A NEW SOLUTION APPROACH FOR THE AIRPORT GATE ASSIGNMENT PROBLEM FOR HANDLING OF UNEVEN GATE DEMANDS

GENÇ, Hakkı Murat, İstanbul Technical University Control Engineering Department and The National Research Institute of Electronics and Cryptology, TÜBİTAK, TURKEY murat.genc@bte.tubitak.gov.tr

ABSTRACT

Airport Gate Assignment problem increases its importance with the growing demand on airport transportation. An efficient gate assignment that satisfies constraints specific to the airport has an important role in the revenue obtained from the airport operations. Besides, if an appropriate objective function is chosen, it may increase customer satisfaction by decreasing passenger walking distances within the airport and reduce the probability of bus transfers that may occur between airplane parking aprons and airport terminal. Constructing performance measures heavily depends on the needs of the operation control unit. Designing and weighting the objectives uniquely defines the problem at hand. Because of this, formulation and solution of airport gate assignment problems can vary quite a lot.

In this work the Airport Gate Assignment Problem (AGAP) is formulated as to minimize the vacant time duration of the gates. That is to say, the algorithm tries to assign planes as to fill up the (preferred) gates as much as possible. The objective of the allocation can be supervised by maximizing the revenue obtained.

Previous work carried on the subject can be investigated in many sub-categories: Single Objective Problems – Multi Objective Problems; Greedy Search Routines – Evolutionary Computation Methods; Deterministic Approaches – Stochastic Approaches; Single Time Slot Models – Multiple Time Slot Models, etc. One can easily extend this classification with respect to objective functions used or mathematical models imposed. The common aspect for all these solution approaches is that they do not credit for the distribution of gate allocation demand throughout the day. But there is relatively considerable accumulation for certain time durations in a complete day. Most frequently these are the morning hours at around 07:00 – 09:00 and 18:00 – 20:00 in the evening. These durations can be called "prime time" of the airports and can vary with respect to the airport specific measures like

capacity and number of connection flights leaving; as well as the date specific measures like seasonal weather, holidays etc.
This study deals with the aforementioned uneven demands on the flight list. The flight lists are ready for scheduling the gates for at least a day before. Then, according to these flight lists, the airplanes are assigned to the gates by considering the mostly demanded time intervals in certain priority. The paper discusses the fuzzy terms "prime time" and "high demand" and investigates the effect of three different approaches on handling the prime time traffic.

Algorithm proposed is simulated with the artificial data produced for different test cases that can be sketchily defined as the following:

- -A lot of planes, not enough gates, homogenous demand
- -A lot of planes, not enough gates, certain peaks on demand
- -A lot of planes, not enough gates, increase in demand in patches
- -Not so much planes, enough gates, increase in demand in patches
- -Not so much planes, enough gates, certain peaks on demand

Then the algorithm results will be given on a real data set obtained from Atatürk Airport of İstanbul.

Keywords: Gate assignment problem, prime time, heuristic assignment algorithms

INTRODUCTION

Airport Gate Assignment (Allocation) is an important area of research in airline operations and scheduling. Although the costs of these activities are generally small portions of the overall airline operation costs, they have a major impact on maintaining the efficiency of flight schedules and passengers (Bazargan, 2004).

The problem is easy to understand: optimize the defined performance index (PI) for the most efficient assignment of flights to the gates. But, both the performance index definition and solution techniques can vary quite a lot. Besides, there are certain constraints to be satisfied that are heavily depending on airport and plane properties. So, the gate assignment problem (GAP) can be categorized as multi-objective, multi constraint optimization problem in very general manner. Possible objective functions can be defined in terms of the gate durations of the planes, number of passengers in aircrafts, the total walking distances for total passengers of all scheduled flights within a specified and closed time interval.

Basic gate assignment problem is NP-hard (non-deterministic polynomial-time hard) (Obata, 1979) quadratic assignment problem. There are various approaches to this problem in the literature with respect to imposed requirements.

The gate assignment problems can be categorized with respect to objective functions, mathematical formulations, time slot models and constraint satisfaction strategies. Besides, the solution approaches have two heavily interacting main branches: rule based expert systems and mathematical models. The common aspect (or common weakness) for all these solution approaches is that they do not credit for the distribution of gate allocation demand throughout the day. But there is relatively considerable accumulation for certain time durations in a complete day. Most frequently these are the morning hours at around 07:00 – 09:00 and 18:00 – 20:00 in the evening. These durations can be called “prime time” of the airports and can vary with respect to the airport specific measures like capacity and number of connection flights leaving; as well as the date specific measures like seasonal weather, holidays etc.

In this study, the GAP objective is to minimize vacant gate time. By using a multiple time slot model, the whole day is divided into *n*-minutes intervals to measure the vacancy amount. For the formulation of the problem only the basic constraint that allows one flight at one gate at one time is utilized.

The rest of the paper is organised as follows. The formulation of the problem is given in the next section. Two previously reported heuristic approach is summarized in section titled “Heuristic Approaches”. “Prime Time Heuristic” is dedicated for the newly suggested heuristic method. In “Simulations” section test data set generation and tests are reported. Short summary and final comments are given in “Conclusion” part.

GATE ASSIGNMENT PROBLEM FORMULATION

In this work the Airport Gate Assignment Problem (AGAP) is formulated as to minimize the vacant time duration of the gates. That is to say, the algorithm tries to assign planes to fill up the (preferred) gates as much as possible. This objective has been selected since gate employment is the most related criterion with the income obtained from airport operations.

The basic constraint of the GAP imposed in the formulation can be stated as follows: one gate can only accommodate a single aircraft at a time and that therefore two flights must not be assigned to the same gate if their staying times overlap in time (Dorndorf et al., 2007). To measure gate allocation success, a day is sampled for *n* minutes, where *n* is chosen as 5. This time interval corresponding to *n* minutes is a time slot and the allocation success is measured by counting up all allocated time slots.

Figure 1 illustrates a sample assignment list for the flights. The vertical axis represents the gates available and the horizontal axis is the time slot index. The list is given for a whole day. In other words, a whole day is divided into (24 * 60 / n) timeslots. The planes, depicted as horizontal bars, are shown to occupy the corresponding gates for certain sojourn. As an example, a fictitious plane starting to use a gate at 7 am. and departing at 8 am. stays for 60 minutes, that is 60 / 5 = 12 time slots. Since minimizing vacant slots means maximizing the occupied time slots, assigning this plane earns 12 points for the maximization problem score. The total score is the sum of all occupied time slots in all preferred gates.

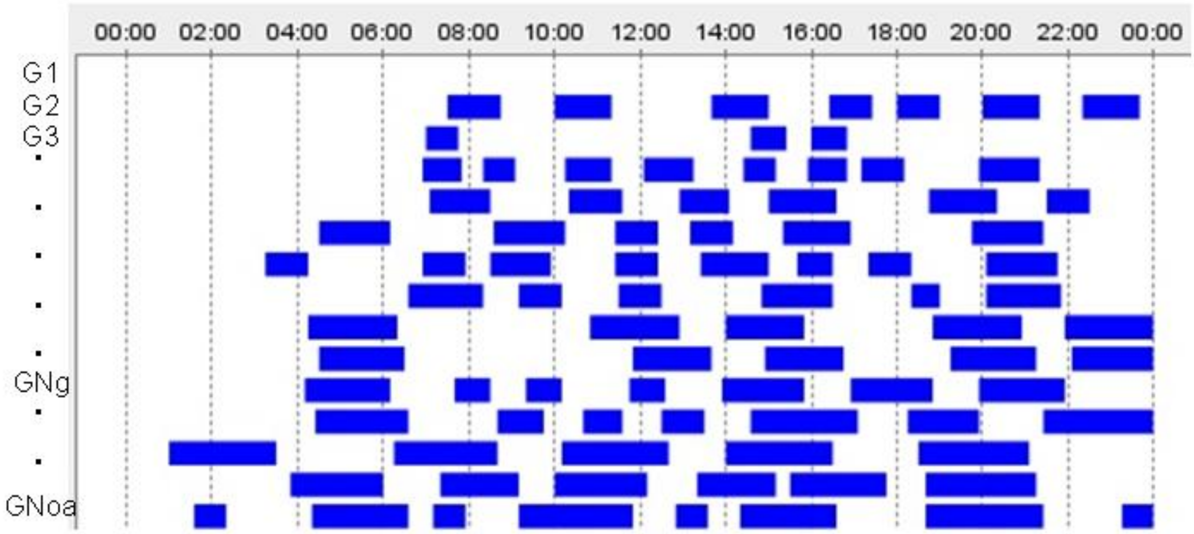

Figure 1 – A sample assignment list.

The basic constraint of the gate assignment problem can be easily verified from the figure. There are no overlapping flights on any gate and a flight is assigned to only one gate.

The problem formulation given above does not concern possible constraints imposed by airport gate characteristics and/or aircraft's physical properties such as length, weight and capacity. The only objective is maximum squeezing of the planes in order to increase total gating time. On the other hand, maximum squeezing means minimum robustness against delays or malfunctioning of any device. That is, the best condition for maximum allocation is to allocate flights end to end with no vacant time slot in between, whereas, this causes need for inevitable reconfiguration process in the occurrence of a delay. This trade off can be handled by defining time windows before arrival and after departure of the flights (Zhu et al. 2003).

HEURISTIC APPROACHES

There are two previously reported approaches designed to optimize given objective function (Genç et al., 2009). They are repeated here for quick referencing. Besides these two methods, Ding et al. (2004; 2004; 2005) proposed a greedy algorithm to minimize the number of the un-gated flights. In these works, the objective is to maximize number of flights assigned to the gates (or in their wording: minimize number of flights assigned to the apron), not the total gate duration, so, this heuristic is omitted here. There are no resembling reported heuristic approaches.

Ground Time Maximization Heuristic

In Ground Time Maximization Heuristic, the longest staying plane is assigned in the highest priority. The algorithm steps can be given as follows:

1. Sort the flights with respect to their planned gate durations.
2. Pick the flight with longest time interval between its arrival and departure.
3. Start from gate #1 (or in other words the most preferred gate).
4. Assign the flight to the gate if possible, else switch to the next gate. Repeat until finding a vacant gate. Remove the flight from the list once it is assigned.
5. Go to step #2 until all the flight have been assigned.

It is appropriate to note that, in this work, there are no further preference levels among the gates. If there are *Ng* gates, they are both the same, in other words, they are equally contributing to the performance score.

Idle Time Minimization Heuristic

Idle Time Minimization algorithm basically tries to assign each plane just after the previous plane's departure. The algorithm objectives to minimize the idle time in between two flights; so, the plane arriving closest to the departure of another is assigned. Algorithm steps can be summarized as follows:

1. Sort the flights according to the planned arrival times.
2. Start from gate #1.
3. Pick the nearest (arriving) flight from the list.
4. Assign it to gate at hand and remove the flight from the list.
5. Pick the nearest coming flight next the previous flight's departure from the list and assign it to the gate at hand and remove the flight from the list. Repeat this procedure until no flight can be assigned to the gate.
6. Increase the gate number by one and repeat the step 3-6 until all flights are assigned.

Actually this heuristic can be configured to fill up a gate beginning from the end of the day in a backward manner. To do so, one must first assign the last departure and then search for the latest departure before the assigned plane's arrival. The algorithm steps are slightly changed,

1. Sort the flights according to the planned **departure** times.
2. Start from gate #1.
3. Pick the **latest (departing)** flight from the list.
4. Assign it to gate at hand and remove the flight from the list.

5. Pick the **latest departing** flight **before the arrival of** the assigned flight from the list and assign it to the gate at hand and remove the flight from the list. Repeat this procedure until no flight can be assigned to the gate.
6. Increase the gate number by one and repeat the step 3-6 until all flights are assigned.

Idle Time Minimization Heuristic is a deterministic method as Ground Time Maximization Heuristic.

Failure Conditions for the Heuristics

Both heuristic approaches are easy to understand; easy to implement and run time is quite satisfactory for real world applications. However one can easily illustrate a possible failure condition. Table I, reports a very simplified flight table available before the day of allocation planning. Let us assume there is only one gate available in this simplified problem. The planes that are not assigned to this gate will be assigned to the apron and will not contribute to scoring. If the GAP solver uses ground time maximization heuristic, it assigns the longest staying flight (*TK 021*) to the available gate. Because of the basic assumption saying one gate can accommodate only one plane at a specific instant, the other planes can not be assigned to the same gate. The total score for this assignment will be 3 hours which in turns equals to be (3 * 60 / 5) 36 time slots in our formulation. However, one can easily observe that assigning both *TK 056* and *TK 221* will yield (165 / 5 + 35 / 5) 40 time slots. Note that in the second assignment alternative, gate durations for the flights are not overlapping and this assignment procedure is optimal for this specific problem with respect to the objective function used in this work.

Table I – An Example Flight List - Ground Time Maximization Heuristic Fails To Find Optimum Solution

Flight	Arrival Time	Departure Time
TK 021	11/07/2010 10:00	11/07/2010 13:00
TK 056	11/07/2010 10:30	11/07/2010 13:15
TK 221	11/07/2010 09:40	11/07/2010 10:15

A similar argument can be carried for the idle time minimization heuristic. Table II reports another sample flight schedule designed to reveal weakness of the algorithm. The example is given for the forward allocating first version of the algorithm. The GAP solver will assign *TK 021* as the first arriving plane. Then, after the departure of this flight the gate can only accommodate *TK 221* which together contributes 16 time slots. If only *TK 056* is assigned instead, the score will be 54 time slots.

Table II – An Example Flight List - Idle Time Minimization Heuristic Fails To Find Optimum Solution

Flight	Arrival Time	Departure Time
TK 021	11/07/2010 10:00	11/07/2010 11:00
TK 056	11/07/2010 10:30	11/07/2010 15:00
TK 221	11/07/2010 11:40	11/07/2010 12:00

PRIME TIME HEURISTIC

The heuristic approaches given in the previous chapter do not credit for the gate demand distribution throughout the day. But there is relatively considerable accumulation for certain time durations in a whole day. These time intervals can be called *prime time (PT)* of the airports. Gate assignment at these high demand time intervals is harder and more profitable in comparison to the sparse time intervals. The importance of the *prime time* handling for any cost function is the same. However, in this work all the allocation scores and simulation examples will be given for the same objective function that tries to maximize total gate duration.

The characteristics of *prime time* regions is actually a function of airport specific measures, like capacity and number of connection flights leaving; as well as the season specific measures like weather, holidays etc. But, the cause of a particular demand peak is not important since the flight lists are ready for scheduling the gates for at least a day before. Then, according to these flight lists, the airplanes are assigned to the gates by considering the mostly demanded time intervals in certain priority.

Prime time flights can be handled with the following algorithm steps:

1. *Find the most popular time slot:* A plane uses all the time slots between its arrival time slot and departure time slot. Most popular time slot is the one used by the most number of planes. It can be easily found by making a histogram for the time slots.
2. *Find all planes using that time slot:* These are the planes that give a vote to the selected time slot in the histogram count. These planes are called prime time planes.
3. *Sort these planes with respect to staying duration:* Even the prime time planes have a priority between them. Longest staying one is assigned with the first priority.
4. *Assign PT planes starting from the one having the longest gate time:* Each one will be assigned to the current "best" gate. Note that, the first *Ng* score contributing gates are all the same in our examples.
5. *Assign the other planes by idle time minimization algorithm*: The planes before the prime time planes will be assigned by the backward version of the algorithm. On the other hand planes that can be assigned after the prime time region are selected by the forward version of the same algorithm.

Careful inspection of the steps 3 and 5, one can deduce that the prime time method is a hybridized version for the ground time maximization algorithm and the idle time minimization algorithm. The prime time flights are sorted with respect to their ground time as in ground time maximization heuristic and assigned respecting the order. Since all these flights are guaranteed to have at least one common time slot, each flight should be assigned to a different gate because of the basic constraint of the problem. That is to say, if number of flights using the most demanded time slot exceeds the number of score contributing gates, there will be inevitable ungated flights. But the ungated flights are guaranteed to be the shortest staying flights. Then, once the prime time flights are assigned, earlier and later time slots are filled up with idle time minimization routine.

SIMULATIONS

This section illustrates the effect of prior handling of prime time intervals. In the following first subchapter, test data generation parameters are given. In the second subchapter, results over generated data sets are reported. Final subchapter reports the results for real – world data collected from İstanbul Atatürk Airport that is one of the most crowded airports of Europe.

Artificial Data Generation

Following parameters are taken into consideration as inputs in test data generation,

1. the proportion in between total time slot demand and total available discrete time slots, ***d*** : total time slot number can be found by

$$\frac{24*60}{n} \quad (1)$$

where n is the time slot length. Total slot demand is the total gate slots of all planned flights.

2. prime time traffic factors, p_1 and p_2 : are the probability measures that a created plane will be staying on the prime time region. The values are between [0, 1]. If both are 0, then a homogenous gate demand is expected.

3. mean staying time for a plane, ***m***: average of staying durations of all planes in terms of time slot count.

4. standard deviation for staying times of all arranged flights, σ: in terms of time slot count.

Simulation Results

Results for Artificially Generated Data

In the simulations, 5 different dataset characteristics are investigated. These datasets have different aspects and are chosen to comprehensively model various kinds of gate demand. The linguistic definitions and the corresponding specifying parameters used are given as follows:

1. A lot of planes, not enough gates, homogenous demand:
   $d = 1.1, m = 10, p_1 = 0, p_2 = 0, \sigma = 10$

2. A lot of planes, not enough gates, certain peaks on demand
   $d = 1.1, m = 10, p_1 = 0, p_2 = 0.2, \sigma = 10$

3. A lot of planes, not enough gates, increase in demand in patches
   $d = 1.15,\ m = 15,\ p_1 = 0.05,\ p_2 = 0.10,\ \sigma = 10$

4. Not so much planes, enough gates, increase in demand in patches
   $d = 0.75,\ m = 10,\ p_1 = 0.05,\ p_2 = 0.10,\ \sigma = 20$

5. Not so much planes, enough gates, certain peaks on demand
   $d = 0.75,\ m = 10,\ p_1 = 0.15,\ p_2 = 0.2,\ \sigma = 20$

All dataset is composed of 30 files generated with respect to the selected parameters. Number of available (score contributing) gates is 15 in all cases. Time slot length is 5 minutes. The results are given in Table III in a compact manner. With no exception, prime time heuristic results are the best among the three. To emphasize the importance of performance improvements, it is convenient to note here that even a random allocation, or a worst possible allocation, can gather around 3000 points for the datasets below. Because of this, the improvements can not be fairly justified by percentages. Also, these slight differences yield considerable amount of income for airline operations.

Table III – Performance Results of the Heuristic Algorithms for the Artificial Datasets (Average of 30 days)

	Dataset - 1	**Dataset - 2**	**Dataset - 3**	**Dataset - 4**	**Dataset - 5**
Ground Time Maximization	3764.7	3640.9	3623.1	3579.4	3563.1
Idle Time Minimization	3876.8	3716.0	3788.5	3618.3	3616.5
Prime Time	3893.9	3721.9	3808.8	3653.5	3647.2

Results for Data from Atatürk Airport

The experiments in this subchapter are performed on the data collected from the Atatürk Airport in İstanbul. Data are for 31 days and an average of nearly 300 planes a day with certain sharp peaks on the demand. The data itself and the characteristic parameters d, m, p_1, p_2 and σ are deliberately omitted here because of the lack of permission. Although the gate and plane constraints can be handled for this specific airport, as mentioned earlier, the problem at hand is squeezing maximum planes to the first *Ng* gates.

The scores for the allocation are given in Table IV. Newly proposed prime time heuristic clearly outperforms the others. Figure 2 visualizes the scores for each method providing comparison of the algorithms in daily basis.

Table IV – Performance Results of the Heuristic Algorithms for the Real Dataset (Average of 31 days)

Method	Performance Results
Ground Time Maximization	3327.4
Idle Time Minimization	3359.6
Prime Time	3466.9

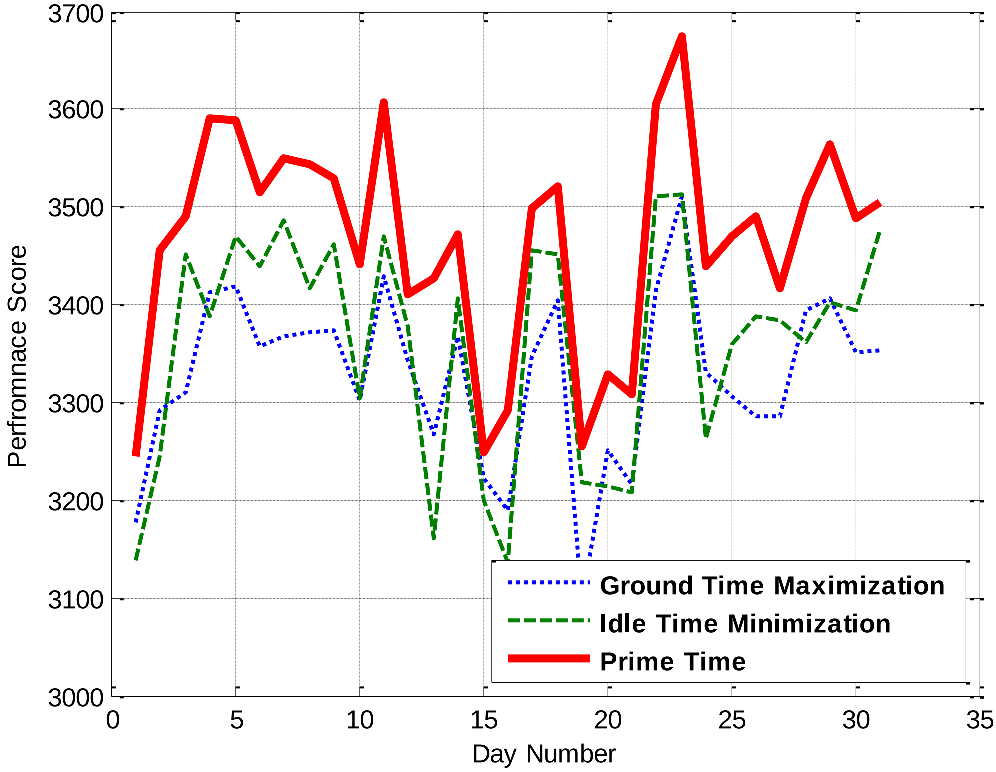

Figure 2 – Comparison of the Heuristic Algorithms.

CONCLUSIONS

In this work airport gate assignment problem is handled with the objective of maximizing total gate time, or in other words, minimizing vacant time intervals. Problem formulation has only the basic constraint imposing only one gate for a flight and one flight for a gate at a particular time instant. Multi time slot model is used throughout the work and 24-hour time interval is discretized with 5 minutes of resolution.

The main contribution of this work is the concept of *prime time*. Prime time of the airports is defined as the relatively high gate demand time intervals during the day. In this work a metric for assigning a degree for the term, *prime term,* is also proposed.

The simulation examples are given both on the artificially generated dataset and the real data set collected from Atatürk Airport. The artificial datasets are created by the dataset generator module designed by the author. The datasets are characterized by five different parameters namely, the proportion in between total time slot demand and total available discrete time slots, two prime time traffic factors, mean staying time for a plane and standard deviation for staying times of all arranged flights.

REFERENCES

Bazargan M. (2004). Airline Operations and Scheduling, Bodmin, Cornwall.

Ding, H., Lim, A., Rodrigues, B., Zhu, Y (2004). Aircraft and gate scheduling optimization at airports. Proceedings of the 37th Hawaii IEEE Int. Conf. on System Sciences, 0-7695-2056-1/04, pp. 1-8

Ding, H., Lim, A., Rodrigues, B., Zhu, Y (2004). New Heuristics for over-constrained Flight to Gate Assignments. Journal of the Operational Research Society, 55, pp. 760 – 768

Ding, H., Lim, A., Rodrigues, B., Zhu, Y (2005). The over-constrained airport gate assignment problem. Computers and Operations Research, 32, pp. 1867 – 1880

Dorndorf, U., Drexl, A., Nikulin Y., Pesch E. (2007). Flight Gate Scheduling: State-of-the-art and Recent Developments, The International Journal Of Management Science, 35, pp. 326 – 334.

Genç, H. M., Erol O. K., Eksin İ. (2009). An Application and Solution to Gate Assignment Problem for Atatürk Airport, DECOMM 2009, Ohrid, Macedonia.

Obata, T. (1979) The quadratic assignment problem: Evaluation of exact and heuristic algorithms. Tech. Report TRS-7901, Rensselaer Polytechnic Institute, Troy, New York.

Zhu, Y., Lim, A., Rodrigues B. (2003). Aircraft and Gate Scheduling with Time Windows. Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI’03), 3-5 November, Sacramento, California, USA.