

Research Article

Multirunway Optimization Schedule of Airport Based on Improved Genetic Algorithm by Dynamical Time Window

Hang Zhou and Xinxin Jiang

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

Correspondence should be addressed to Hang Zhou; zhz@nuaa.edu.cn

Received 15 January 2015; Accepted 24 April 2015

Academic Editor: Wei (David) Fan

Copyright © 2015 H. Zhou and X. Jiang. 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.

Reasonable airport runway scheduling is an effective measure to alleviate air traffic congestion. This paper proposes a new model and algorithm for flight scheduling. Considering the factors such as operating conditions and flight safety interval, the runway throughput, flight delays cost, and controller workload composes a multiobjective optimization model. The genetic algorithm combined with sliding time window algorithm is used to solve the model proposed in this paper. Simulation results show that the algorithm presented in this paper gets the optimal results, the runway throughput is increased by 12.87%, the delay cost is reduced by 61.46%, and the controller workload is also significantly reduced compared with FCFS (first come first served). Meanwhile, compared with the general genetic algorithm, it also reduces the time complexity and improves real-time and work efficiency significantly. The analysis results can provide guidance for air traffic controllers to make better air traffic control.

1. Introduction

With the rapid development of the aviation industry, the passenger throughput of the busiest airport increases significantly and flight delays are becoming a hot focus. Air traffic congestion problem is attracting more and more attention as an important aspect of air traffic control.

Countries in the world take measures to reduce flight delays. The most simple way to solve the problem is to improve hardware facilities, such as making further expansion of airports and runways and improving the performance of communication navigation and surveillance equipment. It needs a long period and large investment to make it and the efficiency is low. By learning a variety of optimization algorithms, we analyze airport capacity and make a model of flight delays and allocate air traffic flow. How to dispatch approach and departure flights in the terminal area reasonably, how to reduce flight delays, and how to make full use of resources in the airspace have become the focus of the study [1]. Therefore, domestic and foreign scholars have done a lot of research: in the foreign countries, Vranas, D'Ariano, and many other scholars made a model of air traffic control in the airspace of the airports and designed appropriate algorithms to reduce the time that the flight waits in the air, optimize

flight scheduling and sequencing, and reduce flight delays, thus increasing air traffic flow [2–8]. In the year 2011, Bennell et al. designed algorithms for runway scheduling, thereby increasing runway capacity and reducing flight delays [9]. In the year 2013, Samà et al. adopted a dynamical time window algorithm to solve the problem of a lot of flight delays in case of heavy traffic [10].

Besides the above methods, many scholars in China also do a lot of research on the flight scheduling. Aircraft terminal controllers (ATC) usually apply FCFS (first come first served) strategy to make flight scheduling, but the fact shows that FCFS is not the best strategy to maximize the use of existing airport capacity and to reduce the average delay costs [11]. In the year 2004, Jiang and Yun presented an efficient optimization algorithm for aircraft landing scheduling problem, based on depth-first-search implicit enumeration algorithm, which is subject to MPS (maximal position shift) limiting, position lock limiting, moving time window, and other ATC restrictions. The results indicated that the algorithm could apply to dynamic ATC automation system with flexible goal cost function, efficiently minimized traffic delay, and improved landing capacity [12]; Wang et al. analyzed the impact of the relationship between the window dimension,

the position constraints, and the step-length, put forward a sliding window method, and gave a theoretical manifestation. The results showed that the quantity of calculation declined, which had been proved in an authentic simulation test [13]. In the year 2005, Jiang and Zhang proposed a new approach scheduling model based on earliest estimated arrival time. A calculation example showed that the new model and its algorithm gave better approach scheduling with obvious less computation [14]. In the year 2009, Yang introduced the sliding window algorithm by considering the total landing time and the amount of calculation. Aiming at the theoretical manifestation and simulation results got in this paper, they analyzed amount of calculation and evaluation versus window dimension and the step-length. The results validated the flexibility and practicability of this algorithm [15]. In the year 2011, Huang et al. developed an object function with total delays and proposed an improved genetic algorithm based on Grefenstette coding and receding horizon control strategy according to the characteristics of ASS problem. Then, the scheduling model with genetic algorithm was simulated. The results of the simulation indicated that the total delays were greatly reduced compared with FCFS algorithm and the proposed algorithm had better optimization performance than traditional genetic algorithm [16]. In the year 2012, J. Zhang and W. Zhang constructed the arriving and leaving aircraft scheduling model based on dynamic optimization. They viewed the scheduling process of arriving and leaving as an integer. So the controller's burden was alleviated and the loss caused by subjective factors was minimized. The model runs effectively and efficiently in the simulation experiment compared with FCFS [17]. In the year of 2014, we considered factors such as operating conditions and safety interval of multirunways; the maximum throughput and minimum flight delays losses as well as robustness were taken as objective functions; the model of optimization scheduling of approach and departure flights was established. The genetic algorithm was introduced to solve the model. Although the algorithm can improve the efficiency and reduce delay losses effectively, the efficiency of algorithm is not high enough [18].

Although the above scholars have done a lot of research on flight scheduling, there is still some deficiency. Some existing models were established in an ideal situation and did not take into account the actual factors such as the weather, air traffic control, and airport surrounding environment. Moreover, most models were only suited for single runway, only a little for multirunway. These models did not meet the development trend of the airport. In addition, the scheduling problem of approach and departure flights on multirunway will be more complex with the increasing number of runways, and the impact of human factors will also be deepened. It is needed to make further improvement to the algorithm and design better optimized program to get closer to the actual operational requirements.

To address the above problems, runway throughput, flight delay cost, and controller workload are created to be optimization objective, a multiobjective optimization model is established, and the genetic algorithm combined with a dynamical time window algorithm is proposed to solve the problem of flight scheduling on multirunway considering

the weather, air traffic control, route, and other factors. The reasons of using this innovative algorithm are shown as follows.

- (i) Genetic algorithm has strong global search capability, which will help to find the optimal results.
- (ii) Dynamical time window algorithm can greatly reduce the amount of calculation and also improve the quality of the solution.
- (iii) Hence, putting these algorithms together to solve the problem of flight scheduling, we cannot only get better results, but also increase operational efficiency and improve the robustness of the flight sequence.

The remainder of this paper is organized as follows. The multiobjective optimization model and the genetic algorithm combined with dynamical time window algorithm are developed in Section 2. After a simulation experiment is conducted, the results analysis and comparison with other algorithms are presented in Section 3. Finally, this paper concludes with a summary and future research direction in Section 4.

2. Model and Methods

2.1. Problem Description. At present, controllers usually use FCFS policy to make flight scheduling in the terminal area, which is mainly based on estimated leaving time or landing time of each flight. It is mostly dependent on the work experience of controllers to make sorting of flight. For the approach flight, controllers need to make a reasonable allocation of their runways and taxiways to make them land smoothly and reach the apron safely; for the departure flight, controllers need to make a reasonable allocation of their taxiways and runways to make them leave the airport safely and take off smoothly. Flight scheduling problem considered in this paper is mainly based on the existing research and combines genetic algorithm with dynamical time window algorithm to solve the problem. Different types of aircraft must comply with specific "minimum safety interval standards," rearrange the order of approach and departure flight in the queue. Dynamic sorting method is used in this paper to search all possible flight sequences and find out the best one with maximal runway throughput, minimal delay cost, and lowest controller workload.

For the mixed types of aircraft, the International Civil Aviation Organization (ICAO) specifies the minimum interval standards, defined as δ_{ij} ; the distance interval and its corresponding time interval are listed in Table 1 (it represents the condition of windless).

Figure 1 presents the approach and departure time of 3 kinds of flight sequences. Queue 1 represents the estimated approach and departure time queue; queue 2 represents the FCFS algorithm queue; queue 3 represents the optimized algorithm queue. The delay time of queue 2 and that of queue 3 are 221 s and 96 s, respectively. Obviously, rearranging the order of the flight can reduce the total delay time as well as delay cost (here, numbers 1 and 2 are approach flights and numbers 3 and 4 are departure flight).

TABLE 1: The safety interval of various types of aircraft.

Types of aircraft	Minimum distance interval/km			Minimum time interval/s		
	L	M	H	L	M	H
Leading	L	3	3	98	74	74
	M	4	3	138	74	74
	H	6	5	167	114	94

H, M, and L represent three kinds of flight type; they are heavy, medium, and light.

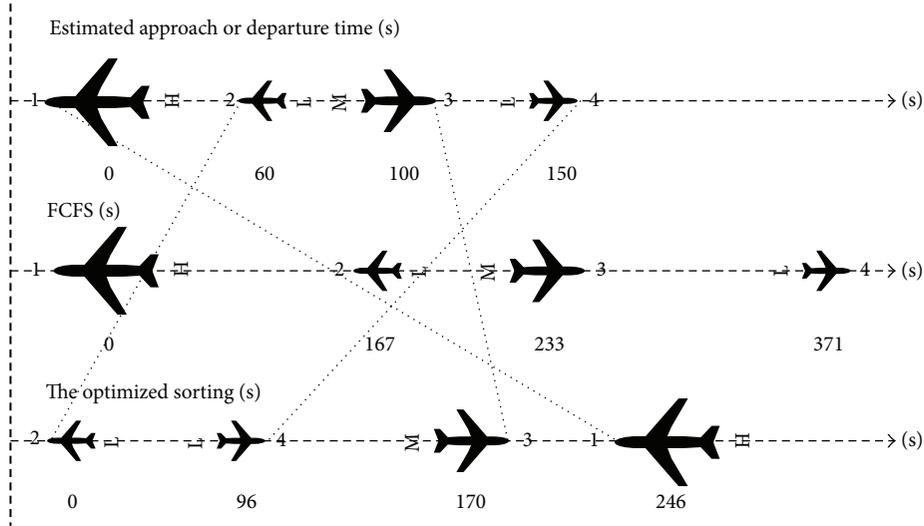


FIGURE 1: Approach and departure time of 3 kinds of flight sequences.

As can be seen from Figure 1, the optimized flight sequence saved 125 s compared with FCFS.

2.2. Model Introduction. Flight scheduling is a dynamic continuous process. It is needed to make adjustments according to the change of real-time information. Assume that there is u approach flight and v departure flight waiting for scheduling. They all need to meet the minimum safety interval. Arrange the order of flight to satisfy the desired object.

In this paper, we studied the approach and departure flight scheduling on multirunway; assume the following:

- (1) There is $u + v$ flight of M airlines waiting for scheduling. Controllers make a dynamic sorting of flight and assign a reasonable time for approach and departure flights.
- (2) The estimated time and actual time of each flight are different and can be determined at the time 0.
- (3) The airport studied in this paper contains multiple parallel runways, and each of them must comply with independent operation standards.
- (4) The information of approach and departure flights (including flight number, flight type, the estimated arriving or leaving time of flight, etc.) is known.

- (5) The capacity of the airport meets the assumption, the number of flight within the capacity of the airport.
- (6) The approach flight does not delay when it takes off at the departure airport and arrives on time at the terminal area waiting for landing.

2.3. Parameters and Symbols. (i) F_{AA} : the collection of approach flight, $F_{AA} = \{f_{A1}, f_{A2}, \dots, f_{Au}\}$.

(ii) F_{DD} : the collection of departure flight, $F_{DD} = \{f_{D1}, f_{D2}, \dots, f_{Dv}\}$.

(iii) F_i : the collection of all the flights f_i , $f_i \in F_i$, $F_{AA} \cup F_{DD} = F_i$, $i = \{1, 2, \dots, u + v\}$.

(iv) L : the collection of independent parallel runways, $L = \{1, 2, \dots, l\}$.

(v) $H_A^l = \begin{cases} 1, & \text{there is a flight arriving on the runway } l \\ 0, & \text{there is not a flight arriving on the runway } l \end{cases}$

(vi) $H_D^l = \begin{cases} 1, & \text{there is a flight leaving the runway } l \\ 0, & \text{there is not a flight leaving the runway } l \end{cases}$

(vii) TAY_{\max} : the maximum delay time of approach flight.

(viii) TDY_{\max} : the maximum delay time of departure flight.

(ix) STA_i : the actual time of flight f_i arriving at the airport.

(x) ETA_i : the estimated time of flight f_i arriving at the airport.

(xi) STD_i : the actual time of flight f_i leaving the airport.

(xii) ETD_i : the estimated time of flight f_i leaving the airport.

(xiii) f_{ip} : the initial position of flight f_i in the flight queue.

(xiv) f_{iq} : the location of flight f_i in the optimized flight queue.

(xv) d : the maximum displacement constraint MPS (maximum position shift).

(xvi) $\alpha_{ij} = \begin{cases} 1, & \text{the flight } f_j \text{ kept landing after flight } f_i \\ 0, & \text{otherwise.} \end{cases}$

(xvii) $\beta_{ij} = \begin{cases} 1, & \text{the flight } f_j \text{ kept leaving after flight } f_i \\ 0, & \text{otherwise.} \end{cases}$

(xviii) $\gamma_{ij} = \begin{cases} 1, & \text{the flights } f_i \text{ and } f_j \text{ are landing on the same runway} \\ 0, & \text{the flights } f_i \text{ and } f_j \text{ are not landing} \\ & \text{on the same runway.} \end{cases}$

(xix) δ_{ij} : the minimum safety interval between two successive approaching flights (they are landing on the same runway and $i \neq j$). Different flights have different safety interval; meanwhile, on the same runway, if the same type of flight has different types of approach and departure, its safety interval is different too.

(xx) C_{AH}^i : the unit time delay cost of approach flight F_{AA} .

(xxi) C_{DH}^i : the unit time delay cost of departure flight F_{DD} .

Different types of flight have different unit time delay costs. In general, the larger flight has higher unit time delay cost.

2.4. Modeling

2.4.1. The Objective Function of Runway Throughput. Suppose that, on runway l , the actual scheduling time of the first flight is T_{fl}^1 and the actual scheduling time of the last flight is T_{fl}^{u+v} , so all the time for scheduling the flight queue is $T_{all} = (T_{fl}^{u+v} - T_{fl}^1)$. In order to maximize the throughput of the runway, it is required that the time of finishing the scheduling of the flight queue is minimum. So the smaller T_{all} is, the larger runway throughput is.

$$T(f_i) = \begin{cases} H_A^l (STA_i - ETA_i - \Delta t) + H_D^l (STD_i - ETD_i - \Delta t) & STA_i - ETA_i > \Delta t \text{ or } STD_i - ETD_i > \Delta t \\ 0 & STA_i < ETA_i \text{ or } STD_i < ETD_i. \end{cases} \quad (4)$$

The delay cost of the flight queue.

Suppose that there is a flight whose actual time is later than estimated time; total delay cost is W_L . Meanwhile, there is b flight whose actual time is earlier than estimated time, because delay time is 0, so total delay cost is $W_E = 0$.

So, the total number of $a+b$ flights is all the approach and departure flight $u+v$:

$$a + b = u + v. \quad (5)$$

The delay cost of all the late flights is

$$W_L = \sum_{l=1}^l \sum_{i=1}^m [C_{AH}^i H_A^l (STA_i - ETA_i - \Delta \tau) + C_{DH}^i H_D^l (STD_i - ETD_i - \Delta \tau)]. \quad (6)$$

The objective function of the maximum runway throughput is

$$\text{Capacity} = \min T_{all} = \min (T_{fl}^{u+v} - T_{fl}^1). \quad (1)$$

2.4.2. The Objective Function of Flight Delays Cost. In most cases, as for approach and departure flights, their actual arriving or leaving time is almost impossible to be the same as their estimated time, so we make the following provision: supposing a certain range, the actual time later than the estimated time within Δt is not regarded as delay. Furthermore, if the actual time is earlier than the estimated approach or departure time, the delay time is a negative value; it is not realistic, so we predetermine that if there is a flight arriving earlier than its estimated time, there is no delay. The total delay time of the whole flight sequence could be expressed as follows.

(1) If flight f_i is an approach flight, the delay time is A_{delay} :

$$A_{\text{delay}} = \begin{cases} STA_i - ETA_i - \Delta t & f_i \in F_{AA}, STA_i - ETA_i > \Delta t \\ 0 & f_i \in F_{AA}, STA_i < ETA_i. \end{cases} \quad (2)$$

(2) If flight f_i is a departure flight, the delay time is D_{delay} :

$$D_{\text{delay}} = \begin{cases} STD_i - ETD_i - \Delta t & f_i \in F_{DD}, STD_i - ETD_i > \Delta t \\ 0 & f_i \in F_{DD}, STD_i < ETD_i. \end{cases} \quad (3)$$

So, the delay time of flight f_i is

The total delay cost of the entire flight queue is

$$W = W_L + W_E. \quad (7)$$

Assume that there are g kinds of sorting programs and the delay cost of each program is W_1, W_2, \dots, W_g , so the objective function of flight delays cost is

$$\text{Cost} = \min (W_1, W_2, \dots, W_g). \quad (8)$$

2.4.3. The Objective Function of Controller Workload. The object of controller workload can be achieved by the number of the flight being adjusted; the larger it is, the heavier workload the controller has. Consequently, in order to reduce the controller workload, we should try to keep the actual scheduling time of the flight consistent with the estimated

time, so that we can reduce the number of flights being adjusted and also improve the robustness of the entire flight queue. The standard deviation of position changing value for each flight in the flight queue is introduced as a measurement for controller workload.

Here, x_i is defined as the position changing value of the flight f_i , so $x_i = |f_{iq} - f_{ip}|$. \bar{x} is the average variable quantity of displacement:

$$\bar{x} = \frac{1}{u+v} \sum_{i=1}^{u+v} x_i = \frac{1}{u+v} \sum_{i=1}^{u+v} |f_{iq} - f_{ip}|. \quad (9)$$

σ^2 is the variance of flight variable quantity of displacement:

$$\sigma^2 = \frac{1}{u+v} \sum_{i=1}^{u+v} (x_i - \bar{x})^2. \quad (10)$$

σ is the standard deviation of flight variable quantity of displacement:

$$\sigma = \sqrt{\frac{1}{u+v} \sum_{i=1}^{u+v} (x_i - \bar{x})^2}. \quad (11)$$

The larger the σ value is, the greater workload the controller has. So, the objective function of controller workload is

$$\text{Work}_{\text{load}} = \min \sigma = \sqrt{\frac{1}{u+v} \sum_{i=1}^{u+v} (x_i - \bar{x})^2}. \quad (12)$$

2.5. Constraints. Consider

$$t_{f_i} - t_{f_{i-1}} \geq \max(m_{i,i-1}, w_{i,i-1}), \quad i = 1, 2, \dots, u+v \quad (13)$$

$$STA_i - ETA_i \leq TAY_{\max}, \quad (14)$$

$$STD_i - ETD_i \leq TDY_{\max}, \quad (15)$$

$$H_A^l \leq 1, \quad (16)$$

$$H_D^l \leq 1, \quad (17)$$

$$STD_j \geq STD_i + \delta_{ij}, \quad i \neq j, \quad (18)$$

$$\alpha_{ij} + \alpha_{ji} \leq 1, \quad (19)$$

$$\beta_{ij} + \beta_{ji} \leq 1, \quad \forall (i, j) \in F_i, \quad i \neq j,$$

$$\sum_{i=1}^u \alpha_{ij} = 1 \quad (20)$$

$$\sum_{i=1}^v \beta_{ij} = 1, \quad \forall j \in F_i, \quad i \neq j$$

$$\sum_{j=1}^u \alpha_{ij} = 1 \quad (21)$$

$$\sum_{j=1}^v \beta_{ij} = 1, \quad \forall i \in F_i, \quad i \neq j$$

$$|f_{iq} - f_{ip}| \leq d. \quad (22)$$

Equation (13) indicates that adjacent flight should meet certain wake separation and vortex separation. $u+v$ is the number of flight, t_{f_i} is the scheduling time allocated to flight f_i , $m_{i,i-1}$ is the vortex separation, and $w_{i,i-1}$ is the wake separation.

Equation (14) is to restrain the delay time of the approach flight; it should not be larger than the maximum approach delay time.

Equation (15) is to restrain the delay time of the departure flight; it should not be larger than the maximum departure delay time.

Equations (16) and (17) are the constraint of runways; each flight only has one runway for arriving or leaving.

Equation (18) indicates the safety interval the flights need to meet if there are two successive flights arriving or leaving the same runway (suppose that flight f_i is the former and the flight f_j is the latter).

Equation (19) indicates that, for two approach flights f_i and f_j , either f_i is the leading and f_j is the trailing or f_i is the trailing and f_j is the leading.

Equations (20) and (21) indicate that any approach flight only has one leading/trailing.

Equation (22) indicates that the variable quantity of displacement should not exceed maximum displacement constraint.

2.6. Design of Sliding Time Window Algorithm. Flight sorting is a continuous dynamic process. Make sorting of $u+v$ approach and departure flights; there will be $(u+v)!$ results. Among these results, there must be a flight sequence meeting all objectives, but the calculation is great and computational complexity is high. However, the dynamical time window algorithm can largely reduce the amount of computation; its idea is as follows: according to the current real-time information, the former p flight waiting to be sorted forms a time window, makes full permutation of these p flights, and selects the q flight with the best result as the former q flight for the final sorting result and keeps its order. Then, the time window moves q positions backward, reconstitutes a time window which contains p flight, seeks for the optimal p flight in the time window, and picks out the best q flight adding to the final flight sorting result. A similar operation is repeated until all flight sorting has been finished, thus achieving the final total flight sorting results. Here, p is the size of time window, q is the sliding step, and $0 < p < n$.

For example, $u+v=9$, $p=5$, and $q=2$; the process is shown in Figure 2.

Step 1. In the initial flight queue, put flights 1–5 into the time window; make full permutation of these five flights to find

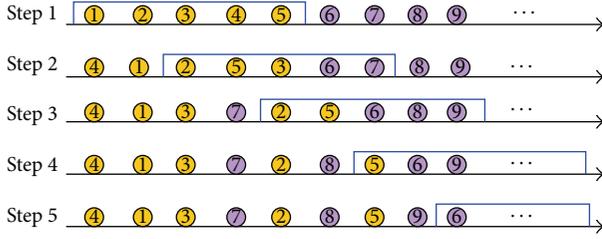


FIGURE 2: The diagrammatic sketch of dynamical time window algorithm.

the best sorting result. At last, put flight 4 and flight 1 as the top two flights in the final result.

Step 2. The time window moves 2 positions backward; flight 6 and flight 7 are included into the time window. Make full permutation of flights 2, 3, 5, 6, and 7 and find the best sorting result. Put the former two flights 6 and 7 into the final result.

Step 3. The time window continues moving 2 positions backward; flight 8 and flight 9 are included into the time window. Make full permutation of flights 2, 5, 6, 8, and 9 and find the best sorting result. Put the former two flights 2 and 8 into the final result.

Step 4. The time window keeps on moving 2 positions backward; the subsequent two flights are included into the time window. Make full permutation of these five flights in the time window and find the best sorting result. Put flight 5 and flight 9 into the final result.

Step 5. The time window goes on moving backward, repeats the above steps until the last flight finishes sorting, and then gets the best final sorting result, that is, 4-1-3-7-2-8-5-9-6.

2.7. Design of Improved Genetic Algorithm. According to the characteristics of multirunway flight scheduling problem, it is needed to make improvement of general genetic algorithm. Conduct encoding, selection, crossover, and mutation operations of the flights in the window, thereby generating optimal sequence.

2.7.1. Coding Scheme. The flight scheduling problem studied in this paper is a multirunway problem, so make improvement of coding scheme and design double string gene coding based on the flight number and runway number. The first code string is flight chromosome, showing flight number; the second code string is runway chromosome, showing the corresponding runway number of the flight (assuming there are two runways, resp., 0 and 1). Assign a runway for each flight randomly as shown in Figure 3.

In the process of genetic operation for double chromosomes, the crossover, mutation operation makes it easy to produce meaningless flight sequence. In order to avoid it, Grefenstette coding method is used to produce efficient chromosomes. So each individual chromosome could represent

3	1	5	2	7	6	4	8
0	1	1	0	0	1	0	1

FIGURE 3

3	1	3	1	3	2	1	1
0	1	1	0	0	1	0	1

FIGURE 4

a meaningful flight sequence and reduce the complexity of the algorithm.

According to Grefenstette coding method, the double chromosome given above could be described as shown in Figure 4.

2.7.2. Generate Initial Population. Apply Grefenstette coding method to generate double chromosomes randomly, and each of them is meaningful.

Judge if the chromosomes are qualified or not: firstly, decode the chromosomes above; then, make sorting of the flights on each runway according to their estimated approach or departure time, and assign scheduling time for each flight on the basis of constraints. If it is prior to the earliest time, make the scheduling time equal to the earliest time. If it is later than the latest time, then the individual is invalid and needs to generate new individual, until it meets the constraints.

2.7.3. Fitness Function. Fitness plays the role of judging if the individual is good or bad and drives the group moving to the best evolution direction in the process of genetic algorithm. Because the fitness function is always nonnegative, under any circumstances, the bigger it is, the better it is. The model established in this paper is a multiobject optimization model; make improvement of general genetic algorithm; the sorting method according to pros and cons of all objectives is determined to calculate fitness. Make descending order of the individual according to their quality of the performance. As for each object $f_i(x)$, all the individuals will generate a feasible sequence X_i on the basis of the value of the object function. Sorting for each object, we can get overall performance of the individual for all object functions. The calculation of fitness can be calculated by [19]

$$F_o(X_j) = \begin{cases} (N - R_o(X_o))^2 & R_o(X_j) > 1 \\ kN^2 & R_o(X_j) = 1 \end{cases} \quad (23)$$

$$F(X_j) = \sum_{i=1}^n F_i(X_j). \quad (24)$$

In (23), $o = 1, 2, \dots, n$; $j = 1, 2, \dots, N$; n is the number of objective functions; N is all individuals of the population; X_j is the individual j of the population; $R_o(X_j)$ is the serial number of the individual j after all individuals are being

sorted in view of the objective o ; $F_o(x_j)$ is the fitness value of the individual X_j on account of the objective o ; $F(X_j)$ is the comprehensive fitness value of the individual X_j on account of all objectives; k is a constant within $(1, 2)$, for increasing the fitness value when the performance of individual is optimal. As can be seen from the equations above, optimal overall performance of individuals can get better fitness to gain more opportunities to participate in evolution.

2.7.4. Selection. Selection is the process to choose the individual with strong vitality to generate new population. Selection operator is used to select the survival of the fitness: the individual with higher fitness has more probability inherited to next generation, while the individual with lower fitness has little probability inherited to next generation. The roulette selection operator is used in this paper; namely, the probability of the fitness in proportion decides the possibility of its descendants going or staying. If a certain individual is X_j , its fitness is $F(X_j)$; the probability to be selected can be expressed as

$$P_{X_j} = \frac{F(X_j)}{\sum_{j=1}^N F(X_j)}. \quad (25)$$

2.7.5. Crossover. Crossover is to pick up two individuals from the population with higher fitness and change a bit or some bits of them to produce new individuals. The offspring inherits the basic characteristics of their parents. Due to the use of Grefenstette coding method, there is no need to worry about the meaninglessness of the chromosome after crossover. Single-point crossover method is used in this paper [19]. For example, write two flight sequences A and B as parents and randomly select crossover point, shown as in Figure 5.

After performing a simple single-point crossover, we can get what is shown in Figure 6.

As for the new generated individuals after crossover, the constraints are applied to make a judgment; if the offspring does not meet the constraints, then execute the crossover operation again, until producing the offspring that is consistent with the constraints.

2.7.6. Mutation. In order to speed up the operation efficiency of the algorithm, design two kinds of mutation operator according to the coding scheme.

- (1) Exchange two consecutive flight chromosomes on the same runway. Select a certain position of a runway sequence randomly and exchange two consecutive flight chromosomes of the position.
- (2) Exchange two flight chromosomes at any position on the different runway. Select a certain position of two runway sequences randomly and exchange their flight chromosomes.

As for the new individuals after mutation, the constraints are applied to make a judgment; if the offspring generated does not meet the constraints, then execute the mutation

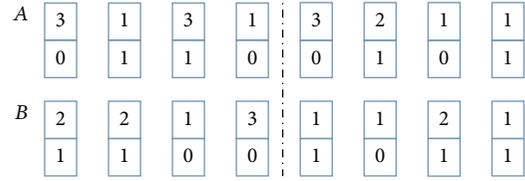


FIGURE 5

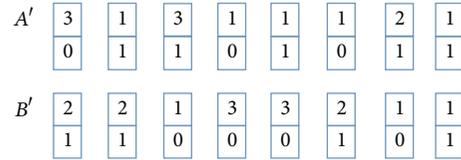


FIGURE 6

operation again, until producing the offspring that is consistent with the constraints.

3. Simulation and Analysis

3.1. Simulation. In order to verify the performance of the algorithm proposed in this paper, the data of 24 flights in a continuous period based on Chengdu Shuangliu International Airport is shown as an example. There are 12 approach flights and 12 departure flights and 2 mutually independent parallel runways. Moreover, $\Delta\tau = 120$ s. The initial flight data is shown in Table 2.

Run the algorithm in this paper. Based on the selected 24 approach and departure flights, set the size of time window $p = 15$ and sliding step $q = 3$. According to the estimated time, number 1–7 flights and number 12–20 flights are included in the window. Apply genetic algorithm in the window for these 15 flights; select the former 3 flights of the best sequence as the former 3 flights of the final sequence. The time window moves 3 positions backward and repeats the above steps until the last flight sorting is finished. Set displacement constraint $d = 8$. In the genetic algorithm, the crossover probability is 0.8, the mutation probability is 0.2, the generation gap is 0.9, the elimination rate is 0.2, k in the fitness function is 1.5, the size of population is 100, and evolution algebra is 2000. After several rounds of selection, crossover, and mutation, the evolutionary trends of the genetic algorithm are shown in Figure 7.

3.2. Analysis. Make sorting of the flight according to the optimization algorithm and select 6 kinds of optimization schemes; their values of each objective function are shown in Table 3.

Draw the comparison diagram of 6 optimization schemes according to Table 3, and compare them with the results of FCFS algorithm. The delay cost of FCFS is 101115.8/CNY; the runway throughput is 1172/s. The comparison results are shown in Figure 8. As can be seen from Figure 8, compared with FCFS, the results of the optimization algorithm have

TABLE 2: The initial data of approach and departure flights.

Airline	Number	Flight number	Type	Unit time delay cost of departure	Estimated departure time	Runway
CES	1	MU5990	M	2	0:00:00	0
CES	2	MU2342	L	1	0:00:00	1
CCA	3	CA2342	L	1.1	0:00:00	0
CSC	4	3U8731	M	2.1	0:00:00	1
CSZ	5	ZH1407	M	2.2	0:04:00	1
CSZ	6	ZH4307	M	2.2	0:04:00	0
CCA	7	CA426	L	1.2	0:04:00	1
CCA	8	CA4307	H	4	0:04:00	0
CSC	9	3U8668	L	1.1	0:08:00	0
CES	10	MU2414	L	1	0:08:00	1
CES	11	MU3005	M	2.1	0:08:00	0
CSC	12	3U8869	H	4.1	0:08:00	1

Airline	Number	Flight number	Type	Unit time delay cost of approach	Estimated approach time	Runway
CSC	13	3U8676	M	42	0:00:00	0
CCA	14	CA4434	H	62	0:01:06	1
CSC	15	3U8648	L	23	0:01:47	0
CCA	16	CA1415	L	23.1	0:02:35	1
CSZ	17	ZH1415	M	42.2	0:03:19	0
CSC	18	CA1945	L	23	0:03:55	1
CES	19	MU5401	M	41.3	0:04:07	0
CSZ	20	ZH1915	H	62.1	0:05:14	1
CSC	21	3U8702	M	42	0:02:52	0
CSC	22	3U8628	L	23	0:06:23	1
CCA	23	CA4392	L	23.1	0:07:34	0
CSZ	24	ZH2306	M	42.2	0:08:48	1

Note: CCA: Air China; CSC: Sichuan Airlines; CSZ: Shenzhen Airlines; CES: China Eastern; CHH: Hainan Airlines.

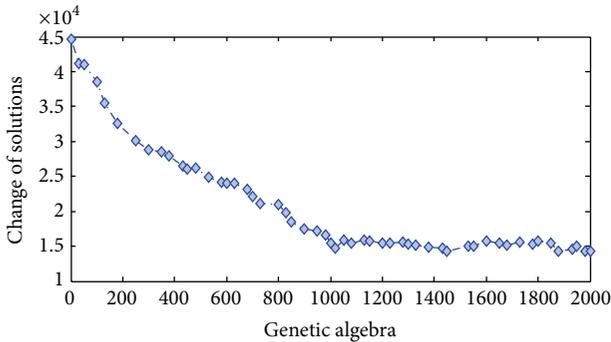


FIGURE 7: The evolution trend graph of genetic algorithm.

been improved greatly compared with the FCFS algorithm with lower delay cost and higher runway throughput.

As for the selected 6 optimization schemes, based on multiobjective decision, the optimal cardinal number summation method is adopted to select the best scheme in this paper [20]. Specific method is described as follows.

According to the value of each objective function in Table 3, suppose that 6 kinds of schemes are x^1, x^2, \dots, x^6 ; the three objective functions are A_1, A_2, A_3 , so that

$$y_1^1 = A_1(x^1) = 1018,$$

$$y_1^2 = A_1(x^2) = 1100,$$

$$y_1^3 = A_1(x^3) = 1072,$$

$$y_1^4 = A_1(x^4) = 1022,$$

$$y_1^5 = A_1(x^5) = 994,$$

$$y_1^6 = A_1(x^6) = 1066,$$

$$y_2^1 = A_2(x^1) = 64988.8,$$

$$y_2^2 = A_2(x^2) = 56622.5,$$

$$y_2^3 = A_2(x^3) = 59488.7,$$

$$\begin{aligned}
 y_2^4 &= A_2(x^4) = 41369.5, \\
 y_2^5 &= A_2(x^5) = 67218.4, \\
 y_2^6 &= A_2(x^6) = 44634.6, \\
 y_3^1 &= A_3(x^1) = 2.7346, \\
 y_3^2 &= A_3(x^2) = 2.6812, \\
 y_3^3 &= A_3(x^3) = 2.3886, \\
 y_3^4 &= A_3(x^4) = 1.9539, \\
 y_3^5 &= A_3(x^5) = 2.8974, \\
 y_3^6 &= A_3(x^6) = 2.0648.
 \end{aligned}
 \tag{26}$$

The quantitative matrix is

$$Y = \begin{bmatrix} 1018 & 1100 & 1072 & 1022 & 994 & 1066 \\ 64988.8 & 56622.5 & 59488.7 & 41369.5 & 67218.4 & 44634.6 \\ 2.7346 & 2.6812 & 2.3886 & 1.9539 & 2.8974 & 2.0648 \end{bmatrix}. \tag{27}$$

Find the minimum value of the objective function; the calculation of optimal cardinal number is shown in the following formula:

$$z_k^i = 1 - \frac{y_k^i}{\sqrt{\sum_{i=1}^s y_k^{i,2}}}. \tag{28}$$

Here, i is the number of optimization schemes; k is the number of object functions; that is, $i = 1, 2, \dots, 6, k = 1, 2, 3$. So the optimal cardinal number matrix is

$$Z = \begin{bmatrix} 0.8421 & 0.8157 & 0.8249 & 0.8409 & 0.8495 & 0.8269 \\ 0.7799 & 0.8329 & 0.8156 & 0.9108 & 0.7646 & 0.8962 \\ 0.7971 & 0.8049 & 0.8452 & 0.8964 & 0.7722 & 0.8843 \end{bmatrix}. \tag{29}$$

Now suppose that the weight coefficient of each object function is 0.2, 0.4, and 0.4. According to formula (30), make weighted sum and sorting of these optimal cardinal numbers:

$$u_i = \sum_{k=1}^m w_k \cdot z_k^i. \tag{30}$$

So,

$$\begin{aligned}
 u_1 &= 0.7992, \\
 u_2 &= 0.8183, \\
 u_3 &= 0.8293, \\
 u_4 &= 0.8911,
 \end{aligned}$$

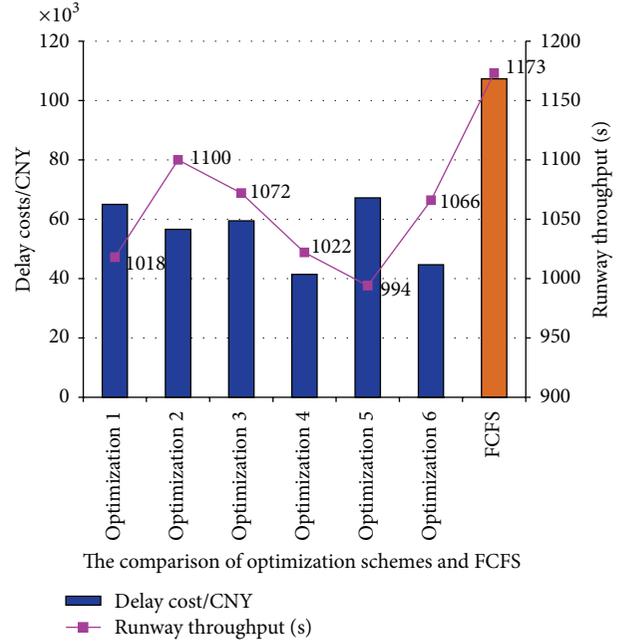


FIGURE 8: Comparison of 6 optimization schemes with the FCFS of object function value.

$$\begin{aligned}
 u_5 &= 0.7846, \\
 u_6 &= 0.8776.
 \end{aligned}
 \tag{31}$$

Because of $u_4 > u_6 > u_3 > u_2 > u_1 > u_5$, the number 4 scheme is the best scheme under the given weight coefficient. In addition, the decision makers can change the weight coefficient of each objective function according to their preferences, thereby selecting the best scheme to meet their demands.

The results of the number 4 scheme are shown in Table 4. The genetic algorithm combined with the dynamical time window algorithm is applied in this paper, making optimization scheduling of the approach and departure flights, and obtaining ideal results. Compared with the FCFS algorithm, the runway throughput is improved by 12.87%; delay cost is reduced by 61.46%. In addition, in order to verify the effect of the algorithm, make a comparison of the genetic algorithm combined with dynamical time window optimization algorithm used in this paper and the general genetic algorithm used in other literature. The comparison results are shown in Table 5.

According to Table 5, the delay cost of three algorithms for each flight is compared; the 24 flight delays' cost comparison chart is shown in Figure 9.

Compared with the FCFS algorithm, the runway throughput of genetic algorithm was improved by 13.21% compared to FCFS; the delay cost decreased by 62.95%; compared with the general genetic algorithm, the runway throughput of the genetic algorithm combined with dynamical time window algorithm only reduced by 0.39%; delay cost only increased by 3.86%, while the controller

TABLE 3: The objective function value of the 6 optimization schemes.

Optimization schemes	1	2	3	4	5	6
Runway throughput/s	1018	1100	1072	1022	994	1066
Delay cost/CNY	64988.8	56622.5	59488.7	41369.5	67218.4	44634.6
Controller displacement	2.7346	2.6812	2.3886	1.9539	2.8974	2.0648

TABLE 4: The optimized flight data.

Runway 0						Runway 1					
Airline	Number	Flight number	Type	Time/s	Delay cost/CNY	Airline	Number	Flight number	Type	Time/s	Delay cost/CNY
CSC	13	3U8676	M	0	0	CSC	15	3U8648	L	0	0
CSZ	17	ZH1415	M	74	0	CCA	3	CA2342	L	96	0
CES	19	MU5401	M	148	0	CCA	14	CA4434	H	162	0
CES	1	MU5990	M	224	208	CES	2	MU2342	L	302	182
CSC	21	3U8702	M	300	336	CCA	16	CA1415	L	398	2841.3
CSC	4	3U8731	M	376	537.6	CCA	8	CA4307	H	464	416
CSZ	24	ZH2306	M	452	0	CSZ	20	ZH1915	H	542	6706.8
CSZ	5	ZH1407	M	528	369.6	CCA	7	CA426	L	682	386.4
CES	11	MU3005	M	602	4.2	CSC	12	3U8869	H	756	639.6
CSC	18	CA1945	L	722	8441	CCA	23	CA4392	L	896	7438.2
CSZ	6	ZH4307	M	788	941.6	UEA	9	EU2705	L	992	431.2
CES	10	MU2414	L	926	326						
CSC	22	3U8628	L	1022	11164						

Runway throughput: 1022 s

Delay cost: 41369.5CNY

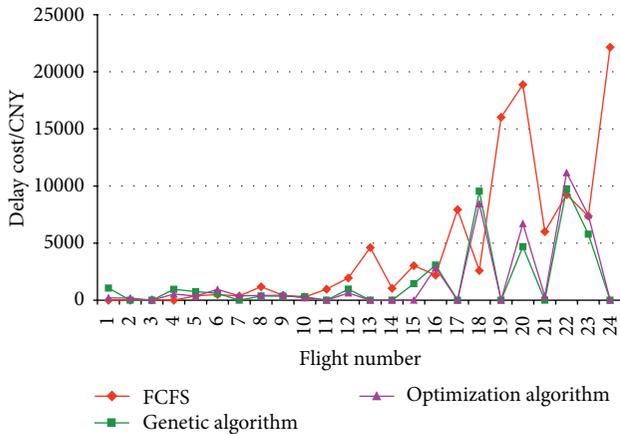


FIGURE 9: Comparison of the delay costs of 24 flights for three algorithms.

workload has decreased by 26.35%; meanwhile, as for the complexity of the algorithm, the optimization algorithm increased nearly by 62.5% compared with general genetic algorithm, which greatly saves the calculation time. Although the optimization algorithm used in this paper did not achieve the best global optimization results as the general genetic algorithm, it reduced the computational complexity, improved efficiency, and reduced controller workload to a certain degree. Thus, the optimization algorithm used in this paper could also get a good optimization result and significantly saved the calculation time.

4. Conclusion

For the air traffic congestion problem, this paper develops a new algorithm based on multiobjective optimization for flight scheduling model. The runway throughput, flight delays cost, and controller workload are regarded as objective, respectively. A new improved genetic algorithm combined with a dynamical time window algorithm is created in the paper. Compared with FCFS and general genetic algorithm, the main advantages of the model and improved algorithm proposed in this paper are as follows:

- (1) The runways studied in this paper are multiple parallel runways, which accords with the development trend of airport nowadays. According to the characteristic of multirunway, double string gene coding based on the flight number and runway is created to make improvement of genetic algorithm, and the optimal results getting in this method are more ideal.
- (2) The multiobjective optimization could find an optimum solution of the flight scheduling. According to the characteristic of multiobjective optimization model, define the fitness value of individual by the comprehensive values of all optimization objects; then, get the better offspring. Besides, the decision makers could change the weight coefficient of the objective functions according to their need, so they can get the satisfying flight sequence.
- (3) The displacement constraint of the flight moving reflected the fairness among the flights taking off

TABLE 5: The delay cost of 3 kinds of algorithms.

Number	FCFS				Genetic algorithm			Optimized algorithm		
	Estimated departure time	Actual departure time	Runway	Delay cost	Departure time	Runway	Delay cost	Optimized departure time	Runway	Delay cost
1	0:00:00	0:00:00	0	0	0:10:50	1	1060	0:03:44	0	208
2	0:00:00	0:00:00	1	0	0:00:00	1	0	0:05:02	1	182
3	0:00:00	0:02:34	0	37.4	0:01:38	1	0	0:01:36	1	0
4	0:00:00	0:02:02	1	4.2	0:09:36	1	957.6	0:06:16	0	537.6
5	0:04:00	0:08:54	1	382.8	0:11:44	0	756.8	0:08:48	0	369.6
6	0:04:00	0:09:43	0	490.6	0:10:30	0	594	0:12:48	0	941.6
7	0:04:00	0:11:12	1	374.4	0:03:14	0	0	0:11:22	1	386.4
8	0:04:00	0:10:57	0	1188	0:07:34	0	376	0:07:44	1	416
9	0:08:00	0:16:29	0	427.9	0:15:20	0	352	0:16:32	1	431.2
10	0:08:00	0:16:41	1	401	0:16:58	0	418	0:15:26	0	326
11	0:08:00	0:17:43	0	972.3	0:08:22	1	0	0:10:02	0	4.2
12	0:08:00	0:17:55	1	1947.5	0:13:56	1	967.6	0:12:36	1	639.6

Number	FCFS				Genetic algorithm			Optimized algorithm		
	Estimated approach time	Actual approach time	Runway	Delay cost	Approach time	Runway	Delay cost	Optimized approach time	Runway	Delay cost
13	0:00:00	0:03:50	0	4620	0:01:14	0	0	0:00:00	0	0
14	0:01:06	0:03:23	1	1054	0:02:44	1	0	0:02:42	1	0
15	0:01:47	0:05:59	0	3036	0:04:50	0	1449	0:00:00	1	0
16	0:02:35	0:06:10	1	2194.5	0:06:28	0	3072.3	0:06:38	1	2841.3
17	0:03:19	0:08:27	0	7933.6	0:00:00	0	0	0:01:14	0	0
18	0:03:55	0:07:48	1	2599	0:12:50	1	9545	0:12:02	0	8441
19	0:04:07	0:12:35	0	16024.4	0:05:52	1	0	0:02:28	0	0
20	0:05:14	0:12:18	1	18878.4	0:08:52	0	4693.6	0:09:02	1	6706.8
21	0:02:52	0:07:15	0	6006	0:04:38	1	0	0:05:00	0	336
22	0:06:23	0:15:05	1	9246	0:16:16	1	9756.2	0:17:02	0	11164
23	0:07:34	0:14:53	0	7368.9	0:13:44	0	5775	0:14:56	1	7438.2
24	0:08:48	0:19:33	1	22155	0:07:06	1	0	0:07:32	0	0

Total			101115.8/CNY			39773.1/CNY			41369.5/CNY		
Controller workload						2.6531			1.9539		

and landing, reduced the workload of the controllers, and improved the robustness of the flight sequence. It improved the efficiency greatly with less time complexity, which provided good guidance for air traffic controllers to make flight scheduling.

However, because of the complexity of flight scheduling of multiobjective on multirunway, it is needed to make adjustment according to real-time information. So, in the future study, make further improvement of the algorithm so as to fit the actual complex scheduling environment and achieve a mutually satisfactory outcome.

Conflict of Interests

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

Acknowledgments

The authors thank the reviewers for helping them to improve this paper. This work is supported by the Fundamental Research Funds for the Central Universities, no. NS2014064.

References

- [1] D. A. Lee, C. Nelson, and G. Shapiro, *The Aviation System Analysis Capability Airport Capacity and Delay Models*, National Aeronautics and Space Administration, Langley Research Center, 1998.
- [2] P. B. Vranas, D. Bertsimas, and A. R. Odoni, "Dynamic ground-holding policies for a network of airports," *Transportation Science*, vol. 28, no. 4, pp. 275–291, 1994.
- [3] J. C. Beck and P. Refalo, "A hybrid approach to scheduling with earliness and tardiness costs," *Annals of Operations Research*, vol. 118, no. 1–4, pp. 49–71, 2003.

- [4] A. Mukherjee, "Dynamic stochastic optimization models for air traffic flow management," Working Papers, Institute of Transportation Studies, Berkeley, Calif, USA, 2004.
- [5] A. D'Ariano, P. D'Urgolo, D. Pacciarelli, and M. Pranzo, "Optimal sequencing of aircrafts take-off and landing at a busy airport," in *Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems (ITSC '10)*, pp. 1569–1574, Funchal, Portugal, September 2010.
- [6] Y. Eun, I. Hwang, and H. Bang, "Optimal arrival flight sequencing and scheduling using discrete airborne delays," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 359–373, 2010.
- [7] Z.-H. Zhan, J. Zhang, Y. Li et al., "An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 399–412, 2010.
- [8] M. Pistelli, A. D'Ariano, and D. Pacciarelli, "Optimization models and algorithms for air traffic control in the airspace of busy airports," Tech. Rep. RT-DIA-185-11, Roma Tre University, 2011.
- [9] J. A. Bennell, M. Mesgarpour, and C. N. Potts, "Airport runway scheduling," *4OR*, vol. 9, no. 2, pp. 115–138, 2011.
- [10] M. Samà, A. D'Ariano, and D. Pacciarelli, "Rolling horizon approach for aircraft scheduling in the terminal control area of busy airports," *Procedia—Social and Behavioral Sciences*, vol. 80, pp. 531–552, 2013.
- [11] M. Ignaccolo and G. Inturri, "Modelli Analitici e di Simulazione per la Valutazione delle Prestazioni in ambito Aeroportuale," in *Metodi e Tecnologie per l'Ingegneria dei Trasporti*, Franco Angeli, Milan, Italy, 2001.
- [12] Y. Jiang and P. Yun, "Mathematical model and analysis of runway landing capacity," *Journal of Southwest Jiaotong University*, vol. 39, no. 1, pp. 42–46, 2004.
- [13] L.-L. Wang, Z.-K. Shi, and Z.-N. Zhang, "Optimizing slip window algorithm for sequencing of scheduled arriving aircrafts," *Journal of Civil Aviation*, vol. 22, no. 6, pp. 18–21, 2004.
- [14] B. Jiang and F.-Q. Zhang, "Approach scheduling model based on earliest estimated arrival time and its algorithm," *Journal of Southwest Jiaotong University*, vol. 40, no. 4, pp. 509–512, 2005.
- [15] K. Yang, "Arriving aircrafts scheduling algorithm based on adaptive slip window," *Journal of Sichuan University (Natural Science Edition)*, vol. 46, no. 4, pp. 963–967, 2009.
- [16] Z. Huang, C. Bai, and W. Zhang, "Genetic algorithm for arrival sequencing and scheduling optimization," *Advances in Aeronautical Science and Engineering*, vol. 2, no. 2, pp. 236–240, 2011.
- [17] J. Zhang and W. Zhang, "Arriving and leaving aircrafts scheduling model based on dynamic optimize," *Journal of Sichuan University (Natural Science Edition)*, vol. 49, no. 1, pp. 90–96, 2012.
- [18] H. Zhou and X. Jiang, "Research on arrival/departure scheduling of flights on multirunways based on genetic algorithm," *Mathematical Problems in Engineering*, vol. 2014, Article ID 851202, 13 pages, 2014.
- [19] R. Han, *Genetic Algorithms and Applications*, Ordnance Industry Press, Beijing, China, 2010.
- [20] Y. Hu, *Multi-Objective Decision: Practical Models and Optimal Selection Method*, Shanghai Science and Technology Press, Shanghai, China, 2010.



Hindawi

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

