{m+1} p_{ij}\, d_{kl}\, y_{ijkl} + \sum_{i=1}^{n} \sum_{k=1}^{m+1} (en_i +\ ex_i)\, ed_k\, x_{ik}$$

The constraint sets of the mixed integer linear programming model are as stated below:

$\sum_{k \in J_D} x_{ik} = 1 \quad (\forall i \in I_D)$ (1a)

$\sum_{k \in J_I} x_{ik} = 1 \quad (\forall i \in I_I)$ (1b)

$\sum_{i=1}^{n} comp_{ir} x_{ik} = 1 \quad (\forall k \in (J \setminus \{(m+1)\}),\ r = 1, \ldots, R\text{-}1)$ (2)

$x_{ik} \in \{0, 1\} \quad (\forall i \in I, \forall k \in J)$ (3)

$y_{ijkl} \geq\ x_{ik} + x_{jl} - 1 \quad (\forall i, \forall j \in I; \forall k, \forall l \in J$ and $i \neq j)$ (4)

$y_{ijkl} \geq 0 \quad (\forall i, \forall j \in I; \forall k, \forall l \in J)$ (5)

3.3. Complexity of the Problem

Obata (1979) shows that the gate assignment problem is NP-hard in the strong sense. Thus, the size of the search grows exponentially with the increases in the problem size.

In our models, we use 8 parameters that are tabulated in Table 3.1.

Table 3.1. *Numbers of Parameters in Nonlinear and Linear Models*

Parameter	Number
$\boldsymbol{n}$	1
$\boldsymbol{m}$	1
$\boldsymbol{d_{kl}}$	$(m+1)^2$
$\boldsymbol{p_{ij}}$	n^2
$\boldsymbol{en_i}$	n
$\boldsymbol{ex_i}$	n
$\boldsymbol{ed_k}$	$m+1$
$\boldsymbol{comp_{ir}}$	n x R
Total	$\boldsymbol{n\,(n+2+R)+}$ $\boldsymbol{(m+2)(m+1)+\ 2}$

The number of decision variables in nonlinear and linear models are tabulated below:

Table 3.2. *Number of Decision Variables in Nonlinear Model*

Decision Variable	Number
$\boldsymbol{x_{ik}}$	n x $(m+1)$
Total	$\boldsymbol{n}$ **x** $\boldsymbol{(m+1)}$

Table 3.3. *Number of Decision Variables in Linear Model*

Decision Variable	**Number**
$\boldsymbol{x_{ik}}$	n x $(m+1)$
$\boldsymbol{y_{ijkl}}$	n^2 x $(m+1)^2$
Total	$\boldsymbol{n\ (m+1)}$ **x** $\boldsymbol{(n\ (m+1)+1)}$

The numbers of constraint sets in nonlinear and linear models are as follows:

Table 3.4. *Number of Constraints in Nonlinear Model*

Constraint Set	**Number**
1a	$\|I_D\|$
1b	$\|I_I\|$
2	m x R
3	n x $(m+1)$
Total	$\boldsymbol{\|I_D\| + \|I_I\| + m}$ **x** $\boldsymbol{R +}$ $\boldsymbol{n}$ **x** $\boldsymbol{(m+1)}$

Table 3.5. *Number of Constraints in Linear Model*

Constraint Set	**Number**
1a	$\|I_D\|$
1b	$\|I_I\|$
2	m x R
3	$n\ (m+1)$
4	$n\ (n-1)$ x $(m+1)\ (m+1)$
5	$n^2\ (m+1)^2$
Total	$\boldsymbol{(m+1)^2\ n^2 +}$ $\boldsymbol{(m+1)^2}$ **x** $\boldsymbol{(n-1)}$ **x** $\boldsymbol{n}$ $\boldsymbol{+ (m+1)}$ **x** $\boldsymbol{n}$ $\boldsymbol{+ \|I_D\| + \|I_I\|}$ $\boldsymbol{+ m}$ **x** $\boldsymbol{R}$

3.4. An Example Problem

In this section, we present an example airport gate assignment problem where $n = 10$, $m = 5$ (+ remote gate).

In order to find a feasible solution, no more data than the information of arrival and departure times is required. The flight schedule and the sets of overlaps for each aircraft is as follows:

Table 3.6. *Domestic and International Aircraft in the Example Problem*

Type of Travel	*Aircraft*
Domestic	1, 3, 5, 7, 9
International	2, 4, 6, 8, 10

Table 3.7. *Domestic and International Gates in the Example Problem*

Terminals	*Gates*
Domestic	1, 3, 5, Remote
International	2, 4, Remote

Table 3.8. *Flight Schedule and Sets of Overlaps in the Example Problem*

Index of Aircraft (#)	*Arrival Time (min)*	*Departure Time (min)*	*Overlapping Aircraft*
1	4	56	2, 3
2	29	80	1, 3, 4
3	43	84	1, 2, 4, 5
4	58	91	2, 3, 5
5	83	130	3, 4, 6
6	127	165	5
7	206	260	8, 9
8	233	282	7, 9
9	239	270	7, 8
10	299	352	none

Using the given data, a feasible assignment is as follows:

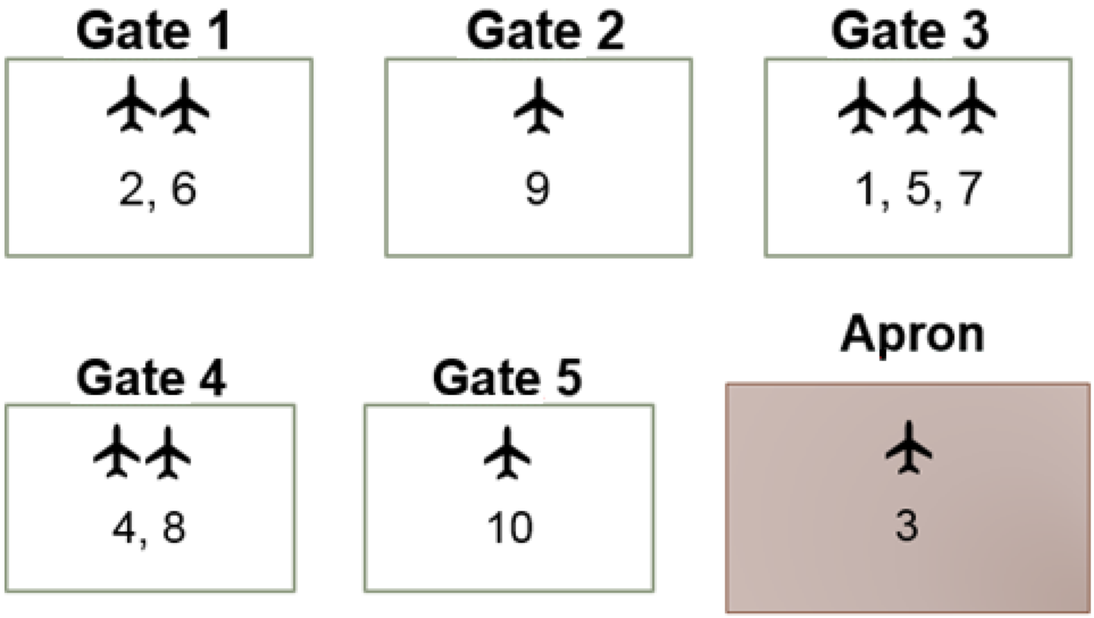

Figure 3.2. Illustration of a Feasible Solution of the Example Problem

To find the optimal solution, number of transit and non-transit passengers, distances between gates and distances from gates to the airport entrance are needed. Number of transit passengers between aircraft are as illustrated below:

Table 3.9. *Number of Transit Passengers Between Aircraft in the Example Problem*

Aircraft	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	26	11	6	5	2	9	17	26	29	14
2	11	9	27	15	28	3	26	8	2	22
3	6	27	0	11	12	15	10	14	7	7
4	5	15	11	29	3	26	16	13	5	2
5	2	28	12	3	21	2	22	22	2	14
6	9	3	15	26	2	19	5	7	21	12
7	17	26	10	16	22	5	1	18	12	24
8	26	8	14	13	22	7	18	13	5	1
9	29	2	7	5	2	21	12	5	1	15
10	14	22	7	2	14	12	24	1	15	20

Number of non-transit passengers in each aircraft is as follows:

Table 3.10. *Number of Non-Transit Passengers in the Aircraft in the Example Problem*

Aircraft	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Passengers	60	47	89	31	60	55	43	19	26	60

Distances between gates and the airport entrance and exit point are as follows:

Table 3.11. *Distances from Gates to Airport Entrance & Exit in the Example Problem*

Gates	**1**	**2**	**3**	**4**	**5**	**Remote**
Distance	34	69	71	83	95	9999

Distances between gates are as illustrated below:

Table 3.12. *Distances between Gates in the Example Problem*

Gates	**1**	**2**	**3**	**4**	**5**	**Remote**
1	0	177	40	76	124	9999
2	177	0	162	150	56	9999
3	40	162	0	27	48	9999
4	76	150	27	0	148	9999
5	124	56	48	148	0	9999
Remote	9999	9999	9999	9999	9999	9999

The optimal solution of the problem is assigning the aircraft 3 and 7 to Gate 1, the aircraft 1, 5 and 9 to Gate 3, the aircraft 4 to Gate 4 and the aircraft 2, 6, 8 and 10 to Gate 5. No aircraft are assigned to Gate 2 and the apron. An illustration of the optimal solution is as follows:

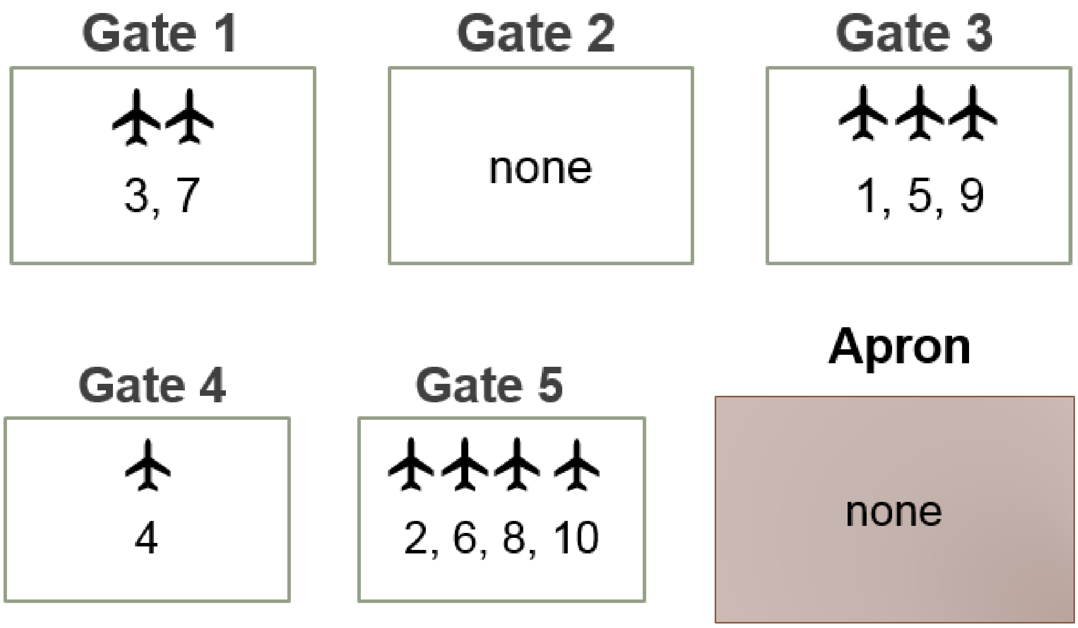

Figure 3.3. Illustration of the Optimal Solution of the Example Problem

CHAPTER 4

BRANCH-AND-BOUND ALGORITHM

The results of our initial experiments have shown that our mathematical model could handle on only small-sized instances. This gives rise to a need for an implicit enumeration technique, like a branch-and-bound algorithm for finding optimal solutions for medium and large-sized instances.

We propose a branch-and-bound algorithm for an implicit enumeration of all feasible solutions. The algorithm uses an efficient branching scheme, lower bounds and an upper bound, each of which is discussed below.

4.1. Branching Scheme

As a pre-processing, we index the gates in nondecreasing order of their closeness to the airport entrance & exit point. Accordingly, gate 1 is the closest gate and gate m+1, i.e. remote gate, is the farthest gate, to the airport entrance & exit point. Moreover, the aircraft are indexed by their arrival times to the airport. Accordingly, aircraft 1 has the earliest arrival time and aircraft n is the latest arriving one.

Level i of the branch-and-bound tree represents the assignment of aircraft i. Node k at level i represents the assignment of gate k to aircraft i. Hence, the branch-and-bound tree has at most n levels and each level has at most m+1 nodes. The nodes representing the infeasible assignments are not created. Level 0 represents the root node with no assignments.

Our branch-and-bound algorithm employs a depth-first strategy. At any level, it selects the lowest index unexplored node and goes to the succeeding levels. Hence, the first assignment is always gate 1 and the next assignment is gate 1 if aircraft 1 and 2 are non-overlapping and gate 2 if they do overlap.

Figure 4.1 illustrates the branch-and-bound tree for $n = 2$ aircraft and $m = 2$ gates. We assume that both aircraft serve domestic flights and they do overlap. Node A represents the remote gate (apron).

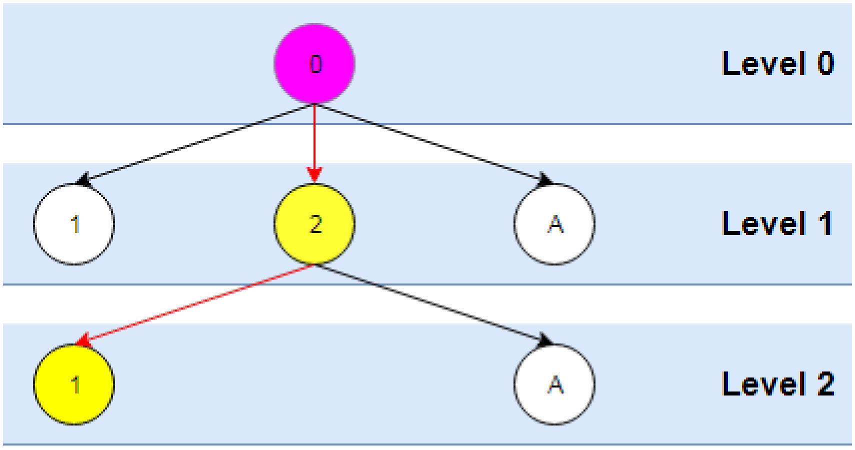

Figure 4.1. A Branch-and-Bound Tree for $n = 2$ and $m = 2$

In Figure 4.1 at level 2, node 2 is infeasible as the aircraft are overlapping, hence it is not created. Yellow nodes and red arcs indicate the selected gates and path, respectively. The selection is gates 2 and 1, for aircraft 1 and 2, respectively.

Now consider an instance with $n = 3$ aircraft serving domestic flights and $m = 2$ gates. The following tree illustrates all possible branchings when all aircraft are overlapping.

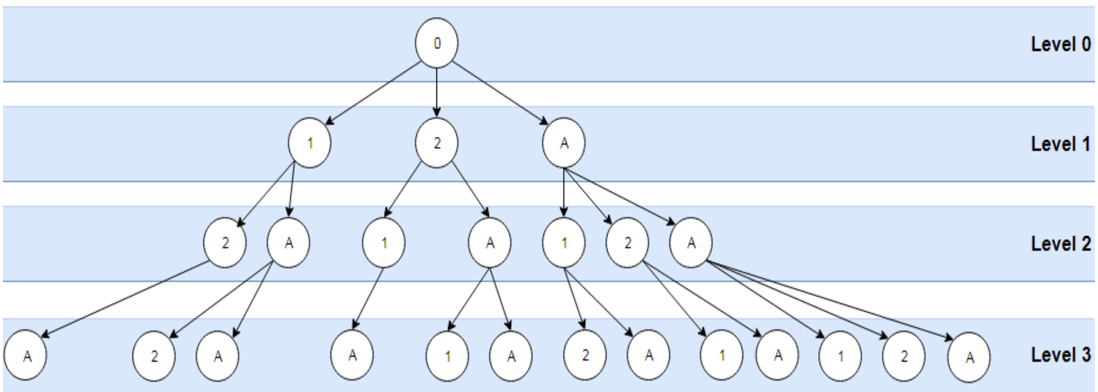

Figure 4.2. A Branch-and-Bound Tree for $n = 3$ All Overlapping Aircraft and $m = 2$

As stated, the search starts with aircraft 1 and gate 1. After the first assignment is done, the first three feasible assignments are shown in Figures 4.3, 4.4 and 4.5. We use notation (i, k) to denote the assignment of aircraft i to gate k.

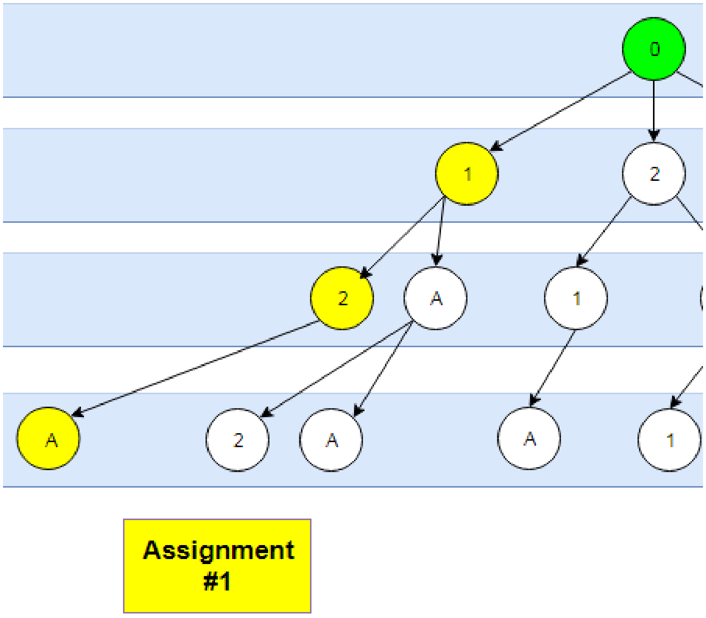

Figure 4.3. A Branch-and-Bound Tree Representing 1st Assignment After (1, 1)

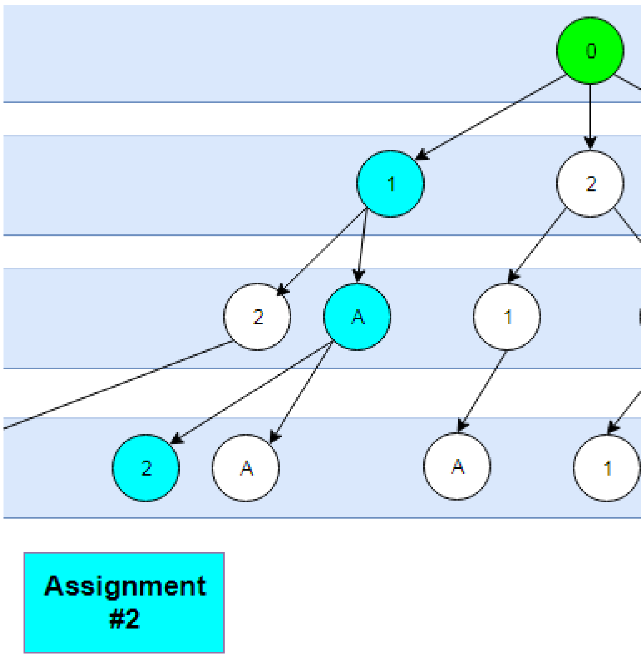

Figure 4.4. A Branch-and-Bound Tree Representing 2nd Assignment After (1, 1)

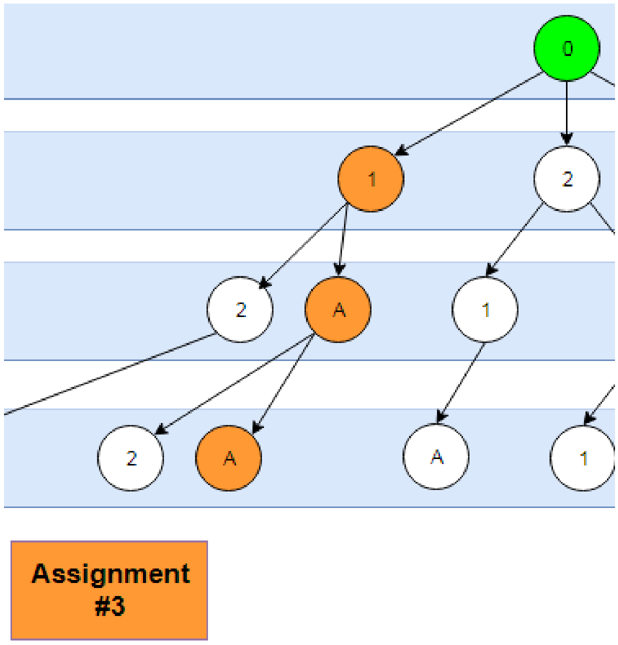

Figure 4.5. A Branch-and-Bound Tree Representing 3rd Assignment After (1, 1)

All feasible solutions are enumerated similarly, and the optimal assignment set is found. The enumeration is implicit as the nonpromising assignments are eliminated via the lower bounds, which are discussed next.

4.2. Lower Bounds

To increase the efficiency of the search process, we calculate a lower bound after each feasible aircraft assignment.

Note that our objective function has two components: one for non-transit passengers and one for transit passengers. Considering this nature, we develop two separate lower bounds. We first discuss the lower bound for the root node and then its extensions to the partial solutions.

4.2.1. Non-Transit Passengers Walking Distance Lower Bound

The distance that would be covered by all non-transit passengers is the number of non-transit passengers of aircraft i times the distance between the gate of aircraft i and the airport entrance & exit point. The distance of the gate of aircraft i is a decision that could be underestimated by the distance of gate 1 to airport entrance & exit point as gate 1 is always the closest gate due to our indexing. Hence a valid lower bound is;

$$\sum_{i=1}^{n} nt_i \text{ x } ed_1 = ed_1 \text{ x} \sum_{i=1}^{n} nt_i = ed_1 \text{ x } NT$$

where;

nt_i: total number of non-transit passengers in aircraft i ($\forall i \in I$)

$$nt_i = en_i + ex_i \ (\forall i \in I)$$

NT: total number of non-transit passengers in all aircraft

$$NT = \sum_{i=1}^{n} nt_i$$

This lower bound can be improved by considering the maximum number of non-transit passengers that can be assigned to gate 1. The maximum number is found through a longest path algorithm where there are connections between two non-overlapping aircraft and the arc weights are represented by the number of non-transit passengers. The arc weight from aircraft i to aircraft j is nt_i. The following network depicts the flows through any gate. Note that as the nodes are ordered, the network is acyclic.

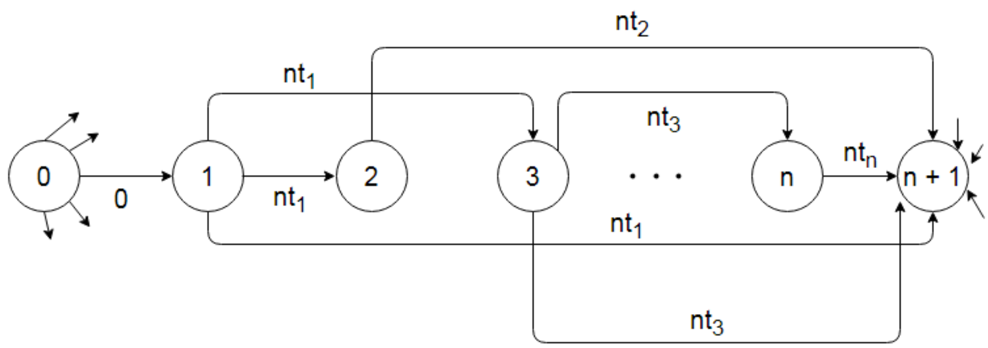

Figure 4.6. A Generalized Network of Non-Transit Passengers

The above network has n+1 arcs departing from node 0, one to each aircraft and one to terminal node n+1. There are n+1 arcs arriving to node (n+1), each from one aircraft and one from the source node (node 0). The longest path between node 0 and node (n+1) gives the maximum number of passengers that can be feasibly assigned (without any overlaps) to any one of the gates, hence to gate 1.

Consider an instance with n = 5 aircraft that serve domestic flights. The data are tabulated as follows:

Table 4.1. *Numbers of Non-Transit Passengers in the Aircraft in Example LP*

Aircraft	**Number of Non-Transit Passengers**
1	15
2	25
3	20
4	15
5	15

Table 4.2. *Lists of Overlapping Aircraft for Each Aircraft in Example LP*

Aircraft	**Overlapping Aircraft**
1	2, 3
2	1, 3, 4, 5
3	1, 2, 4, 5
4	2, 3
5	2, 3

The corresponding network is as follows:

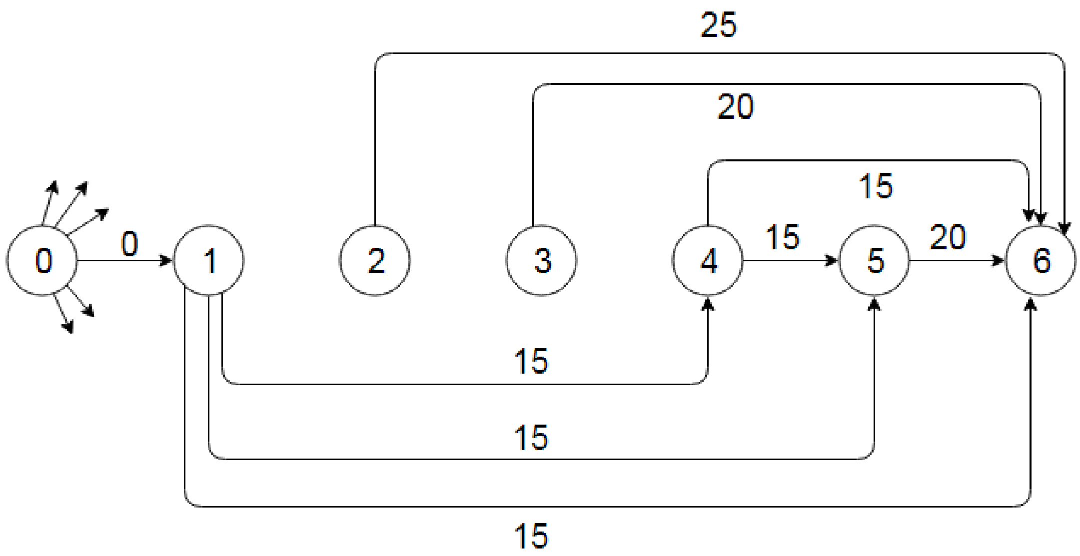

Figure 4.7. A Network of Passengers at the Instance Problem

The longest path is 0 → 1 → 4 → 5 → 6 and TD is its length, hence the maximum number of passengers that can served by one gate. We assume TD passengers are assigned to gate 1 and the rest are assigned to the second closest gate to the airport entrance & exit point, hence gate 2, and get the following expression:

$$LB_{NT} = TD * ed_1 + (NT - TD)\ ed_2$$

LB_{NT} is found for domestic and international aircraft separately and their sum is taken.

For any level s where the first s aircraft are assigned, we find the longest path considering the $(n - s)$ unassigned aircraft. Let TD_s be the length of the longest path for unassigned aircraft at level s.The corresponding lower bound is as follows:

$$LB_{NT} = TD_s\ ed_1 + (\sum_{i=s+1}^{n} nt_i - TD_s)\ ed_2$$

The realized walking distance RC_{NT} is as follows:

$$RC_{NT} = \sum_{i=1}^{s} \sum_{k=1}^{m+1} nt_i\ ed_k\ x_{ik}$$

Hence, the overall lower bound for non-transit passengers walking distance is calculated as follows:

$$LB_{NT} + RC_{NT}$$

We write the longest path algorithm for a generic network with n+1 nodes. For domestic flights the corresponding network will have n_D+1 nodes (indexed from 1 to n_D+1, where n_D is the number of domestically serving aircraft considered), while for international for international flights it will have n_I+1 nodes, (indexed from 1 to n_I+1, where n_I is the number of internationally serving aircraft considered).

$maxT_i$: longest path value from node i to the end node n+1

Let NO_i be the set of non-overlapping aircraft for aircraft i(including dummy aircraft n+1)

The algorithm is based on the following recursive equation:

$$maxT_i = \max_{j \geq i:\, j \in NO_i} nt_i + maxT_j$$

Stepwise description of the algorithm is as follows:

Step 0. $maxT_{n+1} = 0$

Step 1.

for ($i = n,\ i\ \geq 0,\ i = i - 1$) {

$maxT_i = 0$

for ($j = i + 1,\ j \leq n + 1,\ j = j + 1$) {

if ($j \in NO_i$ and $nt_i +\ maxT_j > maxT_i$) {

$maxT_i =\ nt_i +\ maxT_j$} // end if

} // end for j

} // end for i

Step 2. Return $maxT_i$ values.

4.2.2. Transit Passengers Walking Distance Lower Bound

We now explain the transit passenger walking distance lower bound calculations. We first discuss the calculations at level 0 and then at an arbitrary level $s > 0$.

At Level 0: At level 0, when all aircraft are unassigned, the lower bound for the distance travelled by all transfer passengers is found as follows:

Case 1. The aircraft (i, j) are not overlapping.

In that case, any non-overlapping aircraft can be assigned to the same gate with distance 0.

Case 2. The aircraft (i, j) are overlapping.

Overlapping aircraft (i, j) should be assigned to different gates. The minimum distance of travel will be the shortest distance between all gate pairs, i.e. $\min_{(k,l)}\{d_{kl}\}$. The lower bound for the travel distance between the aircraft (i, j) is

$$p_{ij} \text{ x } \min_{(k,l)}\{d_{kl}\}$$

Therefore, the overall lower bound is;

$$LB_T = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} o_{ij}\, p_{ij} \min_{(k,l)}\{d_{kl}\} \qquad (\forall k, \forall l \in J)$$

where $o_{ij} = \begin{cases} 1, if\ aircrafts\ i\ and\ j\ are\ overlapping \\ 0, otherwise \end{cases}$

At Level s > 0: At any level of branch-and-bound tree, LB_T is extended, considering the assigned and not yet assigned aircraft pairs. For level $s \in I$, we consider the following three cases:

Case 1. $i, j \leq s$, they are both assigned. The cost of assignment is the realized travel distance of transit passengers, RC_T, where;

$$RC_T = \sum_{i=1}^{s-1} \sum_{j=i+1}^{s} \sum_{k=1}^{m+1} \sum_{l=1}^{m+1} p_{ij}\, d_{kl}\, x_{ik}\, x_{jl}$$

For the other cases, we find eligible sets for each unassigned aircraft i using the gates of the assigned aircraft that overlap with aircraft i. A gate is said to be eligible for aircraft i if none of its overlapping aircraft have yet been assigned to that gate. We let $e(i)$ be the set of eligible gates for aircraft i. Note that at the root node when all the gates are empty, $e(i)$ includes all gates. Once an assignment to gate k is made by an overlapping aircraft with i, then $e(i)$ is updated as $\{e(i) \setminus k\}$.

Case 2. $i \leq s,\ j > s$, i.e., i is assigned, j is not yet assigned.

Let g_i be the gate that aircraft i is assigned. There are two cases to be considered based on whether i and j are overlapping.

Case 2.1. i and j are non-overlapping.

The minimum distance is zero, as j can be assigned to g_i.

Lower bound contribution, LB_{Tij}, is zero.

Case 2.2. i and j are overlapping

The minimum distance of travel is $\min_{\substack{k \neq g_i \\ k \in e(j)}} \{d_{g_i,k}\}$.

The lower bound on the realized cost is as follows:

$$LB_{Tij} = p_{ij} \text{ x } \min_{\substack{k \neq g_i \\ k \in e(j)}} \{d_{g_i,k}\}$$

Case 3. $i > s, j > s$, i.e., i and j are both unassigned. LB_{Tij} is calculated as follows:

$$LB_{Tij} = p_{ij} \text{ x } \min_{\substack{k \in e(i) \\ l \in e(j)}} \{d_{kl}\}$$

Note that Case 1 produces realized walking distance, Case 2 produces a lower bound using realized assignments and Case 3 produces a lower bound for not-yet-made assignments. Therefore, the overall lower bound is the sum of Case 2 and Case 3 lower bounds, i.e.;

$$LB_T = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} LB_{Tij}$$

Any partial solution for which the following relation holds cannot lead to a unique optimal solution:

UB: the best known objective value

$$LB_P = LB_T + RC_T + LB_{NT} + RC_{NT} \geq UB$$

In our implementation we apply the lower bounds in a sequel. We first find the realized travel distances and fathom the partial solution if the following relation holds:

$$RC_T + RC_{NT} \geq UB$$

If not, the partial solution may lead to an optimal solution. We then calculate the lower bounds in sequel. Firstly, we calculate LB_{NT} and check whether the following relation holds:

$$RC_T + RC_{NT} + LB_{NT} \geq UB$$

If realized travel distances combined with LB_{NT} cannot eliminate the node, then we compute LB_T. Note that we are proceeding from easier to compute bound to a more difficult one in order to increase the speed of elimination. Figure 4.8 below illustrates the hierarchy of our application:

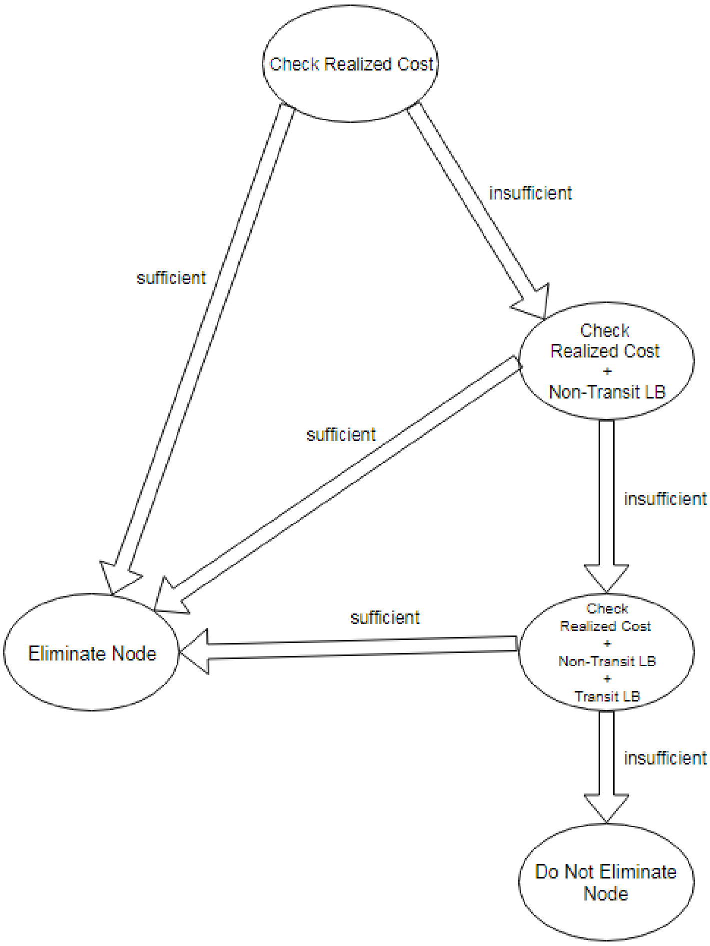

Figure 4.8. The Hierarchy of Using Lower Bounds

If the lower bound at a partial solution is outweighed by the current best known upper bound, we fathom the node and proceed to the next one. If all nodes at a level are eliminated by the upper bound, we backtrack to the previous level.

The upper bound is first found by a heuristic procedure discussed next and it is updated whenever a complete assignment with a better objective function value is reached. The upper bound solution at termination is the optimal solution.

4.3. Initial Upper Bound

Our branch-and-bound algorithm starts with an upper bound. We separate the domestic terminal gate assignments and the international terminal gate assignments and find the upper bounds separately. The sum of these upper bounds gives the global upper bound which is used to start the branch-and-bound algorithm.

Our upper bounding procedure decomposes the problem into subproblems. Each subproblem represents an assignment to a particular gate k. We start with gate 1 and terminate whenever all gates are scheduled, or all aircraft are assigned.

Subproblem 1 considers all aircraft and solves a longest path problem for gate 1. The longest path problem is the same as the one defined for lower bound for non-transit passengers except the arc weights. The arc weights between the aircraft i and j are the number of transit passengers between these two aircraft plus the number of non-transit passengers arriving at and departing by aircraft i. Arc weights between the root node and aircraft i are zero and arc weight between aircraft i and the dummy end node is just the number of non-transit passengers arriving at and departing by aircraft i. Below is the associated longest path network:

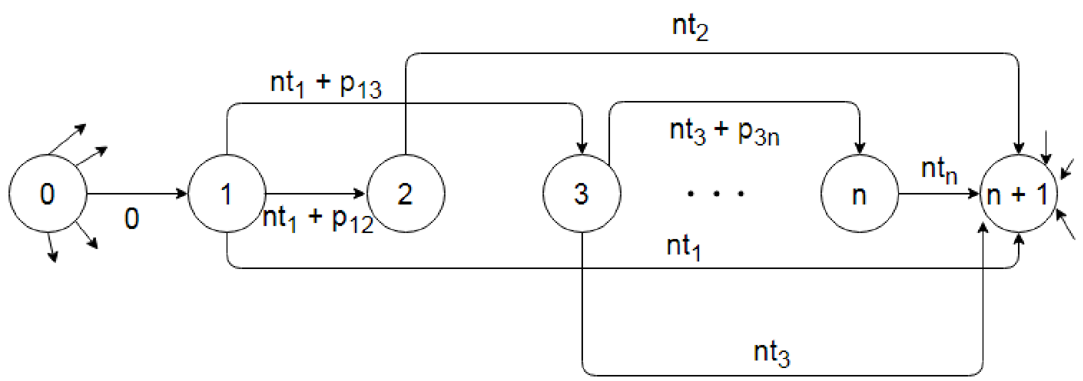

Figure 4.9. Example Network for Upper Bound Evaluation

The length of the longest path from node 0 to node n+1 gives the number of transit passengers whose arrival and departure aircraft are assigned to be same gate once the non-transit passengers are ignored. Maximizing the length of the path would minimize travel by transit passengers. We also count the number of non-transit passengers as their travel distance should also be shortened.

After the aircraft are assigned to gate 1, we update the aircraft set and again solve a longest path problem with the reduced aircraft set for gate 2. Recall that gate 2 is the second closest gate to airport entrance & exit point and should receive higher priority for assignment due to non-transit passengers. We continue for gate 3 and so on, till the assignment schedule is complete. Upper bound is the objective function of the resulting schedule.

Below is the stepwise description of our upper bound heuristic:

Step 0. N: set of aircraft, set k = 1.

Step 1. Solve the longest path problem for gate k with set N with arc weights $nt_i + p_{ij}$ between aircraft i and j, 0 between node 0 and aircraft i and t_i between aircraft i and node n+1.

Step 2. Let S_k be the set of aircraft appearing on the longest path. Set $N = N \setminus S_k$. If $N = \emptyset$ or $k = m$, go to Step 3. Else, set $k = k + 1$ and go to Step 1.

Step 3. Upper bound schedule is formed by S_k (set of aircraft at gate k). UB is the objective function of the schedule.

We illustrate the upper bound heuristic via a 5-aircraft and 2-fixed gate example with the following data:

Table 4.3. *Overlapping Aircraft for Each Aircraft in Example for UB Heuristic*

Aircraft	Overlapping Aircraft
1	2, 3
2	1, 3, 4, 5
3	1, 2, 4, 5
4	2, 3
5	2, 3

Table 4.4. *Non-Transit Passengers in Each Aircraft in Example for UB Heuristic*

Aircraft	Number of Non-Transit Passengers
1	15
2	40
3	45
4	15
5	15

Table 4.5. *Transit Passengers Between Aircraft in Example for UB Heuristic*

Aircraft	*1*	*2*	*3*	*4*	*5*
1	5	15	25	10	15
2	15	10	10	20	5
3	25	10	5	10	30
4	10	20	10	10	15
5	15	5	30	15	10

Table 4.6. *Distances Between Gates in Example for UB Heuristic*

Distance	*1*	*2*	*3 (Apron)*
1	0	50	9999
2	50	0	9999
3	9999	9999	9999

Table 4.7. *Distances Between Gates and Entrance in Example for UB Heuristic*

Gate	Distance to Entrance & Exit
1	50
2	100
3 (Apron)	9999

The longest path network is as follows:

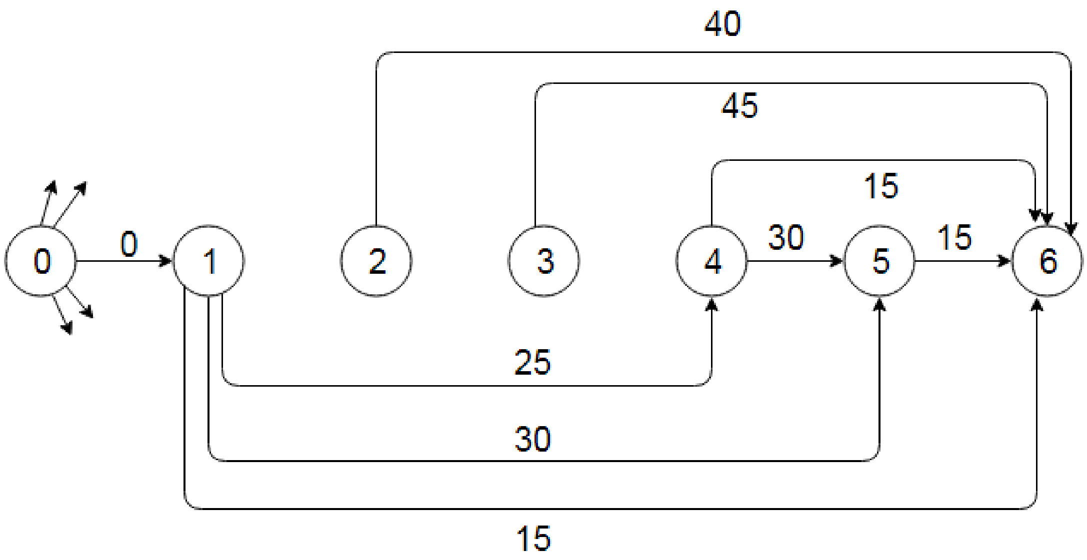

Figure 4.10. Longest Path for Upper Bound Heuristic Example

The longest path to the end node is 0 → 1 → 4 → 5 → 6 with length 70. On the path, the aircraft 1, 4 and 5 are assigned to gate 1 which is the closest to the airport entrance & exit point. The network of unassigned aircraft is as follows:

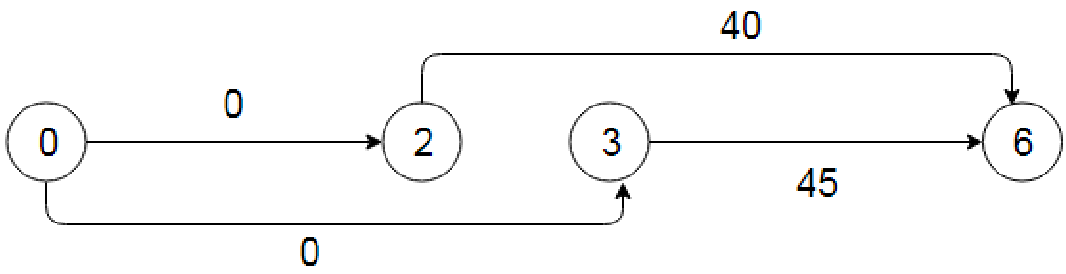

Figure 4.11. Example Network After an Iteration at LP Algorithm for Upper Bound

The longest path is 0 → 3 → 6 for gate 2. The third longest path is 0 → 2 → 6 for apron. Therefore, aircraft 3 is assigned to gate 2 and aircraft 2 is assigned to apron.

The total distance associated with this assignment is an upper bound that is found as 1059895.

CHAPTER 5

COMPUTATIONAL RESULTS

To test the performance of the mathematical model and branch and bound algorithm, we conducted computational experiments on randomly generated instances. We first describe the data generation scheme. Then, we discuss the results of the mathematical model and branch and bound algorithm.

5.1. Data Generation

We randomly generated instances of different sizes using a random number generator on Java.

We use Ankara Esenboğa Airport for the layout data, i.e., the distances between pairs of fixed gates and distances between fixed gates and the airport entrance & exit point. All distances are estimated using public satellite images of the airport building. Due to intense investigation process and additional control points, we assume that all gates located in the international terminal are farther from the airport entrance & exit point compared to the gates in the domestic terminal. Moreover, we assume that the distance between an international gate and a domestic gate is significantly longer than two gates located in the same terminal. The remote gate is assumed to be located extremely far away from all fixed gates; therefore, a big-M number is assigned to every distance parameter it is associated with.

In our test instances, we consider three different layouts, i.e., number of fixed gates (m): 5, 7 and 10. Data for each case (layout) are as follows:

Case 1. m = 5, J_D = {1, 2, 3, 6}, J_I = {4, 5, 6}

$\{ed_k\}$ = {50, 100, 150, 300, 350, M}

Table 5.1. *Distance Matrix of Gates (m = 5)*

Distance	1	2	3	4	5	Remote
1	0	50	100	275	325	M
2	50	0	50	325	375	M
3	100	50	0	375	425	M
4	275	325	375	0	50	M
5	325	375	425	50	0	M
Remote	M	M	M	M	M	M

Case 2. m = 7, J_D = {1, 2, 3, 4, 8}, J_I = {5, 6, 7, 8}

$\{ed_k\}$ = {50, 100, 150, 200, 300, 350, 400, M}

Table 5.2. *Distance Matrix of Gates (m = 7)*

Distance	1	2	3	4	5	6	7	Remote
1	0	50	100	150	275	325	375	M
2	50	0	50	100	325	375	425	M
3	100	50	0	50	375	425	475	M
4	150	100	50	0	425	475	525	M
5	275	325	375	425	0	50	100	M
6	325	375	425	475	50	0	50	M
7	375	425	475	525	100	50	0	M
Remote	M	M	M	M	M	M	M	M

Case 3. m = 10, J_D = {1, 2, 3, 4, 5, 11}, J_I = {6, 7, 8, 9, 10, 11}

$\{ed_k\}$ = {50, 100, 150, 200, 250, 300, 350, 400, 450, 500, M}

Table 5.3. *Distance Matrix of Gates (m = 10)*

Distance	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Remote**
1	0	50	100	150	200	275	325	375	425	475	M
2	50	0	50	100	150	325	375	425	475	525	M
3	100	50	0	50	100	375	425	475	525	575	M
4	150	100	50	0	50	425	475	525	575	625	M
5	200	150	100	50	0	475	525	575	625	675	M
6	275	325	375	425	475	0	50	100	150	200	M
7	325	375	425	475	525	50	0	50	100	150	M
8	375	425	475	525	575	100	50	0	50	100	M
9	425	475	525	575	625	150	100	50	0	50	M
10	475	525	575	625	675	200	150	100	50	0	M
Remote	M	M	M	M	M	M	M	M	M	M	M

For each m value, we created instances by varying the number of aircraft n. The combinations we used are as in Table 5.4. We categorized these instances as small, medium and large sized instances. We generate 10 problem instances for each (n, m) combination.

Table 5.4. *m, n Values of the Test Instances*

<table>
<tr><td rowspan="6">Small-sized</td><td>m</td><td>n</td><td rowspan="6">Medium sized</td><td>m</td><td>n</td><td rowspan="6">Large-sized</td><td>m</td><td>n</td></tr>
<tr><td rowspan="3">5</td><td>10</td><td>5</td><td>25</td><td>5</td><td>30</td></tr>
<tr><td>15</td><td rowspan="2">7</td><td>20</td><td>7</td><td>30</td></tr>
<tr><td>20</td><td>25</td><td rowspan="3">10</td><td>25</td></tr>
<tr><td rowspan="2">7</td><td>10</td><td rowspan="2">10</td><td>15</td><td></td></tr>
<tr><td>15</td><td>20</td><td></td></tr>
</table>

The other parameters are generated using discrete uniform distributions (DU) as follows:

- Number of transit passengers between aircraft i and j (p_{ij}): $DU(0, 200/n)$.
- Number of non-transit passengers in aircraft i (nt_i): $DU(0, 100)$.
- Arrival time of an aircraft (a_i) (minutes): $DU(0, 300)$.
- Duration of stay of an aircraft at the airport (dur_i) (minutes): $30 + DU(0, 30)$.
- Departure time of an aircraft i (d_i) is set to $a_i + dur_i$.

The maximum number of passengers in an aircraft is taken as 300 based on information provided on the website of a commercial airline company (Turkish Airlines, 2019). We assume that total number of transit passengers does not exceed 200 and that a transit passenger has the equal probability of connection to all other aircraft. We assumed that maximum 50 passengers go from an aircraft to airport exit and maximum 50 passengers arrive at airport from outside for the same aircraft. The total number of non-transit passengers in an aircraft is calculated by the summation of these two, hence it is generated using $DU(0, 100)$.

5.2. Computational Results

In this section we provide the results of our computational experiments. We set a time limit of 1 hour for both mathematical model and B&B algorithm. We do not experiment on larger instances when at least 3 out of 10 instances cannot be solved in one hour for an (n, m) combination.

The B&B algorithm is coded in Java and solved by a quad-core (Intel Core i7 2.30 GHz) computer with 8 GB RAM. All models are solved by CPLEX 12.8.0. The solution times are expressed in Central Processing Unit (CPU) seconds.

We first report on the performance of the proposed lower bound (LB), calculated at the root node, in Table 5.5. For each (n, m) combination, the table shows the number of instances that could be solved to optimality by the B&B algorithm within the time limit and the average and minimum (worst case) values of percentage of lower bound to the optimal value (100 x LB / OPT), calculated over these instances.

Table 5.5. *Performance of the Proposed LB Calculated at the Root Node*

n	m	# of instances solved	Average LB / OPT (%)	Min LB / OPT (%)
10	*5*	10	30.932	5.214
	7	10	38.104	32.187
	5	10	27.501	3.254
15	*7*	10	33.565	7.858
	10	10	33.739	30.633
	5	10	4.965	2.827
20	*7*	10	26.962	8.741
	10	2	29.712	28.014
25	*5*	10	2.689	2.011
	7	10	19.481	9.555
30	*5*	8	2.530	2.055
	7	7	7.927	4.399
35	*5*	0	-	-

It is seen that for most of the (n, m) combinations, the average percentage values range from 27% to 38%. It is also seen from the minimum percentage values that there are instances for which the LB/OPT ratio is too low. However, the average values are significantly larger than the minimum values, indicating that the instances with too poor LB performance are quite few. For example, for m = 5, n = 15 the average percentage is 27.5 despite the minimum percentage of 3.2.

It is also observed that for fixed m, the performance of the LB decreases as n increases. This is related to the fact that as n increases, the number of flights that should be assigned to the apron in the optimal solution increases.

As the $\frac{n}{m}$ ratio increases further, the performance of the LB deteriorates significantly; as seen in the problem instances with m = 5, n = 20, 25, 30 and m = 7, n = 30. This is again in line with the observation that apron usage affects the performance of the LB (average number of flights assigned to the apron was 2.4, 5.3, 6.5 and 2.0 for (20, 5), (25, 5), (30, 5) and (30, 7) sets, respectively).

We report on the performance of two solution approaches, the mathematical model and B&B algorithm, in Table 5.6. We report the average and maximum solution times for both approaches and the average and maximum values for the number of nodes in the B&B algorithm. We also report the number of instances that could not be solved optimally within the time limit of 1 hour (in parentheses, in the columns showing maximum CPU Time).

Table 5.6. *Comparison of Two Solution Approaches*

n	m	# of instances solved	Mathematical Model		Branch-and-Bound Algorithm			
			CPU Time (s)		CPU Time (s)		# of Nodes	
			Average	Maximum	Average	Maximum	Average	Maximum
10	*5*	10	0.185	0.225	0.009	0.032	1706	2552
	7	10	0.557	0.623	0.034	0.073	9292	20467
15	*5*	10	0.448	0.734	0.109	0.548	15900	64221
	7	10	3.386	6.534	1.763	7.190	317998	1395240
	10	10	71.206	223.527	15.014	40.219	2397353	6487742
20	*5*	10	2.811	5.717	1.362	3.398	141410	415150
	7	10	67.329	207.807	51.247	105.047	7731576	22071977
	10	2	2289.767	3600 (8)	1593.471	3600 (8)	114750695	130540698
25	*5*	10	56.850	263.286	42.521	151.478	4054587	17305083
	7	10	636.931	1781.858	475.718	2772.459	37548958	178956329
30	*5*	8	419.582	3600 (1)	458.575	3600 (2)	21702949	100908126
	7	7	1559.631	2568.754	1106.757	3600 (3)	81043582	224083837
35	*5*	0		3600 (10)		3600 (10)		

The results show that the B&B algorithm outperforms the mathematical model approach in almost all problem combinations in terms of the solution time. For relatively small-sized instances (m = 5, 7; n = 10, 15, 20) both approaches provide solutions in negligible time, with B&B always being faster. For medium sized instances where m = 5, n = 25; m = 7, n = 25 and m = 10, n = 15, the B&B algorithm has notably lower solution times than the mathematical model. For example, for m = 10, n = 15 combination, even the maximum B&B solution time observed over all instances (40.22 seconds) is lower than the average solution time of the mathematical model (71.21 seconds).

For fixed m (n), the solution times and the tree size increase as n (m) increases, as expected. For larger-sized instances both approaches may fail to return solutions within the 1-hour time limit. In m = 10, n = 20, the B&B algorithm performs better than the mathematical model since both approaches provide optimal solutions within the time limit for the same number of instances while the average solution time of the B&B algorithm is lower. When m = 5; n = 30, the mathematical model seems to perform slightly better as it can solve nine instances to optimality while B&B solves eight instances. When m = 7, n = 30 the mathematical model returns optimal solutions for all instances within the time limit. Note that the B&B still has notable performance in these instances, returning optimal solutions to seven of the ten instances. Overall, one can conclude that, the performances of the mathematical model and the B&B algorithm are comparable in large-sized instances.

Note that our branch-and-bound algorithm is capable of solving the instances with up to 30 aircraft and 7 gates. For real life problem instances of larger sizes, it can be used as a heuristic procedure in the following two ways:

a. Decomposition Approach

The problem can be decomposed into problems of small sizes and each subproblem can be solved to optimality by the branch-and-bound algorithm. Once the subproblem solutions are combined, the resulting overall solution is feasible for the original problem. The performance of this solution can be enhanced by some improvement procedures.

b. Truncated branch-and-bound algorithm

A truncated branch-and-bound algorithm is the one that can be terminated after a specified time limit or specified search size, i.e., number of nodes. One may use our algorithm as a truncated branch and bound algorithm with our time limit of 1 hour. To see the promise of such a truncated algorithm, we analyze the node at which the optimal solution is found, i.e., optimality node. All effort spent after the optimality node is for verifying the optimality.

Table 5.7 reports on the total number of nodes, optimality node and percentage of optimality node to the total number of nodes, for all problem combinations.

Table 5.7. *Number and Percentage of Optimality Nodes for All Combinations*

n	m	# of instances solved	# of nodes		Optimality Node		Opt Node / Nodes (%)	
			Average	Maximum	Average	Maximum	Average	Maximum
10	*5*	10	1706	2552	137	576	7.979	39.588
	7	10	9292	20467	1257	3162	16.222	43.214
15	*5*	10	15900	64221	1937	10144	26.432	93.547
	7	10	317998	1395240	17050	47247	14.332	40.648
	10	10	2397353	6487742	205394	561356	9.710	35.371
20	*5*	10	141410	415150	19921	126656	18.532	79.601
	7	10	7731576	22071977	1318579	3877668	28.57	94.79
	10	2	114750695	130540698	21566352	35320744	20.478	27.057
25	*5*	10	4054587	17305083	303171	704593	15.483	64.384
	7	10	37548958	178956329	9687786	65894125	18.065	54.903
30	*5*	8	21702949	100908126	5987785	34955172	16.603	34.641
	7	7	81043582	224083837	8455002	20219714	15.758	48.465
35	*5*	0	-	-	-	-	-	-

As can be observed from the table, the maximum percentages are high, implying that for some instances the optimal solution is found close to termination. However, the average percentages are low, despite the maximum percentages, for all problem combinations.

As can be observed from the table, the maximum percentages are much higher than the averages, implying that for some instances the optimal solution is reached a bit before the termination of the branch and bound algorithm. However, the average percentages are very low, despite the maximum percentages, for all problem combinations.

Note that when n = 10 and m = 5, the maximum percentage is 39.59%. Despite this high percentage, the average percentage is approximately 8%. That is on average, the optimal solution is reached at the first 8% of the search and 92% of the search is used for the verification of the optimal solution. When n = 25 and m = 5, the maximum percentage is 64.38% and the average is 15.48% despite the too high maximum performance. For the largest solvable size, n = 30 and m = 7, the average performance is 15.76%, i.e., still very low. Similar observations hold for other problem combinations, the majority of the optimal solutions are found before 15% of the search. This can be attributed to the effectiveness of the lower bounds in eliminating non-promising solutions. Hence, our branch-and-bound algorithm with a termination limit of 1 hour can be used as a heuristic to find approximate solutions for larger-sized instances.

Moreover, our upper bound at root node delivers an implementable feasible solution. We assess the performance of the initial upper bound by its deviation from the optimal solution as a percentage of the optimal solution and report the results in Table 5.8. In assessing the performance, only the instances residing the same number of aircraft in the apron at the optimal solution and upper bound solution are considered. This is due to the big-M value used for apron distances that might misevaluate the actual deviations. The number of times the upper bound returns the optimal solution is also included in the table. The table does not report the CPU times, as the upper bounds are found in negligible time.

Table 5.8. *Performance of the Upper Bound (UB) for All Combinations*

n	**m**	**# total instances solved**	**# of times UB & OPT have the same # of apron**	**# of times UB returns the optimum**	**Avg UB Gap (%)**	**Max UB Gap (%)**
10	*5*	10	9	2	6.668	28.006
	7	10	10	2	1.885	5.982
15	*5*	10	4	2	1.458	5.412
	7	10	10	1	1.894	7.976
	10	10	10	0	14.918	19.385
20	*5*	10	5	0	7.007	12.386
	7	10	7	0	2.934	6.389
	10	2	2	0	15.694	17.652
25	*5*	10	3	0	13.122	14.405
	7	10	7	0	4.764	14.591
30	*5*	8	1	0	27.056	27.056
	7	7	2	0	4.975	6.200
35	*5*	0	-	-	-	-

Note from the table that all average deviations are below %15 with one exception. The exception is for n = 30 and m = 5 and is due to a single instance that defines both average and maximum deviation. The performances are slightly getting worse with increases in the problem size. For example, when n = 15 and m = 5, the average deviation is 1.46%, when n = 20 and m = 5, the average deviation is 7.01% and when n = 25 and m = 5, the average deviation increases to 13.22%. Hence, one can conclude that the satisfactory performance of the upper bound used by the branch-and-bound algorithm may justify its use as a heuristic approach for large-sized problem instances.

CHAPTER 6

CONCLUSIONS

This thesis considers an airport gate assignment problem that assigns the aircraft to the gates so as to minimize the total walking distance covered by the passengers throughout the airport. We assume that the fixed gates can handle only one aircraft at a time and the remote gate, so called apron, has an unlimited capacity.

We first formulate the problem as a mixed integer nonlinear programming model and then linearize it. The resulting mixed integer linear programming model could handle small-sized instances; however, not the medium and large-sized instances in our termination limit of one hour. To solve larger sized problem instances, we develop a branch-and-bound algorithm with powerful lower bounding mechanisms and an initial upper bound.

The results of our computational experiments have revealed the superiority of the proposed branch-and-bound algorithm over the mathematical model over all problem combinations. The algorithm could return optimal solutions to the instances with up to 30 aircraft and 7 gates in our termination limit of one hour. For the problem instances of larger sizes, we suggest heuristic algorithms like our initial upper bound or truncated versions of our branch-and-bound algorithm.

To the best of our knowledge, we propose the first optimization algorithm for the total walking distance while considering transit passengers of airport gate assignment problem. Future research may consider the extension of our approach to the multi-objective problems like minimizing total walking distance traveled and weighted number of aircraft assigned to the apron.

REFERENCES

Aktel, A., Yagmahan, B., Özcan, T., Yenisey, M. M., & Sansarcı, E. (2017). The comparison of the metaheuristic algorithms performances on airport gate assignment problem. *Transportation Research Procedia*, *22*, 469-478. doi:10.1016/j.trpro.2017.03.061

Babić, O., Teodorović, D., & Tošić, V. (1984). Aircraft Stand Assignment to Minimize Walking. *Journal of Transportation Engineering*, *110*(1), 55-66. doi:10.1061/(asce)0733-947x(1984)110:1(55)

Bihr, R. A. (1990). A conceptual solution to the aircraft gate assignment problem using 0, 1 linear programming. *Computers & Industrial Engineering*, *19*(1-4), 280-284. doi:10.1016/0360-8352(90)90122-3

Bolat, A. (1999). Assigning arriving flights at an airport to the available gates. *Journal of the Operational Research Society*, *50*(1), 23-34. doi:10.1057/palgrave.jors.2600655

Bolat, A. (2001). Models and a genetic algorithm for static aircraft-gate assignment problem. *Journal of the Operational Research Society*, *52*(10), 1107-1120. doi:10.1057/palgrave.jors.2601190

Brazile, R., & Swigger, K. (1988). GATES: an airline gate assignment and tracking expert system. *IEEE Expert*, *3*(2), 33-39. doi:10.1109/64.2103

Cheng, C., Ho, S. C., & Kwan, C. (2012). The use of meta-heuristics for airport gate assignment. *Expert Systems with Applications*, *39*(16), 12430-12437. doi:10.1016/j.eswa.2012.04.071

Daş, G. S. (2017). New Multi objective models for the gate assignment problem. *Computers & Industrial Engineering*, *109*, 347-356. doi:10.1016/j.cie.2017.04.042

Deng, W., Sun, M., Zhao, H., Li, B., & Wang, C. (2018). Study on an airport gate assignment method based on improved ACO algorithm. *Kybernetes*, *47*(1), 20-43. doi:10.1108/k-08-2017-0279

Ding, H., Lim, A., Rodrigues, B., & Zhu, Y. (2005). The over-constrained airport gate assignment problem. *Computers & Operations Research*, *32*(7), 1867-1880. doi:10.1016/j.cor.2003.12.003

Dorndorf, U., Drexl, A., Nikulin, Y., & Pesch, E. (2007). Flight gate scheduling: State-of-the-art and recent developments. *Omega*, *35*(3), 326-334. doi:10.1016/j.omega.2005.07.001

Genç, H. M., Erol, O. K., Eksin, İ., Berber, M. F., & Güleryüz, B. O. (2012). A stochastic neighborhood search approach for airport gate assignment problem. *Expert Systems with Applications*, *39*(1), 316-327. doi:10.1016/j.eswa.2011.07.021

Gosling, G. D. (1990). Design of an expert system for aircraft gate assignment. *Transportation Research Part A: General*, *24*(1), 59-69. doi:10.1016/0191-2607(90)90071-d

Haghani, A., & Chen, M. (1998). Optimizing gate assignments at airport terminals. *Transportation Research Part A: Policy and Practice*, *32*(6), 437-454. doi:10.1016/s0965-8564(98)00005-6

Hu, X., & Di Paolo, E. (2009). An Efficient Genetic Algorithm with Uniform Crossover for the Multi-Objective Airport Gate Assignment Problem. *Multi-Objective Memetic Algorithms*, 71-89. doi:10.1007/978-3-540-88051-6_4

International Air Transport Association [IATA]. (2019, June 2). Annual growth in global air traffic passenger demand from 2006 to 2019 [chart]. Retrieved from https://www.statista.com/statistics/193533/growth-of-global-air-traffic-passenger-demand

Jiang, Y., Zeng, L., & Luo, Y. (2013). Multiobjective Gate Assignment Based on Passenger Walking Distance and Fairness. *Mathematical Problems in Engineering*, *2013*, 1-7. doi:10.1155/2013/361031

Kim, S. H., Feron, E., & Clarke, J. (2013). Gate Assignment to Minimize Passenger Transit Time and Aircraft Taxi Time. *Journal of Guidance, Control, and Dynamics*, *36*(2), 467-475. doi:10.2514/1.57022

Mangoubi, R. S., & Mathaisel, D. F. (1985). Optimizing Gate Assignments at Airport Terminals. *Transportation Science*, *19*(2), 173-188. doi:10.1287/trsc.19.2.173

Marinelli, M., Dell'Orco, M., & Sassanelli, D. (2015). A Metaheuristic Approach to Solve the Flight Gate Assignment Problem. *Transportation Research Procedia*, *5*, 211-220. doi:10.1016/j.trpro.2015.01.013

Mokhtarimousavi, S., Talebi, D., & Asgari, H. (2018). A Non-Dominated Sorting Genetic Algorithm Approach for Optimization of Multi-Objective Airport Gate Assignment Problem. *Transportation Research Record: Journal of the Transportation Research Board*, *2672*(23), 59-70. doi:10.1177/0361198118781386

Obata, T. (1979). *Quadratic assignment problem: Evaluation of exact and heuristic algorithms* (Doctoral dissertation, Rensselaer Polytechnic Institute Troy, NY, USA).

Pentico, D. W. (2007). Assignment problems: A golden anniversary survey. *European Journal of Operational Research*, *176*(2), 774-793. doi:10.1016/j.ejor.2005.09.014

Srihari, K., & Muthukrishnan, R. (1991). An expert system methodology for aircraft-gate assignment. *Computers & Industrial Engineering*, *21*(1-4), 101-105. doi:10.1016/0360-8352(91)90071-d

Steuart, G. N. (1974). Gate Position Requirements at Metropolitan Airports. *Transportation Science*, *8*(2), 169-189. doi:10.1287/trsc.8.2.169

Su, Y., & Srihari, K. (1993). A knowledge based aircraft-gate assignment advisor. *Computers & Industrial Engineering*, *25*(1-4), 123-126. doi:10.1016/0360-8352(93)90236-q

Turkish Airlines. Retrieved September 1, 2019, from https://www.turkishairlines.com/

Xu, J., & Bailey, G. (2001). The airport gate assignment problem: mathematical model and a tabu search algorithm. *Proceedings of the 34th Annual Hawaii International Conference on System Sciences*, 3032. doi:10.1109/hicss.2001.926327

Yan, S., & Huo, C. (2001). Optimization of multiple objective gate assignments. *Transportation Research Part A: Policy and Practice*, *35*(5), 413-432. doi:10.1016/s0965-8564(99)00065-8

Yan, S., Shieh, C., & Chen, M. (2002). A simulation framework for evaluating airport gate assignments. *Transportation Research Part A: Policy and Practice*, *36*(10), 885-898. doi:10.1016/s0965-8564(01)00045-3

Yu, C., Zhang, D., & Lau, H. (2016). MIP-based heuristics for solving robust gate assignment problems. *Computers & Industrial Engineering*, *93*, 171-191. doi:10.1016/j.cie.2015.12.013

Yu, C., Zhang, D., & Lau, H. Y. (2017). An adaptive large neighborhood search heuristic for solving a robust gate assignment problem. *Expert Systems with Applications*, *84*, 143-154. doi:10.1016/j.eswa.2017.04.050

Şeker, M., & Noyan, N. (2012). Stochastic optimization models for the airport gate assignment problem. *Transportation Research Part E: Logistics and Transportation Review*, *48*(2), 438-459. doi:10.1016/j.tre.2011.10.008