OPTIMIZATION APPROACHES FOR SPECIFIC AIRPORT OPERATIONS

İPEK SÜĞÜT

JUNE 2016

OPTIMIZATION APPROACHES FOR SPECIFIC AIRPORT OPERATIONS

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

IZMIR UNIVERSITY OF ECONOMICS

BY

İPEK SÜĞÜT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE

OF

MASTER OF SCIENCE

IN

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

JUNE 2016

Approval of the Graduate School of Natural and Applied Sciences

(Prof. Dr. İsmihan Bayramoğlu)
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science** in **Industrial Engineering.**

(Prof. Dr. M. Arslan Örnek)
Head of Department

We have read the thesis entitled **Optimization Approaches for Specific Airport Operations** prepared by **İpek SÜĞÜT** under supervision of **Prof. Dr. M. Arslan ÖRNEK** and we here by agree that it is fully adequate, in scope and quality, as a thesis for the degree of **Master of Science** in **Industrial Engineering**.

(Prof. Dr. M. Arslan Örnek)
Supervisor

Examining Committee Members:
(Chairmen, Supervisor and members)

Prof. Dr. M. Arslan ÖRNEK
Industrial Engineering Dept., IUE

Asst. Prof. Dr. Kamil Erkan KABAK
Industrial Engineering Dept., IUE

Assoc. Prof. Dr. Adil ALPKOÇAK
Computer Engineering Dept. DEU

ABSTRACT

OPTIMIZATION APPROACHES FOR SPECIFIC AIRPORT OPERATIONS

Süğüt, İpek

M.Sc. in Industrial Engineering

Graduate School of Natural and Applied Sciences

Supervisor: Prof. Dr. M. Arslan ÖRNEK

June 2016, 107

Many operation managers of airports face daily with some important problems such as the gate assignment problem (GAP), counter assignment problem (CAP), baggage carousels assignment problem (BCAP) etc. In order to solve these problems and tackle its complexity, many researches have been done. The objective of the gate assignment problem (GAP) is assigning each flight to an available gate while maximizing both conveniences to passengers and the operational efficiency of airport. In the counter assignment problem (CAP), the objective is to find a satisfactory allocation, given limited check-in counter resources that can adequately fulfill the requirements of each airline and at the same time meet all other constraints the airport may have. Our study covers both the gate assignment problem (GAP) and counter assignment problem (CAP) with the description of mathematical formulations and resolution methods such as decomposition algorithms.

Keywords: optimization, mathematical model, decomposition heuristic, airport, gate assignment problem, counter assignment problem.

ÖZ

BELİRLİ HAVAALANI OPERASYONLARINDA OPTİMİZASYON YAKLAŞIMLARI

Süğüt, İpek

Endüstri Mühendisliği Yüksek Lisans Programı

Fen Bilimleri Enstitüsü

Tez Danışmanı: Prof. Dr. M. Arslan ÖRNEK

Haziran 2016, 107

Birçok havaalanının operasyon bölümü hergün çok önemli problemleri yürütmek durumundadır. Bunlardan bazıları kapı atama problemi (GAP), kontuar atama problemi (CAP), bagaj atama problemidir (BCAP). Bu tarz problemleri çözebilmek ve karmaşıklığıyla baş edebilmek için birçok çalışma yapılmış ve birçok yaklaşım öne sürülmüştür. Bu tezin içerdiği problemlerden biri olan kapı atama probleminin amacı, havaalanına gelen her bir uçuşu istenen özelliklere göre uygun bir kapıya atamaktır ve aynı zamanda bu atamayı yaparken müşteri memnuniyetini ve havaalanının operasyonel verimliliğini de maksimum seviyede tutmak amaçlanmıştır. Tezin içerdiği ikinci problem olan kontuar atama probleminin amacı ise havaalanında

bulunan sınırlı sayıdaki kontuar kaynağını ihtiyaç belirten havayolu firmalarını uçuşlarının servisini yapmak üzere etkili bir biçimde dağıtmaktır. Bu dağıtımı yaparken havaalanının sahip olduğu bütün kısıtlamalar göz önüne alınır. Belirtildiği gibi tez, kapı atama ve kontuar atama problemleri için matematiksel modeller ve ayrıştırma sezgiselleri sunmaktadır.

Anahtar Kelimeler: optimizasyon, matematiksel model, ayrıştırma sezgiseli, havaalanı, kapı atama problemi, kontuar atama problemi.

ACKNOWLEDGEMENTS

I would never have been able to finish my thesis without the invaluable guidance of my advisor Prof. Dr. Arslan Örnek, my secret advisor Dr. Cemalettin Öztürk, constant support from my family and my friends.

I would like to express my deepest gratitude to my advisors, Prof. Dr. Arslan Örnek and Dr. Cemalettin Öztürk, for their excellent guidance, encouragement and patience. I am extremely thankful and indebted to them for sharing expertise.

I would like to thank my teachers in the department, helping me to develop my background in Industrial Engineering. In addition, I would like to express my gratitude to my manager at Turkish Airlines Company, Özgür Karagöz for his support.

I would also like to thank my friends Tuğçe Üstüner, Hatice Doğramacı, Gülce Haner, Yasemin Aktürk, Hatice Şentürk and so many of them. I thank them for the constant support they gave me throughout the challenges of life.

I would also like to thank my parents Mehmet Ata and Gülhan Süğüt, my brothers İdris and İlkay Süğüt and my cousin Merve İpekşen. They were always supporting me and encouraging me with their best wishes. Finally, I would like to thank to all people that I had a moment in my life.

This thesis I dedicate to my family and my friends.

TABLE OF CONTENTS

ABSTRACT 7
ÖZ... 9
ACKNOWLEDGEMENTS 11
TABLE OF CONTENTS 12
LIST OF TABLES... 14
LIST OF FIGURES 15
CHAPTER 1: INTRODUCTION 16
1.1. Gate Assignment Problem (GAP) 18
1.2. Counter Assignment Problem (CAP)..20
CHAPTER 2: PROBLEM DEFINITION 22
2.1. Introduction 22
2.2. Gate Assignment Problem (GAP) 23
2.3. Counter Assignment Problem (CAP)..23
CHAPTER 3 LITERATURE REVIEW.. 26
3.1. Gate Assignment Problem (GAP) 26
3.2. Counter Assignment Problem (CAP)..41
CHAPTER 4: MATHEMATICAL MODELS: OBJECTIVES & FORMULATIONS... 44
4.1. Introduction 44
4.2. Gate Assignment Problem (GAP) 44
4.2.1. Timetabling Based Integer Programming Model 46
4.2.2. Assignment Based Integer Programming Model 50
4.3. Counter Assignment Problem (CAP) 53
4.4. Decomposition Algorithm for CAP..57
CHAPTER 5: RESULTS & DISCUSSION.. 61
5.1. Gate Assignment Problem (GAP) 61
5.1.1. Timetabling Based Integer Programming Model 62
5.1.2. Assignment Based Integer Programming Model 63

5.2. Counter Assignment Problem (CAP) 64

CHAPTER 6: CONCLUSIONS & FUTURE WORK.................................... 68

REFERENCES 70

APPENDICES... 77

Appendix A: The numerical Instance for Developed Models of GAP 78

Appendix B: The Results of Timetabling Based Integer Programming Model …………………………………………………………………………………90

Appendix C: The Results of Assignment Based Integer Programming Model …………………………………………………………………………………93

Appendix D: The Numerical Instance for Developed Model of CAP..............96

Appendix E: The results for Developed Model of CAP..................................99

LIST OF TABLES

Table 1 Counter demand for six flights between starting and ending period of counter time of that flight 24
Table 2 Mathematical Formulations of GAP and Related Problems 35
Table 3 Resolution Methods for GAP 41
Table 4 Mathematical Formulations and Resolution Methods for CAP 43
Table 5 Theoretical bounds for the size of the models 61
Table 6 Performance evaluation of developed models on a real size instance where * indicates optimality of the solution 62
Table 7 The run-time of the model with some specified flight numbers 65
Table 8 Performance evaluation of sub-problems 66
Table 9 The Weight Logic for Domestic Gates 78
Table 10 The Weight Logic for International Gates 78
Table 11 The weight values for each flight at each gate and the weights for only gate 81
Table 12 The neighbourhood matrix for gates 83
Table 13 The arrival, original departure and departure with extra periods of the flights and the ground service data of flights 85
Table 14 The eligibility matrix for flights to the gates 88
Table 15 Ground Location Service and Night-Stand Flight Data 88
Table 16 The busy gates from the previous day and earliest available period of busy gates 89
Table 17 The bridge-equipped and open-park gates 89
Table 18 The results of assignments of flights to the gates for timetabling based integer programming model 91
Table 19 The assignment of flights on each gate for timetabling based integer programming model 92
Table 20 The results of assignments of flights to the gates for assignment based integer programming model 94
Table 21 The assignment of flights on each gate for assignment based integer programming model 95
Table 22 The arrival, departure, and the number of counters needed of the flights and the firm data of flights 97
Table 23 The neighbourhood matrix for counters (The full matrix is 135R x 135C) 98
Table 24 The results of assignments of flights to the counters for developed model 107
Table 25 The result of each sub-problem and its run-time 108
Table 26 The part of the result for Counter Assignment Problem 109

LIST OF FIGURES

Figure 1 An example of an airport view 16
Figure 2 An aircraft in a bridge-equipped parking position 18
Figure 3 An aircraft in a remote parking position 19
Figure 4 A traditional check-in counter 20
Figure 5 The counter layout of the airport 25
Figure 6 The small part of the results of GAP for Timetabling Based Model based on each gate 63
Figure 7 The small part of the results of GAP for Assignment Based Model based on each gate 64
Figure 8 The small part of the results of CAP based on each counter 67

CHAPTER 1: INTRODUCTION

Airports play a significant role in globalization, connecting cities and countries. They are a major part of a country's infrastructure and foster economic activities by encouraging international commerce and tourism and generating employment. Due to its crucial role in the economy, the complexity of airport management has increased significantly. If its operations are not handled well, flight delays or accidents can happen and the domino effect might exist to influence the whole operations of airport.

Figure 1 An example of an airport view

The airports which have reached to significant performance levels especially need to be managed effectively and taken some precautions to avoid from bad situations such as decreased performance, increased passenger travel times, long waiting times of passengers, and airport traffic congestion etc. In order to prevent these bad performance measures, some efficient strategic decisions must be taken.

The subject of this thesis is the tasks related to gate assignment problem (GAP) and the counter assignment problem (CAP) which are two of the most important daily operations many researches have been published on with the aim of solving these problems in spite of the complexity.

The objective of first problem in this study, the gate assignment problem (GAP), is assigning each flight to an available gate while maximizing both conveniences to passengers and the operational efficiency of airport. This problem has increasing importance due to the increasing passengers in airports. Although this problem is an easily-understood, it is difficult to solve problem. The calculation of number of planes over number of gates gives us the total number of solution candidates, and for a practical airport, the result yields impractical amounts of candidate solutions to be tried. So, the solution space and the existence of some constraints make the problem still difficult to solve for the optimum solution and therefore still up to date.

The second problem in this thesis is about counters which is the first place encountered by the passengers when travelling by air. These counters provide a check-in service. By checking in, the passengers confirm to the airline that they actually have the intention to board the flight for which they have booked a ticket. Moreover, to choose, buy or change a seat, register bags, etc. are the possibilities for a passenger at counters. Today, the process can be done in various ways; online, via self-service kiosks at the airport, and via the traditional check-in desks, where the passengers are served by representatives of the airlines. In general, check-in is performed before the passengers reach the security check. It is often the responsibility of the airport to provide available check-in counters, and the airlines must provide available representatives. A check-in group is a group of flights (departures) that share the same check-in counters for check-in or baggage drop-off. A check-in group can either consist of a single flight or all flights of a specific or multiple airlines. Each check-in group has a counter demand, and the counter assignment problem (CAP) is the problem of allocating check-in groups to available check-in counters.

1.1. Gate Assignment Problem (GAP)

Aircraft parking space assignment is the problem of assigning aircrafts to bridge-equipped or remote parking positions under a number of objectives and constraints. Maximizing the number of aircrafts assigned to bridge-equipped parking positions is the main objective. Besides this, there are also other considerations such as the efficient use of parking positions, walking distance of passengers, timetables of the flights, and compatibility of aircrafts with the parking positions, etc.

Figure 2 An aircraft in a bridge-equipped parking position

Figure 3 An aircraft in a remote parking position

This thesis considers gate assignment problem for an airport, which is regarded to be a highly complex problem with the possibility of application in both planning as well as operations mode. There are various considerations that are involved while assigning gates to incoming and outgoing flights at an airport. Different gates have restrictions, such as adjacency, LIFO and push time, which is known in advance from the structure of the airport. Different optimization models, namely, two alternative integer programming model (IP) are proposed. These models are solved and presented with the results for oneday operation of an airport using real data. In addition, the efficiency of the models is compared.

1.2. Counter Assignment Problem (CAP)

The counter assignment problem (CAP) is the problem of allocating check-in groups to available check-in counters under a number of objectives and constraints. Minimizing the number of unallocated check-in groups is the main objective. Besides this, there are also other considerations such as satisfying airline preferences to the greatest possible extent, etc.

Figure 4 A traditional check-in counter

This study considers counter assignment problem for an airport, which is regarded to be a highly complex problem with the possibility of application in both planning as well as operations mode. There are many considerations that are involved while assigning counters to outgoing flights at an airport. Counters have restrictions, such as adjacency, etc. An optimization model with a decomposition algorithm is proposed. This integer programming model (IP) and the algorithm is solved and presented with the results for oneday operation of an airport using real data. In addition, the efficiency of the model and the algorithm is presented.

The rest of this study is organized as follows. In Chapter 2, the problem definition of Gate Assignment Problem (GAP) and Counter Assignment Problem (CAP) are described. In Chapter 3, the objectives of the problems are presented. In Chapter 4, literature review with previous publications is stated. In Chapter 5, the models are formulated. In Chapter 6, the numerical instances for the problems are introduced and the results are discussed. Finally, the study is concluded in Chapter 7.

CHAPTER 2: PROBLEM DEFINITION

2.1. Introduction

In airport operations, managers face daily with some important issues and try to overcome hypercorrectly. As some of the most crucial issues in this management, the assignment of gates and counters to the flights is very significant for daily management. With an intense air-traffic increase in recent years, improper assignment of gates and counters may result in some problems such as flight delays, inefficient use of the resources, passenger dissatisfaction, etc. Therefore, the need to efficiently use these resources to reduce operating costs, increase customer satisfaction, and lighten congestion has become more common in these days.

There are two steps related in our study to solve the problem. In the first step, we assign the incoming flights to the gates according to the defined objectives between airport managers and airlines. These flights have specific departure times from the airport which are decided by their airlines. Now, incoming flights transform to the outgoing ones. According to these departure times, the need of another resource in the airport, counters, shows up. Therefore, the main work of this second step is the assignment of the flights to the counters. After these related assignments, an important issue which has many difficulties for managers to handle is settled.

In the following sections, the detailed information about related problems, the gate assignment and counter assignment are described.

2.2. Gate Assignment Problem (GAP)

The airport under consideration has a number of open park areas and bridge-equipped gates. Airport management prefers flights to be assigned to bridge-equipped gates as it facilitates embarking and disembarking of passengers. Also, after aircrafts arrive, they need to be refueled, replenished, all the waste has to be taken off-board. When all gates are engaged, then flights are to be assigned to open park areas. Also, night stand flights are assigned to open park areas. Some gates are for emergencies only. These are large enough for allocation of larger planes. For instance, if the bridges 26th and 42nd are full, large planes are assigned to 24th or 25th bridge-equipped parking area. Some airlines have a priority to be assigned to the same gates. Normally, no other flight is assigned to those gates unless that gate is available.

Airline companies that use the same ground handling services firms are assigned adjacent to each other in order to prevent apron traffic. Departure and arrival of a plane is also considered. For instance, if a plane's departure is international, it has a priority for bridge-equipped parking areas in the international terminal. Similarly, if its departure is domestic, it has a priority in the domestic terminal. Some gates have priority due to their proximity to facilities in the airport. Not every plane fits in every parking area. Hence, some flights cannot be assigned to some parking areas, which we call plane-gate eligibility. We consider improving gate utilization as our primary objective.

Due to combinatorial nature of the problem, we provide two different integer programming (IP) formulations, namely timetabling and assignment based, and then compare their performance.

2.3. Counter Assignment Problem (CAP)

The objective of second part of the problem is to allocate the flights of each airline company to the check-in counters observing some restrictions. For each flight we have a counter demand over time. The correct counter demand for a flight is decided by the airline in dialogue with the airport. For instance, Table 3 describes the counter demand for six flights between starting and ending period of counter time of that flight. For this allocation problem, the counter demands are determined and is not a part of the optimization.

FLIGHT	NUMBER OF COUNTERS	COUNTER START TIME	COUNTER END TIME
ABG7702	2	04:50	06:50
ABG7708	2	05:00	07:00
AFL2143	4	10:55	12:55
AFL7221	2	06:10	08:10
AUA2250	3	06:00	08:00
BER2355	3	08:45	10:45

Table 1 Counter demand for six flights between starting and ending period of counter time of that flight

A check-in counter is said to be opened when a representative enters into the check-in system at the counter and is ready to check-in passengers and take their bags. When the representative logs out, the check-in counter is closed. A counter opening is the opening of a given counter and the length is the time from the beginning until it is closed.

Airports can have different layouts and also service quality when planning. In this study, the layout of the airport related with this problem is shown at Figure 5.

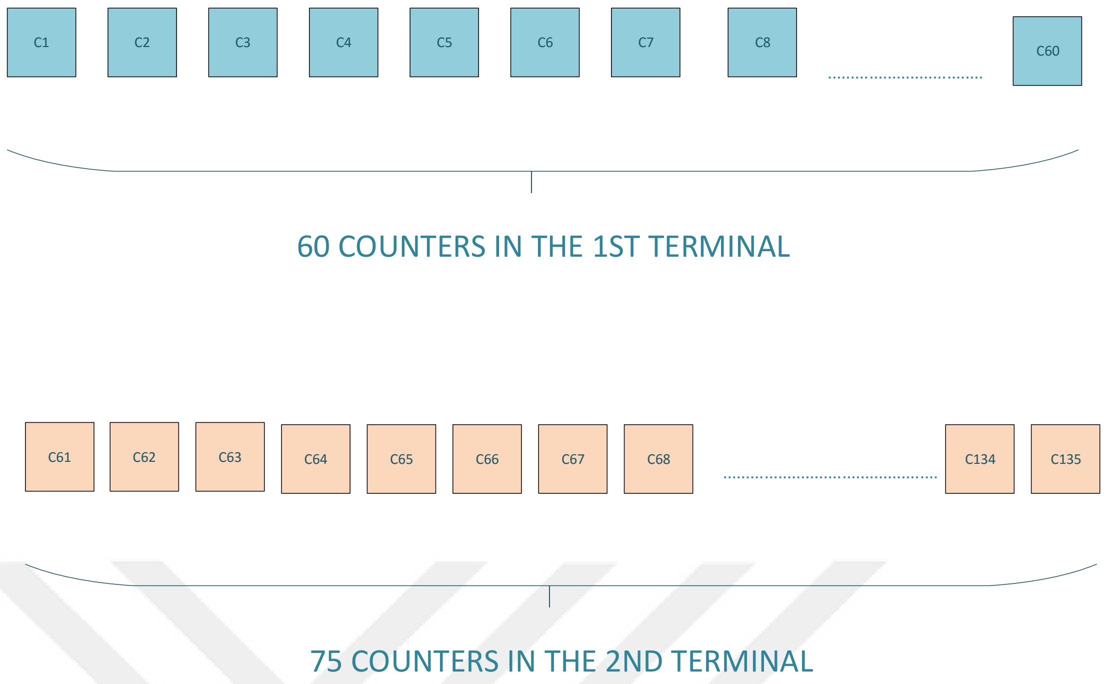


Figure 5 The counter layout of the airport

As one can realize from the figure that the airport has two terminals. In the first terminal, there are 60 counters and the remaining 75 counters are in the second terminal. In this study, our goal is generally assigning flights to the counters in the first terminal which is the preference of the airport.

In addition, flights of the same airline should be assigned to the adjacent counters in order to the practical use of the airline. Moreover, in the following periods, we want to assign flights consecutively to that counter if these flights belong to the same airline. We consider improving counter utilization as our primary objective.

Due to combinatorial nature of the problem, an integer programming formulation (IP) is proposed and the results are shown in the following chapters.

CHAPTER 3 LITERATURE REVIEW

3.1. Gate Assignment Problem (GAP)

One of the problems studied in this thesis is the assignment of aircraft arriving on schedule to available gates is a major issue during the daily airline operations. Innumerable methods have been developed to solve this problem since 1974. A simple stochastic model to find the efficiency use of the gate positions was proposed by Steuart [1]. There were less than 15 publications for 25 years. As a result, the research interest in this field was slow in development but after 2000, the interest to find solutions for this problem increased. In these days, there is a still small growth.

The authors consider some various objective functions to solve this problem and they use a real case or a theoretical instance in their studies. The first point of view is as an airport owner, which is the government. Maximizing the utilization of the available gates and terminal [1-4], minimizing the number of gate conflicts [5], minimizing the number of ungated flights [3,6-9], and minimizing the flights delay [10] are the objectives. An airlines owner is an another point of view. Increasing the customer satisfaction with minimizing the passenger walking distance between gates [3,6,7,11-18] and minimizing the travelling distance from runway to the gate [9] are their goals.

From a mathematical view, GAP has been formulated as integer, binary, or mixed integer, general linear or nonlinear models. Some publications proposed specific formulation as binary or mixed binary quadratic models.

Quadratic assignment problem (QAP), clique partitioning problem (CPP), and scheduling problem which are well-known related problems in combinatorial optimization have been used to formulate GAP. From a different point, stochastic or robust optimization was used on few publications.

The first method from the literature is Integer Linear Programming (ILP) formulations. The references for this formulation type are Lim et al. [24], Diepen et al. [25], and Diepen et al. [26].

The first reference, Lim et al. [24], formulated GAP as an integer programming model and proposed two models with time windows. Minimization of the passenger walking distance (travel time) is the first model while the second model minimized cargo handling costs of the gate assignments. They used an IP solver to find the optimal solution in the first model, however several heuristic algorithms were used to generate solutions in the second model. According to the results, heuristics gave better results than the IP solver in both CPU time and solutions quality.

The second and last reference of Diepen et al. [25,26] formulated GAP as integer linear programming model with a relaxation for the integrality. After relaxed integrality, column generation (CG) exploited the resulting relaxed LP to obtain solutions of ILP. They divided the problem into two phases, planning and attaching. The objective of the first phase is minimizing the cost of a gate plan. The second phase was an assignment in physical gate.

In the last reference for ILP, [26], the solution obtained from their assignment of gates was used as an input to solve bus-planning problem in the same airport.

The second method from the literature is Binary Integer Programming and many authors contribute to this area. The references are Mangoubi and Mathaisel [11], Yan et al. [29], Vanderstraeten and Bergeron [28], Bihr [12], Tang et al. [27], and Prem Kumar and Bierlaire [18].

The first reference for this method, Mangoubi and Mathaisel [11], developed a binary integer model to minimize the passenger total walking distance and proposed a heuristic method to find the solution. According to the comparison with the heuristic method result and the results from a standard IP solver, first one was better if we compare it with the results of LP solver.

In 2002, the other reference, the static GAP as a binary integer programming model was formulated by Yan et al. [29] to serve as a basis of real time gate assignments in a simulation framework which is for analyzing the effects of stochastic flight delays on static gate assignments.

The next reference, Vanderstraeten and Bergeron [28], formulated GAP as a binary integer model with the objective of minimizing the off-gate events and they developed a new heuristic. A real case has been analysed in Canada.

To minimize the passenger walking distance, the other reference, Bihr [12], developed a binary integer model. This model was used to solve a sample problem using primal-dual simplex algorithm.

In 2009, GAP was formulated as a binary integer programming model by Tang et al. [27], another reference for BIP. To generate a lower bound to their original problem, the output model was used.

A binary integer programming model that produced a feasible gate plan in the light of all the business constraints is another reference and was presented by Kumar et al. [18].

The third method from the literature is Mixed Integer Linear Programming (MILP). The references for this method are proposed by one author who is Bolat [30, 31].

The first reference for MILP, Bolat [30], developed a mixed integer program for GAP with the objective of minimizing the slack times (slack time is an idle time between two successive assignment of the gate).

In 2001, another reference, a framework for GAP that transformed the nonlinear binary models into an equivalent linear binary model was presented by Bolat [31] with the objective of minimizing the range or the variance of the idle times. This study includes five mathematical models, where two of them were formulated as a mixed integer linear programming and the others as a mixed integer nonlinear programming. Models 1 to 4 were defined for homogeneous gate while model 5 was defined for heterogeneous gate.

The other method from the literature is Mixed Integer Nonlinear Programming. The references are Li [5, 32], and Bolat [31].

The first reference, Li [5], developed a nonlinear binary mixed integer model with a constraint programming which minimizes the number of gate conflicts of any two adjacent flights that are assigned to the same gate.

Another reference, Bolat [31] also proposed two Mixed Integer Nonlinear Programming formulations as stated in the previous paragraphs.

Another method from the literature is Multiple Objective GAP Formulations. The references are Hu and Di Paolo [36], Wei and Liu [16], B.A.C.o.E.B. Team and A.I.C.o.E. Team [17], Yan and Huo [2], and Kaliszewski and Miroforidis [37].

The first reference for this method, Hu and Di Paolo [36], formulated a mathematical model with an objective of minimization and solved using a new genetic algorithm.

The next reference, Wei and Lui [16], considered GAP as a fuzzy model and applied a hybrid genetic algorithm to solve the model. Minimizing passengers' total walking distance and gates idle times variance are the main objectives.

Wipro Technologies [17], another reference, developed a binary multiple objective integer quadratic programming model with a quadratic objective function.

In 2001, another reference, a model with two objectives was formulated by Yan and Huo [2]. The objectives are minimizing the walking distance and the waiting time for the passengers.

The last reference for this method, a model with the objective of assigning incoming flights to airport gates with some assumptions was developed by Kaliszewski and Miroforidis [37].

The next method from the literature is stochastic models. The references are Yan and Tang [10], Genç et al. [38], and Şeker and Noyan [9].

The first reference, a study for a stochastic GAP was designed by Yan and Tang [10]. In this analysis, the flight delays are stochastic. It had three parts: the gate assignment model, a rule for the reassignments, and two adjustment methods for penalties. The performance was analyzed and evaluated by a simulation-based method.

The next reference, a stochastic model with the objective of minimizing the gate duration (total time of the allocated gates for all flights in a day) was formulated by Genç et al. [38].

The last reference for this method, Şeker and Noyan [9], also formulated a stochastic model which is a minimization of the number of conflicts and the expected variance of the idle times.

Some researchers formulated GAP as a quadratic assignment problem (QAP), clique partitioning problem (CPP) and scheduling problem or even a network representation. On the other hand, some of them formulated GAP as a robust optimization model.

Another method from the literature to solve GAP is Quadratic Assignment Problem (QAP). The references are Drexl and Nikulin [3], and Haghani and Chen [13].

The first reference, Drexl and Nikulin [3], formulated the multicriteria airport gate assignment as a quadratic assignment problem (QAP) and solved it using Pareto simulated annealing. The objectives are: minimizing connection times or total passenger walking distances, maximizing the preferences of total gate assignment, and minimizing the number of ungated flights.

The last reference for this method, Haghani and Chen [13], modeled GAP as QAP in order to minimize the total passenger walking distances.

The next method from the literature is Scheduling Problems. The reference is just one and proposed by Li [39]. In 2010, Li [39] modeled GAP as a parallel machines scheduling problem and applied dynamic scheduling and the direct graph model to solve it. For solving the small size problems, B&B is used while the large size problems were solved by using dynamic scheduling.

Another method for solving GAP is Quadratic Mixed Binary Programming. The references are Bolat [34], Zheng et al. [33], and Xu and Bailey [14].

The first reference, Bolat [34], also developed a mixed binary quadratic programming model in order to minimize the variance of idle times and applied branch and bound algorithm and proposed two heuristics for solving the proposed model.

The next reference, a mixed binary quadratic program with minimizing slack time overall variance as the objective function was formulated by Zheng et al. [33]. There was an assumption which is that the flights are sequenced according to their arrival time (from smallest one to the largest).

Another mixed binary quadratic programming model was formulated by Xu and Bailey [14]. The objective of the study was minimizing the passenger connection time.

Another method for solving GAP is Binary Quadratic Programming and it is just proposed by Ding et al. [6, 7, 35]. In order to minimize the number of ungated flights, Ding et al. [6, 35] formulated a binary quadratic programming model. For an initial solution, a greedy algorithm was used and it was improved by using Tabu Search (TS). In 2005, Ding et al. [7] also developed a binary quadratic programming model for the same objective. The same greedy algorithm was used for an initial solution but for this time, it is improved by first simulated annealing (SA), then a hybrid of simulated annealing (SA) and tabu search (SA-TS).

The next method for GAP is Clique Partitioning Problem (CPP) and is also just proposed by Dorndorf et al. [8] who formulated an optimization model for GAP and converted that model into a CPP model. A heuristic approach developed by Dorndorf and Pesch (1994) was used in order to solve the transformed model.

Another method is Network Representation and the only reference is just proposed by Maharjan and Matis [40]. A binary integer multicommodity network flow model with minimizing the passengers comfort and aircraft fuel burn was formulated by Maharjan and Matis [40].

The last method for GAP from a metmatical perspective is Robust Optimization. The only reference is proposed by Diepen et al. [41] who modeled a completely new integer linear programming formulation with a robust objective function which can be expressed as the maximization of an allocation of a maximum possible idle time between each pair of consecutive flights.

Table 2 gives a brief explanation about all mathematical formulations used recently for GAP.

Formulation	References	Criterion (Comments)	Problem Type
Integer Linear Programming (ILP)	Lim et al. [24]	(1) Minimizing the sum of delay penalties (2) Minimizing the total walking distance	Theoretical
	Diepen et al. [25]	(1) Minimizing the deviation of arrival and departure time (2) Minimizing replanning the schedule	Real case (Amsterdam Airport)
	Diepen et al. [26]	Minimizing the deviations from the expected arrival and departure times	Real case (Amsterdam Airport)
Binary Integer Programming	Mangoubi and Mathaisel [11]; Yan et al. [29]	Minimizing passenger walking distances	Real case (Toronto International Airport); Real case (Chiang Kai-Shek Airport)
	Vanderstraeten and Bergeron [28]	Minimizing the number off-gate event	Theoretical
	Bihr [12]	Minimizing of the total passenger distance	Theoretical
	Tang et al. [27]	Developing a gate reassignment framework and a systematic computerized tool	Real case (Taiwan International Airport)
		(1) Maximizing the gate rest time between two turns (2) Minimizing the cost	

	Prem Kumar and Bierlaire [18]	of towing an aircraft with a long turn (3) Minimizing overall costs that include penalization for not assigning preferred gates to certain turns	Theoretical
Mixed Integer Linear Programming (MILP)	Bolat [30]	Minimizing the range of slack times	Real case (King Khaled International Airport)
	Bolat [31]	Minimizing the variance or the range of gate idle time	Real case (King Khaled International Airport)
Mixed Integer Nonlinear Programming	Li [5 ,32]	Minimizing the number of gate conflicts of any two adjacent aircrafts assigned to the same gate	Real case (Continental Airlines, Houston George Bush Intercontinental Airport)
	Bolat [31]	Minimizing the variance or the range of gate idle time	Real case (King Khaled International Airport)
Multiple Objective GAP Formulations	Hu and Di Paolo [36]	Minimize passenger walking distance, baggage transport distance, and aircraft waiting time on the apron	Theoretical
	Wei and Liu [16]	(1) Minimizing the total walking distance for passengers (2) Minimize the variance of gates idle times	Theoretical
	B.A.C.o.E.B. Team and A.I.C.o.E. Team [17]	(1) Minimizing walking distance (2) Maximizing the number of gated flights (3) Minimizing flight delays	Theoretical
	Yan and Huo [2]	(1) Minimizing passenger walking distances (2) Minimizing the passenger waiting time	Real case (Chiang Kai-Shek Airport)
	Kaliszewski and Miroforidis [37]	Finding gate assignment efficiency which represents rational compromises between waiting time for gate and apron operations	Theoretical
	Yan and Tang [10]	Minimizing the total passenger waiting time	Real case (Taiwan International Airport)

Stochastic Model	Genç et al. [38]	Maximizing gate duration, which is total time of the gates allocated	Theoretical and real case (Ataturk Airport of Istanbul, Turkey)
	Şeker and Noyan [9]	Minimizing the expected variance of the idle time	Theoretical
Quadratic Assignment Problem (QAP)	Drexl and Nikulin [3]	(1) Minimizing the number of ungated flights (2) Minimizing the total passenger walking distances or connection times (3) Maximizing the total gate assignment preferences	Theoretical
	Haghani and Chen [13]	Minimizing the total passenger walking distances	Theoretical
Scheduling Problems	Li [39]	(1) Maximizing the sum of the all products of the flight eigenvalue (2) Maximizing the gate eigenvalue that the flight assigned	Theoretical
Quadratic Mixed Binary Programming	Bolat [34]	Minimizing the variance of idle times	Real case (King Khaled International Airport)
	Zheng et al. [33]	Minimizing the overall variance of slack time	Real case (Beijing International Airport, China)
	Xu and Bailey [14]	Minimizing the passenger connection time	Theoretical
Binary Quadratic Programming	Ding et al. [6, 7, 35]	Minimize the number of ungated flights and the total walking distances or connection times	Theoretical
Clique Partitioning Problem (CPP)	Dorndorf et al. [8]	(1) Maximizing the total assignment preference score (2) Minimizing the number of unassigned flights (3) Minimizing the number of tows (4) Maximizing the robustness of the resulting schedule	Theoretical
Network Representation	Maharjan and Matis [40]	Minimizing both fuel burn of aircraft and the comfort of connecting passengers	Real case (Continental Airlines, Houston George Bush Intercontinental Airport)
		Maximizing the	

Robust Optimization	Diepen et al. [41]	robustness of a solution to the gate assignment problem	Real case (Amsterdam Airport)

Table 2 Mathematical Formulations of GAP and Related Problems

While finding an algorithm that guarantees an optimal solution in polynomial time on the subject of the problem size is the goal of combinatorial optimization research, in practice the main interest is to find a nearly optimal or at least good-quality solution in an acceptable amount of time. Many approaches to solve the GAP have been suggested, changing from Branch and Bound (B&B) to highly mystical optimization methods. The larger part of these methods can be commonly categorized as either “exact” algorithms or “heuristic” algorithms. Exact algorithms are the algorithms that return an optimal solution. Different exact solution techniques have been used to solve the GAP and in some study, the authors used some optimization programming languages like CPLEX and AMPL.

In nature, the GAP is a QAP and it is an NP-hard problem as shown in Obata [21]. Researchers have proposed varied heuristic and metaheuristics approaches for solving GAP because it is NP-hard. With heuristic algorithms, hypothetically there is a chance to find an optimal solution. That chance can be unknown because heuristics usually reach a local optimal solution and get stuck at that period. But metaheuristics or “modern heuristics” provide systematic rules to deal with this problem. These rules can escape from local optima or give the ability of quitting of local optima. The acceptable characteristic of these metaheuristics is the use of some mechanisms to avoid local optima. Metaheuristics achieved in leaving the local optimum by temporarily obtaining moves that cause declining of the objective function value.

Many researches also have been done on the exact, heuristic and metaheuristic approaches for solving GAP and they provided a real or a theoretical case.

For the first method, exact algorithms, the references are Mangoubi and Mathaisel [11], Bihr [12], Yan and Huo [2], Bolat [30, 34], Xu and Bailey [14], and Li [39].

The first reference for this method, Mangoubi and Mathaisel [11], relaxed the integrality of ILP model and solved it by using CPP.

The next reference, Bihr [12], developed a primal-dual simplex algorithm to find the solution.

Yan and Huo [2] as another reference used simplex algorithm with column generation and weighting method to solve the problem.

The other references, Bolat [30,34], Li [39], and Yan and Huo [2], applied branch and bound algorithm to solve the models.

The last reference, Xu and Bailey [14], used branch and bound algorithm and compared the result with tabu search algorithm.

The another method is heuristic algorithms and the references are Thengvall et al. [43], Yan and Tang [10], Ding et al. [6, 35], Lim et al. [24], Diepen et al. [25], Dorndorf et al. [8], Mangoubi and Mathaisel [11], Vanderstraeten and Bergeron [28], Yan et al. [29], Bolat [30], Bolat [34], Haghani and Chen [13], Genç [42], B.A.C.o.E.B. Team and A.I.C.o.E. Team [17], and Bouras et al. [45].

The first reference, Thengvall et al. [43] proposed a heuristic approach for the problem of schedules recovery in airports during hub closures. The approach was a bundle algorithm.

Another reference for this method, Yan and Tang [10], formulated a study to deal with GAP with stochastic flight delays. This developed framework had a heuristic approach.

The next reference, Ding et al. [6, 35], formed a greedy algorithm with an objective of minimizing the number of ungated flights.

Lim et al. [24], as another reference, applied several solution approaches, “Insert Move Algorithm”, “Interval Exchange Move Algorithm”, and “Greedy Algorithm” to solve the developed model for GAP.

Another reference, Diepen et al. [25], used column generation to solve the resulting LP-relaxation and the original ILP model.

The following reference, Dorndorf et al. [8], applied a heuristic approach which was developed by Dorndorf and Pesch (1994) and it was an ejection chain algorithm.

The next reference, Mangoubi and Mathaisel [11], also developed a heuristic approach to solve GAP with the objective of minimizing walking distance for the passengers.

Vanderstraeten and Bergeron [28], as a next reference, formulated a direct assignment of flights to gates algorithm, named ADAP.

The other reference, Yan et al. [29], suggested a simulation study and designed an optimization model and solved the model using two greedy heuristics.

These references, Bolat [30] and Bolat [34] first developed branch and trim heuristic to solve GAP to minimize slack times range and then applied the HBB and SPH heuristics to solve models developed by him for GAP.

The next reference, Haghani and Chen [13], applied a heuristic approach to solve GAP with the objective of minimizing walking distance for the travellers.

Genç [42], as last reference, applied several heuristics, which are the “Ground Time Maximization Heuristic”, “Idle Time Minimization Algorithm”, and “Prime Time Heuristic” to solve GAP with a performance measure which is minimizing the idle gate time (or maximizing the number of assigned flights).

The next method from the literature for GAP is metaheuristics which was succesful at leaving the local optimum by accepting moves that cause worsening of the objective function value. The references are Ding et al. [6, 35], Ding et al. [7], Lim et al. [24], Hu and Di Paolo [36], Drexl and Nikulin [3], Xu and Bailey [14], Bolat [31], Şeker and Noyan [9], Zheng et al. [33], Wei and Liu [16], Gu and Chung [44], Cheng et al. [23], and Bouras et al. [45].

The first reference, Ding et al. [6, 35], developed a tabu search (TS) algorithm to solve GAP and the starting initial solution was found by a designed greedy algorithm.

The next reference, Ding et al. [7], applied a simulated annealing and a hybrid of SA and TS to solve their GAP model.

Lim et al. [24], as a next reference, formulated TS and memetic algorithms to solve GAP.

The following reference, Hu and Di Paolo [36], used a new genetic algorithm with uniform crossover to solve the multiobjective gate assignment problem (MOGAP).

As another reference, Drexl and Nikulin [3], used Pareto simulated annealing to solve multicriteria airport gate assignment.

The other one, Xu and Bailey [14], applied a tabu search algorithm and compared the results of this algorithm with a branch and bound algorithm.

Bolat [31], as a next reference, applied genetic algorithm (GA) to minimize the variance or the range of gate idle time.

For the next reference, Şeker and Noyan [9] formulated stochastic programming models. The developed models were solved by using Tabu Search (TS).

The next reference, Zheng et al. [33], formulated a model for solving GAP and applied a TS algorithm to get solutions of the model.

Another reference, Wei and Liu [16], developed a hybrid genetic algorithm to solve the fuzzy GAP model.

Gu and Chung [44] developed a genetic algorithm model to solve GAP.

For the next one, Cheng et al. [23] analysed the performance of some metaheuristics in solving GAP which were genetic algorithm (GA), tabu search (TS), simulated annealing (SA), and a hybrid of SA and TS.

The last reference, Bouras et al. [45], formulated a parallel machine-scheduling problem with some priority and eligibility to solve GAP. The objectives were total cost, total tardiness, and maximum tardiness. They formed three heuristics and used three metaheuristics (simulated annealing, genetic algorithm, and tabu search).

Table 3 summarizes all solution techniques used recently for GAP.

Method	References	Approach/Results	Problem Type
Exact Algorithms	Mangoubi and Mathaisel [11]	Linear programming relaxation	Real case (Toronto International Airport)
	Bihr [12]	Primal-dual simplex	Theoretical
	Yan and Huo [2]	Simplex Branch and bound	Real case (Chiang Kai-Shek Airport)
	Bolat [30, 34]; Xu and Bailey [14]; Li [39]	Branch and bound	Real case (King Khaled International Airport, KSA); theoretical
Heuristic Algorithms	Thengvall et al. [43]	Bundle algorithm approach	Theoretical
	Yan and Tang [10]	Heuristic approach embedded in a framework designed	Real case (Taiwan International Airport)
	Ding et al. [6, 35]	Greedy algorithm	Theoretical
	Lim et al. [24]	The insert move algorithm The interval exchange move algorithm Greedy algorithm	Theoretical
	Diepen et al. [25]	Column generation	Real case (Amsterdam Airport)
	Dorndorf et al. [8]	Heuristic based on the ejection chain algorithm	Theoretical
	Mangoubi and Mathaisel [11]	Heuristic Approach	Real case (Toronto International Airport)
	Vanderstraeten and Bergeron [28]	ADAP	Theoretical
	Yan et al. [29]	Greedy heuristics	Real case (Chiang Kai-Shek Airport)
	Bolat [30]	Heuristic branch and trim	Real case (King Khaled International Airport, KSA)
	Bolat [34]	Heuristic branch and bound SPH heuristic	Real case (King Khaled International Airport, KSA)
	Haghani and Chen	Heuristic approach	Theoretical

	[13]		
	Genç [42]	Ground time maximization heuristic Idle time minimization heuristic	Theoretical and real case (Ataturk Airport of Istanbul, Turkey)
	B.A.C.o.E.B. Team and A.I.C.o.E. Team [17]	A hybrid heuristics algorithm guided by simulated annealing and greedy heuristic	Theoretical
	Bouras et al. [45]	Heuristic approach	Theoretical
Metaheuristics Algorithms	Ding et al. [6, 35]	Tabu search	Theoretical
	Ding et al. [7]	Simulated annealing Hybrid of simulated annealing and tabu search	Theoretical
	Lim et al. [24]	TS algorithm Memetic algorithm	Theoretical
	Hu and Di Paolo [36]	New genetic algorithm with uniform crossover	Theoretical
	Drexl and Nikulin [3]	Pareto simulated annealing	Theoretical
	Xu and Bailey [14]	Tabu search	Theoretical
	Bolat [31]	Genetic algorithm	Real case (King Khaled International Airport, KSA)
	Şeker and Noyan [9]	Tabu search algorithms	Theoretical
	Zheng et al. [33]	Tabu search algorithm Metaheuristic method	Real case (Beijing International Airport, China)
	Wei and Liu [16]	Hybrid genetic algorithm	Theoretical
	Gu and Chung [44]	Genetic algorithms approach	Theoretical
	Cheng et al. [23]	Genetic algorithm (GA) Tabu search (TS) Simulated annealing (SA) Hybrid approach based on SA & TS	Real case (Incheon International Airport, South Korea)
	Bouras et al. [45]	Genetic algorithm (GA) Tabu search (TS) Simulated annealing (SA)	Theoretical
OPL	Li [5, 32]	Optimization programming language (CPLEX)	Real case (Continental Airlines, Houston George Bush Intercontinental Airport)
	Tang et al. [27]	Using CPLEX 10.0 solver concert with C language	Real case (Taiwan International Airport)
	Prem Kumar and Bierlaire [18]	Optimization programming	Theoretical

		language (OPL)	
	Maharjan and Matis [40]	AMPL/CPLEX 11.2	Real case (Continental Airlines at GeorgeW. Bush Intercontinental Airport in Houston (IAH))

Table 3 Resolution Methods for GAP

3.2. Counter Assignment Problem (CAP)

The related literature for this check-in counter assignment problem is rather sparse; however, a lot of research has been done for airline adn airport optimization.

The problem was first analysed in a paped by Hon [49] who aims to optimize the counter assignment in Hong Kong International Airport. This publication presented a heuristic to solve the stochastic problem where the counter demands can change.

An adjacent resource scheduling problem was presented by Duin and Sluis [50]. Resources that are next to each other are adjacent resources. In this study, the authors provide mathematical formulations for the problem, and show that the decision version of the problem is strongly NP-complete.

Another real case is from Chiang Kai-Shek International Airport (CKS) in Taiwan. Yan et al. [51] formulated an integer programming model to help airport managers to assign common use check-in counters at the airport. In this study, the authors planned the problem monthly by minimizing the total walking distances of passengers. The demand of counters is assumed to be constant.

Yan et al. [52] made a development to their previous study and they formulated a model to minimize total inconsistencies in common-use counter assignments with a different number of counters. The model is binary integer programming and a heuristic was used to solve it.

Another heuristic to solve the problem was proposed by Wang Yeung and Chun [53]. They suggested an airport check-in counter assignment system that uses a genetic algorithm (GA).

Many publications are related to determine the actual counter demand and most of them minimize the number of required counters.

van Dijk and van der Sluis [2006] proposed an approach that decides the number of counters needed as a result of optimization and minimizes the maximum number of counters used at any time.

A network model for common use check-in groups optimization was presented by Tang [55]. In this study, the goal is minimizing the number of counters required for daily operations.

There are also some other references for optimizing the number of counters needed. First one is stated by Parlar and Sharafali [56] which is about a single flight check-in queueing estimation. The next one is proposed by Park and Ahn [57] about passenger arrival and the last one is a simulation paper to determine the counter usage by Chun and Mak [58].

Table 4 summarizes all solution techniques used recently for CAP.

Formulation / Method	References	Approach/Results	Problem Type
Integer Linear Programming (ILP)	Duin and Sluis [50]	Mathematical Formulations	Theoretical
	Yan et al. [51]	Mathematical Formulations	Real case (Chiang Kai-Shek International Airport(CKS))
Binary Integer Programming	Yan et al. [52]	Mathematical Formulations	Real case (Chiang Kai-Shek International Airport(CKS))
Network Representation	Tang [55]	Mathematical Formulations	
Heuristic Algorithms	Hon [49]	Heuristic Approach	Real case (Hong Kong International

			Airport)
	Yan et al. [52]	Heuristic Approach	Real case (Chiang Kai-Shek International Airport(CKS))
	van Dijk and van der Sluis [54]	Heuristic Approach	Theoretical
Metaheuristic Algorithms	Wang Yeung and Chun [53]	Genetic Algorithm	Theoretical

Table 4 Mathematical Formulations and Resolution Methods for CAP

CHAPTER 4: MATHEMATICAL MODELS: OBJECTIVES & FORMULATIONS

4.1. Introduction

The aim of this thesis is to find an accurately and efficient assignment for the resources which are gates and counters. The main motivation is to improve the efficiency of these airport operations by solving these two problems in our proposed methods with the real instance and obtaining effective schedulings for the system.

In this section, the objectives and the mathematical formulations of GAP and CAP are given. The decomposition heuristic for CAP model is also explained.

4.2. Gate Assignment Problem (GAP)

In this problem, the effectiveness of the gate assignment to a flight is measured by the term, *utility*. In other words, this utility value shows that this gate is how appropriate for the flight. Maximizing the total utility of the flight-gate assignment under some restrictions is the main objective for two IP formulations. Each gate has a utility value and also each gate takes a different utility value based on the flight if it is assigned to that gate. These utility values are defined after many observations and the meeting with the operation managers. Therefore; the multiplication of these utility values gives us the total utility of this assignment problem for two IP formulations.

The considered airport has a number of open park areas and bridge-equipped gates. Airport managers prefer flights to be assigned to bridge-equipped gates as it facilitates embarking and disembarking of passengers. In addition, after the arrival of flight, it needs to be refueled, replenished, all the waste has to be taken off-board. If all bridge-equipped gates are engaged, then flights are to be assigned to open park areas same as night stand flights. Some gates are for emergencies only. These are large enough for allocation of larger planes. For instance, if the bridges 26th and 42nd are full, large planes are assigned to 24th or 25th bridge-equipped parking area. Some airlines have a priority to be assigned to the same gates. Airline companies that use the same ground company services firms are assigned adjacent to each other in order to prevent apron traffic. With the information of the ground service firms of flights, departure and arrival of a plane is also considered. For instance, if a plane's departure is international, it has a priority for bridge-equipped parking areas in the international terminal. Similarly, if its departure is domestic, it has a priority in the domestic terminal. Some gates have priority due to their proximity to facilities in the airport. Not every plane fits in every gate. Hence, some flights cannot be assigned to some gates, which we call plane-gate eligibility. Our primary objective is improving gate utilization.

Due to combinatorial nature of the problem, we provide two different integer programming (IP) formulations, namely timetabling and assignment based, and then compare their performance.

Before giving IP model formulations, we introduce the notation used throughout the study.

Sets and Indices

i Index of periods , $i \in S = \{1, 2, ..., |S|\}$

j, b Index of flights, $j, b \in U = \{1, 2, ..., |U|\}$

k, r Index of parking areas, $k, r \in N = \{1, 2, ..., |N|\}$

c Index of ground service firms, $c \in C = \{1, 2, ..., |C|\}$

y Index of night-stand flights, $y \in Y = \{1, 2, ..., |Y|\}$

m Index of bridge-equipped parking areas in which night-stand planes cannot be assigned to, $m \in M = \{1, 2, ..., |M|\}$

d Index of parking areas that are occupied from the previous day, $d \in D = \{1, 2, ..., |D|\}$.

Parameters

a_j Scheduled arrival period of flight $j,\ a_j \in S$

g_j Scheduled departure period of flight $j,\ g_j \in S$. It is assumed that flight *j* left the airport in period $(g_j - 1)$. Hence, the same parking area can be used by another flight starting from the beginning of g_j . In other words, any flight occupies the assigned parking area in time interval $[a_j, g_j)$. Note that buffer periods for changes in flight schedules are also added to the g_j.

f_j Ground service company of flight $j,\ f_j \in C$

$$L_{kr} = \begin{cases} 1 & \text{If parking areas } k \text{ and } r \text{ are adjacent} \\ 0 & o/w \end{cases}$$

$$B_{jk} = \begin{cases} 1 & \text{If flight } j \text{ can be assigned to the parking area } k \\ 0 & o/w \end{cases}$$

t_k Earliest available period of parking area $k \in D,\ t_k \in S$

W_{jk} Utility of assigning flight *j* to parking area *k*, $W_{jk} \in \mathbb{R}^+$

w_k Utility of parking area $k, w_k \in \mathbb{R}^+$

4.2.1. Timetabling Based Integer Programming Model

In this section, we provide a timetabling based IP model for the gate assignment problem (GAP). Although, we assume gates as the limited resources and flights as the resource consumers, different than in the literature, in this model, we initialize a variable for each flight at each eligible gate during the service time. In other words, if a flight *j* can be assigned to gate *k*, we define (g_j-a_j) binary variables for flight *j* at gate *k* for periods [a_j, g_j).

Decision Variables

$$x_{ijk} = \begin{cases} 1 & \text{If plane } j \text{ is assigned to parking area } k \text{ in period } i \\ 0 & \text{O/W} \end{cases}$$

Mathematical Model

$$\text{Maximize} \quad \sum_{j \in U} \sum_{k \in N} W_{jk} w_k x_{a_j jk} \qquad (1)$$

s.t.

$$\sum_{k \in N} x_{a_j jk} = 1, \qquad \forall j \in U \qquad (2)$$

$$x_{ijk} = 0, \qquad \forall i \in S, \forall j \in U, \forall k \in N \mid (i < a_j) \wedge (i \geq g_j) \qquad (3)$$

$$\sum_{j \in U} x_{ijk} \leq 1, \qquad \forall i \in S, \ \forall k \in N \qquad (4)$$

$$\sum_{i \in S \mid a_j \leq i < g_j} x_{ijk} = (g_j - a_j) x_{a_j jk}, \qquad \forall j \in U, \forall k \in N \qquad (5)$$

$$x_{ijr} + x_{ibk} \leq 1, \qquad \forall i \in S, \forall j, b \in U, \forall k, r \in N \mid (j < b) \wedge (a_j \leq i < g_j) \wedge (f_j \neq f_b) \wedge (L_{kr} = 1) \wedge [(a_b \leq a_j \wedge g_j > a_b) \vee (a_j \leq a_b \wedge g_b > a_j)] \qquad (6)$$

$$x_{a_j jk} = 0, \qquad \forall j \in Y, \ \forall k \in M \qquad (7)$$

$$x_{a_j jk} = 0, \qquad \forall j \in U, \ \forall k \in D \mid a_j \leq t_k \qquad (8)$$

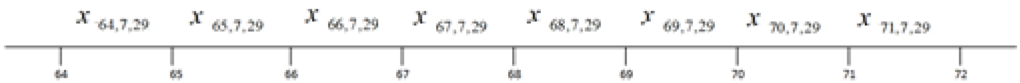
$$x_{a_j jk} \leq B_{jk}, \qquad \forall j \in U, \ \forall k \in N \qquad (9)$$

$$x_{ijk} \in \{0,1\} \qquad \forall j \in U, \forall i \in S, \forall k \in N \qquad (10)$$

Objective function (1) maximizes total utility of flight-gate assignment plan.

Constraints (2) ensure that each flight is assigned to exactly one gate at its arrival time.

Constraints (3) forbid assigning a flight to a gate before its arrival and after its departure. Note that if a flight's arrival and departure periods are, a_j and g_j respectively, assigned gate for the plane of that flight will occupy that gate from start of a_j to the start of g_j. For example, assume that a flight's (flight 7) arrival and departure times are 64th and 72nd time period. Then, corresponding gantt chart for the assigned gate (gate 29) will be as in the following:



$$\sum_{i=a_j \ (i=64)}^{g_j-1 \ (72-1=71)} x_{i,7,29} = (g_7 - a_7)x_{64,7,29}$$

$$x_{64,7,29} + x_{65,7,29} + \ldots + x_{71,7,29} = \left(g_7 - a_7\right)x_{64,7,29}$$

$$8 = 72 - 64$$

This example is taken from gantt chart of the optimal solution. See gantt chart for gate 29, flight 7 and time periods from 64 to 72.

In other words, if a flight is assigned to period *i*, it indicates that the assigned gate is used (i.e., occupied by that plane in between *i* and *i+1*).

Constraints (4) ensure that a parking area is occupied by at most one flight at any period.

Constraints (5) guarantee that the same parking area is used during the service period of the flight. See numerical example given for constraints (3). Since variable is equal to 1 for only one gate (due to constraints (2)), right hand side of constraints (5) will be equal to the service period of the flight for only the assigned parking area. And in that case, left hand side of the constraint will enforce that sum of the binary assignment variables in the consecutive periods will be equal to the service time period. Similarly, if flight *j* is not assigned to parking area *k* at its arrival period a_j, right hand side of the constraint (5) will be "0" and therefore enforce that the sum of the binary variables for the next consecutive periods during the service period will be equal to zero. In other words, that flight cannot be assigned to those periods. Note that this constraint is formulated for the periods in which between the arrival and departure times of the flight and executed once for each flight and parking area when the period is equal to that flights' arrival period.

Constraints (6) ensure that any two flights that are served with different companies and their service periods overlap must be assigned to non-neighbour parking areas.

Constraints (7) forbid assigning night-stand flights to bridge-equipped parking areas.

While constraints (8) ensure that a parking area can't be used before it becomes available, constraints (9) guarantee that each flight is assigned to only eligible parking areas.

Finally, constraints (10) give variable domain.

Proposed model is bounded with $|S| \text{ x } |U| \text{ x } |N|$ variables and $|S| \text{ x } |U|^2 \text{ x } |N|^2$ constraints.

Revised Version of Constraints (6)

$$x_{ijr} + x_{ibk} \leq 1, \qquad \forall i \in S, \forall j,b \in U, \forall k,r \in N \,|\, (j < b) \wedge [(a_j \leq i < g_j) \vee (a_b \leq i < g_b)] \wedge (f_j \neq f_b) \wedge (L_{kr} = 1) \wedge [(a_b \leq a_j \wedge g_b > a_j) \vee (a_j \leq a_b \wedge g_j > a_b)] \quad (6)$$

$(j < b)$: Generate this constraint for flight numbers $j < b$. Do not generate again $j > b$.

$[(a_j \leq i < g_j) \vee (a_b \leq i < g_b)]$: Generate this constraint for the largest time interval from earliest arrival to the largest departure of the flights *j* and *b*. In other words, *i* is within the following arrival $\min(a_j, a_b) \leq i < \max(g_j, g_b)$.

$(f_j \neq f_b)$: Generate this constraint if and only if flights *j* and *b* belong to different ground service companies.

$(L_{kr} = 1)$: Generate this constraint for only adjacent parking areas.

$[(a_b \leq a_j \wedge g_b > a_j) \vee (a_j \leq a_b \wedge g_j > a_b)]$: Generate this constraint if and only if flight *j* arrives while flight *b* has been already in the airport and departure of

b is later than j $(a_b \le a_j \wedge g_b > a_j)$ or vice versa $(a_j \le a_b \wedge g_j > a_b)$. This filtering guarantees that we consider only overlapping periods.

4.2.2. Assignment Based Integer Programming Model

In the second alternative IP model formulation, to reduce the number of binary variables, we define binary variables for only those periods where corresponding flights arrive to the airport. Before giving the new IP model, we introduce re-defined sets, indices, parameters and variables as in the following:

Sets&Indices

S Number of periods in a day

i Index of periods, $i \in S = \{1, 2, ..., |S|\}$

U Set of all flights

j,b Index of flights, $j,b \in U = \{1, 2, ..., |U|\}$

N Set of all parking areas

k,r Index of parking areas, $k,r \in N = \{1, 2, ..., |N|\}$

C Set of ground services firms

c Index of ground services firms, $c \in C = \{1, 2, ..., |C|\}$

Y Set of night-stand flights

y Index of night-stand flights, $y \in Y = \{1, 2, ..., |Y|\}$

MD Set of bridge-equipped parking areas in which night-stand planes cannot be assigned to

m Index of bridge-equipped parking areas in which night-stand planes cannot be assigned to, $m \in MD = \{1, 2, ..., |MD|\}$

DK Set of parking areas that are occupied from the previous day

d Index of parking areas that are occupied from the previous day, $d \in D = \{1, 2, ..., |DK|\}$

Parameters

a_j Scheduled arrival period of flight j, $a_j \in S$

g_j Scheduled departure period of flight j, $g_j \in S$

f_j Ground service company of flight j, $f_j \in C$

$$L_{kr} = \begin{cases} 1 & \text{If parking areas } k \text{ and } r \text{ are neighbours} \\ 0 & \text{O/W} \end{cases}$$

$$B_{jk} = \begin{cases} 1 & \text{If flight } j \text{ can be assigned to the parking area } k \\ 0 & \text{O/W} \end{cases}$$

t_d Earliest available period of parking area $d \in D$, $t_d \in S$

W_{jk} Utility of assigning flight j to parking area k, $W_{jk} \in \mathbb{R}^+$

w_k Utility of parking area k, $w_k \in \mathbb{R}^+$

M A big number (Number of flights is considered as an upper bound)

Decision Variables

$$x_{jk} = \begin{cases} 1 & \text{If plane } j \text{ is assigned to parking area } k \\ 0 & \text{O/W} \end{cases}$$

Mathematical Model

$$\text{Maximize} \quad \sum_{j \in U} \sum_{k \in N} W_{jk} w_k x_{jk} \qquad (1)$$

s.t.

$$\sum_{k \in N} x_{jk} = 1, \qquad \forall j \in U \qquad (2)$$

$$\sum_{u \in U | (u \neq j) \wedge (a_u \leq a_j < g_u)} x_{uk} \leq M(1 - x_{jk}), \qquad \forall j \in U, \forall k \in N \qquad (3)$$

$$x_{jk} + x_{hr} \leq 1, \qquad \forall j,h \in U, \forall k,r \in N \,|\, (j < h) \wedge (f_j \neq f_h) \wedge (L_{kr} = 1) \wedge [(a_h \leq a_j < g_h) \vee (a_j \leq a_h < g_j)] \qquad (4)$$

$$x_{jk} \leq B_{jk}, \qquad \forall j \in U,\ \forall k \in N \qquad (5)$$

$$x_{ym} = 0, \qquad \forall y \in Y,\ \forall m \in MD \qquad (6)$$

$$x_{jd} = 0, \qquad \forall j \in U,\ \forall d \in D \,|\, a_j \leq \mathrm{t}_d \qquad (7)$$

$$x_{jk} \in \{0,1\} \qquad \forall j \in U, \forall k \in N \qquad (8)$$

Objective function (1) maximizes total utility of flight-gate assignment plan.

Constraints (2) ensure that each flight is assigned to exactly one gate.

Constraints (3) forbid assigning another flight to a gate before assigned flight’s arrival and after its departure.

Constraints (4) ensure that any two flights that are served with different companies and their service periods overlap must be assigned to non-neighbour parking areas.

Constraints (5) guarantee that each flight is assigned to only eligible parking areas.

Constraints (6) forbid assigning night-stand flights to bridge-equipped parking areas.

Constraints (7) ensure that a parking area can’t be used before it becomes available.

Finally, constraints (8) give variable domain.

4.3. Counter Assignment Problem (CAP)

In this part of the study, the objectives of the counter assignment problem that we preferred are described. According to the policy of the airport management and contracts with the airlines, there can be many different objectives for making this kind of assignments. In our study, the objectives are generated after the meetings with the operation managers.

Our defined objectives are in the following in the order of preference:

-The flights of the same airline should be assigned into the adjacent counters.
-The flights should be assigned to preferred counters.
-The different number of airlines assigned to each counter should be minimized.

The considered airport has a number of counters. Airport managers assign the flights to the counters under some considerations. The need of counters for a flight begins before 2,5 hours of that flight's departure. The demand of counters of that flight ends before 0,5 hour of that flight's departure. The total number of counters that flight needs is also deterministic and known. The most important constraint for this assignment is assigning the deserved number of counters of a flight to adjacent counters. In addition, if more than one flight of the same airline need some counters at the same time, managers prefer to assign these flights to adjacent counters. One preferation that managers do is assigning flights into counters by starting from counter 1.

Another situation is that if an airline is assigned to a counter and if in the following periods there is a flight of the same airline, managers prefer to assign that coming flight to the assigned counter.

Before giving IP model formulation, we introduce the notation used throughout the study.

Sets & Indices

S Number of periods in a day
i Index of periods, $i \in S = \{1, 2, ..., |S|\}$
U Set of all flights
j, h Index of flights, $j, h \in U = \{1, 2, ..., |U|\}$
N Set of all check in counters
k, r Index of check in counters, $k, r \in N = \{1, 2, ..., |N|\}$
C Set of ground service providing firms
m Index of ground service providing firms, $m \in C = \{1, 2, ..., |C|\}$
DK Set of check in counters in use at the beginning of the planning horizon

Parameters

a_j Scheduled opening period of flight j's counters, $a_j \in S$
g_j Scheduled closing period of flight j's counters, $g_j \in S$
c_j Number of counters that flight j needs, $c_j \in N$
f_j Ground service company of flight j, $f_j \in C$

$$L_{kr} = \begin{cases} 1 & \text{If counters } k \text{ and } r \text{ are adjacent} \\ 0 & \text{O/W} \end{cases}$$

T_k Earliest available period of counter $k \in DK$, $T_k \in S$
M A big number

Decision Variables

$$x_{jk} = \begin{cases} 1 & \text{If flight } j \text{ is assigned to counter } k \\ 0 & \text{O/W} \end{cases}$$

$$y_{jk} = \begin{cases} 1 & \text{If flight } j \text{ is first assigned to counter } k \\ 0 & \text{O/W} \end{cases}$$

$$w_{jkhr} = \begin{cases} 1 & \text{If flight } j \text{ is assigned to counter } k \text{ and flight } h \text{ is assigned to counter } r \\ & \text{where } f_j \neq f_h \text{ and } L_{kr} = 1 \\ 0 & \text{O/W} \end{cases}$$

$$b_{mk} = \begin{cases} 1 & \text{If firm } m \text{ is assigned to counter } k \\ 0 & \text{O/W} \end{cases}$$

Mathematical Model

$$Minimize \quad \sum_{j\in U}\sum_{k\in N}\sum_{h\in U}\sum_{r\in N} w_{jkhr} + \sum_{m\in C}\sum_{k\in N} b_{mk} + \sum_{j\in U}\sum_{k\in N} x_{jk}k \qquad (1)$$

s.t.

$$\sum_{k\in N} x_{jk} = c_j, \qquad \forall j \in U \qquad (2)$$

$$\sum_{k\in N} y_{jk} = 1, \qquad \forall j \in U \qquad (3)$$

$$\sum_{h\in U|(h\neq j)\wedge(a_j \le a_h < g_j)} x_{hk} \le M(1-x_{jk}), \qquad \forall j \in U, \qquad \forall k \in N \qquad (4)$$

$$x_{jk} + x_{hr} \ge 2w_{jkhr}, \qquad \forall j \in U, \qquad \forall k \in N,\ \forall h \in U, \forall r \in N \mid (L_{kr} = 1) \wedge (h > j) \wedge (f_j \ne f_h) \wedge ((a_h \le a_j < g_h) \vee (a_j \le a_h < g_j)) \qquad (5)$$

$$x_{jk} + x_{hr} \le 1 + w_{jkhr}, \qquad \forall j \in U, \qquad \forall k \in N,\ \forall h \in U, \forall r \in N \mid (L_{kr} = 1) \wedge (h > j) \wedge (f_j \ne f_h) \wedge ((a_h \le a_j < g_h) \vee (a_j \le a_h < g_j)) \qquad (6)$$

$$\sum_{r\in N|(r\ge k)\wedge(r\le k+c_j-1)} x_{jr} \ge c_j y_{jk}, \qquad \forall j \in U, \qquad \forall k \in N \qquad (7)$$

$$\sum_{r\in N|(r\ge k)\wedge(r\le k+c_j-1)} x_{jr} \le c_j y_{jk} + M(1-y_{jk}), \qquad \forall j \in U, \qquad \forall k \in N \qquad (8)$$

$$x_{jk} \le b_{f_j k}, \qquad \forall j \in U, \forall k \in N \qquad (9)$$

$$b_{mk} \le \sum_{j\in U|f_j = m} x_{jk}, \qquad \forall m \in C, \forall k \in N \qquad (10)$$

$$x_{jk} = 0, \qquad \forall j \in U, \forall k \in DK \mid a_j < T_k \qquad (11)$$

In the objective function (1), the first summation is for assigning of the same airline into adjacent counters. The second summation is for minimizing the different number of airlines assigned to each counter and the last one is for assigning flights to preferred counters.

The constraints (2) ensure that required number of counters are assigned to each flight.

Tracing the index of the first counter in which each flight is assigned to is done by constraints (3).

Constraints (4) ensure that there are non-overlapping flights.

Constraints (5) and (6) are for tracing that whether any two flights of different airline companies are assigned into adjacent counters or not.

“All counters assigned to a flight must be adjacent” constraint is done by constraints (7) and (8).

Constraints (9) say that assigning a flight to a counter means that its airline company uses that counter.

Constraints (10) ensure that a counter is used by an airline company if and only if at least one flight of that company is assigned to that counter.

Assigning a flight to a counter iff that counter is available at the counter opening period is guaranteed by constraints (11).

This IP model formulation gives optimal solution up to 9 flights in one-hour.

Therefore, we developed an approach which is dividing the problem into solvable small pieces, decomposition.

According to our approach, we first consider a limited sub-period instead of the whole day. As a second, we consider limited number of flights in each sub-problem. Finally, we communicate sub-problems and carry state of counters to the next sub-problem.

4.4. Decomposition Algorithm for CAP

Algorithm 1 Decomposition Algorithm for Assignment of Checkin Counters to Flights

---

$maxflight = M,\ U^* = \emptyset,\ subgroup = 1$
Sort U in non-decreasing order of a_j and non-increasing order of c_j in case of tie
for all $s \in S$ **do**
    $U^* \leftarrow \{j |\ j \in U \land a_j \in s\}$
    if $|U^*| \leq maxflight$ **then**
        $U_{subgroup} \leftarrow U^*$
        $U \leftarrow U \setminus U^*$
        $U^* = \emptyset, subgroup++$
    else
        $subsets = \lceil |U^*| / maxflight \rceil$
        for $i \leftarrow 1$ **to** $subsets$ **do**
            $selected = 1, U_{subgroup} = \emptyset$
            while $|U^*| \neq 0 \lor selected \leq maxflight$ **do**
                $U_{subgroup} \leftarrow U_{subgroup} \cup U^*_{[selected]}$
                $U^* \leftarrow U^* \setminus U_{[selected]}$
                $selected++$
            $subgroup++$
for $s \leftarrow 1$ **to** $subgroup$ **do**
    solve decMIP $(U_{[s]})$
    update availability of counters
return assignment of counters to flights

In this decomposition algorithm, we need to make some modifications in our IP model formulation to communicate sub-problems and carry the state of the counters to the next sub-problem. The revised model formulation is as follows:

Sets & Indices

S Number of periods in a day
i Index of periods, $i \in S = \{1,2,....,|S|\}$
U Set of all flights
j,h Index of flights, $j,b \in U = \{1,2,....,|U|\}$
N Set of all contuars
k,r Index of contuars, $k,r \in N = \{1,2,....,|N|\}$
C Set of ground services firms
m Index of ground services firms, $m \in C = \{1,2,....,|C|\}$
DK Set of contuars assigned in previous subgroups
pF Set of ground service firm numbers of flights assigned in previous subgroup
cF Set of ground service firm numbers of flights in current subgroup

Parameters

a_j Scheduled starting period of flight j's contuars, $a_j \in S$
g_j Scheduled closed period of flight j's contuars, $g_j \in S$
c_j Number of contuars that flight j needs, $c_j \in N$
f_j Ground service company of flight j, $f_j \in C$

$$L_{kr} = \begin{cases} 1 & \text{If contuars } k \text{ and } r \text{ are adjacent} \\ 0 & \text{O/W} \end{cases}$$

T_k Earliest available period of contuar $k \in DK$, $T_k \in S$
z_k Ground service firm number of contuar k assigned in previous subgroups $k \in DK$, $z_k \in C$

$$bp_{mk} = \begin{cases} 1 & \text{If firm } m \text{ is assigned to contuar } k \text{ in previous subgroup} \\ 0 & \text{O/W} \end{cases}$$

M A big number

```
Execute pre{
var hold=0;
for(var r in DK){
    hold = z_r;
    for(var s in C){
        if(s = hold){
            bp_sr = 1;
        }
    }
}
}
```

Decision variables

$$x_{jk} = \begin{cases} 1 & \text{If flight } j \text{ is assigned to contuar } k \\ 0 & \text{O/W} \end{cases}$$

$$y_{jk} = \begin{cases} 1 & \text{If flight } j \text{ is first assigned to contuar } k \\ 0 & \text{O/W} \end{cases}$$

$$w_{jkhr} = \begin{cases} 1 & \text{If flight } j \text{ is assigned to contuar } k \text{ and flight } h \text{ is assigned to contuar } r \\ 0 & \text{O/W} \end{cases}$$

$$b_{mk} = \begin{cases} 1 & \text{If firm } m \text{ is assigned to contuar } k \\ 0 & \text{O/W} \end{cases}$$

dif_{mk} The difference amount at contuar k for firm m

Mathematical Model

Objective Function

$$\text{Minimize} \quad \sum_{j\in U}\sum_{k\in N}\sum_{h\in U}\sum_{r\in N} w_{jkhr} + \sum_{m\in C}\sum_{k\in N} dif_{mk} + \sum_{j\in U}\sum_{k\in N} x_{jk}k \qquad (1)$$

s.t.

$$\sum_{k\in N} x_{jk} = c_j, \qquad \forall j \in U \qquad (2)$$

$$\sum_{k\in N} y_{jk} = 1, \qquad \forall j \in U \qquad (3)$$

$$\sum_{h\in U|(h\neq j)\wedge(a_j \leq a_h < g_j)} x_{hk} \leq M(1 - x_{jk}), \qquad \forall j \in U, \qquad \forall k \in N \qquad (4)$$

$$x_{jk} + x_{hr} \geq 2w_{jkhr}, \qquad \forall j \in U, \qquad \forall k \in N,\ \forall h \in U, \forall r \in N \mid (L_{kr} = 1) \wedge (h > j) \wedge (f_j \neq f_h) \wedge ((a_h \leq a_j < g_h) \vee (a_j \leq a_h < g_j)) \qquad (5)$$

$$x_{jk} + x_{hr} \leq 1 + w_{jkhr}, \qquad \forall j \in U, \qquad \forall k \in N,\ \forall h \in U, \forall r \in N \mid (L_{kr} = 1) \wedge (h > j) \wedge (f_j \neq f_h) \wedge ((a_h \leq a_j < g_h) \vee (a_j \leq a_h < g_j)) \qquad (6)$$

$$\sum_{r\in N|(r\geq k)\wedge(r\leq k+c_j-1)} x_{jr} \geq c_j y_{jk}, \qquad \forall j \in U, \qquad \forall k \in N \qquad (7)$$

$$\sum_{r \in N|(r \geq k) \wedge (r \leq k + c_j - 1)} x_{jr} \leq c_j y_{jk} + M(1 - y_{jk}), \qquad \forall j \in U, \qquad \forall k \in N \qquad (8)$$

$$x_{jk} \leq b_{f_j k}, \qquad \forall j \in U, \forall k \in N \qquad (9)$$

$$b_{mk} \leq \sum_{j \in U | f_j = m} x_{jk}, \qquad \forall m \in C, \forall k \in N \qquad (10)$$

$$x_{jk} = 0, \qquad \forall j \in U, \forall k \in DK \,|\, a_j < T_k \qquad (11)$$

$$\sum_{j \in U} y_{jk} \leq 1, \qquad \forall k \in N \qquad (12)$$

$$b_{mk} - bp_{mk} \leq dif_{mk}, \qquad \forall m \in C, \forall k \in N \,|\, (k \in DK) \wedge (m \in cF) \wedge (m \in pF)$$

(13)

$$bp_{mk} - b_{mk} \leq dif_{mk}, \qquad \forall m \in C, \forall k \in N \,|\, (k \in DK) \wedge (m \in cF) \wedge (m \in pF)$$

(14)

The objective function (1) and the constraints from (2) to (11) have the same role as in the previous model formulation.

Constraints (12) ensure that each counter can have at most one flight assignment in a sub-problem because of the length of the sub-problem which is 2 hours.

Constraints (13) and (14) is for tracing that the assigned flights to a counter have the same or different airlines.

CHAPTER 5: RESULTS & DISCUSSION

5.1. Gate Assignment Problem (GAP)

In this section, the results of implementing developed models by a real case are provided. Before presenting the numerical results, in the following table, the size of proposed models in terms of theoretical bounds in the number of variables and constraints are compared:

Formulation	Model (Approach)	Number of Variables	Number of Constraints
Integer Programming	Timetabling	$\sum_{j \in U} (g_j - a_j) \lvert N \rvert$	$(\lvert U \rvert \lvert N \rvert)(\lvert U \rvert \lvert N \rvert - 1)/2$
	Assignment	$\lvert U \rvert \lvert N \rvert$	$(\lvert U \rvert \lvert N \rvert)(\lvert U \rvert \lvert N \rvert - 1)/2$

Table 5 Theoretical bounds for the size of the models

Developed models are tested using a realistic size instance provided by a main airline operator in Turkey with 35 parking areas in which 19 are bridge-equipped, 105 flights, and four different ground service companies. For the analysis, we used an optimization programming language, IBM ILOG CPLEX Version 12.6.

Performances of developed models are summarized in the following table:

Assignment Model	Number of Variables	Number of Constraints	Number of Nodes	Objective	Time (seconds)
Timetabling	96025	649207	16859	8832	101*
Assignment	3467	202340	31245	8832	116*

Table 6 Performance evaluation of developed models on a real size instance where * indicates optimality of the solution

Although both theoretical and instance based results show that there are more variables and constraints in timetabling based IP formulation, it provides tighter bound for the problem. The reason is that, timetabling model allows a stronger LP-relaxation since it does not rely on big-M based constraints as in the assignment model. Numerical results also show the same results. Although both IP models find the optimal solution, timetabling model takes less time with smaller search tree in terms of number of nodes.

5.1.1. Timetabling Based Integer Programming Model

As discussed in the previous section, timetabling based IP solution gives a tighter bound for the problem; however, it has more variables and constraints. This model allows a stronger LP-relaxation because it does not rely on big-M based constraints as in the assignment model.

The numerical instance is provided in Appendix A.

The results of this integer programming model are shown in Appendix B.

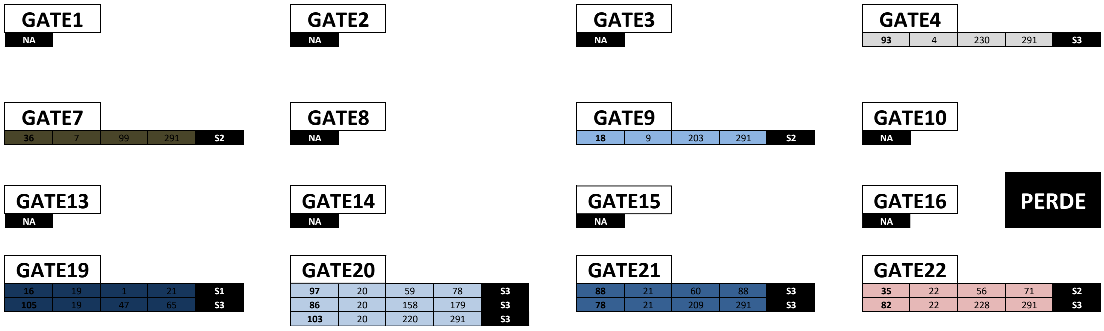


Figure 6 The small part of the results of GAP for Timetabling Based Model based on each gate

The full version of this representation of flights on each gate is shown in Appendix B.

5.1.2. Assignment Based Integer Programming Model

As stated in the previous parts, it has less variables and constraints in contrast to the timetabling based integer programming model. However, the bound is not as good as the one of timetabling based model. The reason also told previously is that this formulation has big-M based constraints.

The numerical instance is the same as in the previous model which is provided in Appendix A.

The results are shown in Appendix C.

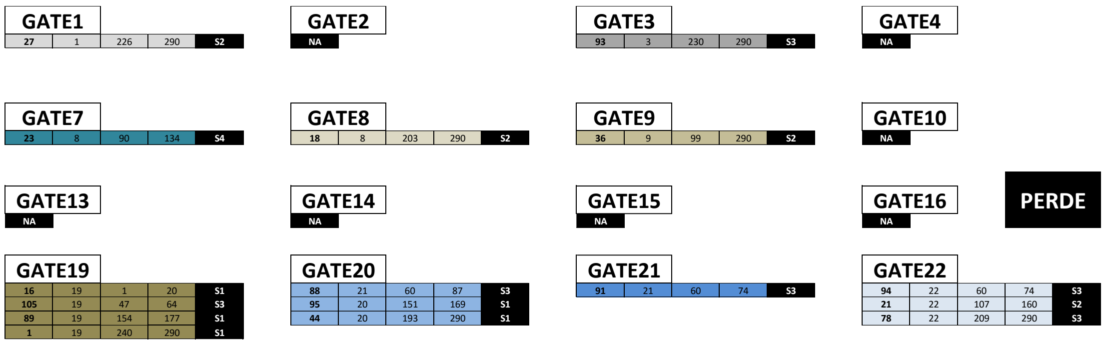


Figure 7 The small part of the results of GAP for Assignment Based Model based on each gate

The full version of this representation of flights on each gate is shon in Appendix C.

5.2. Counter Assignment Problem (CAP)

In this section, the results of implementing developed model by a real case are provided.

The model is tested using a realistic size instance provided by a main airline operator in Turkey with 135 counters, 150 flights, and 52 different companies.

As explained before, the model reached a solution with the help of decomposition algorithm. In the following table; the run-time of the model which has an instance of 5, 6, 7, 8, 9 and 10 flights are provided.

FLIGHT NO	RUN-TIME
5	00:00:18:30
6	00:00:28:13
7	00:00:58:61
8	00:02:37:04
9	00:32:21:06
10	Over 1.5 hour

Table 7 The run-time of the model with some specified flight numbers

In this study, the subproblem size is taken as 5. According to the Table 6, it is the fastest and effective number for the solution. First, all flights are separated into groups and in each group, the difference between the open time of the counters for flights is maximum 2 hours (Table 21). Then, because of the limited model size for flights, each group is also divided into subgroups which consist of 5 flights. The numerical instance is provided in Appendix D.

For the analysis, we used an optimization programming language, IBM ILOG CPLEX Version 12.6.

Performances of the developed model’s sub-problems are summarized in the following table:

MODELS	RUNTIME	OBJECTIVE FUNCTION
1_1	00:00:18:68	81
1_2	00:00:20:31	252
1_3	00:00:29:52	510
1_4	00:00:22:14	839
1_5	00:00:28:96	1077
1_6	00:00:25:29	1066
1_7	00:00:27:89	1435
1_8	00:00:22:41	1435
1_9	00:00:06:00	348
2_1	00:00:30:64	702
2_2	00:00:24:69	1313
2_3	00:00:07:15	318
2_4	00:00:08:50	138
2_5	00:00:18:59	420
2_6	00:00:15:58	801
2_7	00:00:16:08	1082
2_8	00:00:13:79	1552
2_9	00:00:16:18	2176
2_10	00:00:05:69	697
3_1	00:00:09:13	293
3_2	00:00:13:93	569
3_3	00:00:14:21	1681
3_4	00:00:13:93	253
3_5	00:00:13:78	465
3_6	00:00:14:77	664
3_7	00:00:24:16	588
3_8	00:00:28:59	1097
3_9	00:00:27:76	1429
3_10	00:00:06:17	1117
4_1	00:00:24:39	1252
4_2	00:00:23:72	967
4_3	00:00:21:88	966
4_4	00:00:21:76	752
4_5	00:00:08:05	177

Table 8 Performance evaluation of sub-problems

The results are shown in Appendix E.

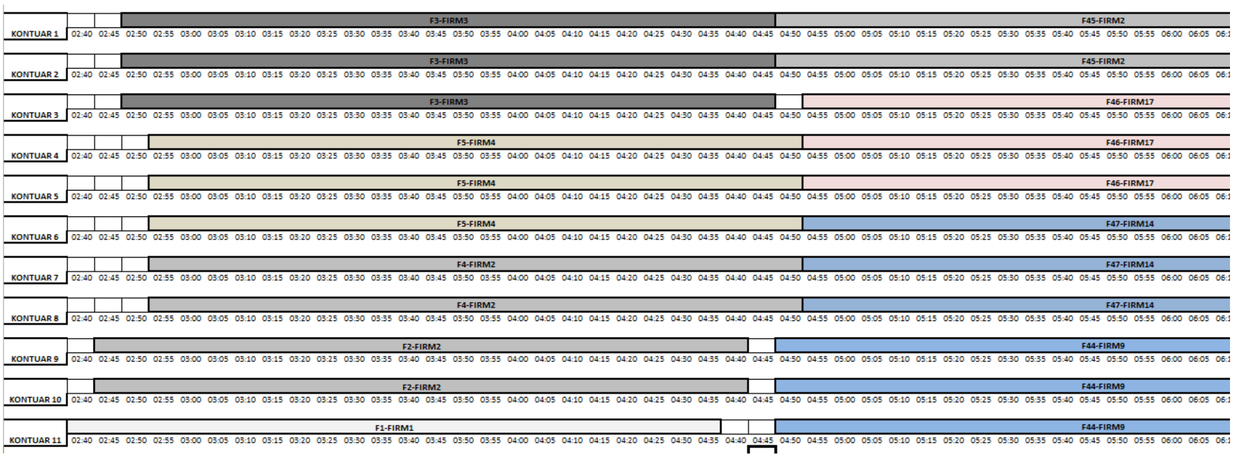


Figure 8 The small part of the results of CAP based on each counter

The full version of this representation of flights on each gate is shown in Appendix E.

CHAPTER 6: CONCLUSIONS & FUTURE WORK

With the increase in the intensity of air-traffic in recent years, the management of airport gates and check-in counters has become more important and complicated. The improper assignment of gates to incoming and outgoing flights may result in flight delays, customer dissatisfaction, and increase in operational costs. The same situation is also undertaken by the unefficient assignment of the counters to the flights in the airport. As a result, many studies have been published to efficiently solve these related problems and use these resources.

In this study, two IP models are proposed for Gate Assignment Problem (GAP) and an IP model with a decomposition algorithm is developed for Counter Assignment Problem (CAP). The models are nearly efficient to solve this highly complicated and over-constrained flight-gate and flight-counter assignment problems to optimality. IP models for Gate Assignment Problem (GAP) are able to find the optimal solution in about 100 seconds, whereas the IP model with decomposition algorithm for Counter Assignment Problem (CAP) does not give the solution at that speed. But according to the complexity of this problem, the solution is still efficient. More than 15 flights, model is hard put to take a solution. Therefore, with the help of the decomposition algorithm, the solution of the problem with taken instance from the airport operator is reached. Even though it is not optimal, it is near to that point. So, good quality solutions are obtained in reasonable time.

For future research directions, the problem of CAP can be extended by changing the size of the sub-problems (from 5 flights to any other one). As a next extension for this problem, you consider expected maximal queue lengths in assignment of flights to the counters or consider baggage handling capacities, or combine the counter assignment solver with a simulation tool for planning counter requirements.

For research directions of solution methods, developing a heuristic algorithm for companies that do not own commercial optimization solvers can be one study gor the Gate Assignment Problem (GAP). In addition to this, hybridizing a constraint programming based multi-dimensional placement model formulation with large neighbourhood search meta-heuristic can be another direction. For Counter Assignment Problem (CAP), the performance of the model and the decomposition algorithm can be improved. A fast and an efficient custom heuristic can be developed for this problem. You can take advantage of global contstraint. With constraint programming technology, multi-dimemsional placement problem formulation with global constraints can be useful. In addition, you can compute good lower bounds for this problem.

REFERENCES

[1] G. N. Steuart, “Gate position requirements at metropolitan airports,” *Transportation Science*, vol.8, no. 2, pp. 169-189, 1974.

[2] S. Yan and C.-M. Huo, “Optimization of multiple objective gate assignments,” *Transportation Research A: Policy and Practice*, vol. 35, no. 5, pp. 413-432, 2001.

[3] A. Drexl and Y. Nikulin, “Multicriteria airport gate assignment and Pareto simulated annealing,” *IIE Transactions,* vol. 40, no. 4, pp. 385-397, 2008.

[4] W. Li and X. Xu, “Optimized assignment of airport gate configuration based on immune genetic algorithm,” in *Measuring Technology and Mechatronics Automation in Electrical Engineering*, pp. 347-355, Springer, 2012.

[5] C. Li, “Airport gate assignment: A hybrid model and implementation,” http://arxiv.org/abs/0903.2528.

[6] H. Ding, A. Lim, B. Rodrigues, and Y. Zhu, “Neew heuristics for over-constrained flight to gate assignments,” *Journal of the Operational Research Society*, vol. 55, no. 7, pp. 760-768, 2004.

[7] H. Ding, A. Lim, B. Rodrigues, and Y. Zhu, “The over-constrained airport gate assignment problem,” *Computers & Operations Research*, vol. 32, no. 7, pp. 1867-1880, 2005.

[8] U. Dorndorf, F. Jaehn, and E. Pesch, “Flight gate scheduling with respect to a reference schedule,” *Annals of Operations Research*, vol. 194, pp. 177-187, 2012.

[9] M. Şeker and N. Noyan, “ Stochastic optimization models for the airport gate assignment problem,” *Transportation Research E: Logistics and Transportation Review*, vol. 48, no. 2, pp. 438-459, 2012.

[10] S. Yan and C.-H. Tang, “A heuristic approach for airport gate assignments for stochastic flight delays,” *European Journal of Operational Research*, vol. 180, no. 2, pp. 547-567, 2007.

[11] R. S. Mangoubi and D. F. X. Mathaisel, “Optimizing gate assignments at airport terminals,” *Transportation Science*, vol. 19, no. 2, pp. 173-188, 1985.

[12] R. A. Bihr, “A conceptual solution to the aircraft gate assignment problem using 0,1 linear programming,” *Computers and Industrial Engineering*, vol. 19, no. 1-4, pp. 280-284, 1990.

[13] A. Haghani and M.-C. Chen, “Optimizing gate assignments at airport terminals,” *Transportation Research A: Policy and Practice*, vol. 32, no. 6, pp. 437-454, 1998.

[14] J. Xu and G. Bailey, “The airport gate assignmentproblem: mathematical model and a tabu search algorithm,” in *Proceedings of the IEEE 34th Annual Hawaii International Conference*, Delta Technology Inc., Atlanta, Ga, USA, 2001.

[15] S. Lam, J. Cao, anf H. Fan, “Development of an intelligent agent for airport gate assignment,” *Journal of Airport Transportation*, vol. 7, no. 2, pp. 103-114, 2002.

[16] D.-X. Wei and C.-Y. Liu, “Fuzzy model and optimization for airport gate assignment problem,” in *Proceedings of the IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS ’09)*, pp. 828-832, Shangai, China, November 2009.

[17] B.A.C.o.E.B. Team and A.I.C.o.E. Team, *Gate Assignment Solution (GAM) Using Hybrid Heuristics Algorithm*, Wipro Technologies (WIT), 2009.

[18] V. Prem Kumar and M. Bierlaire, “Multi-objective airport gate assignment problem in planning and operations,” *Journal of Advanced Transportation*, vol. 48, no. 7, pp. 902-926, 2014.

[19] H. S. Al-Khalifah, *Dynamic Gate Assignment at an Airport*, Department of Industrial Engineering, King Saud University, Riyadh, Saudi Arabia, 2005.

[20] U. Dorndorf, A. Drexl, Y. Nikulin, and E. Pesch, “Flight gate scheduling: state-of-the-art and recent developments,” *Omega*, vol. 35, no. 3, pp. 326-334, 2007.

[21] T. Obata, *Quadratic Assignment Problem: Evaluation of Exact and Heuristic Algorithms*, Rensselaer Polytechnic Institute, Troy, NY, USA, 1979.

[22] U. Dorndorf, F. Jaehn, C. Lin, H. Ma, and E. Pesch, “Disruption management in flight gate scheduling,” *Statistica Neerlendica. Journal of the Netherlands Society for Statistics and Operations Research*, vol. 61, no. 1, pp. 92-114, 2007.

[23] C.-H. Cheng, S. C. Ho, and C.-L. Kwan, “The use of metaheuristics for airport gate assignment,” *Expert Systems with Applications*, vol. 39, no. 16, pp. 12430-12437, 2012.

[24] A. Lim, B. Rodrigues, and Y. Zhu, ”Airport gate scheduling with time windows,” *Artificial Intelligence Review*, vol. 24, no. 1, pp. 5-31, 2005.

[25] G. Diepen, J. M. van den Akker, J. A. Hoogeveen, and J. W. Smeltink, *Using Column Generation for Gate Planning at Amsterdam Airport Schiphol*, Department of Information and Computing Sciences, Utrecht University, 2007.

[26] G. Diepen, J. M. V. D. Akker, and J. A. Hoogeveen, *Integration of Gate Assignment and Platform Bus Planning*, Department of Information and Computing Sciences, Utrecht University, 2008.

[27] C. Tang, S. Yan, and Y. Hou, “A gate reassignment framework for real time flight delays,” *4OR: A Quarterly Journal of Operations Research*, vol.8, no. 3, pp. 299-318, 2010.

[28] G. Vanderstraeten, and M. Bergeron, “Automatic assignment of aircraft to gates at a terminal,” *Computers and Industrial Engineering*, vol. 14, no. 1, pp. 15-25, 1988.

[29] S. Yan, C.-Y. Shieh, and M. Chen, “A simulation framework for evaluating airport gate assignments,” *Transportation Research A: Policy and Practice*, vol. 36, no. 10, pp. 885-898, 2002.

[30] A. Bolat, “Assigning arriving flights at an airport to the available gates,” *Journal of the Operational Research Society,* vol. 50, no. 1, pp. 23-34, 1999.

[31] A. Bolat, “Models and a genetic algorithm for static aircraft-gate assignment problem,” *Journal of the Operations Research Society,* vol. 52, no. 10, pp. 1107-1120, 2001.

[32] C. Li, “Airport gate assignment: new model and implementation,” http://arxiv.org/abs/0811.1618

[33] P. Zheng, S. Hu, and C. Zhang, “Airport gate assignments model and algorithm,” in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT ’10)*, vol. 2, pp. 457-461, IEEE, Chengdu, China, July 2010.

[34] A. Bolat, “Procedures for providing robust gate assignments for arriving aircrafts,” *European Journal of Operational Research*, vol. 120, no. 1, pp. 63-80, 2000.

[35] H. Ding, A. Lim, B. Rodrigues, and Y. Zhu, “Aircraft and gate scheduling optimization at airports,” in *Proceedings of the 37th Annual Hawaii International Conference on System Sciences*, pp. 1185-1192, January 2004.

[36] X. B. Hu, and E. Di Paolo, “An efficient Genetic Algorithm with uniform crossover for the multi-objective airport gate assignment problem,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’07)*, pp. 55-62, September 2007.

[37] I. Kaliszewski , and J. Miroforidis, “On interfacing multiobjective optimisation models the case of the airport gate assignment problem,” in *Proceedings of the International Conference on Application and Theory of Automation in Command and Control Sytems*, London, UK, 2012.

[38] H. M. Genç, O. K. Erol, İ. Eksin, M. F. Berber, and B. O. Güleryüz, “A stochastic neighborhood search approach for airport gate assignment problem,” *Expert Systems with Applications*, vol. 39, no. 1, pp. 316-327, 2012.

[39] W. Li, “Optimized assignment of çivil airport gate,” in *Proceedings of the International Conference on Intelligent System Design and Engineering Application (ISDEA ’10)*, pp. 33-38, Changsha, China, October 2010.

[40] B. Maharjan, and T. I. Matis, “Multi-commodity flow network model of the flight gate assignment problem,” *Computers and Industrial Engineering*, vol. 63, no. 4, pp. 1135-1144, 2012.

[41] G. Diepen, J. M. van den Akker, and J. A. a. Hoogeveen, “Finding a robust assignment of flights to gates at Amsterdam Airport Schiphol,” *Journal of Scheduling*, vol. 15, no. 6, pp. 703-715, 2012.

[42] H. M. Genç, “A new solution approach for the airport gate assignment problem for handling of uneven gate demands,” in *Proceedings of the World Conference of Transport Research (WCTR ’10)*, Lisbon, Portugal, 2010.

[43] B. G. Thengvall, J. F. Bard, and G. Yu, “A bundle algorithm approach for the aircraft schedule recovery problem during hub closures,” *Transportation Science*, vol. 37, no. 4, pp. 392-407, 2003.

[44] Y. Gu, and C. A. Chung, “Genetic algorithm approach to aircraft gate reassignment problem,” *Journal of Transportation Engineering*, vol. 125, no. 5, pp. 384-389, 1999.

[45] A. Bouras, M. A. Ghaleb, U. S. Suryahatmaja, and A. M. Salem, “The airport Gate Assignment Problem (AGAP) as parallel machine scheduling with eligibility: metaheuristic approaches,” Industrial Engineering Technical Report KSUENG-IED-TR-012014-01, Department of Industrial Engineering, King Saud University, Riyadh, Saudi Arabia, 2014.

[46] E. K. Burke, M. Gendreau, M. Hyde et al., “Hyper-heuristics: a survey of the state of the art,” *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695-1724, 2013.

[47] E. Soubeiga, *Development and Application of Hyperheuristics to Personnel Scheduling*, University of Nottingham, 2003.

[48] E. K. Burke, M. Hyde, G. Kendall, G. Ochoa, E. Ozcan, and R. Qu, “A survey of hyper-heuristics,” Tech. Rep. NOTTCS-TR-SUB-0906241418-2747, School of Computer Science and Information Technology, University of Nottingham, 2009.

[49] Hon, W. C., 1996. Scheduling as a multi-dimensional placement problem. Engineering Applications of Artificial Intelligence 9 (3), 261-273.

[50] Duin, C. W., Sluis, E. V. D., 2006. On the complexity of adjacent resource scheduling. Journal of Scheduling 9 (1), 49-62.

[51] Yan, S., Tang, C.-H., Chen, M., 2004. A model and a solution algorithm for airport common use check-in counter assignments. Transportation Research, Part A (Policy and Practice) 38A (2), 101-125.

[52] Yan, S., Chang, K. C. K.-C., Tang, C. H. C.-H., 2005. Minimizing inconsistencies in airport common-use checking counter assignments with a variable number of counters. Journal of Air Transport Management 11 (2), 107-116.

[53] Wang Yeung, B. K., Chun H. W., 1995. Check-in counter allocation using genetic algorithm. EXPERSYS, 197-202.

[54] van Dijk, N. M., van der Sluis, E., 2006. Check-in computation and optimization by simulation and ip in combination. European Journal of Operational Research 171 (3), 1152-1168.

[55] Tang, C.-H., 2010. A network model for airport common use check-in counter assignments . Journal of the Operational Research Society 61 (11), 1607-1618.

[56] Parlar, M., Sharafali, M., 2008. Dynamic allocation of airline check-in counters: A queueing optimization approach. Management Science 54 (8), 1410-1424.

[57] Park, Y., Ahn, S. B., 2003. Optimal assignment for check-in counters based on passenger arrival behavior at an airport. Transportation Planning and Technology 26 (5), 397-416.

[58] Chun, H. W., Mak, 1999. Intelligent resource simulation for an airport check-in counter allocation system. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 29 (3), 325-335.

[59] Ozturk, C.; Süğüt, I. and Ornek, Arslan M., An Optimization Approach To Assignment Of Check-in Counters To Flights in Airports, 27th European Conference on Operational Research, Uni. of Strathclyde, Glasgow, Uk, (2015)

[60] Ozturk, C.; Süğüt, I. and Ornek, Arslan M., Integer and Constraint Programming Model Formulations For Over Constrained Flight-Gate Assignment Problem, CIE45 Proceedings, University of Lorraine, Metz, France, (2015)

APPENDICES

Appendix A: The numerical Instance for Developed Models of GAP

	Arrival	Departure	Parking Area: Bridge-equipped/Open	Has priority?	Weight
PARKING AREAS (DOMESTIC)	D	D	Bridge-equipped	Yes	10
	I	D	Bridge-equipped	Yes	9
	D	D	Bridge-equipped	No	8
	I	D	Bridge-equipped	No	7
	D	I	Bridge-equipped		6
	I	I	Bridge-equipped		5
	D	D	Open		4
	I	D	Open		3
	D	I	Open		2
	I	I	Open		1

Table 9 The Weight Logic for Domestic Gates

	Arrival	Departure	Parking Area: Bridge-equipped/Open	Has priority?	Weight
PARKING AREAS (INTERNATIONAL)	I	I	Bridge-equipped	Yes	10
	D	I	Bridge-equipped	Yes	9
	I	I	Bridge-equipped	No	8
	D	I	Bridge-equipped	No	7
	I	D	Bridge-equipped		6
	D	D	Bridge-equipped		5
	I	I	Open		4
	D	I	Open		3
	I	D	Open		2
	D	D	Open		1

Table 10 The Weight Logic for International Gates

Flight No	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
1	1	1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	0	0	8	8	8	8	8	10	10	10	10	5	5	5	5	5	5	5	5
2	1	1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	0	0	8	8	8	8	8	10	10	10	10	6	6	6	5	5	5	5	5
3	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
4	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
5	1	1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	0	0	8	8	8	8	8	10	10	10	10	5	5	5	5	5	5	5	5
6	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
7	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
8	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	9	9	9	7	7	7	7	7
9	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
10	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
11	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
12	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
13	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
14	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
15	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
16	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
17	1	1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	0	0	8	8	8	8	8	10	10	10	10	5	5	5	5	5	5	5	5
18	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
19	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
20	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
21	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
22	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
23	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100
24	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
25	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
26	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
27	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
28	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
29	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
30	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
31	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	9	9	9	7	7	7	7	7
32	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
33	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	7	7	7	6	6	6	6	6
34	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
35	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	7	7	7	6	6	6	6	6
36	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	9	9	9	7	7	7	7	7
37	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
38	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
39	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	9	9	9	7	7	7	7	7
40	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
41	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	10	10	10	8	8	8	8	8
42	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
43	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
44	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
45	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10

46	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
47	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
48	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
49	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
50	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
51	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
52	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
53	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
54	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	9	9	9	9
55	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	9	9	9	9
56	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
57	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	9	9	9	9
58	1	1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	0	0	8	8	8	8	8	10	10	10	10	5	5	5	5	6	6	6	6
59	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
60	1	1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	0	0	8	8	8	8	8	10	10	10	10	5	5	5	5	5	5	5	5
61	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
62	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
63	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
64	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
65	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
66	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
67	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
68	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
69	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
70	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
71	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
72	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
73	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
74	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
75	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
76	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	10	10	10	10
77	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
78	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
79	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
80	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
81	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
82	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
83	1	1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	0	0	8	8	8	8	8	10	10	10	10	5	5	5	5	5	5	5	5
84	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
85	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
86	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
87	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
88	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
89	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
90	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6

91	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
92	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
93	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
94	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
95	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
96	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
97	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
98	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
99	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
100	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
101	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
102	3	3	3	3	3	3	2	2	2	2	2	2	2	2	2	2	0	0	6	6	6	6	6	6	6	6	6	7	7	7	7	7	7	7	7
103	4	4	4	4	4	4	1	1	1	1	1	1	1	1	1	1	0	0	5	5	5	5	5	5	5	5	5	8	8	8	8	8	8	8	8
104	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	0	0	7	7	7	7	7	9	9	9	9	6	6	6	6	6	6	6	6
105	1	1	1	1	1	1	4	4	4	4	4	4	4	4	4	4	0	0	8	8	8	8	8	10	10	10	10	5	5	5	5	5	5	5	5

Weight of Gate	1	1	2	2	3	3	3	3	3	2	2	2	2	1	1	1	0	0	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Table 11 The weight values for each flight at each gate and the weights for only gate

Neighborhood Matrix																																			
	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Table 12 The neighbourhood matrix for gates

FLIGHT NO	LANDING PERIOD	ORIGINAL DEPARTURE PERIOD	DEPARTURE PERIOD WITH EXTRA 15 MIN	GROUND SERVICE NO
1	240	288	291	1
2	153	163	166	3
3	177	187	190	2
4	147	159	162	2
5	76	86	89	2
6	158	170	173	2
7	64	69	72	2
8	110	115	118	3
9	148	153	156	3
10	170	175	178	2
11	68	106	109	1
12	24	29	32	2
13	206	211	214	2
14	187	192	195	3
15	36	72	75	1
16	1	18	21	1
17	155	168	171	3
18	203	288	291	2
19	83	95	98	1
20	88	93	96	2
21	107	158	161	2
22	180	240	243	2
23	90	132	135	4
24	57	69	72	2
25	103	110	113	2
26	145	152	155	2
27	226	288	291	2
28	49	54	57	2
29	235	288	291	2
30	52	57	60	2
31	109	117	120	3
32	161	166	169	2
33	170	203	206	3
34	210	288	291	2
35	56	68	71	2
36	99	288	291	2
37	44	49	52	2
38	103	111	114	2
39	146	153	156	2
40	186	203	206	3
41	238	288	291	2
42	31	42	45	3
43	48	57	60	3
44	193	288	291	1
45	162	172	175	3
46	55	66	69	3
47	70	79	82	3
48	103	114	117	3
49	205	288	291	3
50	211	288	291	3
51	139	150	153	3
52	171	180	183	3
53	31	40	43	3
54	127	136	139	1
55	115	126	129	1
56	43	54	57	3
57	175	186	189	3
58	244	288	291	3
59	67	78	81	3
60	103	162	165	3

61	253	288	291	3
62	199	207	210	3
63	79	90	93	3
64	187	198	201	3
65	27	36	39	1
66	194	204	207	3
67	130	140	143	3
68	118	134	137	3
69	72	82	85	1
70	141	147	150	2
71	35	41	44	3
72	52	58	61	3
73	97	103	106	3
74	232	288	291	3
75	119	125	128	3
76	186	192	195	2
77	109	118	121	2
78	209	288	291	3
79	55	61	64	3
80	121	136	139	3
81	219	288	291	3
82	228	288	291	3
83	9	30	33	3
84	114	123	126	3
85	51	74	77	3
86	158	176	179	3
87	222	288	291	3
88	60	85	88	3
89	154	175	178	1
90	212	219	222	3
91	60	72	75	3
92	163	176	179	3
93	230	288	291	3
94	60	72	75	3
95	151	167	170	1
96	199	209	212	1
97	59	75	78	3
98	157	177	180	2
99	210	288	291	3
100	7	30	33	3
101	62	73	76	3
102	161	178	181	3
103	220	288	291	3
104	129	138	141	1
105	47	62	65	3

Table 13 The arrival, original departure and departure with extra periods of the flights and the ground service data of flights

FLIGHT NO	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
23	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
25	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
26	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
27	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
28	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
29	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
30	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
32	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
33	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
34	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
35	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
36	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
37	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
38	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
39	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
40	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
41	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
42	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
43	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
44	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
45	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

46	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
47	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
48	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
49	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
50	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
51	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
52	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
53	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
54	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
55	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
56	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
57	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
58	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
59	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
60	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
61	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
62	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
63	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
64	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
65	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
66	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
67	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
68	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
69	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
70	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
71	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
72	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
73	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
74	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
75	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
76	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
77	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
78	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
79	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
80	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
81	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
82	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
83	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
84	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
85	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
86	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
87	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
88	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
89	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
90	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

91	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
92	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
93	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
94	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
95	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
96	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
97	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
98	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
99	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
100	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
101	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
102	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
103	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
104	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
105	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	0	1	0	0	1	1	1	1	1	1	1	1	1	1

Table 14 The eligibility matrix for flights to the gates

GROUND SERVICES	NO
Çelebi	1
Havaş	2
TGS	3
Other	4

NIGHT-STAND FLIGHTS
10

Table 15 Ground Location Service and Night-Stand Flight Data

GATE BUSY FROM THE PREVIOUS DAY	EARLIEST AVAILABLE PERIOD OF THE GATE
4	3
34	6
31	11
1	12
2	12
33	14
32	14
30	17
9	17
29	18
35	18
24	18
5	18
20	19
28	20
23	20
21	27
6	30
22	46

Table 16 The busy gates from the previous day and earliest available period of busy gates

BRIDGE-EQUIPPED GATES	OPEN-PARK GATES
17	1
18	2
19	3
20	4
21	5
22	6
23	7
24	8
25	9
26	10
27	11
28	12
29	13
30	14
31	15
32	16
33	
34	
35	

Table 17 The bridge-equipped and open-park gates

Appendix B: The Results of Timetabling Based Integer Programming Model

PLANE	GATE	ARRIVAL_PERIOD	DEPARTURE_PERIOD
1	26	240	291
2	24	153	166
3	25	177	190
4	35	147	162
5	25	76	89
6	29	158	173
7	29	64	72
8	28	110	118
9	30	148	156
10	5	170	178
11	28	68	109
12	29	24	32
13	28	206	214
14	29	187	195
15	35	36	75
16	19	1	21
17	25	155	171
18	9	203	291
19	35	83	98
20	30	88	96
21	27	107	161
22	35	180	243
23	5	90	135
24	30	57	72
25	30	103	113
26	28	145	155
27	29	226	291
28	26	49	57
29	30	235	291
30	29	52	60
31	29	109	120
32	30	161	169
33	24	170	206
34	6	210	291
35	22	56	71
36	7	99	291
37	28	44	52
38	31	103	114
39	29	146	156
40	28	186	205
41	28	238	291
42	32	31	45
43	33	48	60
44	27	193	291
45	34	162	175
46	32	55	69
47	32	70	82
48	35	103	117
49	32	205	291
50	34	211	291

51	34	139	153
52	33	171	183
53	33	31	43
54	35	127	139
55	32	115	129
56	25	43	57
57	34	176	189
58	25	244	291
59	33	67	81
60	26	103	165
61	35	253	291
62	34	199	210
63	34	79	93
64	33	187	201
65	28	27	39
66	30	194	207
67	29	130	143
68	30	118	137
69	29	73	85
70	33	141	150
71	34	35	44
72	34	52	61
73	33	97	106
74	33	232	291
75	34	119	128
76	32	186	195
77	24	109	121
78	21	209	291
79	28	55	64
80	25	121	139
81	23	219	291
82	22	228	291
83	25	9	33
84	31	115	126
85	24	51	77
86	20	158	179
87	5	222	291
88	21	60	88
89	32	154	178
90	26	212	222
91	27	60	75
92	35	163	179
93	4	230	291
94	25	60	75
95	33	151	170
96	26	199	211
97	20	59	78
98	31	157	180
99	24	210	291
100	34	7	33
101	26	62	76
102	28	161	181
103	20	220	291
104	24	129	141
105	19	47	65

Table 18 The results of assignments of flights to the gates for timetabling based integer programming model

GATE1

NA

GATE2

NA

GATE3

NA

GATE4

93	4	230	291	S3

GATE5

23	5	90	135	S4
10	5	170	178	S2
87	5	222	291	S3

GATE6

34	6	210	291	S2

GATE7

36	7	99	291	S2

GATE8

NA

GATE9

18	9	203	291	S2

GATE10

NA

GATE11

NA

GATE12

NA

GATE13

NA

GATE14

NA

GATE15

NA

GATE16

NA

PERDE

GATE17

NA

GATE18

NA

GATE19

16	19	1	21	S1
105	19	47	65	S3

GATE20

97	20	59	78	S3
86	20	158	179	S3
103	20	220	291	S3

GATE21

88	21	60	88	S3
78	21	209	291	S3

GATE22

35	22	56	71	S2
82	22	228	291	S3

GATE23

81	23	219	291	S3

GATE24

85	24	51	77	S3
77	24	109	121	S2
104	24	129	141	S1
2	24	153	166	S3
33	24	170	206	S3
99	24	210	291	S3

GATE25

83	25	9	33	S3
56	25	43	57	S3
94	25	60	75	S3
5	25	76	89	S2
80	25	121	139	S3
17	25	155	171	S3
3	25	177	190	S2
58	25	244	291	S3

GATE26

28	26	49	57	S2
101	26	62	76	S3
60	26	103	165	S3
96	26	199	211	S1
90	26	212	222	S3
1	26	240	291	S1

GATE27

91	27	60	75	S3
21	27	107	161	S2
44	27	193	291	S1

GATE28

65	28	27	39	S1
37	28	44	52	S2
79	28	55	64	S3
11	28	68	109	S1
8	28	110	118	S3
26	28	145	155	S2
102	28	161	181	S3
40	28	186	205	S3
13	28	206	214	S2
41	28	238	291	S2

GATE29

12	29	24	32	S2
30	29	52	60	S2
7	29	64	72	S2
69	29	73	85	S1
31	29	109	120	S3
67	29	130	143	S3
39	29	146	156	S2
6	29	158	173	S2
14	29	187	195	S3
27	29	226	291	S2

GATE30

24	30	57	72
20	30	88	96
25	30	103	113
68	30	118	137
9	30	148	156
32	30	161	169
66	30	194	207
29	30	235	291

GATE31

38	31	103	114	S2
84	31	115	126	S3
98	31	157	180	S2

GATE32

42	32	31	45	S3
46	32	55	69	S3
47	32	70	82	S3
55	32	115	129	S1
89	32	154	178	S1
76	32	186	195	S2
49	32	205	291	S3

GATE33

53	33	31	43	S3
43	33	48	60	S3
59	33	67	81	S3
73	33	97	106	S3
70	33	141	150	S2
95	33	151	170	S1
52	33	171	183	S3
64	33	187	201	S3
74	33	232	291	S3

GATE34

100	34	7	33	S3
71	34	35	44	S3
72	34	52	61	S3
63	34	79	93	S3
75	34	119	128	S3
51	34	139	153	S3
45	34	162	175	S3
57	34	176	189	S3
62	34	199	210	S3
50	34	211	291	S3

GATE35

15	35	36	75	S1
19	35	83	98	S1
48	35	103	117	S3
54	35	127	139	S1
4	35	147	162	S2
92	35	163	179	S3
22	35	180	243	S2
61	35	253	291	S3

Table 19 The assignment of flights on each gate for timetabling based integer programming model

Appendix C: The Results of Assignment Based Integer Programming Model

PLANE	GATE	ARRIVAL_PERIOD	DEPARTURE_PERIOD
1	19	240	291
2	25	153	166
3	25	177	190
4	35	147	162
5	27	76	89
6	30	158	173
7	28	64	72
8	32	110	118
9	31	148	156
10	6	170	178
11	31	68	109
12	30	24	32
13	30	206	214
14	28	187	195
15	6	36	75
16	19	1	21
17	24	155	171
18	8	203	291
19	30	83	98
20	28	88	96
21	22	107	161
22	31	180	243
23	7	90	135
24	29	57	72
25	29	103	113
26	29	145	155
27	1	226	291
28	26	49	57
29	30	235	291
30	28	52	60
31	31	109	120
32	28	161	169
33	27	170	206
34	28	210	291
35	27	56	71
36	9	99	291
37	29	44	52
38	28	103	114
39	28	146	156
40	29	186	205
41	29	238	291
42	35	31	45
43	34	48	60
44	20	193	291
45	34	162	175
46	33	55	69
47	35	70	82
48	33	103	117
49	35	205	291
50	34	211	291

51	32	139	153
52	33	171	183
53	33	31	43
54	34	127	139
55	35	115	129
56	24	43	57
57	34	176	189
58	24	244	291
59	34	67	81
60	26	103	165
61	32	253	291
62	33	199	210
63	33	79	93
64	35	187	201
65	28	27	39
66	34	194	207
67	31	130	143
68	30	118	137
69	29	73	85
70	34	141	150
71	34	35	44
72	35	52	61
73	35	97	106
74	33	232	291
75	32	119	128
76	32	186	195
77	24	109	121
78	22	209	291
79	31	55	64
80	27	121	139
81	23	219	291
82	25	228	291
83	25	9	33
84	29	115	126
85	23	51	77
86	32	158	179
87	5	222	291
88	20	60	88
89	19	154	178
90	24	212	222
91	21	60	75
92	35	163	179
93	3	230	291
94	22	60	75
95	20	151	170
96	24	199	211
97	25	59	78
98	29	157	180
99	26	210	291
100	34	7	33
101	24	62	76
102	23	161	181
103	6	220	291
104	24	129	141
105	19	47	65

Table 20 The results of assignments of flights to the gates for assignment based integer programming model

GATE1

27	1	226	290	S2

GATE2

NA

GATE3

93	3	230	290	S3

GATE4

NA

GATE5

87	5	222	290	S3

GATE6

15	6	36	74	S1
10	6	170	177	S2
103	6	220	290	S3

GATE7

23	8	90	134	S4

GATE8

18	8	203	290	S2

GATE9

36	9	99	290	S2

GATE10

NA

GATE11

NA

GATE12

NA

GATE13

NA

GATE14

NA

GATE15

NA

GATE16

NA

PERDE

GATE17

NA

GATE18

NA

GATE19

16	19	1	20	S1
105	19	47	64	S3
89	19	154	177	S1
1	19	240	290	S1

GATE20

88	21	60	87	S3
95	20	151	169	S1
44	20	193	290	S1

GATE21

91	21	60	74	S3

GATE22

94	22	60	74	S3
21	22	107	160	S2
78	22	209	290	S3

GATE23

85	23	51	76	S3
102	23	161	180	S3
81	23	219	290	S3

GATE24

56	24	43	56	S3
101	24	62	75	S3
77	24	109	120	S2
104	24	129	140	S1
17	24	155	170	S3
96	24	199	211	S1
90	24	212	221	S3
58	24	244	290	S3

GATE25

83	25	9	32	S3
97	25	59	77	S3
2	25	153	165	S3
3	25	177	189	S2
82	25	228	290	S3

GATE26

28	26	49	56	S2
60	26	103	164	S3
99	26	210	290	S3

GATE27

35	27	56	70	S2
5	27	76	88	S2
80	27	121	138	S3
33	27	170	205	S3

GATE28

65	28	27	38	S1
30	28	52	59	S2
7	28	64	71	S2
20	28	88	95	S2
38	28	103	113	S2
39	28	146	155	S2
32	28	161	168	S2
14	28	187	194	S3
34	28	210	290	S2

GATE29

37	29	44	51	S2
24	29	57	71	S2
69	29	72	84	S1
25	29	103	112	S2
84	29	114	125	S3
26	29	145	154	S2
98	29	157	179	S2
40	29	186	205	S3
41	29	238	290	S2

GATE30

12	30	24	31	S2
19	30	83	97	S1
68	30	118	136	S3
6	30	158	172	S2
13	30	206	213	S2
29	30	235	290	S2

GATE31

79	31	55	63	S3
11	31	68	108	S1
31	31	109	119	S3
67	31	130	142	S3
9	31	148	155	S3
22	31	180	242	S2

GATE32

8	32	110	117	S3
75	32	119	127	S3
51	32	139	152	S3
86	32	158	178	S3
76	32	186	194	S2
61	32	253	290	S3

GATE33

53	33	31	42	S3
46	33	55	68	S3
63	33	79	92	S3
48	33	103	116	S3
52	33	171	182	S3
62	33	199	209	S3
74	33	232	290	S3

GATE34

100	34	7	32	S3
71	34	35	43	S3
43	34	48	59	S3
59	34	67	80	S3
54	34	127	138	S1
70	34	141	149	S2
45	34	162	174	S3
57	34	175	188	S3
66	34	194	206	S3
50	34	211	290	S3

GATE35

42	35	31	44	S3
72	35	52	60	S3
47	35	70	81	S3
73	35	97	105	S3
55	35	115	128	S1
4	35	147	161	S2
92	35	163	178	S3
64	35	187	200	S3
49	35	205	290	S3

Table 21 The assignment of flights on each gate for assignment based integer programming model

Appendix D: The Numerical Instance for Developed Model of CAP

SP1 (FIRST 2 HOURS)			
START_TIME	CLOSED_TIME	# OF COUNTERS	FIRM_NO
0	24	2	1
1	25	2	2
2	26	3	3
3	27	2	2
3	27	3	4
5	29	3	4
8	32	3	4
8	32	2	5
8	32	3	6
8	32	2	5
10	34	3	7
10	34	3	8
10	34	3	4
10	34	3	4
10	34	3	6
11	35	4	9
11	35	4	9
11	35	3	10
11	35	3	8
12	36	3	11
13	37	4	9
13	37	3	4
13	37	2	2
14	38	3	12
15	39	4	9
15	39	2	13
15	39	2	13
16	40	2	5
17	41	4	9
17	41	3	10
18	42	3	11
18	42	3	10
19	43	3	7
19	43	3	8
20	44	3	12
22	46	2	2
22	46	3	11
22	46	3	8
23	47	2	14
23	47	3	6
24	48	3	15

SP2 (NEXT 2 HOURS)			
START_TIME	CLOSED_TIME	# OF COUNTERS	FIRM_NO
25	49	3	6
25	49	2	16
26	50	4	9
26	50	2	2
27	51	3	17
27	51	3	14
27	51	2	18
28	52	2	19
28	52	3	7
28	52	3	17
29	53	2	18
29	53	2	5
30	54	2	2
30	54	2	18
31	55	4	9
31	55	3	7
32	56	2	20
33	57	3	21
33	57	3	21
33	57	3	22
35	59	3	21
35	59	2	23
35	59	3	10
35	59	2	23
37	61	3	15
38	62	2	20
38	62	3	7
38	62	3	3
39	63	5	24
40	64	4	9
40	64	3	25
40	64	3	3
41	65	3	8
41	65	3	26
41	65	5	24
41	65	2	20
41	65	3	4
43	67	4	9
43	67	8	14
43	67	2	14
44	68	4	9
45	69	4	9
46	70	8	14
47	71	3	3
47	71	2	2
48	72	2	20
48	72	4	9

SP3 (NEXT 2 HOURS)			
START_TIME	CLOSED_TIME	# OF COUNTERS	FIRM_NO
49	73	2	27
50	74	2	5
50	74	8	14
50	74	3	21
51	75	2	28
52	76	2	28
52	76	2	28
52	76	2	29
52	76	2	20
53	77	2	5
53	77	3	6
53	77	2	30
55	79	3	12
55	79	3	31
55	79	2	32
56	80	2	30
57	81	3	33
57	81	2	20
58	82	2	28
59	83	3	10
59	83	3	34
61	85	3	8
62	86	2	35
62	86	3	17
62	86	3	22
63	87	5	36
63	87	3	15
63	87	3	22
64	88	4	37
64	88	2	28
65	89	2	38
65	89	1	39
66	90	2	16
66	90	2	40
67	91	3	41
67	91	4	9
68	92	4	37
68	92	2	20
69	93	2	42
69	93	3	43
70	94	3	22
70	94	2	38
70	94	4	9
71	95	4	37
71	95	3	33
71	95	8	14
72	96	3	44

SP4 (LAST 2 HOURS)			
START_TIME	CLOSED_TIME	# OF COUNTERS	FIRM_NO
73	97	3	22
73	97	4	9
73	97	2	45
73	97	2	20
75	99	2	46
75	99	2	47
75	99	3	22
76	100	4	37
77	101	3	37
77	101	4	37
77	101	2	47
77	101	2	48
79	103	2	48
79	103	3	8
80	104	2	42
80	104	3	11
80	104	2	49
81	105	2	50
81	105	2	51
82	106	2	35
83	107	2	1
83	107	3	52
83	107	3	7

Table 22 The arrival, departure, and the number of counters needed of the flights and the firm data of flights

L	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0
64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0
66	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0
67	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0
68	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Table 23 The neighbourhood matrix for counters (The full matrix is 135R x 135C)

Appendix E: The results for Developed Model of CAP

FLIGHT NO	RESULTS	FIRMS
1	FLIGHT 1 (0,24) WITH FIRM 1 TO GATE 11 is 1	**F1-FIRM1**
	FLIGHT 1 (0,24) WITH FIRM 1 TO GATE 12 is 1	**F1-FIRM1**
2	FLIGHT 2 (1,25) WITH FIRM 2 TO GATE 9 is 1	**F2-FIRM2**
	FLIGHT 2 (1,25) WITH FIRM 2 TO GATE 10 is 1	**F2-FIRM2**
3	FLIGHT 3 (2,26) WITH FIRM 3 TO GATE 1 is 1	**F3-FIRM3**
	FLIGHT 3 (2,26) WITH FIRM 3 TO GATE 2 is 1	**F3-FIRM3**
	FLIGHT 3 (2,26) WITH FIRM 3 TO GATE 3 is 1	**F3-FIRM3**
4	FLIGHT 4 (3,27) WITH FIRM 2 TO GATE 7 is 1	**F4-FIRM2**
	FLIGHT 4 (3,27) WITH FIRM 2 TO GATE 8 is 1	**F4-FIRM2**
5	FLIGHT 5 (3,27) WITH FIRM 4 TO GATE 4 is 1	**F5-FIRM4**
	FLIGHT 5 (3,27) WITH FIRM 4 TO GATE 5 is 1	**F5-FIRM4**
	FLIGHT 5 (3,27) WITH FIRM 4 TO GATE 6 is 1	**F5-FIRM4**
6	FLIGHT 1 (5,29) WITH FIRM 4 TO GATE 20 is 1	**F6-FIRM4**
	FLIGHT 1 (5,29) WITH FIRM 4 TO GATE 21 is 1	**F6-FIRM4**
	FLIGHT 1 (5,29) WITH FIRM 4 TO GATE 22 is 1	**F6-FIRM4**
7	FLIGHT 2 (8,32) WITH FIRM 4 TO GATE 23 is 1	**F7-FIRM4**
	FLIGHT 2 (8,32) WITH FIRM 4 TO GATE 24 is 1	**F7-FIRM4**
	FLIGHT 2 (8,32) WITH FIRM 4 TO GATE 25 is 1	**F7-FIRM4**
8	FLIGHT 3 (8,32) WITH FIRM 5 TO GATE 18 is 1	**F8-FIRM5**
	FLIGHT 3 (8,32) WITH FIRM 5 TO GATE 19 is 1	**F8-FIRM5**
9	FLIGHT 4 (8,32) WITH FIRM 6 TO GATE 13 is 1	**F9-FIRM6**
	FLIGHT 4 (8,32) WITH FIRM 6 TO GATE 14 is 1	**F9-FIRM6**
	FLIGHT 4 (8,32) WITH FIRM 6 TO GATE 15 is 1	**F9-FIRM6**
10	FLIGHT 5 (8,32) WITH FIRM 5 TO GATE 16 is 1	**F10-FIRM5**
	FLIGHT 5 (8,32) WITH FIRM 5 TO GATE 17 is 1	**F10-FIRM5**
11	FLIGHT 1 (10,34) WITH FIRM 7 TO GATE 35 is 1	**F11-FIRM7**
	FLIGHT 1 (10,34) WITH FIRM 7 TO GATE 36 is 1	**F11-FIRM7**
	FLIGHT 1 (10,34) WITH FIRM 7 TO GATE 37 is 1	**F11-FIRM7**
12	FLIGHT 2 (10,34) WITH FIRM 8 TO GATE 38 is 1	**F12-FIRM8**
	FLIGHT 2 (10,34) WITH FIRM 8 TO GATE 39 is 1	**F12-FIRM8**
	FLIGHT 2 (10,34) WITH FIRM 8 TO GATE 40 is 1	**F12-FIRM8**
13	FLIGHT 3 (10,34) WITH FIRM 4 TO GATE 26 is 1	**F13-FIRM4**
	FLIGHT 3 (10,34) WITH FIRM 4 TO GATE 27 is 1	**F13-FIRM4**
	FLIGHT 3 (10,34) WITH FIRM 4 TO GATE 28 is 1	**F13-FIRM4**
14	FLIGHT 4 (10,34) WITH FIRM 4 TO GATE 29 is 1	**F14-FIRM4**
	FLIGHT 4 (10,34) WITH FIRM 4 TO GATE 30 is 1	**F14-FIRM4**
	FLIGHT 4 (10,34) WITH FIRM 4 TO GATE 31 is 1	**F14-FIRM4**
15	FLIGHT 5 (10,34) WITH FIRM 6 TO GATE 32 is 1	**F15-FIRM6**
	FLIGHT 5 (10,34) WITH FIRM 6 TO GATE 33 is 1	**F15-FIRM6**
	FLIGHT 5 (10,34) WITH FIRM 6 TO GATE 34 is 1	**F15-FIRM6**
16	FLIGHT 1 (11,35) WITH FIRM 9 TO GATE 47 is 1	**F16-FIRM9**
	FLIGHT 1 (11,35) WITH FIRM 9 TO GATE 48 is 1	**F16-FIRM9**
	FLIGHT 1 (11,35) WITH FIRM 9 TO GATE 49 is 1	**F16-FIRM9**
	FLIGHT 1 (11,35) WITH FIRM 9 TO GATE 50 is 1	**F16-FIRM9**
17	FLIGHT 2 (11,35) WITH FIRM 9 TO GATE 51 is 1	**F17-FIRM9**
	FLIGHT 2 (11,35) WITH FIRM 9 TO GATE 52 is 1	**F17-FIRM9**
	FLIGHT 2 (11,35) WITH FIRM 9 TO GATE 53 is 1	**F17-FIRM9**
	FLIGHT 2 (11,35) WITH FIRM 9 TO GATE 54 is 1	**F17-FIRM9**

18	FLIGHT 3 (11,35) WITH FIRM 10 TO GATE 41 is 1	**F18-FIRM10**
	FLIGHT 3 (11,35) WITH FIRM 10 TO GATE 42 is 1	**F18-FIRM10**
	FLIGHT 3 (11,35) WITH FIRM 10 TO GATE 43 is 1	**F18-FIRM10**
19	FLIGHT 4 (11,35) WITH FIRM 8 TO GATE 55 is 1	**F19-FIRM8**
	FLIGHT 4 (11,35) WITH FIRM 8 TO GATE 56 is 1	**F19-FIRM8**
	FLIGHT 4 (11,35) WITH FIRM 8 TO GATE 57 is 1	**F19-FIRM8**
20	FLIGHT 5 (12,36) WITH FIRM 11 TO GATE 44 is 1	**F20-FIRM11**
	FLIGHT 5 (12,36) WITH FIRM 11 TO GATE 45 is 1	**F20-FIRM11**
	FLIGHT 5 (12,36) WITH FIRM 11 TO GATE 46 is 1	**F20-FIRM11**
21	FLIGHT 1 (13,37) WITH FIRM 9 TO GATE 61 is 1	**F21-FIRM9**
	FLIGHT 1 (13,37) WITH FIRM 9 TO GATE 62 is 1	**F21-FIRM9**
	FLIGHT 1 (13,37) WITH FIRM 9 TO GATE 63 is 1	**F21-FIRM9**
	FLIGHT 1 (13,37) WITH FIRM 9 TO GATE 64 is 1	**F21-FIRM9**
22	FLIGHT 2 (13,37) WITH FIRM 4 TO GATE 58 is 1	**F22-FIRM4**
	FLIGHT 2 (13,37) WITH FIRM 4 TO GATE 59 is 1	**F22-FIRM4**
	FLIGHT 2 (13,37) WITH FIRM 4 TO GATE 60 is 1	**F22-FIRM4**
23	FLIGHT 3 (13,37) WITH FIRM 2 TO GATE 69 is 1	**F23-FIRM2**
	FLIGHT 3 (13,37) WITH FIRM 2 TO GATE 70 is 1	**F23-FIRM2**
24	FLIGHT 4 (14,38) WITH FIRM 12 TO GATE 71 is 1	**F24-FIRM12**
	FLIGHT 4 (14,38) WITH FIRM 12 TO GATE 72 is 1	**F24-FIRM12**
	FLIGHT 4 (14,38) WITH FIRM 12 TO GATE 73 is 1	**F24-FIRM12**
25	FLIGHT 5 (15,39) WITH FIRM 9 TO GATE 65 is 1	**F25-FIRM9**
	FLIGHT 5 (15,39) WITH FIRM 9 TO GATE 66 is 1	**F25-FIRM9**
	FLIGHT 5 (15,39) WITH FIRM 9 TO GATE 67 is 1	**F25-FIRM9**
	FLIGHT 5 (15,39) WITH FIRM 9 TO GATE 68 is 1	**F25-FIRM9**
26	FLIGHT 1 (15,39) WITH FIRM 13 TO GATE 85 is 1	**F26-FIRM13**
	FLIGHT 1 (15,39) WITH FIRM 13 TO GATE 86 is 1	**F26-FIRM13**
27	FLIGHT 2 (15,39) WITH FIRM 13 TO GATE 83 is 1	**F27-FIRM13**
	FLIGHT 2 (15,39) WITH FIRM 13 TO GATE 84 is 1	**F27-FIRM13**
28	FLIGHT 3 (16,40) WITH FIRM 5 TO GATE 74 is 1	**F28-FIRM5**
	FLIGHT 3 (16,40) WITH FIRM 5 TO GATE 75 is 1	**F28-FIRM5**
29	FLIGHT 4 (17,41) WITH FIRM 9 TO GATE 76 is 1	**F29-FIRM9**
	FLIGHT 4 (17,41) WITH FIRM 9 TO GATE 77 is 1	**F29-FIRM9**
	FLIGHT 4 (17,41) WITH FIRM 9 TO GATE 78 is 1	**F29-FIRM9**
	FLIGHT 4 (17,41) WITH FIRM 9 TO GATE 79 is 1	**F29-FIRM9**
30	FLIGHT 5 (17,41) WITH FIRM 10 TO GATE 80 is 1	**F30-FIRM10**
	FLIGHT 5 (17,41) WITH FIRM 10 TO GATE 81 is 1	**F30-FIRM10**
	FLIGHT 5 (17,41) WITH FIRM 10 TO GATE 82 is 1	**F30-FIRM10**
31	FLIGHT 1 (18,42) WITH FIRM 11 TO GATE 90 is 1	**F31-FIRM11**
	FLIGHT 1 (18,42) WITH FIRM 11 TO GATE 91 is 1	**F31-FIRM11**
	FLIGHT 1 (18,42) WITH FIRM 11 TO GATE 92 is 1	**F31-FIRM11**
32	FLIGHT 2 (18,42) WITH FIRM 10 TO GATE 87 is 1	**F32-FIRM10**
	FLIGHT 2 (18,42) WITH FIRM 10 TO GATE 88 is 1	**F32-FIRM10**
	FLIGHT 2 (18,42) WITH FIRM 10 TO GATE 89 is 1	**F32-FIRM10**
33	FLIGHT 3 (19,43) WITH FIRM 7 TO GATE 99 is 1	**F33-FIRM7**
	FLIGHT 3 (19,43) WITH FIRM 7 TO GATE 100 is 1	**F33-FIRM7**
	FLIGHT 3 (19,43) WITH FIRM 7 TO GATE 101 is 1	**F33-FIRM7**
34	FLIGHT 4 (19,43) WITH FIRM 8 TO GATE 96 is 1	**F34-FIRM8**
	FLIGHT 4 (19,43) WITH FIRM 8 TO GATE 97 is 1	**F34-FIRM8**
	FLIGHT 4 (19,43) WITH FIRM 8 TO GATE 98 is 1	**F34-FIRM8**
35	FLIGHT 5 (20,44) WITH FIRM 12 TO GATE 93 is 1	**F35-FIRM12**
	FLIGHT 5 (20,44) WITH FIRM 12 TO GATE 94 is 1	**F35-FIRM12**
	FLIGHT 5 (20,44) WITH FIRM 12 TO GATE 95 is 1	**F35-FIRM12**

36	FLIGHT 1 (22,46) WITH FIRM 2 TO GATE 111 is 1	**F36-FIRM2**
	FLIGHT 1 (22,46) WITH FIRM 2 TO GATE 112 is 1	**F36-FIRM2**
37	FLIGHT 2 (22,46) WITH FIRM 11 TO GATE 105 is 1	**F37-FIRM11**
	FLIGHT 2 (22,46) WITH FIRM 11 TO GATE 106 is 1	**F37-FIRM11**
	FLIGHT 2 (22,46) WITH FIRM 11 TO GATE 107 is 1	**F37-FIRM11**
38	FLIGHT 3 (22,46) WITH FIRM 8 TO GATE 108 is 1	**F38-FIRM8**
	FLIGHT 3 (22,46) WITH FIRM 8 TO GATE 109 is 1	**F38-FIRM8**
	FLIGHT 3 (22,46) WITH FIRM 8 TO GATE 110 is 1	**F38-FIRM8**
39	FLIGHT 4 (23,47) WITH FIRM 14 TO GATE 113 is 1	**F39-FIRM14**
	FLIGHT 4 (23,47) WITH FIRM 14 TO GATE 114 is 1	**F39-FIRM14**
40	FLIGHT 5 (23,47) WITH FIRM 6 TO GATE 102 is 1	**F40-FIRM6**
	FLIGHT 5 (23,47) WITH FIRM 6 TO GATE 103 is 1	**F40-FIRM6**
	FLIGHT 5 (23,47) WITH FIRM 6 TO GATE 104 is 1	**F40-FIRM6**
41	FLIGHT 1 (24,48) WITH FIRM 15 TO GATE 115 is 1	**F41-FIRM15**
	FLIGHT 1 (24,48) WITH FIRM 15 TO GATE 116 is 1	**F41-FIRM15**
	FLIGHT 1 (24,48) WITH FIRM 15 TO GATE 117 is 1	**F41-FIRM15**
42	FLIGHT 1 (25,49) WITH FIRM 6 TO GATE 120 is 1	**F42-FIRM6**
	FLIGHT 1 (25,49) WITH FIRM 6 TO GATE 121 is 1	**F42-FIRM6**
	FLIGHT 1 (25,49) WITH FIRM 6 TO GATE 122 is 1	**F42-FIRM6**
43	FLIGHT 2 (25,49) WITH FIRM 16 TO GATE 118 is 1	**F43-FIRM16**
	FLIGHT 2 (25,49) WITH FIRM 16 TO GATE 119 is 1	**F43-FIRM16**
44	FLIGHT 3 (26,50) WITH FIRM 9 TO GATE 9 is 1	**F44-FIRM9**
	FLIGHT 3 (26,50) WITH FIRM 9 TO GATE 10 is 1	**F44-FIRM9**
	FLIGHT 3 (26,50) WITH FIRM 9 TO GATE 11 is 1	**F44-FIRM9**
	FLIGHT 3 (26,50) WITH FIRM 9 TO GATE 12 is 1	**F44-FIRM9**
45	FLIGHT 4 (26,50) WITH FIRM 2 TO GATE 1 is 1	**F45-FIRM2**
	FLIGHT 4 (26,50) WITH FIRM 2 TO GATE 2 is 1	**F45-FIRM2**
46	FLIGHT 5 (27,51) WITH FIRM 17 TO GATE 3 is 1	**F46-FIRM17**
	FLIGHT 5 (27,51) WITH FIRM 17 TO GATE 4 is 1	**F46-FIRM17**
	FLIGHT 5 (27,51) WITH FIRM 17 TO GATE 5 is 1	**F46-FIRM17**
47	FLIGHT 1 (27,51) WITH FIRM 14 TO GATE 6 is 1	**F47-FIRM14**
	FLIGHT 1 (27,51) WITH FIRM 14 TO GATE 7 is 1	**F47-FIRM14**
	FLIGHT 1 (27,51) WITH FIRM 14 TO GATE 8 is 1	**F47-FIRM14**
48	FLIGHT 2 (27,51) WITH FIRM 18 TO GATE 125 is 1	**F48-FIRM18**
	FLIGHT 2 (27,51) WITH FIRM 18 TO GATE 126 is 1	**F48-FIRM18**
49	FLIGHT 3 (28,52) WITH FIRM 19 TO GATE 123 is 1	**F49-FIRM19**
	FLIGHT 3 (28,52) WITH FIRM 19 TO GATE 124 is 1	**F49-FIRM19**
50	FLIGHT 4 (28,52) WITH FIRM 7 TO GATE 130 is 1	**F50-FIRM7**
	FLIGHT 4 (28,52) WITH FIRM 7 TO GATE 131 is 1	**F50-FIRM7**
	FLIGHT 4 (28,52) WITH FIRM 7 TO GATE 132 is 1	**F50-FIRM7**
51	FLIGHT 5 (28,52) WITH FIRM 17 TO GATE 127 is 1	**F51-FIRM17**
	FLIGHT 5 (28,52) WITH FIRM 17 TO GATE 128 is 1	**F51-FIRM17**
	FLIGHT 5 (28,52) WITH FIRM 17 TO GATE 129 is 1	**F51-FIRM17**
52	FLIGHT 1 (29,53) WITH FIRM 18 TO GATE 20 is 1	**F52-FIRM18**
	FLIGHT 1 (29,53) WITH FIRM 18 TO GATE 21 is 1	**F52-FIRM18**

53	FLIGHT 2 (29,53) WITH FIRM 5 TO GATE 133 is 1	**F53-FIRM5**
	FLIGHT 2 (29,53) WITH FIRM 5 TO GATE 134 is 1	**F53-FIRM5**
54	FLIGHT 1 (32,56) WITH FIRM 20 TO GATE 22 is 1	**F58-FIRM20**
	FLIGHT 1 (32,56) WITH FIRM 20 TO GATE 23 is 1	**F58-FIRM20**
55	FLIGHT 2 (33,57) WITH FIRM 21 TO GATE 16 is 1	**F59-FIRM21**
	FLIGHT 2 (33,57) WITH FIRM 21 TO GATE 17 is 1	**F59-FIRM21**
	FLIGHT 2 (33,57) WITH FIRM 21 TO GATE 18 is 1	**F59-FIRM21**
56	FLIGHT 3 (33,57) WITH FIRM 21 TO GATE 13 is 1	**F60-FIRM21**
	FLIGHT 3 (33,57) WITH FIRM 21 TO GATE 14 is 1	**F60-FIRM21**
	FLIGHT 3 (33,57) WITH FIRM 21 TO GATE 15 is 1	**F60-FIRM21**
57	FLIGHT 1 (35,59) WITH FIRM 21 TO GATE 34 is 1	**F62-FIRM21**
	FLIGHT 1 (35,59) WITH FIRM 21 TO GATE 35 is 1	**F62-FIRM21**
	FLIGHT 1 (35,59) WITH FIRM 21 TO GATE 36 is 1	**F62-FIRM21**
58	FLIGHT 2 (35,59) WITH FIRM 23 TO GATE 24 is 1	**F63-FIRM23**
	FLIGHT 2 (35,59) WITH FIRM 23 TO GATE 25 is 1	**F63-FIRM23**
59	FLIGHT 3 (35,59) WITH FIRM 10 TO GATE 28 is 1	**F64-FIRM10**
	FLIGHT 3 (35,59) WITH FIRM 10 TO GATE 29 is 1	**F64-FIRM10**
	FLIGHT 3 (35,59) WITH FIRM 10 TO GATE 30 is 1	**F64-FIRM10**
60	FLIGHT 4 (35,59) WITH FIRM 23 TO GATE 26 is 1	**F65-FIRM23**
	FLIGHT 4 (35,59) WITH FIRM 23 TO GATE 27 is 1	**F65-FIRM23**
61	FLIGHT 5 (37,61) WITH FIRM 15 TO GATE 31 is 1	**F66-FIRM15**
	FLIGHT 5 (37,61) WITH FIRM 15 TO GATE 32 is 1	**F66-FIRM15**
	FLIGHT 5 (37,61) WITH FIRM 15 TO GATE 33 is 1	**F66-FIRM15**
62	FLIGHT 1 (38,62) WITH FIRM 20 TO GATE 52 is 1	**F67-FIRM20**
	FLIGHT 1 (38,62) WITH FIRM 20 TO GATE 53 is 1	**F67-FIRM20**
63	FLIGHT 2 (38,62) WITH FIRM 7 TO GATE 37 is 1	**F68-FIRM7**
	FLIGHT 2 (38,62) WITH FIRM 7 TO GATE 38 is 1	**F68-FIRM7**
	FLIGHT 2 (38,62) WITH FIRM 7 TO GATE 39 is 1	**F68-FIRM7**
64	FLIGHT 3 (38,62) WITH FIRM 3 TO GATE 40 is 1	**F69-FIRM3**
	FLIGHT 3 (38,62) WITH FIRM 3 TO GATE 41 is 1	**F69-FIRM3**
	FLIGHT 3 (38,62) WITH FIRM 3 TO GATE 42 is 1	**F69-FIRM3**
65	FLIGHT 4 (39,63) WITH FIRM 24 TO GATE 43 is 1	**F70-FIRM24**
	FLIGHT 4 (39,63) WITH FIRM 24 TO GATE 44 is 1	**F70-FIRM24**
	FLIGHT 4 (39,63) WITH FIRM 24 TO GATE 45 is 1	**F70-FIRM24**
	FLIGHT 4 (39,63) WITH FIRM 24 TO GATE 46 is 1	**F70-FIRM24**
	FLIGHT 4 (39,63) WITH FIRM 24 TO GATE 47 is 1	**F70-FIRM24**
66	FLIGHT 5 (40,64) WITH FIRM 9 TO GATE 48 is 1	**F71-FIRM9**
	FLIGHT 5 (40,64) WITH FIRM 9 TO GATE 49 is 1	**F71-FIRM9**
	FLIGHT 5 (40,64) WITH FIRM 9 TO GATE 50 is 1	**F71-FIRM9**
	FLIGHT 5 (40,64) WITH FIRM 9 TO GATE 51 is 1	**F71-FIRM9**
67	FLIGHT 1 (40,64) WITH FIRM 25 TO GATE 68 is 1	**F72-FIRM25**
	FLIGHT 1 (40,64) WITH FIRM 25 TO GATE 69 is 1	**F72-FIRM25**
	FLIGHT 1 (40,64) WITH FIRM 25 TO GATE 70 is 1	**F72-FIRM25**
68	FLIGHT 2 (40,64) WITH FIRM 3 TO GATE 57 is 1	**F73-FIRM3**
	FLIGHT 2 (40,64) WITH FIRM 3 TO GATE 58 is 1	**F73-FIRM3**
	FLIGHT 2 (40,64) WITH FIRM 3 TO GATE 59 is 1	**F73-FIRM3**
69	FLIGHT 3 (41,65) WITH FIRM 8 TO GATE 54 is 1	**F74-FIRM8**
	FLIGHT 3 (41,65) WITH FIRM 8 TO GATE 55 is 1	**F74-FIRM8**
	FLIGHT 3 (41,65) WITH FIRM 8 TO GATE 56 is 1	**F74-FIRM8**
70	FLIGHT 4 (41,65) WITH FIRM 26 TO GATE 60 is 1	**F75-FIRM26**
	FLIGHT 4 (41,65) WITH FIRM 26 TO GATE 61 is 1	**F75-FIRM26**
	FLIGHT 4 (41,65) WITH FIRM 26 TO GATE 62 is 1	**F75-FIRM26**

71	FLIGHT 5 (41,65) WITH FIRM 24 TO GATE 63 is 1	**F76-FIRM24**
	FLIGHT 5 (41,65) WITH FIRM 24 TO GATE 64 is 1	**F76-FIRM24**
	FLIGHT 5 (41,65) WITH FIRM 24 TO GATE 65 is 1	**F76-FIRM24**
	FLIGHT 5 (41,65) WITH FIRM 24 TO GATE 66 is 1	**F76-FIRM24**
	FLIGHT 5 (41,65) WITH FIRM 24 TO GATE 67 is 1	**F76-FIRM24**
72	FLIGHT 1 (41,65) WITH FIRM 20 TO GATE 71 is 1	**F77-FIRM20**
	FLIGHT 1 (41,65) WITH FIRM 20 TO GATE 72 is 1	**F77-FIRM20**
73	FLIGHT 2 (41,65) WITH FIRM 4 TO GATE 73 is 1	**F78-FIRM4**
	FLIGHT 2 (41,65) WITH FIRM 4 TO GATE 74 is 1	**F78-FIRM4**
	FLIGHT 2 (41,65) WITH FIRM 4 TO GATE 75 is 1	**F78-FIRM4**
74	FLIGHT 3 (43,67) WITH FIRM 9 TO GATE 76 is 1	**F79-FIRM9**
	FLIGHT 3 (43,67) WITH FIRM 9 TO GATE 77 is 1	**F79-FIRM9**
	FLIGHT 3 (43,67) WITH FIRM 9 TO GATE 78 is 1	**F79-FIRM9**
	FLIGHT 3 (43,67) WITH FIRM 9 TO GATE 79 is 1	**F79-FIRM9**
75	FLIGHT 4 (43,67) WITH FIRM 14 TO GATE 80 is 1	**F80-FIRM14**
	FLIGHT 4 (43,67) WITH FIRM 14 TO GATE 81 is 1	**F80-FIRM14**
	FLIGHT 4 (43,67) WITH FIRM 14 TO GATE 82 is 1	**F80-FIRM14**
	FLIGHT 4 (43,67) WITH FIRM 14 TO GATE 83 is 1	**F80-FIRM14**
	FLIGHT 4 (43,67) WITH FIRM 14 TO GATE 84 is 1	**F80-FIRM14**
	FLIGHT 4 (43,67) WITH FIRM 14 TO GATE 85 is 1	**F80-FIRM14**
	FLIGHT 4 (43,67) WITH FIRM 14 TO GATE 86 is 1	**F80-FIRM14**
	FLIGHT 4 (43,67) WITH FIRM 14 TO GATE 87 is 1	**F80-FIRM14**
76	FLIGHT 5 (43,67) WITH FIRM 14 TO GATE 88 is 1	**F81-FIRM14**
	FLIGHT 5 (43,67) WITH FIRM 14 TO GATE 89 is 1	**F81-FIRM14**
77	FLIGHT 1 (44,68) WITH FIRM 9 TO GATE 90 is 1	**F82-FIRM9**
	FLIGHT 1 (44,68) WITH FIRM 9 TO GATE 91 is 1	**F82-FIRM9**
	FLIGHT 1 (44,68) WITH FIRM 9 TO GATE 92 is 1	**F82-FIRM9**
	FLIGHT 1 (44,68) WITH FIRM 9 TO GATE 93 is 1	**F82-FIRM9**
78	FLIGHT 2 (45,69) WITH FIRM 9 TO GATE 94 is 1	**F83-FIRM9**
	FLIGHT 2 (45,69) WITH FIRM 9 TO GATE 95 is 1	**F83-FIRM9**
	FLIGHT 2 (45,69) WITH FIRM 9 TO GATE 96 is 1	**F83-FIRM9**
	FLIGHT 2 (45,69) WITH FIRM 9 TO GATE 97 is 1	**F83-FIRM9**
79	FLIGHT 3 (46,70) WITH FIRM 14 TO GATE 105 is 1	**F84-FIRM14**
	FLIGHT 3 (46,70) WITH FIRM 14 TO GATE 106 is 1	**F84-FIRM14**
	FLIGHT 3 (46,70) WITH FIRM 14 TO GATE 107 is 1	**F84-FIRM14**
	FLIGHT 3 (46,70) WITH FIRM 14 TO GATE 108 is 1	**F84-FIRM14**
	FLIGHT 3 (46,70) WITH FIRM 14 TO GATE 109 is 1	**F84-FIRM14**
	FLIGHT 3 (46,70) WITH FIRM 14 TO GATE 110 is 1	**F84-FIRM14**
	FLIGHT 3 (46,70) WITH FIRM 14 TO GATE 111 is 1	**F84-FIRM14**
	FLIGHT 3 (46,70) WITH FIRM 14 TO GATE 112 is 1	**F84-FIRM14**
80	FLIGHT 4 (47,71) WITH FIRM 3 TO GATE 100 is 1	**F85-FIRM3**
	FLIGHT 4 (47,71) WITH FIRM 3 TO GATE 101 is 1	**F85-FIRM3**
	FLIGHT 4 (47,71) WITH FIRM 3 TO GATE 102 is 1	**F85-FIRM3**
81	FLIGHT 5 (47,71) WITH FIRM 2 TO GATE 98 is 1	**F86-FIRM2**
	FLIGHT 5 (47,71) WITH FIRM 2 TO GATE 99 is 1	**F86-FIRM2**
82	FLIGHT 1 (48,72) WITH FIRM 20 TO GATE 103 is 1	**F87-FIRM20**
	FLIGHT 1 (48,72) WITH FIRM 20 TO GATE 104 is 1	**F87-FIRM20**

83	FLIGHT 2 (48,72) WITH FIRM 9 TO GATE 113 is 1	**F88-FIRM9**
	FLIGHT 2 (48,72) WITH FIRM 9 TO GATE 114 is 1	**F88-FIRM9**
	FLIGHT 2 (48,72) WITH FIRM 9 TO GATE 115 is 1	**F88-FIRM9**
	FLIGHT 2 (48,72) WITH FIRM 9 TO GATE 116 is 1	**F88-FIRM9**
84	FLIGHT 1 (49,73) WITH FIRM 27 TO GATE 117 is 1	**F89-FIRM27**
	FLIGHT 1 (49,73) WITH FIRM 27 TO GATE 118 is 1	**F89-FIRM27**
85	FLIGHT 2 (50,74) WITH FIRM 5 TO GATE 1 is 1	**F90-FIRM5**
	FLIGHT 2 (50,74) WITH FIRM 5 TO GATE 2 is 1	**F90-FIRM5**
86	FLIGHT 3 (50,74) WITH FIRM 21 TO GATE 9 is 1	**F92-FIRM21**
	FLIGHT 3 (50,74) WITH FIRM 21 TO GATE 10 is 1	**F92-FIRM21**
	FLIGHT 3 (50,74) WITH FIRM 21 TO GATE 11 is 1	**F92-FIRM21**
87	FLIGHT 4 (51,75) WITH FIRM 28 TO GATE 3 is 1	**F93-FIRM28**
	FLIGHT 4 (51,75) WITH FIRM 28 TO GATE 4 is 1	**F93-FIRM28**
88	FLIGHT 1 (52,76) WITH FIRM 28 TO GATE 119 is 1	**F94-FIRM28**
	FLIGHT 1 (52,76) WITH FIRM 28 TO GATE 120 is 1	**F94-FIRM28**
89	FLIGHT 2 (52,76) WITH FIRM 28 TO GATE 121 is 1	**F95-FIRM28**
	FLIGHT 2 (52,76) WITH FIRM 28 TO GATE 122 is 1	**F95-FIRM28**
90	FLIGHT 3 (52,76) WITH FIRM 29 TO GATE 7 is 1	**F96-FIRM29**
	FLIGHT 3 (52,76) WITH FIRM 29 TO GATE 8 is 1	**F96-FIRM29**
91	FLIGHT 4 (52,76) WITH FIRM 20 TO GATE 5 is 1	**F97-FIRM20**
	FLIGHT 4 (52,76) WITH FIRM 20 TO GATE 6 is 1	**F97-FIRM20**
92	FLIGHT 5 (53,77) WITH FIRM 5 TO GATE 19 is 1	**F98-FIRM5**
	FLIGHT 5 (53,77) WITH FIRM 5 TO GATE 20 is 1	**F98-FIRM5**
93	FLIGHT 1 (53,77) WITH FIRM 6 TO GATE 130 is 1	**F99-FIRM6**
	FLIGHT 1 (53,77) WITH FIRM 6 TO GATE 131 is 1	**F99-FIRM6**
	FLIGHT 1 (53,77) WITH FIRM 6 TO GATE 132 is 1	**F99-FIRM6**
94	FLIGHT 2 (53,77) WITH FIRM 30 TO GATE 126 is 1	**F100-FIRM30**
	FLIGHT 2 (53,77) WITH FIRM 30 TO GATE 127 is 1	**F100-FIRM30**
95	FLIGHT 3 (55,79) WITH FIRM 12 TO GATE 123 is 1	**F101-FIRM12**
	FLIGHT 3 (55,79) WITH FIRM 12 TO GATE 124 is 1	**F101-FIRM12**
	FLIGHT 3 (55,79) WITH FIRM 12 TO GATE 125 is 1	**F101-FIRM12**
96	FLIGHT 4 (55,79) WITH FIRM 31 TO GATE 133 is 1	**F102-FIRM31**
	FLIGHT 4 (55,79) WITH FIRM 31 TO GATE 134 is 1	**F102-FIRM31**
	FLIGHT 4 (55,79) WITH FIRM 31 TO GATE 135 is 1	**F102-FIRM31**
97	FLIGHT 5 (55,79) WITH FIRM 32 TO GATE 128 is 1	**F103-FIRM32**
	FLIGHT 5 (55,79) WITH FIRM 32 TO GATE 129 is 1	**F103-FIRM32**
98	FLIGHT 1 (56,80) WITH FIRM 30 TO GATE 21 is 1	**F104-FIRM30**
	FLIGHT 1 (56,80) WITH FIRM 30 TO GATE 22 is 1	**F104-FIRM30**
99	FLIGHT 2 (57,81) WITH FIRM 33 TO GATE 12 is 1	**F105-FIRM33**
	FLIGHT 2 (57,81) WITH FIRM 33 TO GATE 13 is 1	**F105-FIRM33**
	FLIGHT 2 (57,81) WITH FIRM 33 TO GATE 14 is 1	**F105-FIRM33**
100	FLIGHT 3 (57,81) WITH FIRM 20 TO GATE 15 is 1	**F106-FIRM20**
	FLIGHT 3 (57,81) WITH FIRM 20 TO GATE 16 is 1	**F106-FIRM20**

101	FLIGHT 4 (58,82) WITH FIRM 28 TO GATE 17 is 1	**F107-FIRM28**
	FLIGHT 4 (58,82) WITH FIRM 28 TO GATE 18 is 1	**F107-FIRM28**
102	FLIGHT 5 (59,83) WITH FIRM 10 TO GATE 23 is 1	**F108-FIRM10**
	FLIGHT 5 (59,83) WITH FIRM 10 TO GATE 24 is 1	**F108-FIRM10**
	FLIGHT 5 (59,83) WITH FIRM 10 TO GATE 25 is 1	**F108-FIRM10**
103	FLIGHT 1 (59,83) WITH FIRM 34 TO GATE 34 is 1	**F109-FIRM34**
	FLIGHT 1 (59,83) WITH FIRM 34 TO GATE 35 is 1	**F109-FIRM34**
	FLIGHT 1 (59,83) WITH FIRM 34 TO GATE 36 is 1	**F109-FIRM34**
104	FLIGHT 2 (61,85) WITH FIRM 8 TO GATE 26 is 1	**F110-FIRM8**
	FLIGHT 2 (61,85) WITH FIRM 8 TO GATE 27 is 1	**F110-FIRM8**
	FLIGHT 2 (61,85) WITH FIRM 8 TO GATE 28 is 1	**F110-FIRM8**
105	FLIGHT 3 (62,86) WITH FIRM 35 TO GATE 32 is 1	**F111-FIRM35**
	FLIGHT 3 (62,86) WITH FIRM 35 TO GATE 33 is 1	**F111-FIRM35**
106	FLIGHT 4 (62,86) WITH FIRM 17 TO GATE 29 is 1	**F112-FIRM17**
	FLIGHT 4 (62,86) WITH FIRM 17 TO GATE 30 is 1	**F112-FIRM17**
	FLIGHT 4 (62,86) WITH FIRM 17 TO GATE 31 is 1	**F112-FIRM17**
107	FLIGHT 5 (62,86) WITH FIRM 22 TO GATE 37 is 1	**F113-FIRM22**
	FLIGHT 5 (62,86) WITH FIRM 22 TO GATE 38 is 1	**F113-FIRM22**
	FLIGHT 5 (62,86) WITH FIRM 22 TO GATE 39 is 1	**F113-FIRM22**
108	FLIGHT 1 (63,87) WITH FIRM 36 TO GATE 40 is 1	**F114-FIRM36**
	FLIGHT 1 (63,87) WITH FIRM 36 TO GATE 41 is 1	**F114-FIRM36**
	FLIGHT 1 (63,87) WITH FIRM 36 TO GATE 42 is 1	**F114-FIRM36**
	FLIGHT 1 (63,87) WITH FIRM 36 TO GATE 43 is 1	**F114-FIRM36**
	FLIGHT 1 (63,87) WITH FIRM 36 TO GATE 44 is 1	**F114-FIRM36**
109	FLIGHT 2 (63,87) WITH FIRM 15 TO GATE 45 is 1	**F115-FIRM15**
	FLIGHT 2 (63,87) WITH FIRM 15 TO GATE 46 is 1	**F115-FIRM15**
	FLIGHT 2 (63,87) WITH FIRM 15 TO GATE 47 is	**F115-FIRM15**
110	FLIGHT 3 (64,88) WITH FIRM 37 TO GATE 48 is 1	**F117-FIRM37**
	FLIGHT 3 (64,88) WITH FIRM 37 TO GATE 49 is 1	**F117-FIRM37**
	FLIGHT 3 (64,88) WITH FIRM 37 TO GATE 50 is 1	**F117-FIRM37**
	FLIGHT 3 (64,88) WITH FIRM 37 TO GATE 51 is 1	**F117-FIRM37**
111	FLIGHT 4 (64,88) WITH FIRM 28 TO GATE 52 is 1	**F118-FIRM28**
	FLIGHT 4 (64,88) WITH FIRM 28 TO GATE 53 is 1	**F118-FIRM28**
112	FLIGHT 1 (65,89) WITH FIRM 38 TO GATE 62 is 1	**F119-FIRM38**
	FLIGHT 1 (65,89) WITH FIRM 38 TO GATE 63 is 1	**F119-FIRM38**
113	FLIGHT 2 (65,89) WITH FIRM 39 TO GATE 61 is 1	**F120-FIRM39**
114	FLIGHT 3 (66,90) WITH FIRM 16 TO GATE 54 is 1	**F121-FIRM16**
	FLIGHT 3 (66,90) WITH FIRM 16 TO GATE 55 is 1	**F121-FIRM16**
115	FLIGHT 4 (66,90) WITH FIRM 40 TO GATE 59 is 1	**F122-FIRM40**
	FLIGHT 4 (66,90) WITH FIRM 40 TO GATE 60 is 1	**F122-FIRM40**
116	FLIGHT 5 (67,91) WITH FIRM 41 TO GATE 56 is 1	**F123-FIRM41**
	FLIGHT 5 (67,91) WITH FIRM 41 TO GATE 57 is 1	**F123-FIRM41**
	FLIGHT 5 (67,91) WITH FIRM 41 TO GATE 58 is 1	**F123-FIRM41**
117	FLIGHT 1 (67,91) WITH FIRM 9 TO GATE 75 is 1	**F124-FIRM9**
	FLIGHT 1 (67,91) WITH FIRM 9 TO GATE 76 is 1	**F124-FIRM9**
	FLIGHT 1 (67,91) WITH FIRM 9 TO GATE 77 is 1	**F124-FIRM9**
	FLIGHT 1 (67,91) WITH FIRM 9 TO GATE 78 is 1	**F124-FIRM9**

118	FLIGHT 2 (68,92) WITH FIRM 37 TO GATE 64 is 1	**F125-FIRM37**
	FLIGHT 2 (68,92) WITH FIRM 37 TO GATE 65 is 1	**F125-FIRM37**
	FLIGHT 2 (68,92) WITH FIRM 37 TO GATE 66 is 1	**F125-FIRM37**
	FLIGHT 2 (68,92) WITH FIRM 37 TO GATE 67 is 1	**F125-FIRM37**
119	FLIGHT 3 (68,92) WITH FIRM 20 TO GATE 71 is 1	**F126-FIRM20**
	FLIGHT 3 (68,92) WITH FIRM 20 TO GATE 72 is 1	**F126-FIRM20**
120	FLIGHT 4 (69,93) WITH FIRM 42 TO GATE 73 is 1	**F127-FIRM42**
	FLIGHT 4 (69,93) WITH FIRM 42 TO GATE 74 is 1	**F127-FIRM42**
121	FLIGHT 5 (69,93) WITH FIRM 43 TO GATE 68 is 1	**F128-FIRM43**
	FLIGHT 5 (69,93) WITH FIRM 43 TO GATE 69 is 1	**F128-FIRM43**
	FLIGHT 5 (69,93) WITH FIRM 43 TO GATE 70 is 1	**F128-FIRM43**
122	FLIGHT 1 (70,94) WITH FIRM 22 TO GATE 82 is 1	**F129-FIRM22**
	FLIGHT 1 (70,94) WITH FIRM 22 TO GATE 83 is 1	**F129-FIRM22**
	FLIGHT 1 (70,94) WITH FIRM 22 TO GATE 84 is 1	**F129-FIRM22**
123	FLIGHT 2 (70,94) WITH FIRM 38 TO GATE 89 is 1	**F130-FIRM38**
	FLIGHT 2 (70,94) WITH FIRM 38 TO GATE 90 is 1	**F130-FIRM38**
124	FLIGHT 3 (70,94) WITH FIRM 9 TO GATE 91 is 1	**F131-FIRM9**
	FLIGHT 3 (70,94) WITH FIRM 9 TO GATE 92 is 1	**F131-FIRM9**
	FLIGHT 3 (70,94) WITH FIRM 9 TO GATE 93 is 1	**F131-FIRM9**
	FLIGHT 3 (70,94) WITH FIRM 9 TO GATE 94 is 1	**F131-FIRM9**
125	FLIGHT 4 (71,95) WITH FIRM 37 TO GATE 85 is 1	**F132-FIRM37**
	FLIGHT 4 (71,95) WITH FIRM 37 TO GATE 86 is 1	**F132-FIRM37**
	FLIGHT 4 (71,95) WITH FIRM 37 TO GATE 87 is 1	**F132-FIRM37**
	FLIGHT 4 (71,95) WITH FIRM 37 TO GATE 88 is 1	**F132-FIRM37**
126	FLIGHT 5 (71,95) WITH FIRM 33 TO GATE 79 is 1	**F133-FIRM33**
	FLIGHT 5 (71,95) WITH FIRM 33 TO GATE 80 is 1	**F133-FIRM33**
	FLIGHT 5 (71,95) WITH FIRM 33 TO GATE 81 is 1	**F133-FIRM33**
127	FLIGHT 1 (71,95) WITH FIRM 14 TO GATE 95 is 1	**F134-FIRM14**
	FLIGHT 1 (71,95) WITH FIRM 14 TO GATE 96 is 1	**F134-FIRM14**
	FLIGHT 1 (71,95) WITH FIRM 14 TO GATE 97 is 1	**F134-FIRM14**
	FLIGHT 1 (71,95) WITH FIRM 14 TO GATE 98 is 1	**F134-FIRM14**
	FLIGHT 1 (71,95) WITH FIRM 14 TO GATE 99 is 1	**F134-FIRM14**
	FLIGHT 1 (71,95) WITH FIRM 14 TO GATE 100 is 1	**F134-FIRM14**
	FLIGHT 1 (71,95) WITH FIRM 14 TO GATE 101 is 1	**F134-FIRM14**
	FLIGHT 1 (71,95) WITH FIRM 14 TO GATE 102 is 1	**F134-FIRM14**
128	FLIGHT 2 (72,96) WITH FIRM 44 TO GATE 103 is 1	**F135-FIRM44**
	FLIGHT 2 (72,96) WITH FIRM 44 TO GATE 104 is 1	**F135-FIRM44**
	FLIGHT 2 (72,96) WITH FIRM 44 TO GATE 105 is 1	**F135-FIRM44**
129	FLIGHT 1 (73,97) WITH FIRM 22 TO GATE 106 is 1	**F136-FIRM22**
	FLIGHT 1 (73,97) WITH FIRM 22 TO GATE 107 is 1	**F136-FIRM22**
	FLIGHT 1 (73,97) WITH FIRM 22 TO GATE 108 is 1	**F136-FIRM22**
130	FLIGHT 2 (73,97) WITH FIRM 9 TO GATE 113 is 1	**F137-FIRM9**
	FLIGHT 2 (73,97) WITH FIRM 9 TO GATE 114 is 1	**F137-FIRM9**
	FLIGHT 2 (73,97) WITH FIRM 9 TO GATE 115 is 1	**F137-FIRM9**
	FLIGHT 2 (73,97) WITH FIRM 9 TO GATE 116 is 1	**F137-FIRM9**
131	FLIGHT 3 (73,97) WITH FIRM 45 TO GATE 109 is 1	**F138-FIRM45**
	FLIGHT 3 (73,97) WITH FIRM 45 TO GATE 110 is 1	**F138-FIRM45**
132	FLIGHT 4 (73,97) WITH FIRM 20 TO GATE 111 is 1	**F139-FIRM20**
	FLIGHT 4 (73,97) WITH FIRM 20 TO GATE 112 is 1	**F139-FIRM20**
133	FLIGHT 5 (75,99) WITH FIRM 46 TO GATE 1 is 1	**F140-FIRM46**
	FLIGHT 5 (75,99) WITH FIRM 46 TO GATE 2 is 1	**F140-FIRM46**

134	FLIGHT 1 (75,99) WITH FIRM 47 TO GATE 3 is 1	**F141-FIRM47**
	FLIGHT 1 (75,99) WITH FIRM 47 TO GATE 4 is 1	**F141-FIRM47**
135	FLIGHT 2 (75,99) WITH FIRM 22 TO GATE 9 is 1	**F142-FIRM22**
	FLIGHT 2 (75,99) WITH FIRM 22 TO GATE 10 is 1	**F142-FIRM22**
	FLIGHT 2 (75,99) WITH FIRM 22 TO GATE 11 is 1	**F142-FIRM22**
136	FLIGHT 3 (76,100) WITH FIRM 37 TO GATE 5 is 1	**F143-FIRM37**
	FLIGHT 3 (76,100) WITH FIRM 37 TO GATE 6 is 1	**F143-FIRM37**
	FLIGHT 3 (76,100) WITH FIRM 37 TO GATE 7 is 1	**F143-FIRM37**
	FLIGHT 3 (76,100) WITH FIRM 37 TO GATE 8 is 1	**F143-FIRM37**
137	FLIGHT 4 (77,101) WITH FIRM 37 TO GATE 130 is 1	**F144-FIRM37**
	FLIGHT 4 (77,101) WITH FIRM 37 TO GATE 131 is 1	**F144-FIRM37**
	FLIGHT 4 (77,101) WITH FIRM 37 TO GATE 132 is 1	**F144-FIRM37**
138	FLIGHT 5 (77,101) WITH FIRM 37 TO GATE 117 is 1	**F145-FIRM37**
	FLIGHT 5 (77,101) WITH FIRM 37 TO GATE 118 is 1	**F145-FIRM37**
	FLIGHT 5 (77,101) WITH FIRM 37 TO GATE 119 is 1	**F145-FIRM37**
	FLIGHT 5 (77,101) WITH FIRM 37 TO GATE 120 is 1	**F145-FIRM37**
139	FLIGHT 1 (77,101) WITH FIRM 47 TO GATE 19 is 1	**F146-FIRM47**
	FLIGHT 1 (77,101) WITH FIRM 47 TO GATE 20 is 1	**F146-FIRM47**
140	FLIGHT 2 (77,101) WITH FIRM 48 TO GATE 121 is 1	**F147-FIRM48**
	FLIGHT 2 (77,101) WITH FIRM 48 TO GATE 122 is 1	**F147-FIRM48**
141	FLIGHT 3 (79,103) WITH FIRM 48 TO GATE 123 is 1	**F148-FIRM48**
	FLIGHT 3 (79,103) WITH FIRM 48 TO GATE 124 is 1	**F148-FIRM48**
142	FLIGHT 4 (79,103) WITH FIRM 8 TO GATE 125 is 1	**F149-FIRM8**
	FLIGHT 4 (79,103) WITH FIRM 8 TO GATE 126 is 1	**F149-FIRM8**
	FLIGHT 4 (79,103) WITH FIRM 8 TO GATE 127 is 1	**F149-FIRM8**
143	FLIGHT 5 (80,104) WITH FIRM 42 TO GATE 21 is 1	**F150-FIRM42**
	FLIGHT 5 (80,104) WITH FIRM 42 TO GATE 22 is 1	**F150-FIRM42**
144	FLIGHT 1 (80,104) WITH FIRM 11 TO GATE 133 is 1	**F151-FIRM11**
	FLIGHT 1 (80,104) WITH FIRM 11 TO GATE 134 is 1	**F151-FIRM11**
	FLIGHT 1 (80,104) WITH FIRM 11 TO GATE 135 is 1	**F151-FIRM11**
145	FLIGHT 2 (80,104) WITH FIRM 49 TO GATE 128 is 1	**F152-FIRM49**
	FLIGHT 2 (80,104) WITH FIRM 49 TO GATE 129 is 1	**F152-FIRM49**
146	FLIGHT 3 (81,105) WITH FIRM 50 TO GATE 14 is 1	**F153-FIRM50**
	FLIGHT 3 (81,105) WITH FIRM 50 TO GATE 15 is 1	**F153-FIRM50**
147	FLIGHT 4 (81,105) WITH FIRM 51 TO GATE 12 is 1	**F154-FIRM51**
	FLIGHT 4 (81,105) WITH FIRM 51 TO GATE 13 is 1	**F154-FIRM51**
148	FLIGHT 5 (82,106) WITH FIRM 35 TO GATE 16 is 1	**F155-FIRM35**
	FLIGHT 5 (82,106) WITH FIRM 35 TO GATE 17 is 1	**F155-FIRM35**
149	FLIGHT 1 (83,107) WITH FIRM 52 TO GATE 23 is 1	**F157-FIRM52**
	FLIGHT 1 (83,107) WITH FIRM 52 TO GATE 24 is 1	**F157-FIRM52**
	FLIGHT 1 (83,107) WITH FIRM 52 TO GATE 25 is 1	**F157-FIRM52**
150	FLIGHT 2 (83,107) WITH FIRM 7 TO GATE 34 is 1	**F158-FIRM7**
	FLIGHT 2 (83,107) WITH FIRM 7 TO GATE 35 is 1	**F158-FIRM7**
	FLIGHT 2 (83,107) WITH FIRM 7 TO GATE 36 is 1	**F158-FIRM7**

Table 24 The results of assignments of flights to the counters for developed model

MODELS	RUNTIME	OBJECTIVE FUNCTION
1_1	00:00:18:68	81
1_2	00:00:20:31	252
1_3	00:00:29:52	510
1_4	00:00:22:14	839
1_5	00:00:28:96	1077
1_6	00:00:25:29	1066
1_7	00:00:27:89	1435
1_8	00:00:22:41	1435
1_9	00:00:06:00	348
2_1	00:00:30:64	702
2_2	00:00:24:69	1313
2_3	00:00:07:15	318
2_4	00:00:08:50	138
2_5	00:00:18:59	420
2_6	00:00:15:58	801
2_7	00:00:16:08	1082
2_8	00:00:13:79	1552
2_9	00:00:16:18	2176
2_10	00:00:05:69	697
3_1	00:00:09:13	293
3_2	00:00:13:93	569
3_3	00:00:14:21	1681
3_4	00:00:13:93	253
3_5	00:00:13:78	465
3_6	00:00:14:77	664
3_7	00:00:24:16	588
3_8	00:00:28:59	1097
3_9	00:00:27:76	1429
3_10	00:00:06:17	1117
4_1	00:00:24:39	1252
4_2	00:00:23:72	967
4_3	00:00:21:88	966
4_4	00:00:21:76	752
4_5	00:00:08:05	177

Table 25 The result of each sub-problem and its run-time

KONTUAR 1	F3-FIRM3 · F45-FIRM2	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:
KONTUAR 2	F3-FIRM3 · F45-FIRM2	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:
KONTUAR 3	F3-FIRM3 · F46-FIRM17	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:
KONTUAR 4	F5-FIRM4 · F46-FIRM17	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:
KONTUAR 5	F5-FIRM4 · F46-FIRM17	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:
KONTUAR 6	F5-FIRM4 · F47-FIRM14	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:
KONTUAR 7	F4-FIRM2 · F47-FIRM14	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:
KONTUAR 8	F4-FIRM2 · F47-FIRM14	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:
KONTUAR 9	F2-FIRM2 · F44-FIRM9	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:
KONTUAR 10	F2-FIRM2 · F44-FIRM9	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:
KONTUAR 11	F1-FIRM1 · F44-FIRM9	02:40 02:45 02:50 02:55 03:00 03:05 03:10 03:15 03:20 03:25 03:30 03:35 03:40 03:45 03:50 03:55 04:00 04:05 04:10 04:15 04:20 04:25 04:30 04:35 04:40 04:45 04:50 04:55 05:00 05:05 05:10 05:15 05:20 05:25 05:30 05:35 05:40 05:45 05:50 05:55 06:00 06:05 06:

Table 26 The part of the result for Counter Assignment Problem