

A Robust Approach for the Airport Gate Assignment

Liang Xu¹, Fan Wang², and Zhou Xu¹

¹Department of Logistics and Maritime Studies, The Hong Kong Polytechnic University, Hong Kong
{08900198r,lgtx}@polyu.edu.hk

²Department of Management Science, School of Business, Sun Yat-sen University, China
Fanwang@gmail.com

Abstract

Airport gate assignment is a critical issue for the operation management of an airport. Airport gate assignment is to assign flights to gates according to their real-time arrival time and departure time, such that each flight is assigned to exactly one gate, and there is no conflict between two consecutive flights assigned to the same gate. We formulate the airport gate assignment as a stochastic binary integer programming, in which the real-time arrival and departure time are stochastic parameters. Concerning the real-time flight disturbance, a robust approach is introduced to protect the airport gate assignment from flight disturbance such as flight delay or early arrival. Instead of making strong assumption on the distribution of the real-time arrival and departure time of a flight, we assume that they belong to pre-specified uncertainty sets. The robust approach is to make sure that the airport gate assignment is feasible for all possible value for the real-time arrival and departure time within their uncertainty sets. Under these uncertainty sets, we can transform the stochastic binary integer programming to a mix integer programming. The computational results on the real-life test data from Hong Kong International Airport demonstrate that our robust approach can avoid realtime gate conflict efficiently.

Keywords: airport gate assignment, robust optimization, uncertainty set

1. Introduction

Airport gate assignment (AGA for short) is a critical issue for the operation management of an airport. In practice, flights are assigned to airport gates according to their schedule, i.e., their scheduled arrival time and departure time. Table 1 illustrates the scheduled arrival and departure time, and the real-time arrival and departure time for six flights in the Hong Kong International airport.

Table 1: Arrival and departure time of six flights of Hong Kong International Airport

Flight	Arrival	Departure	Real-time Arrival	Real-time Departure	Route	Airline
CA101/102	11:25	12:45	11:23	12:50	Beijing-Hong Kong	Air China
LH738/739	11:30	13:10	11:31	13:30	Frankfurt-Hong Kong	Lufthansa
TG600/601	11:45	12:45	11:50	12:55	Bangkok-Hong Kong	Thai Airway
JL710/702	13:15	15:00	13:15	15:14	Osaka-Hong Kong	Japan Airlines
BR869/870	14:25	15:30	14:23	15:32	Taipei-Hong Kong	EVA Air
SQ862/861	14:20	16:00	14:22	16:10	Singapore-Hong Kong	Singapore Airlines

The airport operation center normally locks the gate for a flight at its arrival time and release the gate at its departure time. Thus, a feasible airport gate assignment must satisfy the following two constraints:

1. Each flight is assigned to exactly one gate.
2. No two flights can be assigned to the same gate concurrently. In other words, if a gate is locked for one flight, it can not serve another flight until it is released.

However, in real-time operation, flight delays or early arrivals often occur. According to the schedules of two flights assigned to the same gate, even though there is no overlap of time durations they occupy the gate, there

may be conflict between the two flights due to flight delay or early arrival. For example, according to the scheduled arrival and departure time, flight LH738/739 and flight JL710/702 can be assigned to the same gate without gate conflict, but due to the flight delay of LH738/739, there is gate conflict in real-time operation. After JL710/702 arrives at Hong Kong International Airport, it is forced to wait on the ramp or even in the air. Therefore, considering the real-time flight disturbance, we should consider the arrival time and departure time of a flight as random parameters rather than deterministic ones, and design a robust airport gate assignment (RAGA for short) framework to protect the system from uncertainty in the operation.

Consider n gates, in which all gates have a uniform service starting time s and service finishing time t ($0 \leq s < t \leq 24$). Consider a flight set $F = \{1, \dots, m\}$ of m flights, in which each flight i has a scheduled arrival time a_i and a scheduled departure time d_i ($s \leq a_i < d_i \leq t$).

Let \tilde{a}_i denote the real-time arrival time for flight i and let \tilde{d}_i denote the real-time departure time for flight i . Accordingly, let $\tilde{lij} = \tilde{a}_j - \tilde{d}_i$ denote the real-time gap between any two flights. We can thus define a feasible solution to the AGA as: n sequences $\{S_1, \dots, S_n\}$ which consist of all elements of F , and each element of F appears exactly once in a sequence; there is no gate conflict between two consecutive flights which are assigned to the same gate, which means, for any two consecutive elements i and j in a sequence, the real-time gap $\tilde{lij} \geq 0$.

Related Work

In the literature, the airport gate assignment problem which is to minimize the total walking distance of customers has been deeply researched. This kind of problem has been studied by Babic et.al [1], Xu and Bailey [10], and Zhu et.al [13] with binary integer program-ming, tabu search, and generic algorithm, respectively. Furthermore, airport gate assignment problem with multiple objective was formulated by Yan et.al [11] as a multi-commodity network flow model, and was solved by Lagrangian Relaxation.

For deterministic airport gate assignment problem, buffer time between two flights was adopted by Hassounah and Steuart [8], and Yan and Chang [11] to avoid real-time flight con- flicts. Buffer time is not flexible enough to address the flight disturbance in the real-time opera- tion. Therefore, Yan and Tang [12] considered stochastic flight delays and provided a gate-flow network model.

Robust airport gate assignment was initiated by Lim and Wang [9]. They modeled the robust airport gate assignment as a stochastic programming model and transformed it into a binary programming model by introducing the un-supervised estimation functions without knowing any information on the real-time arrival and departure time of flights in advance. In their paper, due to the NP-hardness of the graph coloring model, they proposed a hybrid meta-heuristic combining a tabu search and a local search to solve their model.

Recently, Ben-Tal and Nemirovski [2–4] provided a framework for convex optimization prob- lems for which the data is not specified exactly and it is only known to belong to a given uncertainty set. Moreover, Bertsimas and Sim [5, 7, 6] developed a robust approach for linear optimization and extended it for discrete optimization. Based on their work, we can assume that the real-time gap between two flights belongs to a pre-specified uncertainty set and establish our robust approach for the AGA.

Our Results and Significants

The main results of this paper and their significants are as follows:

1. We give a new formulation for the AGA. Compared to the graph coloring model raised in Lim and Wang [9], our model determines not only the flights to be served by a gate but also the order of these flights to be served within a gate.
2. Based on the formulation of the AGA, we further develop the formulation for the RAGA, which takes the uncertainty sets of the random parameters into consideration. Based on the weak duality, we can transform the AGA with random parameters into a mix integer programming.
3. The RAGA model can help the airport operation center to evaluate the gate capacity when there is new flights to be added to the airport.

4. The RAGA model also has high impact to help airlines to estimate how many fixed gates should buy or rent from airport to serve their own flights.

Organization

Section 2 introduces a “general” AGA model defined on a set which consists all the gates and flights. Based on a feasible solution to the “general” AGA, a feasible solution to the AGA can be obtained. In Section 3, the robust airport gate assignment (RAGA) is introduced. In section 4, numerical experiments based on the data collected from Hong Kong International Airport are conducted to show the efficiency of our RAGA model. Finally, this paper is concluded in Section 5.

2. Preliminaries

Since each gate has a service starting time s and a service finishing time t , then it can be considered as a flight with real-time arrival time t and departure time s , which indicates that each flight i can be arranged after the gate if $\tilde{a}^i \geq s$ and each flight i can be arranged before the gate if $\tilde{d}^i \leq t$. Therefore, let us consider a general flight set $F' = \{1, \dots, m, m+1, \dots, m+n\}$ which consists m flights and n gates. For $m+1 \leq i \leq m+n$, we define $\tilde{a}^i = t$ and $\tilde{d}^i = s$. We can then give the formulation of the general airport gate assignment (GAGA).

$$\sum_{i=1}^{m+n} y_{ij} = 1, \forall j \quad (1)$$

$$\sum_{j=1}^{m+n} y_{ij} = 1, \forall i \quad (2)$$

$$\sum_{j=1}^{m+n} y_{ij} \tilde{l}_{ij} \geq 0, \forall i \quad (3)$$

$$y_{ij} \in \{0,1\}, \forall i \forall j \quad (4)$$

in which y_{ij} is the decision variable representing that whether or not flight i is followed immediately by flight j in the same gate. Constraint (1) and (2) indicate that each flight follows one flight and is followed by one flight. Constraint (3) indicates that any two consecutive flights which are assigned to the same gate must have non-negative real-time gap.

Based on a feasible solution $y = \{y_{ij} : 1 \leq i \leq m+n, 1 \leq j \leq m+n\}$ which satisfies constraints (1)-(4), we define a mapping $\sigma : F' \rightarrow F'$, in which $\sigma(i) = j$ if $y_{ij} = 1$. Since y satisfies constraint (1) and (2), it is easy to verify that σ is bijective. We can then construct a feasible solution to the AGA from y by Algorithm 1.

Algorithm 1

Input: A feasible solution y to the GAGA, in which the general flight set consists of m flights and n gates.

Output: A sequence set $S = \{S_1, \dots, S_n\}$ which consists of m flights.

1. Set $i = m+1$.
2. Repeat the following while $i = m+n$.
 - (a) Set $S_i = \emptyset$, set $k = i$
 - (b) Repeat the following until $\sigma(k) \in \{m+1, \dots, m+n\}$:
 - i. Add $\sigma(k)$ to S_i as the last element.
 - ii. Set $k = \sigma(k)$
 - (c) Set $i = i+1$.
 3. Return $S = \{S_1, \dots, S_n\}$.

In Algorithm 1, for each gate i , where $m+1 \leq i \leq m+n$, we find the flight immediately follows i and add it to the sequence S_i , and in each iteration of step 2(b), we find the flight which immediately follows the last element of sequence S_i until the last element of S_i is directly followed by a gate. In the following theorem, one can see that the sequence set $S = \{S_1, \dots, S_n\}$ returned by Algorithm 1 is a feasible solution to the AGA.

Theorem 1. Given a feasible solution y to the GAGA, Algorithm 1 returns a feasible solution to the AGA.

Proof. Since y satisfies constraint (3), we can prove that in each iteration of Step 2(b) of Algorithm 1, the element $\sigma(k)$ added to S_i is different from all elements which have already existed in S_i as follows. By contradiction, suppose $\sigma(k)$ is added after the element t , and $\sigma(k)$ appears before t in the sequence S_i .

According to constraint (3), we have $\tilde{l}_{\sigma(k)t} \geq 0$, i.e., $\tilde{a}_t - \tilde{d}_{\sigma(k)} \geq 0$, and we have $\tilde{l}_{t\sigma(k)} \geq 0$, i.e., $\tilde{a}_{\sigma(k)} - \tilde{d}_t \geq 0$. Since $\tilde{a}_t < \tilde{d}_t$, we can conclude that $\tilde{a}_{\sigma(k)} \geq \tilde{d}_{\sigma(k)}$, which is a contradiction. Since in each iteration of Step 2(b) of Algorithm 1, a new element of $\{1, \dots, m\}$ is added to S_i , there are at most m iterations for each i . Since the mapping σ is bijective and a new element is added to S_i in each iteration of Step 2(b), we can see each element of $\{1, \dots, m\}$ appears exactly once in a sequence. Due to constraint (3), we can see that any two consecutive element i and j in the same sequence satisfies that $\tilde{l}_{ij} \geq 0$. Therefore, the sequence set $S = \{S_1, \dots, S_n\}$ is a feasible solution to the AGA.

Example 1

We use the following example to show how to transform a feasible solution y of the GAGA to a feasible solution of the AGA by Algorithm 1. Consider an airport with two gates and three flights. The general flight set can be constructed as $F' = \{1, 2, \dots, 5\}$, in which flight 4 and 5 in F' represent the two gates. If there is a feasible solution y to the GAGA as listed in Table 2

Table 2: A feasible solution to the GAGA

y_{ij}	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$
$i = 1$	0	0	0	1	0
$i = 2$	0	0	1	0	0
$i = 3$	0	0	0	0	1
$i = 4$	1	0	0	0	1
$i = 5$	0	1	0	0	0

For Algorithm 1, since $m = 3$ and $n = 2$, in step 1, set $i = m + 1 = 4$. In the first iteration of step 2, we find all flights which are served by the first gate. Since $\sigma(4) = 1$, in step 2(b), add 1 to S_1 as the last element and set $k = \sigma(4) = 1$. Since $\sigma(1) \in \{4, 5\}$, then set $i = 5$ and Algorithm 1 goes to the next iteration of step 2. Since $i = 5 = m + n$, then Algorithm 1 goes to step 3 after this iteration. In the second iteration of step 2, we find all flights which are served by the second gate. Similarly, since $\sigma(5) = 2$, add 5 to S_2 as the last element and set $k = 2$. Since $\sigma(2) = 3$, add 3 to S_2 as the last element and set $k = 3$. Since $\sigma(3) = 5 \in \{4, 5\}$, then this iteration of step 2 terminates and Algorithm 1 goes to step 3. In step 3, return $\{S_1 = \langle 1 \rangle, S_2 = \langle 2, 3 \rangle\}$.

3. Robust Airport Gate Assignment

Based on Theorem 1, we can only consider the general flight set F with $|F| = m$ which consists all the gates and flights, and base the robust airport gate assignment on the GAGA. Note that the real-time arrival time \tilde{a}^i and the departure time \tilde{d}^i of the flight i are random parameters. Clearly, if flight i in the general flight set F represents a gate, we have $\tilde{a}^i = t$ and $\tilde{d}^i = s$. Instead of making a strong assumption on the distribution of the real-time gap between two flight, we assume that $\tilde{l}_{ij} = \tilde{a}^j - \tilde{d}^i$ belongs to a symmetric interval $[\bar{l}_{ij} - \hat{l}_{ij}, \bar{l}_{ij} + \hat{l}_{ij}]$, where \bar{l}_{ij} is the mean for \tilde{l}_{ij} , and \hat{l}_{ij} is the worse-case deviation from its mean. Let $\tilde{l}_i = (\tilde{l}_{i1}, \dots, \tilde{l}_{im})$. We can define the uncertainty set L_i for \tilde{l}_i as

$$L_i = \{\tilde{l}_i : \tilde{l}_i \in Z^m, \bar{l}_{ij} - \hat{l}_{ij} \leq \tilde{l}_{ij} \leq \bar{l}_{ij} + \hat{l}_{ij}\} \quad (5)$$

Since L_i is the worse-case uncertainty set for \tilde{l}_i , it is conservative to consider all possible value in this uncertainty set. Instead, let

$$z_{ij} = \frac{\bar{l}_{ij} - \hat{l}_{ij}}{\hat{l}_{ij}}$$

denote the deviation of \tilde{l}_{ij} from its mean \bar{l}_{ij} . Accordingly, let $z_i = (z_{i1}, \dots, z_{im})$. Restrict the norm of z_i as $\|z_i\| \leq \Gamma$. We can obtain a uncertainty set $L_i(\Gamma)$ as follow

$$L_i(\Gamma) = \{\tilde{l}_i: \tilde{l}_i \in Z^m, \bar{l}_{ij} - \hat{l}_{ij} \leq \tilde{l}_{ij} \leq \bar{l}_{ij} + \hat{l}_{ij}, \|z_i\| \leq \Gamma\} \quad (6)$$

Recall that the AGA requires any two consecutive flights which are assigned to the same gate to have non-negative real-time gap. Since the real-time gap \tilde{l}_i is a random vector, under the assumption that \tilde{l}_i belongs to the uncertainty set $L_i(\Gamma)$, our robust approach for the AGA is to ensure that constraint (3) is not violated for any possible value in $L_i(\Gamma)$, which means, for any possible value in $L_i(\Gamma)$, flight i and the flight which follows i have no gate conflict.

Furthermore, since the norm Γ of z_i is the budget of uncertainty, to maximize the reliability of the airport gate assignment is to guarantee that there exists a solution which is feasible for all possible value in the largest uncertainty set. Thus, the goal of the RAGA is to maximize the budget of uncertainty Γ . Therefore, we can define our robust airport gate assignment (RAGA) as follows:

$$\max \Gamma \quad (7)$$

$$\sum_{i=1}^m y_{ij} = 1, \forall i \quad (8)$$

$$\sum_{j=1}^m y_{ij} = 1, \forall j \quad (9)$$

$$\sum_{j=1}^m y_{ij} \tilde{l}_{ij} \geq 0, \forall i, \forall \tilde{l}_i \in L_i(\tilde{\Gamma}) \quad (10)$$

$$y_{ij} \in \{0,1\}, \forall i, \forall j \quad (11)$$

Since $\tilde{l}_{ij} = \bar{l}_{ij} + z_{ij} \hat{l}_{ij}$, then constraint (10) is equivalent to

$$\sum_{j=1}^m y_{ij} \bar{l}_{ij} - \max_{z_i} \sum_{j=1}^m z_{ij} y_{ij} \hat{l}_{ij} \geq 0 \quad (12)$$

$$\|z_i\| \leq \Gamma \quad (13)$$

How to choose the norm $\|\cdot\|$ is closely related to the tractability of the RAGA. By choosing $\|z_i\| = \|z_i\|_1 = \sum_j |z_{ij}|$, we can transform the RAGA to a mix integer programming (MIP).

Theorem 2. *Under the uncertainty set $L_i(\Gamma)$, the RAGA can be transformed into a MIP by replacing constraint (10) as the constraints below*

$$\sum_{j=1}^m y_{ij} \bar{l}_{ij} - \Gamma p_i - \sum_{j=1}^m q_{ij} \geq 0, \forall i \quad (14)$$

$$p_i + q_{ij} \geq \hat{l}_{ij} y_{ij}, \forall i, \forall j \quad (15)$$

$$p_i \geq 0, \forall i \quad (16)$$

$$q_{ij} \geq 0, \forall i, \forall j \quad (17)$$

Proof: Considering the problem

$$\begin{aligned} \max_{z_i} \sum_{j=1}^m z_{ij} y_{ij} \hat{l}_{ij} \geq 0 \\ \|z_i\|_1 \leq \Gamma, \end{aligned}$$

Since $y_{ij} \hat{l}_{ij} \geq 0$, for all $1 \leq i, j \leq m$, it is equivalent to

$$\max_{z_i} \sum_{j=1}^m z_{ij} y_{ij} \hat{l}_{ij} \geq 0,$$

$$\sum_{j=1}^m |z_{ij}| \leq \Gamma$$

$$0 \leq z_{ij} \leq 1 \quad \forall j.$$

Its dual problem is

$$\min (\Gamma p_i + \sum_{j=1}^m q_{ij})$$

$$p_i + q_{ij} \geq \tilde{l}_{ij} y_{ij}, \quad \forall i, \forall j$$

$$p_i \geq 0, \quad \forall i$$

$$q_{ij} \geq 0, \quad \forall i, \forall j.$$

Thus, due to the weak duality, for the RAGA, constraint (12) and (13) are equivalent to constraints (14)-(17) under $\|\cdot\|_1$, which in turns implies that constraint (10) is equivalent to constraints (14)-(17).

In order to construct the RAGA, it is necessary to estimate the parameters \bar{l}_{ij} and \hat{l}_{ij} . Suppose we have N historical data $l_{ij}^{(k)} = a_j^{(k)} - d_i^{(k)}$ for \tilde{l}_{ij} , for $1 \leq k \leq N$. Thus, we can use the sample mean for \tilde{l}_{ij} to estimate \bar{l}_{ij} . Furthermore, according to the 3σ principle, for the random parameter \tilde{l}_{ij} , the probability that \tilde{l}_{ij} deviates from its mean more than three times the variance is less than 0.003. Thus, we can use three times the sample variance for \tilde{l}_{ij} to estimate the worse-case deviation \hat{l}_{ij} .

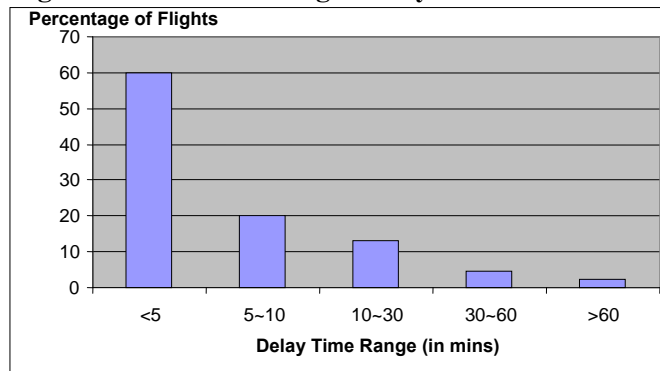
After we specify all the parameters of the RAGA, we can not solve the RAGA directly because both Γ and p_i are decision variables, which indicates that the RAGA is a quadratic programming. It is observed that for any given Γ , the RAGA is a linear programming. Therefore, since Γ belongs to a bounded interval $[0, 1]$, a binary search is conducted to obtain a close optimal value for Γ . The optimal Γ^* is returned as the greatest $\Gamma \in [0, 1]$ such that the RAGA has a feasible solution.

4. Numerical Experiments

The data from Hong Kong International Airport is collected as the test data. The data consists of all records of real-time arrival time and departure time of all flights for 2 weeks. There are 255 flights to be served in the test data. There are in total 48 frontal gates and 27 aprons for Hong Kong International Airport.

Figure 1 shows the distribution of flight delay in the test data. For all records in the 2 weeks, 60% of flights delay less than five minutes, and 20% of flights delay from five to ten minutes, and 13% of flights delay from ten to thirty minutes, and only 4.5% flights delay from half an hour to one hour.

Fig. 1: Distribution of flight delay



We use all records for \tilde{a}^j and \tilde{d}^i to estimate \bar{t}_{ij} and \hat{t}_{ij} , and solve the RAGA by ILOG CPLEX 9.0 to obtain a airport gate assignment. It takes 1.618 seconds for ILOG CPLEX 9.0 to obtain a feasible solution to the RAGA. We use the data in the two weeks to test the efficiency of the airport gate assignment. First, the average gate conflict and stand deviation of gate conflict under our proposed RAGA model are calculated. Second, we solve the AGA model according to the scheduled arrival time and scheduled departure time, and calculate the average gate conflict and the standard deviation of the gate conflict. Figure 2 compares the average gate conflict of these two approaches, and Figure 3 compares the standard deviation of gate conflict of these two approaches.

From these two figures, we can conclude that: our proposed RAGA model can develop robust airport gate assignment to deal with real-time flight disturbance. Only 60 gates are sufficient for the airport to avoid real-time gate conflict for most of the test data while there are in total 78 gates in the Hong Kong International Airport. When we solve the AGA according to the scheduled arrival and departure time of all flights, the the real-time flight disturbance is ignored.

Fig. 2. Average of gate conflict of RAGA and AGA (schedule)

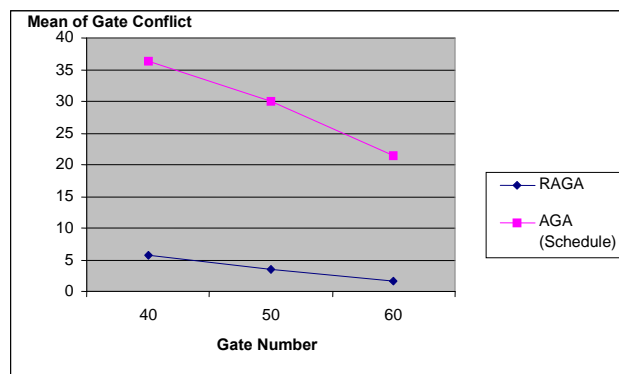
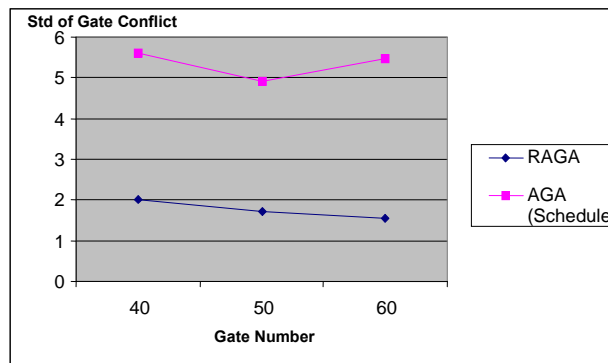


Fig. 3. Standard deviation of gate conflict of RAGA and AGA (schedule)



Our proposed RAGA model performs better than AGA model without considering uncertainty. Moreover, when the RAGA model is restricted on flights belonging to a airline, the minimum number of gates which can serve all the flights without gate conflict can be estimated. Thus, airlines can evaluate the minimum number of gates they should buy or rent from the airport according to our proposed RAGA model. On the other hand, our proposed RAGA model can be used to evaluate the gate capacity when there are new flights to be added in the airport.

In addition, we will compare our proposed RAGA model and the approach proposed by Lim and Wang [9]. They define $p(i, j)$ as the probability distribution function on gate conflict between two flights i and j if they are assigned to the same gate. In their work, they assign flights to airport gate without determining the sequence of all flights assigned to the same gate. Therefore, they formulated the airport gate assignment as a graph coloring model, in which all flights are colored by different colors. Flights with the same color are assigned to the same gate, and the total expected probability of gate conflict within all the flights assigned

to the same gate is calculated. The goal is to minimize the total expected probability of gate conflict for all the gates. Since the graph coloring model does not determine the sequence of all flights assigned to the same gate, the expected probability of gate conflict is calculated by summing up the expected probability of gate conflict for every two flights assigned to the same gate. However, for airport gate assignment, the probability of gate conflict between two consecutive flights is much more important. The drawback of the graph coloring model lies in that the objective function does not estimate the expected probability of the gate conflict accurately. Therefore, we combine the definition of the $p(i, j)$ and our AGA model to establish the following stochastic programming model for the robust airport gate assignment which minimizes the total expected gate conflict probability (RAGA-Lim and Wang):

$$\max \sum_{i,j} E(p(i,j)) y_{ij} \quad (18)$$

$$\sum_{i=1}^m y_{ij} = 1 \quad \forall i \quad (19)$$

$$\sum_{j=1}^m y_{ij} = 1 \quad \forall j \quad (20)$$

$$\sum_{j=1}^m y_{ij} \tilde{l}_{ij} \geq 0 \quad \forall i, \forall \tilde{l}_i \in L_i(\Gamma) \quad (21)$$

$$y_{ij} \in \{0,1\} \quad \forall i \forall j \quad (22)$$

It is noticed that for this model, the objective function consists of the expected probability of gate conflict for consecutive flights which are assigned to the same gate.

Lim and Wang introduced an estimation function $e(i, j)$ to estimate the expected value of the probability of the gate conflict between flight i and j based on lij . This un-supervised estimation function $e(i, j)$ is only based on the scheduled time gap between flight i and j without considering the historical data. In their work, they set $e(i, j) = \exp(-\beta lij)$, where $\beta = -0.03$, and they set

$$E(p(i, j)) = \frac{\max\{s(i,j)\} - s(i,j)}{\max\{s(i,j)\} - \min\{s(i,j)\}}$$

Figure 4 compares the average gate conflict of these two approaches, and Figure 5 compares the standard deviation of gate conflict of these two approaches.

Fig. 4: Average of gate conflict of RAGA and RAGA-Lim and Wang

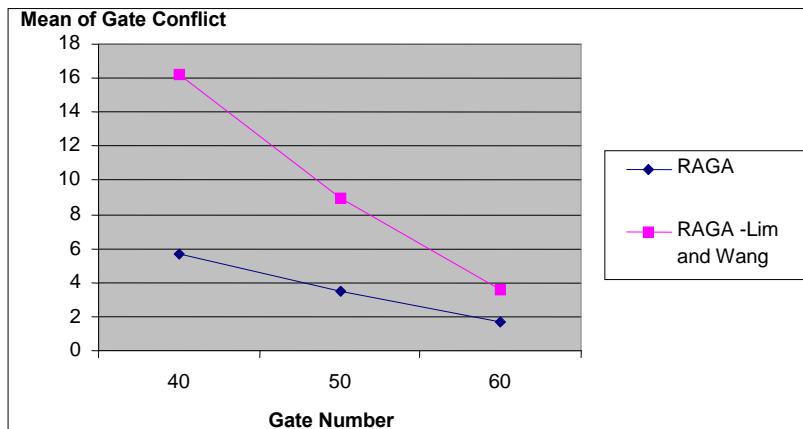
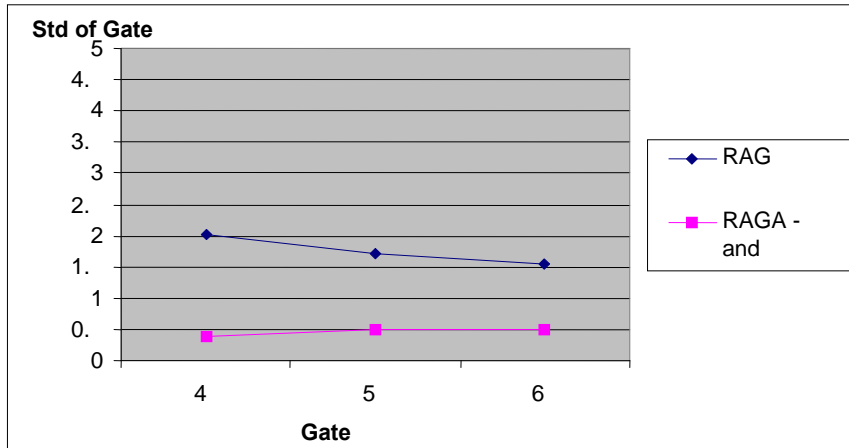


Fig. 5: Standard deviation of gate conflict of RAGA and RAGA-Lim and Wang



From these two figures, we can conclude that: both of the two approaches have small standard deviation which indicates that these two approaches have stable performance. Moreover, our proposed RAGA model has a better average gate conflict than RAGA-Lim and Wang, especially when the number of gate is small.

5. Conclusion

In this paper, we proposed a new robust approach for the airport gate assignment, which can deal with the real-time flight disturbance. Computational results on real-life data demonstrate that the proposed RAGA model can efficiently avoid gate conflict.

Since the worse-case deviation is to some extent conservative for the uncertainty set of the real-time gap between two flights, in practice, more related features such as weather, air traffic control, peak time should be considered for a more accurate estimation of the uncertainty set for the real-time gap. Our future research will focus on constructing more accurate estimation of the uncertainty set for the real-time gap.

References

- O. Babić and D. Teodorović. Aircraft stand assignment to minimize walking. *Journal of Transportation Engineering*, 110:55, 1984.
- A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.
- A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–14, 1999.
- A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424, 2000.
- D. Bertsimas, D. Pachamanova, and M. Sim. Robust linear optimization under general norms. *Operations Research Letters*, 32(6):510–516, 2004.
- D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71, 2003.
- D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004.
- MI Hassounah and GN Steuart. Demand for aircraft gates. *Transportation Research Record*, 1423:26–33, 1993.
- A. Lim and F. Wang. Robust airport gate assignment. *17th IEEE International Conference on Tools with Artificial Intelligence*, pages 74–81, 2005.
- J. Xu and G. Bailey. The airport gate assignment problem: Mathematical model and a tabu search algorithm. *34th Hawaii International Conference on System Sciences*, 3:3032–3040, 2001.
- S. Yan and C.M. Chang. A network model for gate assignment. *Journal of Advanced Transportation*, 32(2):176–189, 1998.
- S. Yan and C.H. Tang. A heuristic approach for airport gate assignments for stochastic flight delays. *European Journal of Operational Research*, 180(2):547–567, 2007.
- Y. Zhu, A. Lim, and B. Rodrigues. Aircraft and gate scheduling with time windows. In *15th IEEE International Conference on Tools with Artificial Intelligence, 2003. Proceedings*, pages 189–193, 2003.