Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 854364, 9 pages
http://dx.doi.org/10.1155/2013/854364

Research Article

A Collaborative Optimization Model for Ground Taxi Based on Aircraft Priority

Yu Jiang, Zhihua Liao, and Honghai Zhang

College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China

Correspondence should be addressed to Zhihua Liao; lzh0909nuaa@163.com

Received 3 July 2013; Revised 9 October 2013; Accepted 11 October 2013

Academic Editor: John Gunnar Carlsson

Large hub airports have gradually become the "bottleneck" of the air transport network. To alleviate the "bottleneck" effect, optimizing the taxi scheduling is one of the solutions. This paper establishes a scheduling optimization model by introducing priority of aircraft under collaborative decision-making mechanism, and a genetic algorithm is designed to verify the scheduling model by simulating. Optimization results show that the reliability of the model and the adjusted genetic algorithm have a high efficiency. The taxiing time decreases by 2.26% when compared with an empirical method and the flights with higher priorities are assigned better taxi routes. It has great significance in reducing flight delays and cost of operation.

1. Introduction

Civil aviation transportation industry in China has developed into an important period of rapid growth in recent years. From 2006 to 2012, the average growth rate of total turnover volume is 12.4%. The contradiction between rapid development of air transport and supply of transport infrastructure has become increasingly acute, and airport has gradually become the "bottleneck" of air transport network. More and more attentions are paid to airport scene resource scheduling, especially the runway and taxiway system resource scheduling problem. The rate of utilization of taxiway will be improved, and the available capacity will also be increased by using scheduling optimization technology. Meanwhile, it will be more conductive to achieve fair, efficient operation and ease the contradiction between traffic flow and the available capacity by introducing aircraft priority under collaborative decision-making mechanism. The study from the perspective of system will reduce flight delays and fuel consumption while taxiing in the whole. The airport scene resources scheduling optimization problem has already become one of the hot topics in the study of domestic and foreign scholars now.

In the study of foreign scholars, Gotteland and Durand [1] presented an optimization model with safety separation and runway capacity constraints and set taxi time minimum as the optimization objective. They solved the model by using genetic algorithm. The model did not take the aircraft priority and taxi waiting problem caused by confliction into account. Marín [2] established a time-space network to describe confliction and congestion during taxiing. A network flow model is used to optimize the scheduling with real data from Madrid-Barajas airport. But the constraint conditions in the model are too complex. Ravizza et al. [3] put forward a stand holding model, and its objective function was to minimize taxi time and fuel consumption. Anderson and Milutinovi [4] introduced deviation probability to control the security constraints, and an uncertainty-based mixed integer linear programming model was established. But this article ignored the deadlock which may be caused by some conflicts. Clare and Richards [5] studied scheduling optimization problem on runway and taxiway by a MILP model. The interaction effect between arrival and departure aircrafts was considered. Keith et al. [6] presented a MILP model based on conflict-free. In fact, it allowed for some conflictions and a holding strategy made the result more optimal. Burgain et al. [7] analyzed departure aircrafts in congested airports by using queuing optimization. Collaborative decision-making concept was introduced in this paper. An aircraft taxi time estimate technique was studied in paper [8] which was based on fuzzy rule system. Some papers [9–13] just presented other new optimization methods, for example, cellular automata model. Some papers [14–17] presented new methods to analyze and solve the problem, for example, linear method, statistical analysis, quantitative analysis, and empirical methods, for example, iteration plan. In these studies, though they proposed different kinds of models, the constraints of taxi rules were the same and the substance of optimization objective was similar. With the implementation of collaborative decision-making (CDM) mechanism, the codecision results from multiorganizations (such as ATC, airlines, and airports) must be taken into consideration when scheduling in busy airports. A normal way to make fair decisions is to introduce some priorities. The priority can guarantee benefit of all parties and make taxi schedule more smooth. The flight priorities are usually determined by the type of flight, aircraft type, or the airlines they belonged to. In this paper, the flight priorities are added into constrains directly and we will not discuss how they are calculated. The computational complexity of accurate computation is higher than intelligence algorithm. But intelligence algorithm may be unstable, because the computation result this time may be not the same as the next time. The efficiency of algorithm still can be improved.

In the study of domestic scholars, a mixed integer programming model algorithm is proposed by Zhang et al. [18]. The paper optimizes taxi time under the conditions of basic safety separation and conflict-free. Apparently, this method cannot verify whether the taxi paths are optimal or not. You and Han [19] present a multiagent model. The aircrafts invariably look for the shortest path from current node to the destination node in the process of optimization on a simulation platform. So the waiting-taxiing balance problem still exists. Wang et al. [20] put forward a dynamic path algorithm based on conflict-free. This approach can effectively avoid conflicts, but the result may not be optimal. An optimization scheduling algorithm based on genetic algorithm is studied by Liu et al. [21]. From these studies, three points are summarized. (1) The waiting-taxiing optimization is not enough. (2) Genetic algorithm has advantages in large-scale scheduling problems, but the efficiency of the algorithm can still be improved. (3) On the CDM platform, every aircraft is given a specific priority and the taxi scheduling will reduce flight delays on the whole.

Taking all the elements in the taxi scheduling into consideration, the paper sets the total taxi time minimum as the optimization goal. The basic safety separation is considered and the aircraft priority and taxiway-waiting strategy are introduced. A linear programming model is established and a genetic algorithm is designed to simulate. The method can not only improve the algorithm efficiency, but also get scheduling path directly.

2. Modelling

2.1. Description and Analysis. Taxi scheduling optimization can be defined as the work in which each aircraft is given a specific path on a certain taxiway network structure without deadlock conflict and make the total taxi time minimal.

Table 1: Minimum safety separation standards (unit: m).

After	Front		
	Heavy	Medium	Light
Heavy	300	200	100
Medium	300	200	100
Light	300	200	100

The taxi system in the airport is composed of runway passageway, taxiway, and parking apron. For a departure flight, after finishing the work in an assigned stand, such as cleaning, on-off passengers, catering, and fuelling, the aircraft will wait for controller's command. The air traffic control (ATC) in the tower will give commands about taxi path as well as take-off runway and entrance. The aircraft will be pushed out and begins to taxi on taxiway. In general, more than two aircrafts taxi on the taxiway at the same time; a basic safety separation between aircrafts is required. According to the aircraft operation management manual, a minimum safety separation is regulated between different types of aircraft (including heavy, medium-size, and light aircraft). Table 1 gives the minimum safety separation between different types of aircrafts.

During taxiing, the pilot can keep safety separation with the following aircraft by adjusting aircraft speed. Only one aircraft is allowed to pass the same node at one time and other aircrafts are required to wait to ensure safety. When two aircrafts need to taxi on the same segment of taxiway from different nodes, one aircraft must hold and wait at the entrance node if the minimum safety separation is not satisfied. If an aircraft arrives at the assigned runway entrance, it can enter runway and take off when ATC allows.

For the arrival flight, the aircraft enters taxiway from assigned runway and exits according to ATC instructions. The taxi path and stand are assigned before the aircraft enters taxi system. The taxi is over when the aircraft arrives at the stand. Figure 1 shows the whole operation flow of aircraft in airport.

Taxi paths in taxi system are very complex, and all aircrafts must keep the basic safety separation, so conflicts tend to occur during taxiing. In order to make the model not too complex, rear-end conflict and intersectional conflict are classified as node-conflict. Deadlock conflict is classified as edge-conflict. So there are two types of conflicts: node-conflict and edge-conflict.

Node-conflict happens when two or more than two aircrafts taxi through a common node without keeping the minimum safety separation (see Figure 2).

Edge-conflict happens when two or more than two aircrafts taxi through the common segment but with opposite direction. One aircraft must hold and wait at the entrance node if the minimum safety separation is not satisfied (see Figure 3).

For most aircrafts, they cannot taxi backward; once edge-conflict happens, the common segment will come to a deadlock. So edge-conflict can also be called deadlock conflict. In general, preventive measures must be taken if edge-conflict is likely to happen; for example, one aircraft is

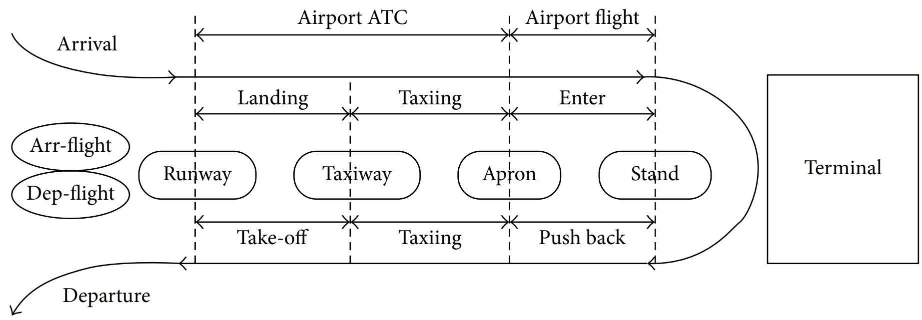

Figure 1: The whole operation flow in the airport.

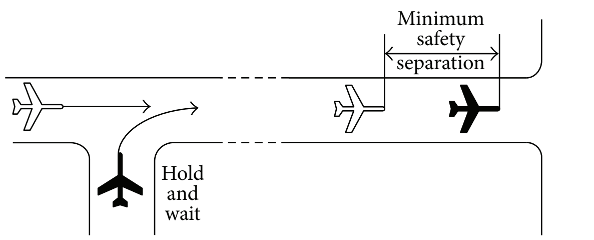

Figure 2: Node-conflict.

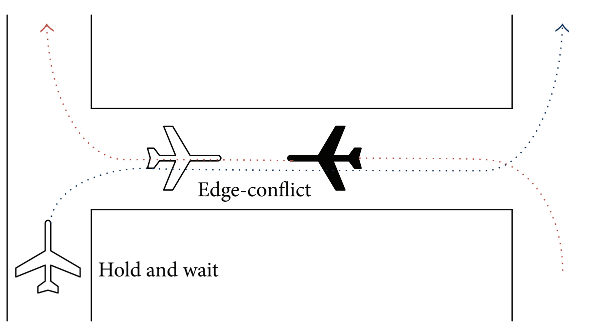

Figure 3: Edge-conflict.

not allowed to taxi through until the common segment is not in use if the aircraft is estimated to arrive later than another or aircrafts are given different priorities and only the aircraft with a higher priority can taxi through the common segment at one time.

Some scholars put forward a dynamic path algorithm based on conflict-free. They want to avoid all the conflicts. The dynamic path algorithm can find the shortest path in real time by a sliding time window. Though the method can make optimal decision, sometimes it is more optimal if a taxiway-waiting strategy is taken. Obviously, taxiway-waiting strategy should be taken into consideration. In this paper, node-conflict is allowed, because aircrafts can resolve conflicts easily and this type of conflict has little effect on taxiway system. Edge-conflict should be avoided as far as possible. Part of taxi system or even the whole taxi system may come to a deadlock once edge-conflict happens. Besides, solving a deadlock is costly. Therefore, an edge-conflict constraint is added to the model to avoid this type of conflicts as far as possible.

2.2. Model Assumption. The paper mainly studies taxi optimization between stand and runway passageway. The method of how aircrafts choose taxi path and how to avoid conflicts is analyzed. The objective of path choice is to make the total taxi path length minimum, but the conflicts are also considered during taxiing. In general, aircrafts can avoid conflicts by adjusting taxi speed or waiting at an intersectional node. If an aircraft reduces its speed to keep the safety separation, it means that the aircraft will arrive at the next node later compared with normal condition. The time difference of arrival can be equivalent to waiting time at the destination node. So the objective function is to make the total time cost of all aircrafts minimum. Three assumptions are made for the model based on the above analysis.

(1) Generally, all the aircrafts taxi at the same maximum speed.

(2) When it is likely to conflict, aircrafts can adjust speed rapidly. So the acceleration is ignored.

(3) When node-conflict happens, hold and wait strategy is always efficient whether the aircraft is large or small.

2.3. Objective Function. Usually, more than two aircrafts need an assigned taxi path at the same time. The path scheduling can be evaluated by the length of path, the type of conflicts, the time of conflict, and the degree of conflict. The total time cost of all aircrafts reflects partly the path scheduling. So the optimization objective in this paper is to make the total time cost of all aircrafts minimum. Consider

$$\operatorname{Min} T = \sum \left(\frac{S_{n^i_{\text{start}} n^i_{\text{end}}}}{v_i} + \sum_{n=\text{start}}^{\text{end}} T_{f_{ni}} + \sum_{n_x=\text{start}}^{\text{end}-1} T_{c_{n_x n_{x+1}}} \right). \quad (1)$$

2.4. Constraints. Variable f_{ijn} is used to detect whether node-conflict happens, and it must satisfy the following constraint:

$$f_{ijn} = \begin{cases} 1 & \left| t_{ni} - t_{nj} \right| \le t_0 \\ 0 & \text{others} \end{cases} \quad \forall n \in R_i \cap R_j, i \ne j. \quad (2)$$

Variable w_{ij} is used to compare priority between aircraft i and j, and it must satisfy the following constraint.

$$w_{ij} = \begin{cases} 1 & p_i \ge p_j \\ 0 & \text{others} \end{cases} \quad i \ne j. \quad (3)$$

Variable x_{ijn} is used to detect the sequencing of aircraft i and j, and it must satisfy the following constraint:

$$x_{ijn} = \begin{cases} 1 & t_{ni} \leq t_{nj} \\ 0 & \text{others} \end{cases} \quad \forall n \in R_i \cap R_j, i \neq j. \tag{4}$$

If node-conflict happens at node n, the hold and wait time $T_{f_{ni}}$ must satisfy the following constraint:

$$T_{f_{ni}} = f_{ijn}\left(1 - w_{ij}\right)\left(t_0 + t_{nj} - t_{ni}\right), \quad \forall n \in R_i \cap R_j, i \neq j. \tag{5}$$

If edge-conflict happens at edge (m, n), the hold and wait time $T_{c_{mni}}$ must satisfy the following constraint:

$$T_{c_{n_x n_{x+1} i}} = \left(1 - w_{ij}\right)\left(\frac{S_{n_x n_{x+1}}}{v_j} + t_{n_x j} - t_{n_{x+1} i}\right), \quad \forall \left(n_x, n_{x+1}\right) \in R_i \cap R_j, i \neq j. \tag{6}$$

S_{mn}, the path length from node m to node n, must satisfy the following constraint:

$$S_{mn} = \sum_{x=m}^{n} S_{n_x n_{x+1}}. \tag{7}$$

The time of arrival at node n_x, $t_{n_x i}$, must satisfy the following condition:

$$t_{n_x i} = \frac{S_{n_1 n_x}}{v_i} + \sum_{\text{start}}^{x} T_{f_{n_x i}} + \sum_{\text{start}}^{x-1} T_{c_{n_x n_{x+1} i}} + T_i. \tag{8}$$

In addition to the above basic constraints, there are four other constraints during taxiing.

The minimum safety time interval constraint is as follows:

$$t_{n_x i} - t_{n_x j} \geq t_0, \quad \forall n_x \in R_i \cap R_j, i \neq j. \tag{9}$$

The minimum safety time interval at intersectional node is as follows:

$$t_{n_x j} \geq x_{ijn_x}\left(t_{n_x i} + t_0\right), \quad \forall i, j \in F, i \neq j. \tag{10}$$

Rear-end conflict constraint is as follows:

$$x_{ijn_x} - x_{ijn_{x+1}} = 0, \quad \forall i, j \in F, i \neq j, \quad \forall \left(n_x, n_{x+1}\right) \in R_i, \quad \forall \left(n_x, n_{x+1}\right) \in R_j. \tag{11}$$

Deadlock conflict constraint is as follows:

$$x_{ijn_x} - x_{ijn_{x+1}} = 0, \quad \forall i, j \in F, i \neq j, \quad \forall \left(n_x, n_{x+1}\right) \in R_i, \quad \forall \left(n_{x+1}, n_x\right) \in R_j. \tag{12}$$

3. Design and Genetic Algorithm

3.1. Coding. In order to make the results evident, segmented real-coded method is chosen in the algorithm. Every gene in chromosome stands for corresponding node in taxi path. In this way, every chromosome can signify multipaths in a more direct way. For example, supposing that all departure aircrafts taxi from node 5 to node 1 and arrival aircrafts are on the contrary, the path codes of two departure aircrafts and two arrival aircrafts can be expressed as follows:

5	3	4	1	0	5	2	1	0	0	1	3	4	5	0	1	4	2	5	0

(13)

The numbers 1–5 stand for the node numbers of taxi path and 0 is a pad character.

3.2. Population Initialization. Population initialization of traditional genetic algorithm is completely random to some extent. Supposing that there is a taxiway with n nodes, m aircrafts need to be assigned a path from node A to node B. There are r feasible taxi paths between A and B. Then the probability of obtaining a feasible path is r/n^{mn}. For r is far smaller than n^{mn}, this method is ineffective in population initialization.

In order to improve the efficiency of algorithm, a traversal algorithm is used to compute all of the feasible paths between start node and destination node. The initial chromosome is produced by selecting feasible paths randomly. In this way, the initial chromosome is a set of feasible solution. So the population initialization is based on the set of feasible solution in this method. Apparently, the change will be conductive to the implementation of genetic operations and impel the population evolve rapidly. The efficiency of the algorithm is improved.

3.3. Crossover Operation. The paper chooses segmented real-coded method, so multipoint matched crossover method is the best choice. The multipoint crossover means that multiple genes implement crossover operation at the same time. Suppose that there are two parent chromosomes: parent1 and parent2

Parent1	7	5	2	0	1	9	0	4	0	6
Parent2	6	3	2	1	0	9	8	4	5	7

(14)

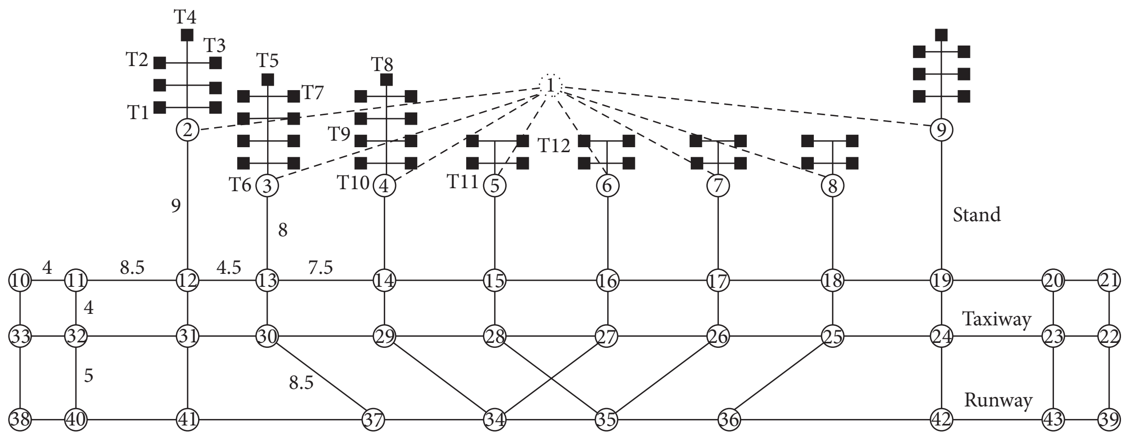

FIGURE 4: The taxiway network.

Three genes in parent1 and parent2 are matched, and they are nodes 2, 9, and 4. New generation is produced after crossover operation

New-generation1	7	5	2	1	0	9	0	4	5	7
New-generation2	6	3	2	0	1	9	8	4	0	6

(15)

Multipoint matched crossover method can prevent some superior chromosomes from being destroyed to some extent. Meanwhile, this method can ensure that the new generation is still a feasible path and prevents population size reducing sharply.

3.4. Fitness Function. Suppose that all the aircrafts taxi at the same speed, then the total time cost of aircraft can easily be converted to the length of path (hold and wait time is also converted to path length). So optimization objective is to make the total length of path minimum. The fitness function can be divided into three parts: the actual path length; node-conflict path length; edge-conflict path length. The actual path length is the sum of every edge length in its taxi path. If the estimated time of two aircrafts which arrive at a common node is dissatisfied with t_0, one aircraft must adjust speed or hold and wait at the common node. It leads to the extension of its total taxi time. The extended time of taxi is the difference value between theoretical taxi time and actual taxi time. Apparently, the extended taxi time is converted into node-conflict path length. The computing method of edge-conflict path length is the same as node-conflict path length.

When the time is converted into path length, the fitness function can be expressed as follows:

$$f_{\text{fitness}} = \frac{\alpha}{S_{\text{route length}} + \beta * S_{V\text{-length}} + \gamma * S_{E\text{-length}}}. \quad (16)$$

In the formula, α, β, and γ are unknown parameters; $S_{\text{route length}}$ is the actual path length; $S_{V\text{-length}}$ is node-conflict path length; $S_{E\text{-length}}$ is edge-conflict path length.

TABLE 2: Flight scheduling information.

	Flight no.	ETD	Type	Priority	Stand	D/A
1	MU5178	10.35	320	4	T1	D
2	CZ3118	10.35	330	4.5	T2	D
3	CZ6218	10.35	330	4.5	T7	D
4	MU2078	10.35	320	4	T8	D
5	CA1605	10.35	737	5	T5	D
6	CA1802	10.35	738	3.5	T4	A
7	MF8115	10.35	737	3	T9	A
8	HU7196	10.35	734	2.5	T12	A
9	GS6574	10.35	319	2	T11	A

4. Simulation Verification

In order to verify whether the algorithm is feasible and effective or not, one large hub airport in China is chosen as an example. We consider flight scheduling at the same time in a rush hour. According to actual operation, taxi routes commonly used are limited. So we choose one runway with partial taxiways as an example. Figure 4 shows the network of the scene. A virtual node is introduced in the example. The virtual node can transform the problem into path problem between any two nodes. Besides, the length between virtual node and stand node can be regarded as the waiting time of push back. The initial value of the length is 0. An adjusted genetic algorithm is programmed in C++ programming language on VC++ 6.0 platform.

The minimum safety time interval is set as 4 units (the conversion of 300 meters). The average speed of aircraft is set as 36 Km/h, which means that the minimum safety time interval is 30 seconds. According to the flight scheduling in this airport, 9 flights need to be scheduled at 10:35 am. The specific flight information is listed in Table 2.

The flight priorities are associated with the type of flight, aircraft type, and the airlines they belonged to. The flight priorities are given directly in scheduling information table.

Table 3: Scheduling by experience.

Flight no.	Assigned routes	Stand
MU5178	2 → 12 → 11 → 10 → 33 → 38	T1
CZ3118	2 → 12 → 11 → 10 → 33 → 38	T2
CZ6218	3 → 13 → 12 → 11 → 10 → 33 → 38	T7
MU2078	4 → 14 → 13 → 12 → 11 → 10 → 33 → 38	T8
CA1605	3 → 13 → 12 → 11 → 10 → 33 → 38	T5
CA1802	37 → 30 → 31 → 12 → 2	T4
MF8115	37 → 30 → 29 → 14 → 4	T9
HU7196	37 → 30 → 29 → 28 → 15 → 5	T12
GS6574	37 → 30 → 29 → 28 → 15 → 5	T11

4.1. Experience Scheduling. When the ground controller assigns paths by experience method, some routes are preferred and waiting phenomenon is universal. Assume the ground controller takes FCFS (first come first service) strategy and assigns routes with experience. One scheduling may be like the following in Table 3.

The distribution of arrival time at each node is shown in Figure 5. It is easy to find that conflict happens at node 10, 11, 12, 13, and 33 theoretically.

In the actual operation, some aircrafts wait at node in order to avoid confliction. This increases the whole taxi time. More information about the experience scheduling is listed in Table 4. The average length of taxi route is 2366.3 m and the average waiting time is about 28.3 s. The actual average taxi time is about 275.6 s and confliction happens 6 times. Though the priority of CA1605 is higher than others, it conflicts with MU2078 and CZ6218 at nodes 3 and 12. As a result, CA1605 waits 94 s in the whole. It is obvious that the scheduling can still be improved.

The distribution of actual arrival time at each node is shown in Figure 6. All the flights arrive at each node with time interval no smaller than the minimum safety time interval. The whole taxi time is increased by 255 s.

4.2. Genetic Algorithm Scheduling. The population size is set as 20, crossover probability is 0.618, and mutation probability is 0.025. The maximum iteration is 100. The initial population shows that the sum of fitness is 101.26 and the average fitness is 5.06. The maximum fitness is 5.13 and the shortest path is 27300 m. The best assigned routes in the initial population are listed in Table 5.

To solve the problem with genetic algorithm, we programmed in C++ programming language on VC++ 6.0 platform and work in a computer with dual core processor of Inter(R) Core(TM) i3 and 2 G RAM. After 100 iterations, the program output the results. The solving time is about 12 s. Table 6 is about the optimized results.

The optimized population shows that the sum of fitness is 131.57 and the average fitness is 6.58. The maximum fitness is 6.62 and the shortest path is 21297 m.

More information is shown in Table 7. The average length of taxi route is 2366.3 m, and the average taxi time is about 269.3 s (decreased by 10.3% compared with an experience value of 5 min). Confliction happens 5 times and the whole

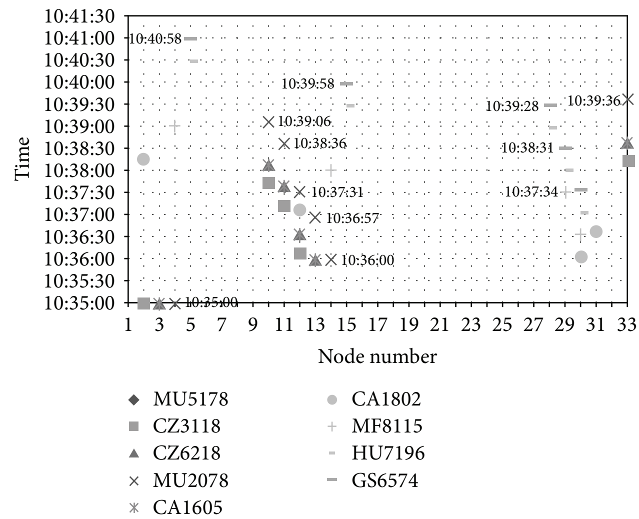

Figure 5: The distribution of arrival time at each node.

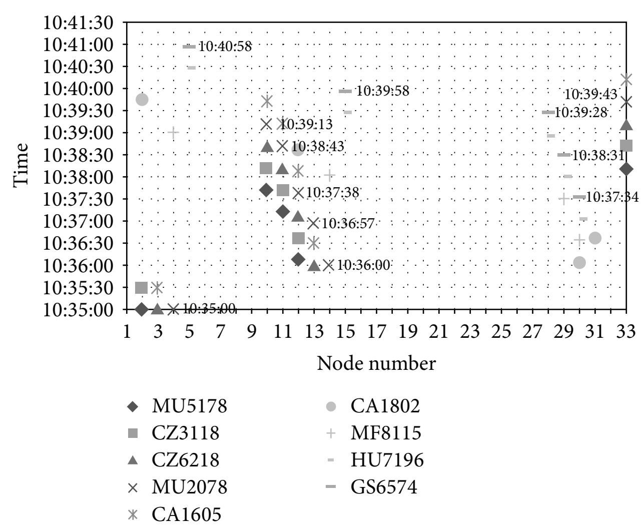

Figure 6: The distribution of actual arrival time at each node.

waiting time is about 199 s with an average of 22.1 s. The waiting time of CA1605 is only 4 s and this is mostly because of its high priority.

The optimized distribution of actual arrival time at each node is shown in Figure 7. All flights which arrive at each node satisfy the minimum time interval. The whole taxi time is increased by 199 s.

Genetic evolution process is shown in Figure 8. It can be clearly seen how the population average fitness changes. As the initial chromosome is produced by selecting feasible paths randomly, in the process of evolution, the average fitness is close to the optimal solution after 37 iterations. The average fitness is stable after 65 iterations; the maximum average fitness is about 6.58.

4.3. Comparisons. Two methods are used to analyze the problem; the results are listed in Table 8.

Table 4: The result of experience scheduling.

Flight no.	Assigned routes	Length	Waiting (s)	Actual time (s)
MU5178	2 → 12 → 11 → 10 → 33 → 38	2287	0	231
CZ3118	2 → 12 → 11 → 10 → 33 → 38	2287	30	261
CZ6218	3 → 13 → 12 → 11 → 10 → 33 → 38	2550	34	291
MU2078	4 → 14 → 13 → 12 → 11 → 10 → 33 → 38	3112	7	321
CA1605	3 → 13 → 12 → 11 → 10 → 33 → 38	2250	94	351
CA1802	37 → 30 → 31 → 12 → 2	1950	90	286
MF8115	37 → 30 → 29 → 14 → 4	1537	0	211
HU7196	37 → 30 → 29 → 28 → 15 → 5	2662	0	264
GS6574	37 → 30 → 29 → 28 → 15 → 5	2662	0	264
Average		2366.3	28.3	275.6

Table 5: The best flight scheduling in initial population.

Flight no.	Assigned routes	Stand
MU5178	2 → 12 → 11 → 32 → 33 → 38	T1
CZ3118	2 → 12 → 13 → 30 → 31 → 32 → 33 → 38	T2
CZ6218	3 → 13 → 30 → 31 → 12 → 11 → 10 → 33 → 38	T7
MU2078	4 → 14 → 29 → 30 → 31 → 12 → 11 → 10 → 33 → 38	T8
CA1605	3 → 13 → 12 → 11 → 32 → 33 → 38	T5
CA1802	37 → 30 → 31 → 32 → 11 → 12 → 2	T4
MF8115	37 → 30 → 29 → 28 → 15 → 14 → 4	T9
HU7196	37 → 30 → 13 → 14 → 29 → 28 → 15 → 5	T12
GS6574	37 → 30 → 31 → 12 → 13 → 14 → 15 → 5	T11

Table 6: Scheduling by genetic algorithm.

Flight no.	Assigned routes	Stand
MU5178	2 → 12 → 11 → 10 → 33 → 38	T1
CZ3118	2 → 12 → 31 → 32 → 33 → 38	T2
CZ6218	3 → 13 → 12 → 11 → 10 → 33 → 38	T7
MU2078	4 → 14 → 13 → 12 → 31 → 32 → 33 → 38	T8
CA1605	3 → 13 → 12 → 11 → 10 → 33 → 38	T5
CA1802	37 → 30 → 13 → 12 → 2	T4
MF8115	37 → 30 → 29 → 14 → 4	T9
HU7196	37 → 30 → 29 → 28 → 15 → 5	T12
GS6574	37 → 30 → 29 → 14 → 15 → 5	T11

Though the actual taxi lengths are equal in two methods, the confliction times and the whole waiting time are decreased in genetic algorithm method. The waiting time has decreased by 56 s and the whole taxi time (waiting time included) has decreased by 2.26%. For CA1605 has the highest priority, the optimized result shows that the waiting time is 4 s. The waiting time of CA1605 in the experience method is 94 s. So the important flights are guaranteed with better routes for their priorities.

For the efficiency of different algorithm, You and Han (2009) proposed a route optimization algorithm based on multiagent. That paper solves a scheduling problem with 3 flights and 14 nodes. The comparison of two algorithms is listed in Table 9.

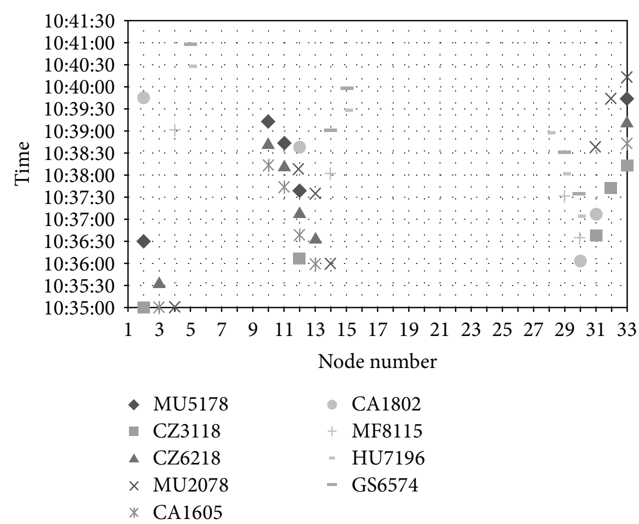

Figure 7: The optimized distribution of actual arrival time at each node.

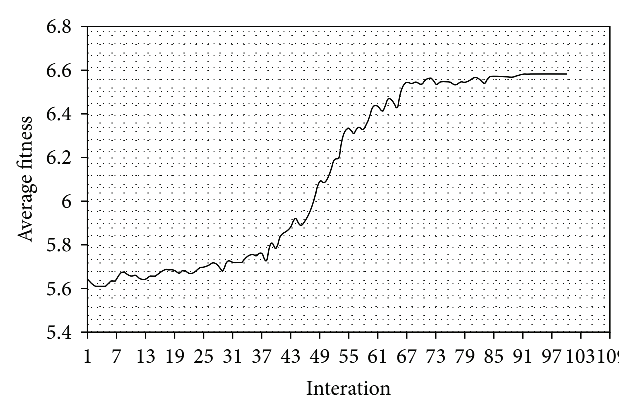

Figure 8: Genetic evolution process.

The number of flights and nodes is less than that in this paper. It is no doubt that the scale in this paper is much bigger. The solving time of the genetic algorithm is about 12 s, but the multiagent takes about 95 s.

Table 7: The optimized result.

Flight no.	Assigned route	Length	Waiting (s)	Actual time (s)
MU5178	$2 \to 12 \to 11 \to 10 \to 33 \to 38$	2287	90	321
CZ3118	$2 \to 12 \to 31 \to 32 \to 33 \to 38$	2287	0	231
CZ6218	$3 \to 13 \to 12 \to 11 \to 10 \to 33 \to 38$	2550	34	291
MU2078	$4 \to 14 \to 13 \to 12 \to 31 \to 32 \to 33 \to 38$	3112	37	351
CA1605	$3 \to 13 \to 12 \to 11 \to 10 \to 33 \to 38$	2250	4	261
CA1802	$37 \to 30 \to 13 \to 12 \to 2$	1950	34	230
MF8115	$37 \to 30 \to 29 \to 14 \to 4$	1537	0	211
HU7196	$37 \to 30 \to 29 \to 28 \to 15 \to 5$	2662	0	264
GS6574	$37 \to 30 \to 29 \to 14 \to 15 \to 5$	2662	0	264
Average		2366.3	22.1	269.3

Table 8: The comparison of two methods.

	Length (m)	Conflict times	Waiting (s)	Whole taxi time (s)
Experience method	21297	6	255	2480
Genetic algorithm	21297	5	199	2424

Table 9: The comparison of two algorithms.

Algorithm	The scale of problem		Solving time (s)
	Nodes	Flights	
Multiagent	14	3	95
Genetic algorithm	43	9	12

From the optimized results and the comparison of different methods, the study of scheduling problems in large hub airports makes great practical significance and the genetic algorithm has great advantage in solving such big scale problems. From the economic view, the fuel consumption will be greatly reduced by decreasing the total time cost of all aircrafts during taxiing. On the one hand, conflicts happen rarely; on the other hand, the operation cost will also be reduced in airlines. From environmental protection point of view, the aircraft engine emissions of nitrogen oxides are reduced and it is beneficial to reduce environmental pollution. From the perspective of operation and management, the use of new scheduling technology will help to improve work efficiency and management level, especially for large-scale scheduling problems.

5. Conclusions

The CDM mechanism will raise a higher requirement for the airport scene management level. The use of a more efficient scheduling technology will help to make decision fairer than experience, reduce flight delays in the whole, and decrease the cost of flight delay and fuel consumption. The paper proposes a taxiing scheduling optimization model based on adjusted genetic algorithm. The results show that the algorithm is efficient.

In fact, aircraft taxiing speed is different and it is related to the aircraft type. Taking aircraft taxiing speed into consideration, we will get a more optimized result.

Symbol Description

- $G(V, E)$: Taxi network structure
- V: Set of all nodes
- E: Set of all edges
- n^i_{start}: The start node of taxi i
- n^i_{end}: The destination node of aircraft i
- R: Set of feasible taxi path for all aircrafts, $R_i \in R$
- n_i: Node in taxi network, $n_i \in V$, $R_i = \{n^i_{\text{start}}, n^i_2, \ldots, n^i_{\text{end}}\}$
- F: Set of all aircrafts, $i \in F$
- P: Set of aircraft priorities, P_i is the priority of aircraft i
- T_i: The release time of aircraft i
- v_i: The taxi speed of aircraft i
- S_{mn}: The edge length between node m and node n
- t_{mi}: The time of arrival at node m
- t_0: The minimum safety time interval
- $T_{f_{ni}}$: The hold and wait time at node n for node-conflict
- $T_{c_{mni}}$: The hold and wait time at edge (m, n) for edge-conflict
- f_{ijn}: Node-conflict detection 0-1 variables
- w_{ij}: Priority comparison 0-1 variables
- x_{ijn}: Arrival sequence detection 0-1 variables.

Acknowledgments

This work was supported in National Natural Science Foundation of China and Civil Aviation Administration of China (no. U1333117); China Postdoctoral Science Foundation (no. 2012M511275) and the Fundamental Research Fund for the Central Universities (nos. NS2013067, NN2012019, and NS2012115).

References

[1] J. B. Gotteland and N. Durand, "Genetic algorithms applied to airport ground traffic optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '03)*, vol. 1, pp. 544–551, December 2003.

[2] A. G. Marín, "Airport management: taxi planning," *Annals of Operations Research*, vol. 143, no. 1, pp. 191–202, 2006.

[3] S. Ravizza, J. A. D. Atkin, and E. K. Burke, "A more realistic approach for airport ground movement optimisation with stand holding," *Journal of Scheduling*, 2013.

[4] R. Anderson and D. Milutinovi, "An approach to optimization of airport taxiway scheduling and traversal under uncertainty," *Proceedings of the Institution of Mechanical Engineers G*, vol. 227, no. 2, pp. 273–284, 2013.

[5] G. L. Clare and A. G. Richards, "Optimization of taxiway routing and runway scheduling," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1000–1013, 2011.

[6] G. Keith, J. Tait, and A. Richards, "Efficient path optimization with terrain avoidance," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 2940–2949, August 2007.

[7] P. Burgain, E. Feron, and J. P. Clarke, "Collaborative virtual queue: benefit analysis of a collaborative decision making concept applied to congested airport departure operations," *Air Traffic Control Quarterly*, vol. 17, no. 2, pp. 195–222, 2009.

[8] J. Chen, S. Ravizza, and J. A. D. Atkin, "On the utilisation of fuzzy rule-based systems for taxi time estimations at airports," in *Proceedings of the 11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*, 2011.

[9] J. W. Smeltink, M. J. Soomer, P. R. de Waal, and R. D. van der Mei, *An Optimisation Model for Airport Taxi Scheduling*, Elsevier Science, 2004.

[10] R. Mori, "Aircraft ground-taxiing model for congested airport using cellular automata," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 1, pp. 180–188, 2013.

[11] S. Rathinam, J. Montoya, and Y. Jung, "An optimization model for reducing aircraft taxi times at the Dallas Fort Worth International Airport," in *Proceedings of the 26th International Congress of the Aeronautical Sciences (ICAS '08)*, pp. 14–19, 2008.

[12] J. W. Smeltink, M. J. Sooner, P. R. de Waal, and R. D. van der Mei, "An Optimization Model for Airport Taxi Scheduling," in *Proceedings of the INFORMS Annual Meeting (INFORMS '04)*, Denver, Colo, USA, 2004.

[13] R. Anderson and D. Milutinovic, *Optimization of Taxiway Traversal at Congested Airports*, American Institute of Aeronautics and Astronautics, 2010.

[14] P. C. Roling and H. G. Visser, "Optimal airport surface traffic planning using mixed-integer linear programming," *International Journal of Aerospace Engineering*, vol. 2008, Article ID 732828, 11 pages, 2008.

[15] C. Lesire, "Iterative planning of airport ground movements," in *Proceedings of the 4th International Conference on Research in Air Transportation*, pp. 147–154, 2010.

[16] D. B. Rappaport, P. Yu, K. Griffin, and C. Daviau, "Quantitative analysis of uncertainty in airport surface operations," in *Proceedings of the AIAA Aviation Technology, Integration and Operations Conference*, September 2009.

[17] S. Ravizza, J. A. D. Atkin, M. H. Maathuis, and E. K. Burke, "A combined statistical approach and ground movement model for improving taxi time estimations at airports," *Journal of the Operational Research Society*, vol. 64, no. 9, pp. 1347–1360, 2013.

[18] Y. Zhang, M. H. Hu, and Y. J. Wang, "The ground skidding time in aeronef airport is excellent to turn pattern of search," *Journal of Civil Aviation Flight University of China*, vol. 17, no. 5, pp. 3–6, 2006.

[19] J. You and S. C. Han, "Application of MAS to airport surface route optimization," *Computer and Communications*, vol. 26, no. 6, pp. 61–64, 2008.

[20] Y. Wang, M. Hu, and W. Su, "Dynamic taxiway routing algorithm based on conflict avoidance," *Journal of Southwest Jiaotong University*, vol. 44, no. 6, pp. 933–939, 2009.

[21] Z. Liu, H. Ge, and F. Qian, "Airport scheduling optimization algorithm based on genetic algorithm," *Journal of East China University of Science and Technology*, vol. 34, no. 3, pp. 392–394, 2008.