

Robust Airport Gate Assignment

Andrew Lim and Fan Wang

Department of Industrial Engineering and Logistics Management,
The Hong Kong University of Science and Technology,
Hong Kong, China

Corresponding author: Dr. Fan Wang (fanwang@ust.hk)

Abstract

In this paper, we propose a new strategy for the robust constraint resource assignment problem and apply it to solve the Robust Airport Gate Assignment (RAGA). RAGA attempts to accurately build an evaluation criteria for the ability of an aircraft-to-gate assignment to handle uncertainty on aircraft schedule; and to accurately and effectively search the most robust airport gate assignment. We model the RAGA by a stochastic programming model and transform it into a binary programming model by introducing the unsupervised estimation functions without knowing any information on the real-time arrival and departure time of aircrafts in advance. Moreover, a partition-based search space encoding, two neighborhood operators for single or multiple aircrafts reassignment, and a hybrid meta-heuristic combining a tabu search and a local search are proposed to solve RAGA efficiently. Experimental results on the real-life test data from Hong Kong International Airport demonstrate that the proposed RAGA model provides a valuable tool for the airport to improve its robustness in uncertain operations.

1 Introduction

Efficient airport operation depends in part on gating aircraft for a smooth flow of arriving and departing flights. Figure 1 illustrates the layout of airport gates within the premises of Hong Kong International Airport with 48 frontal gates and 27 aprons in total. Each aircraft will be assigned to a frontal or apron gate after it lands providing there is a suitable gate available for it. Otherwise, the aircraft will be forced to wait on the ramp or even in the air, sometimes for a long period of time. An aircraft has three types of departure after finishing the current flight: stops at the same gate

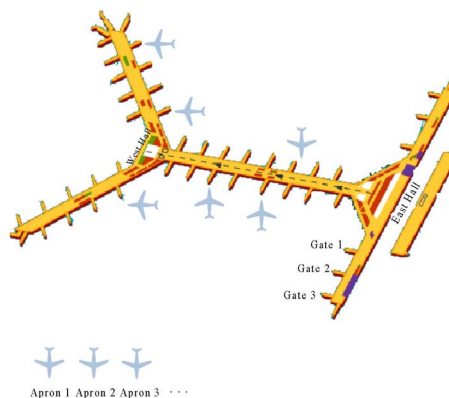


Figure 1. Outline of Hong Kong International Airport

for immediately taking on its next flight to other destination, or temporally moves to an available apron, or enters the warehouse for maintenance. For safety and security, the airport operation center normally helps aircraft to lock the gate assigned for it earlier than its arriving time and to release the gate later than its departure time.

The *airport gate assignment problem*, whose purpose is to assign a suitable gate for each aircraft, is one of core components in the field of airport resource management [3]. In addition, although some major airports currently suffer from capacity constraints at the ramps or the adjacent airspace rather than the gate capacity, the airport gate assignment problem has also high-impact to help airlines estimate on how many fixed gates should buy or rent from airports to server their own aircrafts. The airport gate assignment problem can be considered as a case of the *constraint resource assignment problem* where the gates are *resources* and aircrafts are *resource consumers* under the following

two hard constraints:

1. single: every aircraft must be assigned to one and only one gate;
2. feasible: no two aircrafts can be assigned to the same gate concurrently. In other words, if a gate is locked for one aircraft, it cannot be assigned for another aircraft until it has been released.

In addition, both airports and airlines consider several another soft constraints in real practice to improve their image and service quality, such as gating aircrafts by specified gates, gating aircrafts by frontal gates rather than aprons, minimizing the number of aircraft forced to wait, and minimizing the total walking distances of passenger transferring.

In the body of available literature, airport gate assignment problem on minimizing passengers' walking distance has been well studied by linear binary programming [2], tabu search [6], memetic algorithm [10], and multi-objective programming [8]. However, all the above published research only focus on *static modelling* that approaches optimization without considering any uncertainty of aircraft schedule (such as flight delay or early arrival) that is quite common in real-life airports. Therefore, airport decision support systems solely based on the above aircraft-to-gate optimization cannot accurately be performed in real-life environment. To solve the mismatch between the planned schedule and the on-line schedule, three strategies have been studied in the published research:

1. reassignment: upon the decision support system receives an schedule modification from an aircraft, the reassignment component will reassign relative aircrafts to alternative gates based on the current aircraft-to-gate assignment. Gu and Chung have developed a genetic algorithm [4] for reassignment. However, reassignment is time-consuming and not effective to apply to a real-time system;
2. buffer time: Hassounah and Steuart showed that inserting a large buffer time could improve schedule punctuality in flight operation [5]. Yan and Change inserted a fixed buffer time between two continuous flights assigned to the same gate to absorb the stochastic flight delays [7]. However, the length of the fixed buffer time is hard to determine and such an uniform length possibly results in inferior operating performance.
3. simple on-line assignment rules: the system assigns an alternative gate to any delayed aircraft upon the arrival of the aircraft by simple rules if its original assigned gate has been locked. Yan *et*

al. studied a simulation framework and compared various on-line assignment rules [9]. Although this method is simple and fast to implement, airports cannot publish the aircraft-to-gate assignment information until the aircraft had arrived and therefore this strategy definitely affects other airport operations and the service qualities of both airports and airlines.

In this paper, we attempt real-time airport gate assignment by building a robust aircraft-to-gate assignment based on the planed schedule to minimize the number of gate-reassigned aircrafts. It is an entirely different strategy to the published research. We first model the *Robust Airport Gate Assignment* (RAGA) by a stochastic programming model. Then, we introduce the estimation function on gate conflict to transfer the stochastic programming model into a binary programming model. Since solving the binary programming model is NP-hard and it is also impossible to find an accurate lower bound or upper bound, we propose a hybrid meta-heuristic integrating a tabu search and a local search to solving the RAGA effectively in the decision support system development, rather than directly solving the binary programming model by techniques on Operations Research or Constraint Programming. Experimental results on real-life data collected from Hong Kong International Airport demonstrate that the RAGA model efficiently well handles uncertainty. Moreover, our proposed hybrid meta-heuristic solves the RAGA efficiently. In fact, our proposed RAGA model with its hybrid meta-heuristic solution can be directly applied to produce robust solutions for other cases of the constraint resource assignment problem.

2 Model

Notations:

Resources (Gates Set): $G = \{g_1, g_2, \dots, g_c\}$ where c is the number of gates;

Consumers (Aircrafts Set): $F = \{f_1, f_2, \dots, f_n\}$ where n is the number of aircrafts. For each aircraft f_i ($1 \leq i \leq n$):

- a_i : scheduled arriving time;
- d_i : scheduled departure time;
- a'_i : real arriving time;
- d'_i : real arriving time;

Airport Configurations:

b : buffer time between two continues aircrafts assigned to the same gate. In other words, the gate is

locked for serving aircraft f_i in $[a'_i - b, d'_i + b]$;

Decision Variables:

$x_{ik} = 1$ if we assign aircraft f_i to gate c_k ; otherwise $x_{ik} = 0$ ($1 \leq i \leq n, 1 \leq k \leq c$). We also define the auxiliary variables $y_{ij} = 1$ if $\exists k, x_{ik} = x_{jk} = 1$; Otherwise, $y_{ij} = 0$ ($1 \leq i, j \leq n$).

DEFINITION: Gate Conflict

Two aircrafts f_i and f_j have gate conflict if both the following two conditions hold:

- aircraft f_i and f_j are assigned to the same gate, e.g., $y_{ij} = 1$;
- there is an overlap between the two time durations during which the aircraft will lock the gate, e.g., $[a'_i - b, d'_i + b] \cap [a'_j - b, d'_j + b] \neq \emptyset$;

RAGA aims at minimizing the number of gate conflicts that depends on the gate assignment and the on-line aircraft arrival and departure time. Since aircraft on-line arrival and departure time are unknown exactly in advance, we build an estimation model for on-line arrival and departure time and formulate the RAGA by a stochastic programming model. We define $p(i, j)$ is the probability distribution function on gate conflict between two aircrafts f_i and f_j if they would use the same gate. Then a stochastic programming model for the RAGA is formulated as follows.

$$\min R = \sum_{1 \leq i < j \leq n} (E(p(i, j))y(i, j)) \quad (1)$$

subject to

$$\sum_{k=1}^c x_{ik} = 1, \forall 1 \leq i \leq n \quad (2)$$

$$\text{Additional Soft Constraints} \quad (3)$$

$$x_{ik} \in \{0, 1\}, \forall 1 \leq i \leq n, \forall 1 \leq k \leq c \quad (4)$$

Equation (1) indicates the objective function. Equation (2) guarantees that each aircraft is assigned to one and only one gate. Equation (3) represents various additional soft constraints in real practice, such as that the specified aircrafts should be assigned to the VIP gate. Equation (4) represents the decision variables.

One challenge in solving the above stochastic model is how to define $p(i, j)$ to approach a'_i and d'_i by the known information a_i and d_i solely. Alternatively, we go another way to estimate $E(p(i, j))$ directly. We first define $l(i, j) = a_j - d_i - 2b$ as the gap of gate locking time between two aircrafts f_i and f_j , assuming $a_j < d_i$. Hence, $l(i, j) > 0$ means no conflict between f_i and f_j on the schedule. Then, we introduce an estimation

function, $e(i, j)$, to estimate the mean of the probabilities of the gate conflict between f_i and f_j based on $l(i, j)$, on an obvious assumption that the larger gaps of gate locking results in the smaller probability of real gate conflict, i.e., $e(i, j) < e(s, t)$ if and only if $l(i, j) > l(s, t)$. Finally, we apply Equation (5) to obtain $E(p(i, j))$ by normalizing $e(i, j)$:

$$E(p(i, j)) = \frac{\max \{e(i, j)\} - e(i, j)}{\max \{e(i, j)\} - \min \{e(i, j)\}} \quad (5)$$

Definition of $e(i, j)$ actually depends on the application domains and historical data analysis. In our implementation for the robust airport gate assignment, we test the following four kinds of un-supervised estimation functions.

1. Inverse function:

$$e^1(i, j) = \begin{cases} \frac{b}{l(i, j) + b} & \text{if } l(i, j) > 0, \\ 1 & \text{otherwise.} \end{cases};$$

2. Linear function:

$$e^2(i, j) = -l(i, j);$$

3. Sublinear function:

$$e^3(i, j) = \begin{cases} \cos(\gamma\pi l(i, j)) & \text{if } l(i, j) > 0, \\ 1 & \text{otherwise.} \end{cases}$$

$$\text{where } \gamma = \frac{1}{\max \{l(i, j)\}};$$

4. Exponential function: $e^4(i, j) = \exp(-\beta l(i, j))$

where β is a fixed constant parameter to control the curvature rate of the exponential function.

We compare the characteristics of $E(p(i, j))$ in Figure 2, where the time unit is minute and parameters are set as $b = 15$ and $\beta = 0.05$. If two aircrafts f_i and f_j have gate conflict already ($l(i, j) \leq 0$) subject to their estimated arrival and departure time, their probability of gate conflict is fixed to one. In contrast, the probability of gate conflict is measured by the gap on the gate locking time between two aircrafts. Generally speaking, with the increasing of the gap, the mean of the probability of gate conflict decreases slowly by the sublinear function; linearly by the liner function; rapidly by the inverse function; and significantly by the exponential function.

Now, we are going to illustrate a numerical example of the RAGA. We consider an airport with three gates and a schedule of six aircrafts shown in Table 1.

To apply the proposed RAGA model, we first calculate the matrix of $E(p(i, j))$ (we set $b = 15$ and use the exponential function with $\beta = 0.03$ for illustration). Then, we can assess the robustness of an aircraft-to-gate assignment and search the most robust assignment. For instance, Figure 3 and Figure 4 illustrate two feasible assignments, where a rectangle along

Table 1. Schedule in the numerical example

Aircraft	Est. Arrival	Est. Departure	Route	Airline
CA101/102	11:25	12:45	Beijing - Hong Kong - Beijing	Air China
LH738/739	11:30	13:25	Frankfurt - Hong Kong - Frankfurt	Lufthansa
TG600/601	11:45	12:45	Bangkok - Hong Kong - Bangkok	Thai Airway
JL710/702	13:15	15:00	Osaka - Hong Kong - Osaka	Japan Airlines
BR869/870	14:25	15:30	Taipei - Hong Kong - Taipei	EVA Air
SQ862/861	14:20	16:00	Singapore - Hong Kong - Singapore	Singapore Airlines

Table 2. Matrix of $E(p(i, j))$ for the numerical example

	CA101/102	LH738/739	TG600/601	JL710/702	BR869/870	SQ862/861
CA101/102	-	0.47	0.30	0.04	0.00	0.00
LH738/739	-	-	1.00	0.16	0.01	0.02
TG600/601	-	-	-	0.05	0.01	0.01
JL710/702	-	-	-	-	0.35	0.41
BR869/870	-	-	-	-	-	1.00
SQ862/861	-	-	-	-	-	-

Note: $b = 15, \beta = 0.03$

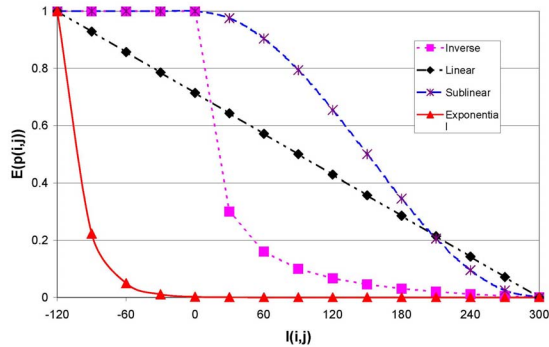


Figure 2. Illustration of $E(p(i, j))$ on various estimation functions

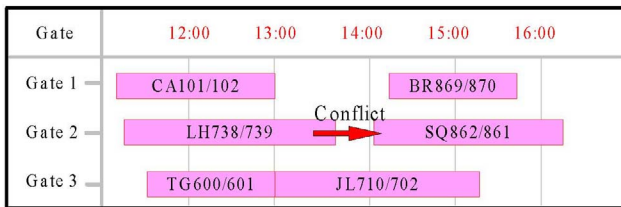


Figure 3. Illustration of Solution 1

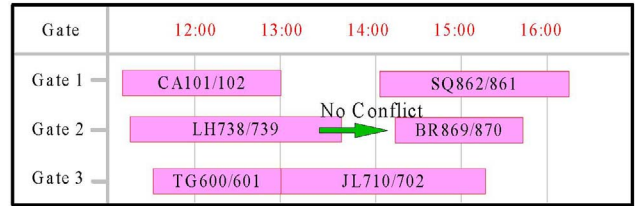


Figure 4. Illustration of Solution 2

the time axis represents the gate locking duration for an aircraft. In Solution 1, CA101/102 and BR869/870 are assigned to Gate 1; LH738/739 and SQ862/861 are assigned to Gate 2; and TG600/601 and JL710/702 are assigned to Gate 3. Different to Solution 1, SQ862/861 is assigned to gate 1 while BR869/870 is assigned to Gate 2 in Solution 2. Therefore, the robustness of Solution 1 and Solution 2 could be calculated by Equation 1 based on the matrix of $E(p(i, j))$.

$$R(\text{Solution1}) = E(p(1, 5)) + E(p(2, 6)) + E(p(3, 4)) = 0.00 + 0.02 + 0.05 = 0.07$$

$$R(\text{Solution2}) = E(p(1, 6)) + E(p(2, 5)) + E(p(3, 4)) = 0.00 + 0.01 + 0.05 = 0.06$$

$R(\text{Solution2}) < R(\text{Solution1})$ indicates that Solution 2 is more robust than Solution 1 to handle uncertain aircraft delay and early arrival. In the scenario that aircraft LH738/739 delays 40 minutes

to departure (13:25 is modified to 14:15), Solution 1 results in one gate conflict on Gate 1. In contrast, Solution 2 can handle such a delay without any gate conflict.

3 Hybrid Meta-heuristic

RAGA is NP-hard and there is no literature on the approximation solutions for solving it to our knowledge (the NP-hardness proof is omitted in this version). Now we are going to propose a hybrid meta-heuristic pipelining a tabu search and a local search to solve the RAGA, with a new partition-based search space encoding, two single and multiple gates reassignment based neighborhood operators.

We first present the partition-based encoding to construct the search space for solving the RAGA. The aircrafts set are partitioned into c subsets and each subset implies one gate. In other words, a solution of the RAGA can be represented as $F = \{F_1, F_2, \dots, F_c\}$ where $f_i \in F_k$ means that aircraft f_i is assigned to gate g_k . For instance, Solution 1 for the numerical example in Section 2 is represented as $\{\{f_1, f_5\}, \{f_2, f_6\}, \{f_3, f_4\}\}$ and Solution 2 is represented as $\{\{f_1, f_6\}, \{f_2, f_5\}, \{f_3, f_4\}\}$.

We then develop two operators for the neighborhood construction: single aircraft reassignment and multiple aircrafts reassignment by improvement graph. The former simply reassigns only one aircraft to a random alternative gate each time. In contrast, the later finds multiple aircrafts and cyclically exchanges their gates.

In multiple aircraft reassignment, we build an improvement graph to search which aircrafts will be reassigned and how to reassign them. Each solution of the RAGA is transformed by the following rules to generate an improvement graph (V, E) :

- Vertices set: $V = F = \{f_1, f_2, \dots, f_n\}$;
- Directed arcs set: $(i, j) \in E$ if and only if $f_i \in F_k$ and $f_j \in F_{k'}$ for all $k \neq k'$;
- Weight of each arc: $w(i, j)$ is defined as the reduction (positive or negative) of R when we clean the original assigned gate of f_i and reassign the gate of f_j to f_i .

We define an *exchange-cycle* in an improvement graph as a simple cycle that links multiple distinct vertices by successive arcs and satisfies that the two aircrafts represented by any pair of vertices in the cycle receive different gates. Therefore, an exchange-cycle means an aircraft-to-gate reassignment. Equivalently, if the exchange cycle is $(f_{i_1} \rightarrow f_{i_2} \rightarrow \dots \rightarrow f_{i_j} \rightarrow$

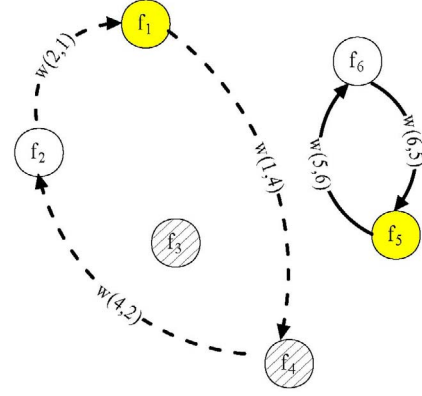


Figure 5. Two exchange-cycles in the improvement graph of Solution 2

$\dots \rightarrow f_{i_1})$ and $C(f_{i_j})$ ($C'(f_{i_j})$) represents the original (new) gate for aircraft f_{i_j} , the gate reassignments are $C'(f_{i_{j+1}}) = C(f_{i_j})$. For instance, in Figure 5, we illustrate two exchange-cycles in the improvement graph according to Solution 1. The exchange-cycle denoted by the real line ($f_5 \rightarrow f_6 \rightarrow f_5$) exchanges the gates of f_5 and f_6 , e.g., Solution 1 becomes Solution 2. Another exchange-cycle denoted by the dashed line ($f_1 \rightarrow f_4 \rightarrow f_2 \rightarrow f_1$) transforms Solution 1 to another solution $\{\{f_4, f_5\}, \{f_1, f_6\}, \{f_2, f_3\}\}$.

In addition, the sum of arcs weights along the exchange-cycle implies the total reduction of R after the gate reassignment according to the exchange-cycle. Obviously, the exchange-cycle with the maximum reduction of R is considered as the best operator of multiple aircrafts gate assignment. However, seeking the best exchange-cycle is NP-complete since it is the same as the *graph longest path problem*. To efficiently seek a near-optimal exchange-cycle, we use the modification algorithm based on the well-known label-correcting approach for the graph shortest path problem [1].

A hybrid two-stage meta-heuristic algorithm solves the RAGA. The first stage is a tabu search by operator single aircraft reassignment to generate an accurate initial solution. We set two tabu lists: *TabuF* and *TabuFG*. One dimension *TabuF* defines the tabu memory for reassignment of each aircraft. During the tabu search, if the current iteration index is smaller than $TabuF[i]$, aircraft f_i is tabued to be reassigned to any alternative gate in this iteration. On the other hand, two dimension *TabuFG* defines the tabu memory for aircraft-to-gate. If the current iteration index is smaller than $Tabu[i, k]$, the aircraft f_i is tabued to be reassigned to the gate g_k in this iteration. The

tabu search proceeds iteratively from one solution to another solution by single aircraft reassignment until the stopping condition meets. In each iteration, if a RAGA solution is found to be better than the current best solution, the current best solution will be updated and the reassignment action to the relevant aircraft and gate will be tabued. In the second stage, a local search by operator multiple aircraft reassignment by improvement graph improves the above initial solution. Generally, the different impacts of the two neighborhood operators motivated us to build such a two-stage approach. Single aircraft assignment can modify the number of aircrafts assigned to each gate but involves only one aircraft each time. In contrast, multiple aircrafts reassignment keeps the number of aircrafts assigned to each gate unchanged while involving multiple aircrafts.

4 Computational Results

A full day real-life schedule of Hong Kong International Airport in August 2004 is collected as test data to evaluate the performance of our proposed RAGA model. 285 aircrafts requested gates in the test data. We counted up gate conflicts by checking each aircraft based on its real-time arrival and departure time. Figure 6 shows the distribution of flight delay in the test data. In all 285 aircrafts, 2% of them early arrived by more than ten minutes. If we call an aircraft on-time if the time difference between its on-line arrival (departure) and its scheduled arrival (departure) does not exceed ten minutes, 59% of aircrafts are on-time. For delay, 21% of aircrafts delayed from five to ten minutes; 13% of aircrafts delayed from ten to thirty minutes; and 4.5% of aircrafts delayed from half to one hour. Moreover, five aircrafts delayed more than one hour, where the longest delay time is three hours and thirty-four minutes.

We import the scheduled arrival time and schedule departure time of all aircrafts into the RAGA model and solved the RAGA by the proposed hybrid meta-heuristic. We first compare our proposed meta-heuristic with a Branch-and-Bound which directly solves the integer programming model of the RAGA in Section 2. The Branch-and-Bound was implemented by the commercial state-of-the-art integer programming solver, ILOG CPLEX 8.0. We set the same stopping conditions for the tabu search and the local search in the proposed hybrid meta-heuristic where the search is stopped when there are 1000 continuous iterations without improvement. For the Branch-and-Bound by CPLEX, we stop the search after running time of two hours. Computational results shows

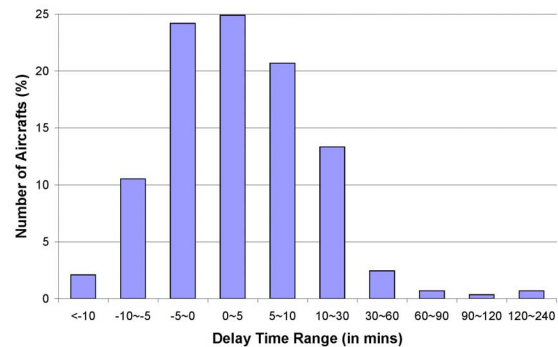


Figure 6. Distribution of aircraft delay in test data

that the hybrid meta-heuristic efficiently solves the RAGA with more accuracy and significantly faster running time (within two minutes) than those of the Branch-and-Bound. (because of the page limit, the detail computational results are omitted in this paper).

The performances of the RAGA by the four estimation functions are compared in Figure 7. Generally, the number of gate conflicts decreased with the increasing of the number of gates. The exponential function outperformed the others, which controlled the number of gate conflicts below five even for the scenario of five gates available. Performances of the inverse function and the linear function were close to that of the exponential function, except that the linear function had poor ability for scenarios of small number of gates. The sublinear function achieved the worst performance, especially for scenarios with 40 or less gates. However, if the number of gates increased to more than 40, performance of the sublinear function becomes remarkable. From the above results, we conclude that:

1. Our proposed RAGA model can produce robust assignments to deal with uncertainty on aircraft schedule. Only 44 gates are sufficient for the test data to avoid any gate conflict by the RAGA model on any of the four estimation functions. In contrast, Hong Kong International Airport used 52 gates in total to assign all 285 aircrafts at that day. Actually, the curves of the number of gate conflicts vs. the number of gates in Figure 7 provides us an useful tool for decision support on how many gates should be prepared. Airlines can also evaluate how many fixed gates should buy or rent from the air-

port to serve its own aircrafts by this tool. On the other hand, the tool can assess airport capacity, especially when a huge number of temporal additional flights will be added into the schedule for peaks such as Christmas Day and Chinese Lunar New Year. In our experiments, we also tested the performance of the RAGA model by keeping the number of gates in 44 while randomly generating 50 additional aircrafts to the schedule. The result shows that the gate assignment produced by the RAGA model only led to 3 extra gate conflicts for the 50 additional aircrafts.

2. The performance of our RAGA model depends on the estimation function when the number of gates is small. For instance, in figure 7, RAGA with the exponential function provided the best robust assignment for scenarios with small numbers of gates. Because we selected the unsupervised estimation functions for the RAGA in our implementation, we did not need to analysis the data behaviors or train the estimation function by the historical data in advance. In fact, our research proposed in this paper focus on the general robustness modelling and the hybrid meta-heuristic solution rather than the estimation function, since the performance of the estimation function depends on the application domains and data behaviors.

In addition, to evaluate the impact of parameter setting in the estimation function, we tested the number of gate conflicts vs. the number of gates by the exponential estimation function on different β for $\beta = 0.05, 0.35, 0.65$. We show the standard deviations of the number of gate conflicts on different β in Figure 8. Clearly, the impact of β is limited. In other words, we can apply the proposed RAGA model by the exponential estimation function without tuning parameter β since parameter tuning is definitely a hard and sometimes high-risk task in real-life system development.

5 Conclusion

In this paper, we have proposed a new strategy for robust resource assignment problem and applied it to build a robust aircraft-to-gate assignment model to maximize its ability to handle real-life uncertainty on aircraft schedule. The proposed RAGA model not only minimizes the number of reassigned gates, but also enables airports to publish gating schedule as early as possible, compared with other research work. RAGA is first formulated by a stochastic programming model and simplified into a binary programming model by

introducing the estimation function on the gate conflict. Computational results on real-life data demonstrate that the proposed RAGA model and its hybrid meta-heuristic solution method efficiently improve the airport robustness.

Future research will focus on constructing accurate estimation function by considering much more related features such as peak and off-peak, on-time rate of different airlines. Another direction is data-driven supervised modelling and training for estimation function. In addition, the RAGA model can be also directly applied to solve other real applications of the constraint resource assignment problem.

References

- [1] R. Ahuja, J. Orlin, and D. Sharma. multi-exchange neighborhood structures for the capacitated minimum spanning tree problem. *Mathematical Programming - A*, 91:71–97, 2001.
- [2] O. Babic, D. Teodorovic, and V. Tosic. Aircraft stand assignment to minimize walking. *Journal of Transportation Engineering*, 110(1):55–66, 1984.
- [3] A. Chun, S. Chan, F. Tsang, and D. Yeung. Hkias: A constraint-based airport stand allocation system developed with software components. In *Proceeding of AAAI/IAAI 1999*, pages 786–793, 1999.
- [4] Y. Gu and C. Chung. Genetic algorithm approach to aircraft gate reassignment problem. *Journal of Transportation Engineering*, 125:384–389, 1999.
- [5] M. Hassounah and G. Steuart. Demand for aircraft gates. *Transportation Research Record*, 1423:26–33, 1993.
- [6] J. Xu and G. Bailey. The airport gate assignment problem: mathematical model and a tabu search algorithm. In *Proceedings of the 34th Hawaii International Conference on System Sciences*, 2001.
- [7] S. Yan and C. Chang. A network model for gate assignment. *Journal of Advanced Transportation*, 32:176–189, 1998.
- [8] S. Yan and C. Huo. Optimization of multiple objective gate assignments. *Transportation Research - A*, 35:413–432, 2001.
- [9] S. Yan, C.-Y. Shieh, and M. Chen. A simulation framework for evaluating airport gate assignments. *Transportation Research - A*, 36:885–898, 2002.
- [10] Y. Zhu, A. Lim, and B. Rodrigues. Airport gate scheduling with time windows. *Artificial Intelligence Review*, In press.

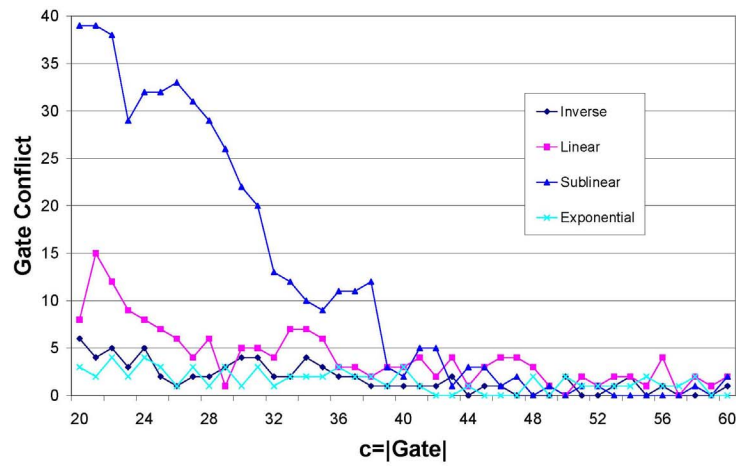


Figure 7. Gate conflicts v.s. various estimation functions

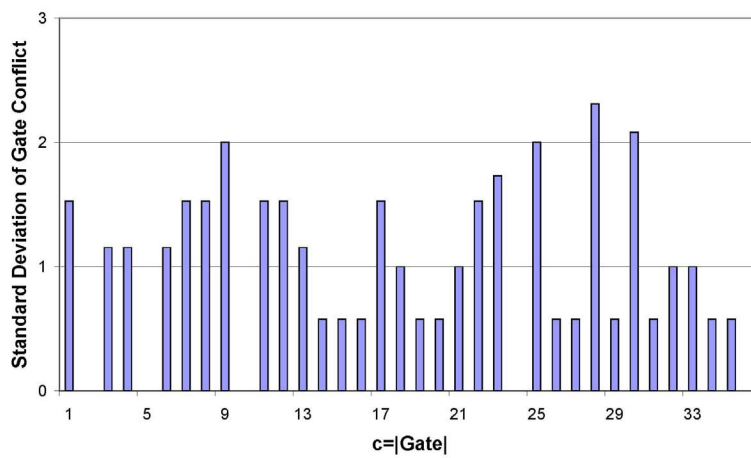


Figure 8. Standard Deviation of |Gate Conflicts|