

Research Article

Ant Colony Algorithm and Simulation for Robust Airport Gate Assignment

Hui Zhao and Liqin Cheng

College of Economics and Management, Civil Aviation University of China, Tianjin 300300, China

Correspondence should be addressed to Hui Zhao; zhaohv@foxmail.com

Received 14 February 2014; Accepted 18 August 2014; Published 25 September 2014

Academic Editor: Albert Victoire

Copyright © 2014 H. Zhao and L. Cheng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Airport gate assignment is core task for airport ground operations. Due to the fact that the departure and arrival time of flights may be influenced by many random factors, the airport gate assignment scheme may encounter gate conflict and many other problems. This paper aims at finding a robust solution for airport gate assignment problem. A mixed integer model is proposed to formulate the problem, and colony algorithm is designed to solve this model. Simulation result shows that, in consideration of robustness, the ability of antidisturbance for airport gate assignment scheme has much improved.

1. Introduction

With the rapid growth of air transport traffic, airport managers are widely concerned with airport operation efficiency. Airport gate assignment scheme is directly related to airport operation efficiency. Purpose of airport gate assignment is to allocate gate for every arrival and departure aircraft. Not only does airport gate assignment affect efficiency of ground service, but also good assignment scheme can prevent congestion at airport.

Airport gate assignment problem can be seen as a kind of resource assignment problem under certain restraints. Airport gate can be seen as resource and aircraft can be seen as resource consumer. Hard constraints of this problem are that aircraft must be assigned to one gate only once, and the gate cannot be assigned to any other aircraft after it is released. In addition to hard constraints, there are many soft constraints that can be considered during gate scheduling process, such as minimizing passenger walking distance and minimizing departure delay.

Mathematical models of airport gate assignment differ from different objectives. Lim and Wang [1] established model to minimize number of conflicted gates; the stochastic factor in arrival and departure time of aircraft is considered.

They show that this model is NP-hard, and heuristic algorithm is used to solve it. Yan and Tang [2] established a simulation framework for gate assignment problem, and waiting time of all passengers is considered. Drexler and Nikulin [3] add the objective of maximizing gate performance scores into model established by Ding, and optimal result is obtained by annealing method. Tang [4] considers gate shortage and flight delay into objective, and evaluation method was used for optimal solution searching. Cheng et al. [5] consider efficient use of resources and passengers' satisfaction as objective and established heuristics algorithm to find optimal solution. Kim and Feron [6] establish mathematical model to minimize number of gate conflicts that are caused by aircraft departure metering.

Deterministic models for aircraft gate assignment seem to get optimal assignment scheme during planning process, but most of these schemes are not really optimal during airport operation process. There are many disturbances that will affect gate assignment scheme; gate assignment scheme may be infeasible. Scheme planners have to adjust gate assignment scheme to make it proof against small deviations from original scheme [7].

In this paper, we will focus on the RAGA (robust airport gate assignment) problem. A mixed integer programming

model is proposed; objective of this model is to find balance between service index and robust index. It is clearly noted that the RAGA model is NP-hard. The ACO (Ant Colony Optimization) algorithm is used to solve this model according to its complexity. Simulation experiments are used to test robustness of solution for RAGA model. Research result in this paper will provide a reference for airport gate assignment management in practice.

This paper is organized as follows. In Section 2 the RAGA mathematic model is established; in Section 3 ACO algorithm is designed for RAGA model; Section 4 is the simulation experiments result and analysis; Section 5 concludes the paper.

2. Problem Formulation

2.1. Description of the Problem. Airport gate assignment is a core task during airport resource scheduling process; its purpose is to determine a certain position for every arrival aircraft, and airport ground operations are scheduled on the basis of gate assignment scheme. Research of RAGA is to find airport gate schedule scheme, which has certain antidisturbance ability for the uncertain factors during airport operation process.

Due to the limited number of airport gate, airport needs to arrange gate in advance for aircraft that will arrive in a certain period. Thus the airport ground services can be scheduled based on airport gate assignment scheme. Airport gate assignment scheme is formulated based on arrival time, leave time of aircraft in flight schedule, and estimation of ground service duration. But during the actual operation process of airport, aircrafts do not always be punctual in the arrival and departure time in flight schedule; there is always a certain deviation between actual arrival and departure time and the scheduled time. For example: delay of aircraft take off that is caused by bad weather may result in arrival delay and departure delay. The arrival delay and departure delay may result in infeasibility of gate assignment scheme; gate planners have to spend several hours on adjusting gate assignment scheme. The adjustment of gate assignment scheme may bring change to related ground service schedule scheme, thus to reduce the overall operation efficiency of airport. So, the ability of antidisturbance should be considered during airport gate assignment scheme making process, namely, the robustness of airport gate assignment scheme.

Figure 1 is the comparison of two assignment schemes for three gates and six aircrafts. The x -axis represents airport gate and the y -axis represents time. The lower edge and upper edge of rectangle represent arrival time and departure time of every aircraft. Arrival time and departure time of every aircraft in these two schemes are exactly the same; these two assignment schemes have no difference on the criterion of punctuality. But on the circumstance of arrival delay or departure delay, scheme 2 has higher antidisturbance ability than scheme 1. For example: when the departure delay of aircraft 1 happened, the gate assignment of aircraft 4 must be adjusted in assignment scheme 1. But in assignment scheme 2, buffer time between departure time of aircraft 1 and arrival

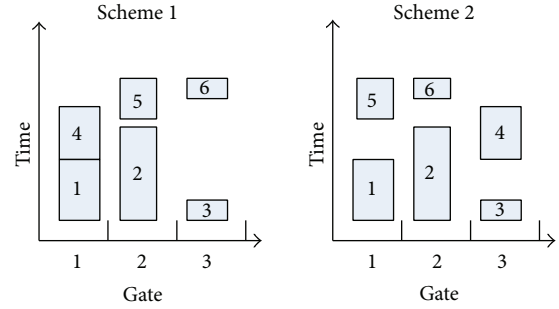


FIGURE 1: Comparison of two assignment schemes.

time of aircraft 5 can absorb certain delay influence; thus assignment scheme 2 is more robust.

Example above shows that adding certain buffer time after departure of every craft can improve robustness of schedule scheme. But too long buffer time may cause new arrival delay or departure delay, and too short buffer time cannot absorb disturbance caused by arrival or departure delay. To find airport gate assignment scheme in consideration of service criterion and robustness criterion is the research aim of this paper.

2.2. Assumption of the Model. In order to establish RAGA model, the following assumptions are made.

Assumption 1. Every gate in airport at most can berth one aircraft at any time.

Assumption 2. Airport gates are divided into two types of large gate and small gate. Aircrafts can be divided into two types of large scale and small scale. Large gate can berth large scale and small scale aircraft and small gate can only berth small scale aircraft.

Assumption 3. Each arrival aircraft must be assigned to a certain gate for berth.

Assumption 4. Arrival time of aircraft, ground operation duration, and passenger boarding duration may be delayed by uncertain factors, which may lead to the delay of aircraft departure.

Assumption 5. Minimum time interval exists between aircraft departure time and berth time in any airport gate.

Assumption 6. Departure time of any aircraft cannot be earlier than departure time in flight schedule.

2.3. Notations Related to the Model

F : the set of all the aircrafts, $F = \{f_1, f_2, \dots, f_n\}$;

P : the set of all airport gates, $P = \{p_1, p_2, \dots, p_m\}$;

u_i : if aircraft f_i is large scale then $u_i = 1$; otherwise $u_i = 0$;

v_i : if p_i is large gate then $v_i = 1$; otherwise $v_i = 0$;

a_i : the arrival time of aircraft f_i in flight schedule;

a'_i : the berth time of flight aircraft f_i in gate arrange scheme;

d_i : the departure time of aircraft f_i in flight schedule;

d'_i : the departure time of flight aircraft f_i in gate arrange scheme;

ξ : the duration of ground operation and passenger boarding;

ϑ : the minimum time interval between aircraft departure time and berth time in the same gate;

x_{ij} : if aircraft f_i is allocated to gate p_i , then $x_{ij} = 1$; otherwise $x_{ij} = 0$;

y_{ijk} : if aircraft f_i and f_j are allocated to gate p_k and f_i follows f_j , then $y_{ijk} = 1$; otherwise $y_{ijk} = 0$;

z_{ij} : if large scale aircraft f_i is allocated to small gate p_j , then $z_{ij} = 1$; if small scale aircraft f_i is allocated to large gate p_j , then $z_{ij} = 0.5$; otherwise $z_{ij} = 0$;

θ : the buffer time between aircraft departure time and berth time in the same gate.

2.4. Formulation of the Model. The mathematical model for RAGA is now presented, followed by step by step explanation. Objective of the model is to find the optimal gate assignment scheme which can ensure service and has certain ability of antidisturbance. Objective function (1) is to minimize the weighted sum of departure delay, buffer time, and matching degree of aircraft with gate. The weights are α , β , and γ , respectively:

$$\min f = \alpha \sum_{i=1}^n (d'_i - d_i) - \beta \theta + \gamma \sum_{i=1}^n \sum_{j=1}^m x_{ij} z_{ij}, \quad (1)$$

subject to:

$$\sum_{k=1}^m x_{ik} = 1, \quad (2)$$

$$x_{ik} \geq \sum_{j=1}^n y_{ijk}, \quad (3)$$

$$x_{jk} \geq \sum_{i=1}^n y_{ijk}, \quad (4)$$

$$d'_i = \begin{cases} a'_i + \xi & a'_i + \xi > d_i \\ d_i & \text{otherwise,} \end{cases} \quad (5)$$

$$a'_j = \begin{cases} d'_i + \vartheta + \theta & \text{if } y_{ijk} = 1, \text{ and } d'_i + \vartheta + \theta > a_j \\ a_j + \vartheta & \text{if } y_{ijk} = 1, \text{ and } d'_i + \vartheta + \theta \leq a_j \\ a_j & \text{if } \sum_{i=1}^n \sum_{k=1}^m y_{ijk} = 0, \end{cases} \quad (6)$$

$$\sum_{k=1}^m z_{ik} < 1, \quad (7)$$

$$\theta \geq 0. \quad (8)$$

Constraint (2) ensures that each aircraft must be and only be arranged to one certain gate. Constraint (3) restricts that each aircraft at a certain gate can at most be followed by one aircraft arranged to that gate. Constraint (4) restricts that each aircraft at a certain gate can at most follow one aircraft arranged to that gate. Constraint (5) is the calculation method for airport departure time. If aircraft berth time adding duration of ground operation and passenger boarding has not exceed departure time in flight schedule, then select the flight schedule departure time as aircraft departure time. Otherwise, select calculation time as aircraft departure time. Constraint (6) is the calculation method for airport berth time. Constraint (7) ensures that there is no large scale aircraft arranged to small gate. Constraint (8) is the range of buffer time.

3. Solving Raga by ACO

ACO is a kind of global algorithm that is put forward by Professor Dorigo in the year of 1992. ACO simulates the process of path formulation during ant search for food and has been widely used in solving many kinds of combinatorial optimization problems [8].

3.1. The Design of the Algorithm ACO

3.1.1. Node Set. Optimal solution of RAGA is related to aircraft and gate, so node for ACO should represent the message of both aircraft and gate. Nodes in path are represented by matrix C:

$$C = \begin{bmatrix} c_{11} & \cdots & c_{1m} \\ \vdots & \vdots & \vdots \\ c_{n1} & \cdots & c_{nm} \end{bmatrix} \quad (9)$$

C is a matrix of order $n \times m$, where m is the number of aircrafts and n is the number of gates. Elements in matrix C represent $n \times m$ nodes in ant searching path. Element c_{ij} in matrix C represents that aircraft i is allocated to gate j . Nodes in matrix C are divided into two types. If $z_{ij} = 1$, then node c_{ij} is called abnormal node; otherwise c_{ij} is called normal node. Abnormal node represents that the node is infeasible because aircraft of large scale is allocated to a small gate.

3.1.2. Tabu Table. Tabu table is established for every ant; Tabu(k) represents nodes that the ant whose code is k cannot

travel. Before node choice of ant k , nodes in matrix C whose value equals Φ are added into $\text{Tabu}(k)$. During node choice process, if node c_{ij} is chosen by ant k , then nodes in matrix C whose row number is i are added into $\text{Tabu}(k)$. The design of tabu table ensured that every aircraft can only be assigned to one certain gate.

3.1.3. Path Table. Path table is established for every ant; nodes that traveled by ant k will be record in $\text{Path}(k)$ sequentially. During initial position selection process for ant k , normal nodes in matrix C are selected as initial position set. One node is randomly selected in initial position set for ant k as the start position, and the node is added into $\text{Path}(k)$ accordingly. During path selection process, when node c_{ij} is traveled by ant k , then the node is added into $\text{Path}(k)$. When all rows in matrix C are traveled by ant k , travel path of ant k can be acquired by sequential read nodes in $\text{Path}(k)$. Airport gate assignment that represented by $\text{Path}(k)$ will be acquired when it is decoded.

3.1.4. Pheromone Design. Pheromone is the information that stays on path when the path is traveled by ant. More pheromones are left when the path is traveled by more ants. During path selection process of every ant, path is selected randomly based on the quantity of its pheromone. The more the quantity pheromone on path, the more the probability the path is selected.

During path selection process, ant at any node has the probability of jumping to any other node normally in matrix C . Pheromone matrix $P(t)$ represents pheromones for ants during path selection process of step t . Matrix $P(t)$ is composed of $n \times m$ submatrices and the order of submatrix is $n \times m$. Submatrix in matrix $P(t)$ is represented by $p_{ij}(t)$:

$$p_{ij}(t) = \begin{bmatrix} \tau_{ij,11}(t) & \tau_{ij,12}(t) & \cdots & \tau_{ij,1m}(t) \\ \tau_{ij,21}(t) & \tau_{ij,22}(t) & \cdots & \tau_{ij,2m}(t) \\ \vdots & \vdots & \vdots & \vdots \\ \tau_{ij,n1}(t) & \tau_{ij,n2}(t) & \cdots & \tau_{ij,nm}(t) \end{bmatrix}. \quad (10)$$

Elements in matrix $p_{ij}(t)$ represent the probability of jumping to other nodes from node c_{ij} at the t jumping step. Before path selection process of ants, matrix $P(0)$ should be initialized, and ants will begin their path selection step according to matrix $P(0)$ from their initial positions. Ants cannot jump between nodes in the same row of matrix C and cannot jump to abnormal nodes either. In the initialization process of matrix $P(0)$, elements in matrix $P(0)$ that represent pheromones between nodes in the same row are set to zero, and elements that represent pheromones between normal nodes and abnormal nodes are set to zero too; other elements are set to 1. Initialization process of matrix $P(0)$ ensured that ants can select node as the next step choice with the same probability in feasible node set.

When ant jumped to another node from its current position, it will leave pheromones on the way. And elements in pheromone matrix $P(t)$ should be updated accordingly.

Formula (11) is the calculation method for pheromone matrix element update:

$$\tau_{ij,rs}(t+1) = (1-\rho)\tau_{ij,rs}(t) + \Delta\tau_{ij,rs}, \quad (11)$$

where

$$\Delta\tau_{ij,rs} = \sum_{k \in K} \Delta\tau_{ij,rs}^k \quad (12)$$

$$\Delta\tau_{ij,rs}^k = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ travels from } c_{ij} \text{ to } c_{rs} \\ 0 & \text{otherwise.} \end{cases}$$

In formula (11) the first item is retained from original pheromone; the second item is the increment of pheromone that is caused by ant travel. Coefficient $1-\rho$ is the description of pheromone reduction, parameter L_k is aircraft total departure delay in path that is formed by travel of ant k , and parameter Q is a constant.

3.1.5. Node Selection Strategy. During node selection process, every ant decides to which node to jump according to pheromone matrix C . Formula (13) is the node selection probability for ant k and its current position is node c_{ij} :

$$p_{ij,rs}^k(t) = \begin{cases} \frac{[\tau_{ij,rs}(t)]^\lambda \times [\eta_{ij,rs}(t)]^\mu}{\sum_{c_{uv} \notin \text{Tabu}(k)} [\tau_{ij,uv}(t)]^\lambda \times [\eta_{ij,uv}(t)]^\mu} & \text{when } c_{rs} \notin \text{Tabu}(k) \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

λ : weight of pheromone;

μ : weight of heuristic factor;

$\eta_{ij,rs}(t)$: $\eta_{ij,rs}(t) = (1/(\text{delay}(c_{rs}) + \zeta)) \text{delay}(c_{rs})$ which is the berth delay in node c_{rs} and is a constant.

Formula (13) is the combination of pheromone factor and heuristic factor. The use of heuristic factor provided possibility of jumping from local extreme value. Parameters λ and μ are used to adjust the weight of pheromone factor and heuristic factor, respectively.

3.1.6. Termination Condition. Since it is difficult to determine scope of optimal solution, the maximum number of generation is used as the algorithm end condition.

3.2. Algorithm Running Step. In robust airport gate assignment problem, robustness of gate assignment scheme is proportional to the length of buffer time. But with the increment of buffer time, aircraft departure delay will increase too. In the model solving process, we should balance the buffer time. Model solving process includes the following steps.

- (1) Set value of maximum buffer time θ_{\max} , set minimum buffer time as θ_{\min} where $\theta_{\min} = 0$, set value of unit interval $\Delta\theta$, and set parameter $e = (\theta_{\max} - \theta_{\min})/\Delta\theta$.

TABLE 1: Calculation result comparison of three methods.

n		ACO		GAA		CPLEX	
		T_{delay} (min)	T_{calc} (sec)	T_{delay} (min)	T_{calc} (sec)	T_{delay} (min)	T_{calc} (sec)
10	Average	32.7	3.1	60.3	<1	32.7	205
	Minimum	15	1	50	<1	15	198
	Maximum	48	5	70	<1	48	213.5
20	Average	126.5	33	179.9	<1	126.5	681
	Minimum	113	30	169	<1	113	601.2
	Maximum	140	38	195	<1	140	722.9
30	Average	393.5	80	755.7	<1	393.5	3526
	Minimum	369	88	747	<1	369	3129.6
	Maximum	426	96	766	<1	426	4022.9

- (2) Set $q = 0$ and set $\theta = \theta_{\min} + q\Delta\theta$ as buffer time.
- (3) Use ACO algorithm for solving RAGA model on condition of given buffer time θ . ACO algorithm includes the following steps.
 - (i) Initialize parameter values of RAGA model, such as number of aircrafts, number of gates, scheduled arrival time and departure time of aircraft, and other parameters.
 - (ii) Initialization of ACO algorithm: set size of ant colony, initialize value of elements in pheromone matrix, initialize tabu table for every ant, and set the maximum iteration generation of ACO algorithm.
 - (iii) Set initial node for every ant in ant colony, and every ant in ant colony begins its path finding process. After each step forward of any ant, update tabu table for that ant correspondingly. When all of the ants finish their travel, update value of elements in pheromone matrix and select the best solution as output of this generation.
 - (iv) If the iteration generation reaches the maximum generation, then select the best solution in all generations as output of ACO algorithm. Otherwise, go to step (iii).
- (4) Put model solution in step 3 into result set R . If $q \neq e$, set $q = q + 1$ and $\theta = \theta_{\min} + q\Delta\theta$, go to step (2); otherwise go to step (5).
- (5) Select best solution in result set R as aircraft assignment scheme, and select the corresponding θ as buffer time for RAGA model.

4. Computational Experiments

4.1. *Experiment without Buffer Time.* Experiment without buffer time is to validate optimal solution search ability of ACO algorithm for RAGA model.

In simulation experiment without buffer time, we simulate a schedule period of one day, which is a period of 1440 minutes. Arrival time a_i of aircraft i is generated in the time

interval of $[0, 1440]$ randomly. Departure time d_i of large aircraft i is generated in the time span of $[a_i + 180, a_i + 300]$ and departure time d_i of small aircraft i is generated in the time interval of $[a_i + 60, a_i + 200]$. Assume there are 4 large gates and 2 small gates.

The algorithm of GAA (Greedy Assignment Algorithm) is commonly used in airport actual operation process. The scheduling process of greedy algorithm is based on the aircrafts' arrival sequence. It will assign aircraft to the gate which can satisfy constraints and cause minimal departure delay currently. CPLEX is a kind of optimization software; it can find optimal solution for linear model and satisfactory solution for nonlinear model. In order to validate optimal solution search ability of ACO, each example is calculated by three methods separately.

Generate 10 examples when the scale of arrival aircraft n is 10, 20, and 30, respectively. Calculate these examples by ACO, GAA, and CPLEX, respectively, and calculation results comparison is shown in Table 1.

The symbol of T_{delay} indicates the departure delay under certain assignment scheme and the symbol of T_{calc} indicates the calculation time. Table 1 shows the average, minimum, and maximum result of T_{delay} , T_{sec} for three methods in case of different aircraft scale. From result in Table 1, we can draw the following conclusions.

- (1) The ACO algorithm has higher ability for optimal solution search. Under the circumstance of $n = 10, 20,$ and 30 , calculation results of T_{delay} by ACO algorithm are equal to results that are calculated by CPLEX. We can say that on case of aircraft scale less than 30, ACO algorithm can get high quality solutions. The values of T_{delay} calculated by ACO algorithm are obviously less than values that are calculated by GAA algorithm under the same aircraft scale circumstance. We can say that ACO algorithm is better than GAA algorithm. When the aircraft scale continues growing, CPLEX cannot find solution in permitted time; it is unable to compare solution quality continually. But ACO algorithm can get solution for RAGA when the aircraft scale continues growing, and quality of the solution is better than solution that was calculated by GAA algorithm.

- (2) The ACO algorithm has shorter calculation time. The GAA algorithm has the shortest calculation time; it can get solution in one second. Calculation time of CPLEX increases with scale of aircraft; when aircraft scale reaches 30, it needs at least 35 minutes to find solution for model. Calculation time of ACO algorithm increases with scale of aircraft too, but the longest time is less than 10 minutes. When the scale of aircraft exceeds 50, CPLEX cannot find solution for model in two hours. But ACO algorithm can find solution in less than 15 minutes. Calculation time of ACO algorithm increases with aircraft scale due to the increasing number of nodes in matrix C .
- (3) The value of T_{delay} increases with aircraft scale. With increasing of aircraft scale, the scheduled departure time delay is increasing accordingly. It shows that the airport gates are gradually approaching their maximum capacity.

4.2. *Experiment with Buffer Time.* Experiment with buffer time is to validate robustness of solution that is calculated by ACO algorithm.

On the simulation circumstance of experiment without buffer time, generate 10 examples for the aircraft scale as 10, 20, and 30, respectively. Every example is calculated on circumstance of $\theta = 0$ and $\theta > 0$ by ACO algorithm, respectively.

Random arrival delay and departure delay are used as disturbance factor to test the robustness of solution calculated by ACO algorithm. The arrival time of aircraft is randomly generated in span $[a_i, a_i + 30]$ to simulate aircraft arrival delay and the departure time of aircraft is randomly generated in span $[d_i, d_i + 30]$ to simulate aircraft departure delay. Aircrafts are assigned to gates according to solution calculated by AOC algorithm. Departure delays are calculated under disturbance circumstance. Simulation results for different aircraft scale are shown in Figures 2, 3, and 4.

The symbol of SD represents scheduled delay, and AD represents actual delay. Figures above show scheduled departure delay and actual departure delay on circumstance of $\theta = 0$ and $\theta > 0$ when $n = 10, 20,$ and $30,$ respectively.

Table 2 is the statistical result for average delay on circumstance of $\theta = 0$ and $\theta > 0$ when $n = 10, 20,$ and $30,$ respectively.

From simulation result and statistical result, we can draw the following conclusions.

- (1) The actual departure delay is longer than scheduled departure delay. When $\theta = 0$ the reason is that robustness factor has not been considered during model solving process, and schedule scheme cannot absorb departure delay caused by random disturbance. When $\theta > 0$ the reason is that although robustness factor has been considered during model solving process, the flight departure delay cannot be absorbed completely by buffer time.
- (2) Actual departure delay increases with scale of aircraft. Actual delay of optimal model is the accumulated

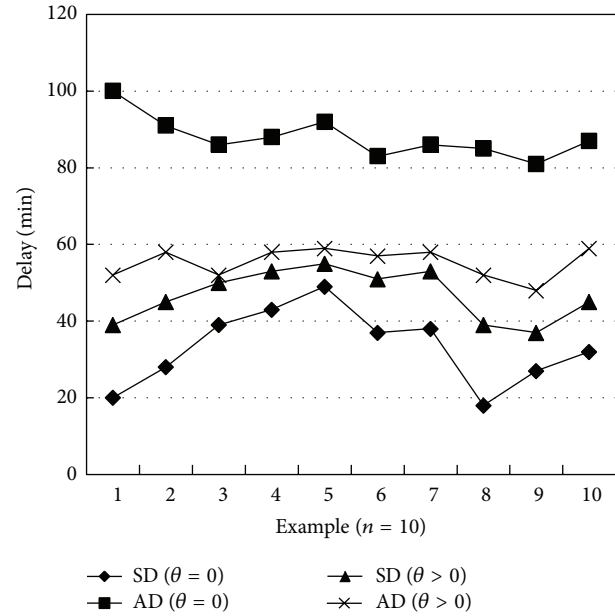


FIGURE 2: Simulation result for $n = 10$.

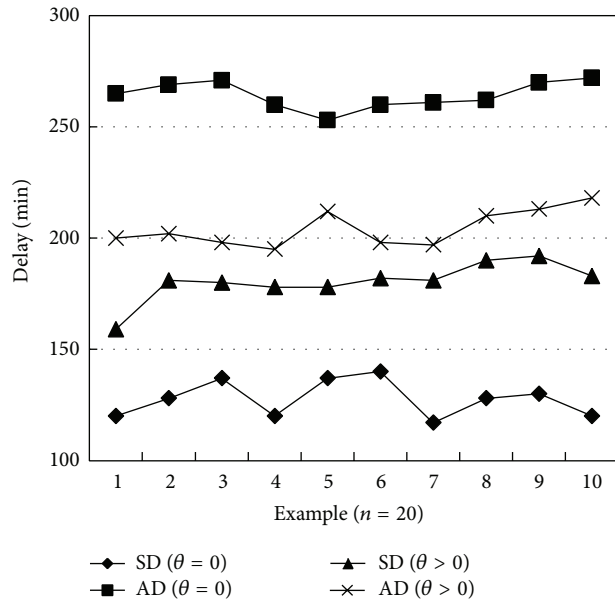


FIGURE 3: Simulation result for $n = 20$.

departure delay of every aircraft. Although actual delay can be partly absorbed by buffer time, the actual delay that cannot be absorbed is increasing with the increase of aircraft scale.

- (3) Robustness of assignment scheme with buffer time is better than that without buffer time. Compare the difference between scheduled delay and actual delay in case of $\theta = 0$ and $\theta > 0$; the difference in case of $\theta > 0$ is smaller than difference in case of $\theta = 0$. We can say that when buffer time is considered, robustness of assignment scheme has improved. When $\theta > 0$,

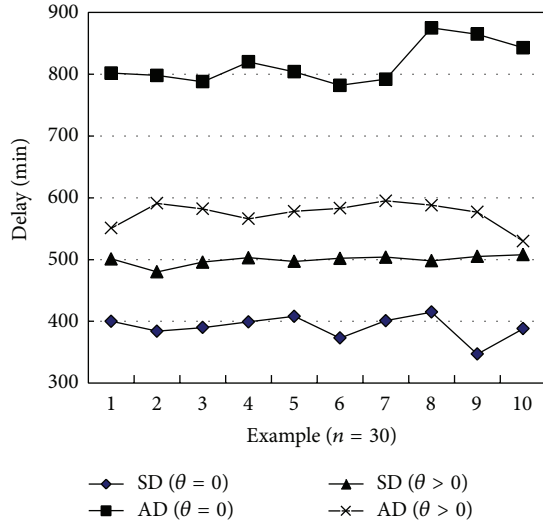
FIGURE 4: Simulation result for $n = 30$.

TABLE 2: Statistical result of three scales.

n	$\theta = 0$		$\theta > 0$	
	Avg _{sd} (min)	Avg _{AD} (min)	Avg _{sd} (min)	Avg _{AD} (min)
10	33.1	87.9	46.7	55.3
20	127.7	264.3	180.4	204.3
30	398.2	823.7	501.6	578.9

most departure delay is absorbed by buffer time; departure delay that cannot be absorbed results in enhancement of actual departure delay on basis of scheduled departure delay. When $\theta = 0$, the actual departure delay of every aircraft cannot be absorbed, and delay will be enlarged during its propagation process.

5. Conclusion

In this paper, buffer time is introduced to the establishment process of robust airport gate assignment model, in order to enhance its ability of resisting uncertainty factors during airport actual operation process. The RAGA model established in this paper does not only consider three optimal aims, that are minimizing departure delay, maximizing buffer time, and maximizing matching degree of aircraft with gate. The ACO algorithm is used to solve the gate assignment model. Model solving process based on ACO is designed in detail. Simulation experiments are used to test optimal solution search ability of ACO algorithm and robustness of calculation result. Simulation result shows that the ACO algorithm can approach optimal solution with high precision, and robustness of gate assignment scheme has improved in consideration of buffer time.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is financially supported by National Nature Science Foundation of China under the Grant 71002077.

References

- [1] A. Lim and F. Wang, "Robust airport gate assignment," in *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '05)*, pp. 74–81, November 2005.
- [2] 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.
- [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] C.-H. Tang, "A gate reassignment model for the taiwan taoyuan airport under temporary gate shortages and stochastic flight delays," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 41, no. 4, pp. 637–650, 2011.
- [5] C.-H. Cheng, S. C. Ho, and C.-L. Kwan, "The use of meta-heuristics for airport gate assignment," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12430–12437, 2012.
- [6] S. H. Kim and E. Feron, "Impact of gate assignment on gate-holding departure control strategies," in *Proceeding of the 31st Digital Avionics Systems Conference: Projecting 100 Years of Aerospace History into the Future of Avionics (DASC '12)*, pp. 4E3-1–4E3-8, Williamsburg, Va, USA, October 2012.
- [7] H. M. Genç, O. K. Erol, B. Eksin, M. F. Berber, and B. O. Güleriyüz, "A stochastic neighborhood search approach for airport gate assignment problem," *Expert Systems with Applications*, vol. 39, no. 1, pp. 316–327, 2012.
- [8] S. Lorpunmance and A. Sap, "An ant colony optimization for dynamic job scheduling in grid environment," *International Journal of Computer and Information Science and Engineering*, vol. 1, pp. 207–214, 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

