Proceeding of the 11th World Congress on Intelligent Control and Automation
Shenyang, China, June 29 - July 4 2014

An Online Method for the Real-time Aircraft Arrival Sequencing and Scheduling Problem

Xiaopeng Ji
School of Electronic and Information Engineering
Beihang University
Beijing 100191, PR China
jixiaopengyz@163.com

Jing Fang and Ran Yan
Aviation Data Communication Corporation

Beijing 100000, PR China
{fangj & yanr}@adcc.com.cn

***Abstract* - Aircraft arrival sequencing and scheduling (ASS) is a hot topic in air traffic control, which has been proven to be an NP-hard problem. So far, many efforts have been made by modeling this problem in a static case, in which the information of all the landing aircrafts is known in advance. However, the air traffic environment in the airport is dynamic. As new aircrafts are arriving at the airport continually, the corresponding adjustment should be considered for the scheduling. From this point of view, an online method which is based on estimation of distribution algorithm (EDA) is introduced in this paper. At any moment in the sequencing operation, the method only focuses on those aircrafts which have already arrived at the airport but have not been assigned to land. Experiments show that the proposed method is effective and efficient to achieve a better result in solving the real-time ASS.**

Index Terms* - *Air traffic control (ATC), arrival sequencing and scheduling (ASS), permutation-based problem, estimation of distribution algorithm (EDA)

I. Introduction

As the incessant increase in air traffic over the past few decades, the workload of airport terminal areas is increasing heavily. When the terminal area (TMA) is busy, congestion will be generated frequently, which leads to less operation efficiency and more airborne delay. In order to ensure the safety and order of the airports, one efficient method is to reassign the landing sequences and landing times for arrivals in the TMA, which is called Arrival Sequencing and Scheduling (ASS). Due to its importance and complexity, ASS has become a hot topic in air traffic control [1].

In airports, a simple way to solve the ASS problem is to schedule the arriving aircrafts on a rule of first-come-first-served (FCFS) by the predicted landing time (PLT). Although FCFS scheduling gives a fair order, it ignores large amounts of useful information that can make use of the capacity of the airport [2, 3]. For instance, shifting aircraft position according to a landing time interval (LTI), that is the minimum permissible time interval between two successive landings, will significantly reduce airborne delay [4-6]. According to this “position-shifting” characteristic, many research efforts have been made to tackle the ASS problem, which include a variety of models and various methods.

Most of the studies assume that the information of all the landing aircrafts is known in advance and then these aircrafts are scheduled in the same solving process. However, there is inevitably unreliable information in an actual dynamic air traffic environment, such as delay of the aircrafts, cancelation of flights, and emergency landings of some unanticipated aircrafts. Air traffic controllers cannot possibly get accurate information of all these aircrafts in advance. On the other hand, since a huge number of aircrafts arrive at a busy terminal area every day, it is inefficient to consider so many aircrafts at the same time because of the large search space. For this reason, how to design a method that could solve the real-time ASS problem robustly, effectively and efficiently, is worth of further study.

Current research on the real-time ASS problem is mainly from the studies of RHC [7-10]. The RHC is an N-step-ahead optimization strategy which focus on the total airborne delay of the landing aircraft and it divides the ASS problem into a number of sub-problems. This has been extended to a binary-representation-based GA (BRGA) [9] and an ant colony system (ACS) [10] with enhanced search ability for optimal ASS. The results show that RHC and its extensions perform effectively to reduce the total airborne delay.

In this paper, a real-time model which is closer to the reality is established to formulate the ASS problem. In the operation process of the terminal area, there are flights which have already been arranged to land in a specific time. At the same time, new flights are also entering the TMA and waiting to land. Imitating the real situation, a real-time method is designed in our work, which is able to get an optimal sequence according to the current time information. Unlike most of the previous studies, not all the aircrafts are considered in the solving process. At any moment in the sequencing operation, only the aircrafts which have not been assigned to land are needed to be rearranged. And those aircrafts which have already been assigned a landing time will be ignored.

Under the condition of above, a final landing sequence is formed gradually. As only a part of information can be achieved in the solving process, the relationship of the current and assigned plane should be fully considered to get a better solution. For this reason, the algorithm designed should have the ability to schedule the aircrafts dynamically and incrementally. Meanwhile, the final sequence, which has included all the aircrafts in the TMA, should be close to the global optimal solution as far as possible.

Due to the large-scale feature of the problem, the estimation of distribution algorithm (EDA) is adopted as the searching module due to its great generality. For those aircrafts which are needed to be scheduled, the algorithm designed by EDA searches the space of the all the possible landing sequence and find the optimal answer. Some of these aircrafts will be

978-1-4799-5825-2/14/$31.00 ©2014 IEEE

assigned to land and others will be scheduled again with the new entering flights. By this way, a final landing sequence will be determined.

II. Problem Formulation

2.1 Real-time ASS

Due to various kinds of uncertain circumstances, air traffic controllers cannot possibly get the accurate information of the aircrafts which have not arrived at the TMA. The decision made by controllers only depends on the judgment of current situation. At any moment, only part of the flights is assigned to land. For those aircrafts which are about to land immediately, reassigning them may bring great security risks. Therefore, only the aircrafts which still need a longer time to land and those new flights which have just arrived are considered in the scheduling operation.

A simple diagram to show the basic principle of our algorithm is illustrated in Fig. 1. The squares in the front represent the aircrafts which have already been assigned to land. In other words, they will not participate in the next scheduling operation. The triangles denote the aircrafts which have been scheduled once in the last process. And the circles are new flights entering the TMA. The main focus of this paper is aimed at scheduling the latter two in a same time in order to get a better optimal sequence.

As containing part of useful information in the last solving process, the aircrafts which have been scheduled once will be considered in the next scheduling operation. However, the sequence that last process achieved has not taken the aircrafts which have not arrived into consideration. So we have reason to believe that a better optimal sequence can be obtained if the previous and new flights are scheduled together. That is why these aircrafts are considered to be retained to enter the next solving process instead of being ignored.

2.2 Optimization Model

Many researches have focus on the total airborne delay of the arrival sequence. In fact, when the terminal area is in congestion, the air traffic controllers always tend to land the aircrafts as soon as possible since new flights are entering continuously. It may lead to more serious congestion and airborne delay if current waiting aircrafts have not landed in time. From this point of view, the emphasis of our work is to minimize the time needed for all related landing aircrafts.

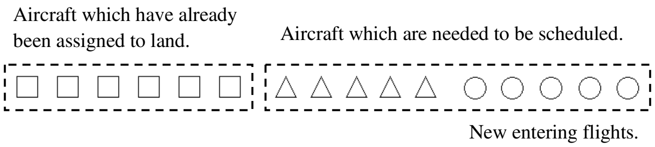
Aircraft which have already been assigned to land.
Aircraft which are needed to be scheduled.
New entering flights.

Fig. 1 The basic principle of our method.

Suppose there are N aircrafts going to land in the airport. Some researchers have focused on the total landing time of the N aircrafts, which is described as

$$T_{length} = \max[ALT(1), \ldots ALT(N)] - \min[ALT(1), \ldots ALT(N)]$$

where $ALT(i)$ denotes the assigned landing time of the i^{th} aircraft. However, there is a problem with this objective function that the first aircraft in the optimal sequence may be scheduled to land quite late. In this case, the landing time of all the aircrafts need to be postponed. Instead, we set the assigned landing time (ALT) of the last aircraft in the sequence as the objective function which can be stated as follows:

$$T = \max[ALT(1), \ldots ALT(N)]. \tag{1}$$

Furthermore, in order to ensure safety, aircrafts are not supposed to land in advance. Otherwise, it will bring security risks and increase the difficulty of the air traffic control. As the aircrafts cannot land before their predicted landing time (PLT), the landing time constraints will be

$$ALT(\prod(i)) \geq PLT(\prod(i)), \quad i = 1, \ldots, N \tag{2}$$

Meanwhile, the LTI which is the minimum safety separation between two successive landing aircrafts must also be satisfied. The LTI is related to the types of the two consecutively landing aircrafts, as illustrated in Table 1. Then the safety separation constraint can be described as

$$ALT(\prod(i+1)) - ALT(\prod(i)) \geq \delta_{i,i+1}, \quad i = 2, \ldots, N \tag{3}$$

In a specific landing sequence $\prod$ after scheduling, $\prod(i)$ denotes the number of the i^{th} aircraft in the original sequence, and $\delta_{i,i+1}$ denotes the minimal LTI between aircraft *i* and *i+1*. By considering the landing time constraints (2) and minimum safety separation (3), the assigned landing time (ALT) for each aircraft can be calculated as

$$ALT(\prod(i)) = \begin{cases} PLT(\prod(i)) & i = 1 \\ \max\{PLT(\prod(i)), ALT(\prod(i-1)) + \delta_{i-1,i}\} & i > 1 \end{cases}. \tag{4}$$

Under this condition, current flights can be scheduled as soon as time permits.

III. An Online Algorithm Based on EDA For The Real-time ASS Problem

In this section, an algorithm based on EDA is designed for the real-time ASS. As the ASS is a permutation-based problem, the Mallows Model and Kendall-τ Distance, which is introduced by Lozano et al. [12], are used as the basic module of our algorithm as they are applicable for the ASS formulation. In addition, a dynamic traffic environment of the airport is also simulated in order to test the efficacy of our online algorithm.

TABLE 1
Minimal LTI between the aircrafts [5]

LTI (i, j) Seconds		Type of the Later Landing Aircraft j			
		1	2	3	4
Type of the Earlier Landing Aircraft i	1	96	200	181	228
	2	72	80	70	110
	3	72	100	70	130
	4	72	80	70	90
Note: (1) Boeing 747; (2) Boeing 727; (3) Boeing 707; (4) Mc Donnel Douglas DC9.					

3.1 EDA

EDA is a kind of novel stochastic optimization algorithms, which has recently become a hot topic in field of evolutionary computation [14]. Different from other algorithms, a probabilistic model is learnt from the current population to express the interrelations between the solutions in EDA. By sampling this probabilistic model, new offsprings are obtained. The algorithm iterates to get the optimal solution until the pre-defined stop point is met. EDA offers a general framework to deal with complex system optimization without lying on detailed domains. In addition, it can settle the problem which has multivariate interaction itself and does not need any information except the objective function. What's more, EDA can also search the optimal value regardless that the search space is continuous or not. In view of these characteristics, the feature of EDA is quite consistent with our need.

3.2 Optimal Sequence Searching

In recent years, many efforts have been made to deal with permutation-based problems by means of EDA, such as Univariate Marginal Distribution Algorithm (UMDA), Estimation of Bayesian Networks Algorithm (EBNAs), and Mutual Information Maximization for Input Clustering (MIMIC) [13-15]. In this paper, the Mallows Model and Kendall- τ Distance [12] are used to search an optimal sequence. Specifically, a central permutation σ_0 is needed to be found in each generation. Then new sequences are generated randomly according to the probability distribution *P*, which can be obtained from the Mallow's model and Kendall- τ distance.

After new sequences are generated, we need to convert these sequences into a feasible solution form depending on the scenario described in the ASS problem, namely the problem formulation, and then evaluates the quality of the sequences. The sequences with a higher fitness will be retained in the next population. This iteration ends when a given stop point is met. In our method, it ends when new flights are entering the airport that we need to restart the scheduling process.

3.3 Dynamic environment

As stated in Section 2, there is great uncertainty in the operation of the airport environment. In the algorithm, we simulate this real-time scenario. At any moment, only the aircrafts which are waiting to land and those just entering the airport are considered to be scheduled. In other word, the aircrafts which have already been assigned to land and those have not arrived will be ignored.

For example, suppose there are *k* aircrafts arriving at the airport. In the last scheduling process, an optimal sequence has already been obtained. We need to assign first *k* aircrafts of the sequence to land and put the new entering *k* flights at the end of the sequence. Then the next optimal sequence searching process begins.

3.4 Algorithm Process

Integrating the modules mentioned above, the process of the online algorithm for ASS problems is achieved as follows:

TABLE 3
Effectiveness of the online algorithm

	Worst	Best	Mean	CPU time (s)
Static	2739	2632	2693.8	172.35
Online	**2836**	**2679**	**2748.1**	**117.96**

1) Population generating: Randomly generate the initial population *pop* according to the aircrafts in the current scheduling process, which contains *M* different sequences. *M* is the size of the population;

2) Initialization: Initialize the probability model *P* based on Mallow's model and Kendall- τ distance. Then set *gen=0*;

3) Sequence searching: Get the central permutation σ_0 of the current population, and then randomly generate *M* new sequences according to the probability distribution *P*;

4) Sequence evaluation: Sort the *2M* individuals according to their fitness. Then keep *M* higher fitness individuals and update *pop*. Then gen=gen+1;

5) Real-time module: If *k* new flights are entering, take out the best sequence in *pop*, and set the first *k* aircrafts in the sequence as assigned. For each individual in *pop*, remove these *k* aircrafts and put the new aircrafts at the end of the sequence. Then return to Step 1). Otherwise, skip this step and return to Step 3);

6) If the simulation ends, output the best solution in *pop*.

IV. Simulation Results

In this part, the online algorithm is applied to solve the ASS problem. In addition, it is also compared with the results obtained by the static case in which all the information of the landing aircrafts is known in advance. Four kinds of aircrafts are considered as shown in Table 1. All the aircrafts will arrive at the airport at their PLTs and an optimal landing sequence is needed to be achieved. An experiment result which solves the ASS problem of 30 aircrafts in both static case and real-time scenario is represented in Table 2. Where, "No." and "Cat." stand for the flight number and category of the aircraft separately. The simulation is implemented in MATLAT 7.1 on a PC running an Intel Core i7 CPU at 1.60GHz with 4.0-GB random access memory.

In our algorithm, the spread parameter θ of Mallow's model is definitely the key to control the search performance. As the value of θ increases, the probability tends to concentrate on the central permutation σ_0. In the experiment we have found that a better result can always be achieved by setting $\theta = 0.1$. In addition, the population in a generation is set as 100, and the maximum number of generations in the static case is also set as 100. The number of aircrafts in the searching process is set as 10, namely only 10 aircrafts is considered in the solving process. Table 2 shows that the online algorithm has the ability to get an optimal answer and it is not far worse than the static case, which shows the great efficacy and efficiency of our method.

In order to verify the stability of the algorithm, the results of 10 independent runs are collected, as shown in Table 3. In this table, "Best" and "Mean" stand for the best and average results obtained from 10 independent runs. We can see that the average performance of the online EDA is quite similar to the static one and the computing time is much shorter, which proves the stability of our online algorithm.

TABLE 2
The best result compared with the static case

Predicted Arrival Flow (PAF)			Static			Online		
No.	Cat.	PLT	No.	Cat.	ALT	No.	Cat.	ALT
1	4	17	1	4	17	1	4	17
2	2	50	2	2	97	2	2	97
3	1	117	4	3	167	4	3	167
4	3	129	7	3	237	7	3	237
5	4	143	9	3	307	10	4	367
6	1	174	5	4	437	11	4	457
7	3	217	10	4	527	5	4	547
8	1	260	13	2	607	12	2	627
9	3	288	3	1	679	13	2	707
10	4	340	8	1	755	14	4	817
11	4	350	18	1	871	15	4	907
12	2	505	6	1	967	16	3	977
13	2	526	16	3	1148	9	3	1047
14	4	541	20	3	1218	20	3	1117
15	4	554	12	2	1318	19	2	1217
16	3	590	24	2	1398	23	4	1327
17	4	648	19	2	1478	25	4	1417
18	1	651	27	2	1558	17	4	1507
19	2	669	11	4	1668	24	2	1587
20	3	691	17	4	1758	22	3	1657
21	1	707	14	4	1848	8	1	1729
22	3	762	25	4	1938	3	1	1825
23	4	769	29	4	2028	18	1	1921
24	2	792	15	4	2118	6	1	2017
25	4	859	23	4	2208	21	1	2113
26	1	993	28	4	2298	26	1	2209
27	2	1064	22	3	2368	28	4	2437
28	4	1077	30	1	2440	29	4	2527
29	4	1207	26	1	2536	27	2	2607
30	1	1401	21	1	2632	30	1	2679
Final answer					2632			**2679**

V. Conclusion

In order to solve the real-time ASS problem, a mathematical model which is much closer to the reality is established and an online algorithm based on EDA is designed. A dynamic environment of the airport is also simulated so that the algorithm solves the ASS problem gradually according to the known information. Simulation results show that our method can always get an optimal solution which is not much worse than the static case. Furthermore, this method is far more efficient and much closer to the reality than static algorithms that it shows great application value in the air traffic control.

Acknowledgment

This work was supported in part by the National Key Technology R&D Program under Grant 2011BAH24B13, and the National Key Scientific Instrument and Equipment Development Project under Grant 2011YQ040083.

References

[1] M. Pelegrin, "Towards global optimization for air traffic management," AGARD-AG-321, 1994.

[2] Carr G C, Erzberger H, Neuman F. Delay exchanges in arrival sequencing and scheduling [J]. Journal of Aircraft, 1999, 36(5): 785-791.

[3] Carr G C, Erzberger H, Neuman F. Fast-time study of airline-influenced arrival sequencing and scheduling [J]. Journal of Guidance, Control, and Dynamics, 2000, 23(3): 526-531.

[4] Bianco L, Bielli M. System aspects and optimization models in ATC planning [M]//Large Scale Computation and Information Processing in Air Traffic Control. Springer Berlin Heidelberg, 1993: 47-99.

[5] Bianco L, Dell'Olmo P, Giordani S. Scheduling models and algorithms for TMA traffic management [M]//Modeling and simulation in air traffic management. Springer Berlin Heidelberg, 1997: 139-167.

[6] Bianco L, Ricciardelli S, Rinaldi G, et al. Scheduling tasks with sequence□dependent processing times [J]. Naval Research Logistics (NRL), 1988, 35(2): 177-184.

[7] Hu X B, Chen W H. Receding horizon control for aircraft arrival sequencing and scheduling [J]. Intelligent Transportation Systems, IEEE Transactions on, 2005, 6(2): 189-197.

[8] Hu X B, Chen W H. Genetic algorithm based on receding horizon control for arrival sequencing and scheduling [J]. Engineering Applications of Artificial Intelligence, 2005, 18(5): 633-642.

[9] Hu X B, Di Paolo E. Binary-representation-based genetic algorithm for aircraft arrival sequencing and scheduling [J]. Intelligent Transportation Systems, IEEE Transactions on, 2008, 9(2): 301-310.

[10] Zhan Z H, Zhang J, Li Y, et al. An efficient ant colony system based on receding horizon control for the aircraft arrival sequencing and scheduling problem [J]. Intelligent Transportation Systems, IEEE Transactions on, 2010, 11(2): 399-412.

[11] Guo Y, Cao X, Zhang J. Constraint handling based multi-objective evolutionary algorithm for aircraft landing scheduling [J]. International Journal of Innovative Computing, Information and Control, 2009, 5(8): 2229-2238.

[12] Ceberio J, Mendiburu A, Lozano J A. Introducing the mallows model on estimation of distribution algorithms [C]//Neural Information Processing. Springer Berlin Heidelberg, 2011: 461-470.

[13] Bengoetxea E, Larrañaga P, Bloch I, et al. Inexact graph matching by means of estimation of distribution algorithms [J]. Pattern Recognition, 2002, 35(12): 2867-2880.

[14] Larrañaga, P., & Lozano, J. A. (Eds.). Estimation of distribution algorithms: A new tool for evolutionary computation [M]. Springer, 2002.

[15] Robles V, de Miguel P, Larranaga P. Solving the traveling salesman problem with EDAs [M]//Estimation of Distribution Algorithms. Springer US, 2002: 211-229.