Optimization of Aircraft Landing Route and Order: An approach of Hierarchical Evolutionary Computation

Akinori Murata[1]†, Masaya Nakata[1], Hiroyuki Sato[1], Tim Kovacs[2], Keiki Takadama[1]

[1] Department of Informatics
The University of Electro-Communications
1-5-1 Chofugaoka, Chofu
Tokyo, Japan
Tel:+81-42-443-5808,

[2] Department of Computer Science
The University of Bristol
Merchant Venturers Building, Woodland Road
Bristol, BS8 1UB, United Kingdom
Tel:+44 117 95-45145

kouho.aki@gmail.com,{m.nakata@cas.hc,sato@hc,keiki@inf}.uec.ac.jp, Tim.Kovacs@bristol.ac.uk

ABSTRACT

This paper focuses on the aircraft landing optimization problem where both the landing routes and the landing order of aircrafts should be optimized to minimize an occupancy time of airport, and proposes its optimization method which is robust to dynamical situations such as weather condition change and other aircrafts' landing routes change. As a difficulty of this optimization problem, appropriate landing routes of aircrafts change depending on such an environment change. To tackle this problem, this paper proposes the hierarchical evolutionary computation to solve the aircraft landing optimization problem. Specifically, our method firstly generates candidates of main landing route of all aircrafts with their own additional sub-routes, which can be applied into the main routes depending on the current environmental situation. Secondly, our method evolves the good combination of landing routes (including their sub-routes) of all aircrafts to minimize an occupancy time of airport. Through the intensive experiment on a benchmark problem, the following implications have been found: (1) our method successfully generates robust landing routes including some sub-routes, which are flexible depending on environmental situations; and (2) Our method can finds an adequate landing order which contributes to reducing the occupancy time.

Categories and Subject Descriptors

I.2.8 [Problem Solving, Control Methods, and Search]: Heuristic methods

General Terms

Control

Keywords

Aircraft Scheduling, Evolutionary Computation, Landing Path Planning

1. INTRODUCTION

In aircraft landing, air traffic controllers should determine both the landing routes of aircrafts and their landing order to minimize an occupancy time of the airport as the air transport service [1]. Such landing route and landing order are important issue because of directly affecting the occupancy time. This problem is called as the *aircraft landing problem* [2]. For this issue, the conventional research tackled this problem by dividing into the following two problems: (1) the landing route optimization problem and (2) the landing order optimization problem. Regarding the first problem, Tajima proposed the real-time path planning method that can evolve the landing routes of the aircraft by evolutionary computation [3]. Regarding the second problem, on the other hand, Xiao proposed the binary-representation-based genetic algorithm method that can evolve the landing orders of the aircraft by evolutionary computation as the aircraft arrival sequencing and scheduling problem [4]. What should be noted here is that there is no method that can cope with the aircraft landing problem including both the landing route and landing order optimization.

However, such an integrated method is indispensable for actual situations such as weather condition change (*e.g.*, turbulence) and other aircrafts' landing routes change**.** This is because the pre-optimized landing routes of aircrafts are not necessary any longer to optimize their landing order due to the situation change. From this fact, the appropriate combination of the landing route of aircrafts and their landing order should be generated together for safe landing. However, it is generally very hard for the air traffic controllers to adaptively and quickly determine the landing routes and their order depending on the situation changes (e.g., avoiding sudden turbulence, obstacles and a congestion of aircrafts).

To tackle this problem, this paper proposes the method that can optimize both the landing routes of aircrafts and their landing order simultaneously to minimize the occupancy time of the airport even in the situation changes. For this purpose, the landing route for each aircraft in our method is designed to have some number sub-routes (*i.e.*, the short detour routes) which can be applied into the main route to be flexibly changed depending on situation changes. By employing such landing route including sub-routes, our method optimizes the landing order depending on the situations to minimize the occupancy time of the airport (*i.e.*, our method decides which sub-route for each aircraft should be employ to minimize the occupancy time). In detail, our method firstly evolves possible landing routes of aircrafts, each of which consists of a main route including sub-routes as options. Secondly, our method finds an adequate landing order based on the customized route for aircrafts to minimize the occupancy time.

BICT 2015, December 03-05, New York City, United States

DOI 10.4108/eai.3-12-2015.2262953

This paper is organized as follows. Section 2 gives a brief description of related works on the aircraft landing problem. Section 3, proposes the hierarchical evolutionary computation method. Section 4 conducts the experiments in order to investigate the effectiveness of the proposed method. Section 5 conducts the additional experiment and discusses its result. Finally, we summarize the contribution of this paper and show future works in Section 6.

2. RELATED WORKS

The aircraft landing problem can be further classified to two specific problems; the landing order optimization problem [4] and the landing route optimization problem [6][7]. The landing order optimization problem aims at finding the shortest landing route avoiding obstacles and wake turbulence; the landing order optimization problem focuses on finding an adequate problem. Many previous works have dealt with either one of those specific problems.

2.1 Aircraft landing order optimization

L. Bianco proposed a scheduling model for aircraft which simulates a real environment to generate landing order [5]. H. Xiao proposed an optimization method which is based on Genetic Algorithm (GA) using Bianco's model. His optimization method successfully produced a landing order to minimize the occupancy time of a destination airport because this method takes into consideration a size of aircraft [4]. The Size is a significant impact on landing time interval (LTI) , also different interval by the leading and follower of the aircraft. For example, LTI of 228s is needed for a category 1 to follow a category 4. This asymmetric relation is a key factor in this optimization. To solve ALP, the total airborne delay has to be minimized. The following formula reoresents the total airbone delay.

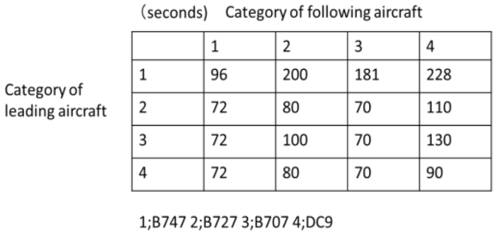
$$A_{Q(n)} = \begin{cases} P_{Q(n),} & n=1 \\ \max\left(P_{Q(n)}, A_{Q(n-1)} + S\left(C_{Q(n-1)}, C_{Q(n)}\right)\right). & n > 1 \end{cases}$$

$$D_i = A_i - P_i, \qquad i = 1, \ldots., N_{AC}$$

$$\min_{Q(1),\ldots.Q(N_{AC})} J_1 = \min_{Q(1),\ldots.Q(N_{AC})} \sum_{i=1}^{N_{AC}} D_i$$

N_{AC} aircrafts are planning to land an destination airport during operation day, and $C_i, P_i and\ A_i$ are the category of aircraft size, the predicted landing time and the ALT of the ith aircraft in the first predicted arrival order. $Q(n)$ is the nth aircraft in the optimized landing order. $S(i,j)$ is the LTI for an aircraft of category j to follow an aircraft of carefory i to land. D_i is a difference of ith aircraft optimization time and the original prediction arrical time. The initial landing time is setting on by the principle of FCFS. Note that this method does not consider the landing route of aircraft. This means, it can produce an adequate landing order so that reducing the occupancy time, and so it is still unclear how a landing route should be designed to follow the decided the landing order.

Table 1 Minimum LTI



（seconds） Category of following aircraft

Category of leading aircraft		1	2	3	4
	1	96	200	181	228
	2	72	80	70	110
	3	72	100	70	130
	4	72	80	70	90

1;B747 2;B727 3;B707 4;DC9

2.2 Navigation route optimization method

2.2.1 GA Based Model

For the landing path planning, T. Tajima proposed an optimization method based on GA [3]. As shown in Figure 1, He introduced a grid-like map where the aircraft was flying. Noted that the black squares represent obstacles which indicate a prohibit area. As shown in Figure 2, a gene represented a combination of some waypoints the aircraft would be flying (denoted by the yellow symbols in Figure 1). The genetic locus indicated the coordinate of a waypoint on the grid map. Additionally, the gene has a variable length. Thus a gene having a long length indicates a complex landing path including many waypoints. The landing route could be represented by connecting the waypoints as shown by the red line in Figure 1.

Tajima showed his optimization method generates the shortest route of aircraft avoiding the prohibit area, and this would contribute to reduce the occupancy time of the airport. However, this method optimizes the landing route for single aircraft; thus, it is limited in optimize the landing routes for multiple aircrafts. In fact, the landing route for each aircraft is designed considering the other aircraft's landing routes.

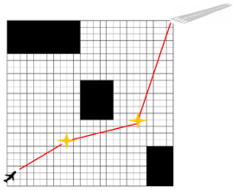
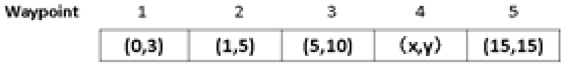


Figure 1 Representation of landing route [4].



Waypoint	1	2	3	4	5
	(0,3)	(1,5)	(5,10)	(x,y)	(15,15)

Figure 2 Gene structure [4].

Two individuals are selected based on fitness value and applied to crossover operation.

Using crossover, off springs can take over good features for solution. Cross over operation is different from the number of waypoints. There are three types of method of crossover operation and we show the way following.

Table 2 Crossover pattern

Parent-1 Total number of waypoints	Parent-1 Total number of waypoints	Crossover operation
1	1	**Figure 3**,**Figure 4**
1	n(>1)	
n(>1)	1	
m(>1)	n(>1)	**Figure 5**

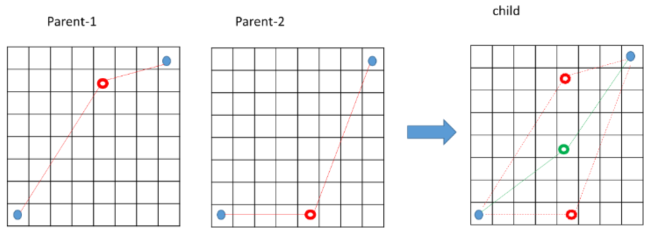


Figure 3 Average position of waypoint

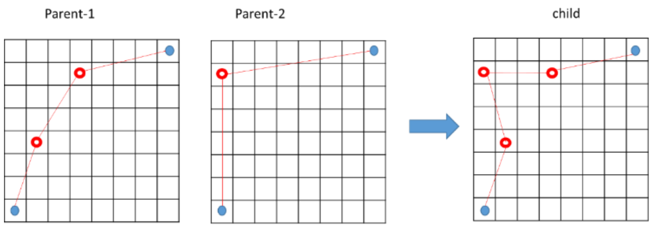


Figure 4 Connection between waypoints

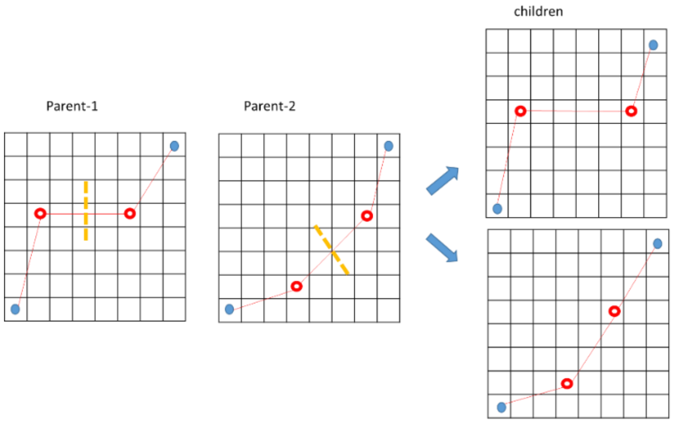


Figure 5 Recombination of waypoint

Crossover operation

① Average position of waypoints: Getting one waypoint of each path plan and calculating average position between waypoints. Then Average position become new waypoint.

② Connection between waypoints: combining two waypoints of each route into one route.

③ Recombination of waypoint: Dividing two route into multi part of routes, then recombination of these routes.

Mutation operator

① Addition to waypoint: Adding to waypoint on grid map randomly with a constant probability

② Change of waypoints : Changing position of one waypoint with a constant probability

③ Deletion of waypoints: Deleting one waypoint randomly with a constant probability

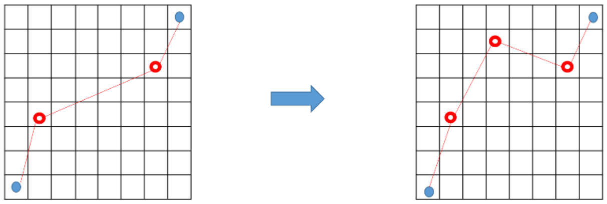


Figure 6 Addition to waypoint

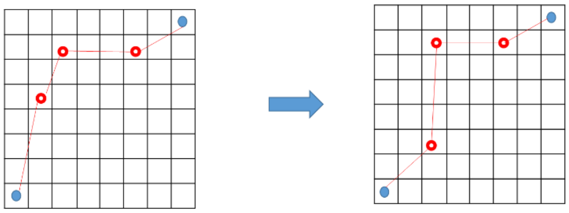


Figure 7 Deletion of waypoints

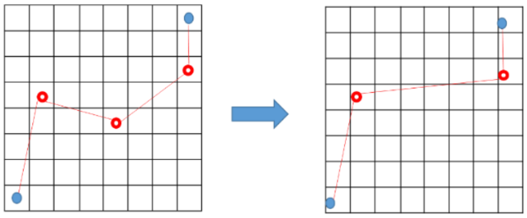


Figure 8 Change of waypoints

2.2.2 Ant Colony Optimization Based Model

Chari proposed the path planning method by Ant colony optimization (ACO) [8]. This method used a grid-based model to indicate action space of robots. This map is described a 2-D map and added to obstacles. In this case, it is more difficult to search the shortest path than the map where there is no obstacles. In order to find feasible solution easily, nodes which are called sensor nodes are set in random location on the map (see Figure 9). And path is decided by combination of nodes and edges. In this method, it is effective to search shortest path. But when the environment is changeable, this method cannot deal with the changeable situation.

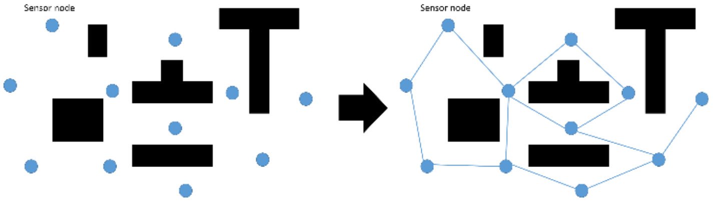


Figure 9 Environment

3. PROPOSED METHOD

Compared with the related works as described in Section 2, our approach aims at optimizing both landing routes and the landing order together to minimize the occupancy time of airport. Specifically, different from the Xiao's method, our method optimizes not only the landing order but also the landing routes that follow the optimized landing order; unlike the Tajima's method, our method optimizes the landing routes for multiple aircrafts.

As shown in Figure 12, our method produces the robust landing routes that can be flexibly customized depending on the changeable situations (i.e., wake turbulence and a congestion of aircrafts). Our method is based on NSGA-II as a multi-objective optimization technique [9] and composed of the following two units;

1. Landing route generating unit: which makes candidates of landing route from the current position to the destination airport. Each landing route is represented by a gene as in Tajima's method (see Figure 2); the gene represents a set of waypoints. Then, some sub-routes are added to each generated main route as options that enable the main route to be flexibly customized depending on the situations.
2. Arrival sequence generating unit: which selects the candidates of main route or customizes the main landing route by replacing with the sub-routes depending on the situations. Then, it optimizes a combination of landing routes of multiple aircrafts in order to minimize the occupancy time of destination's runway

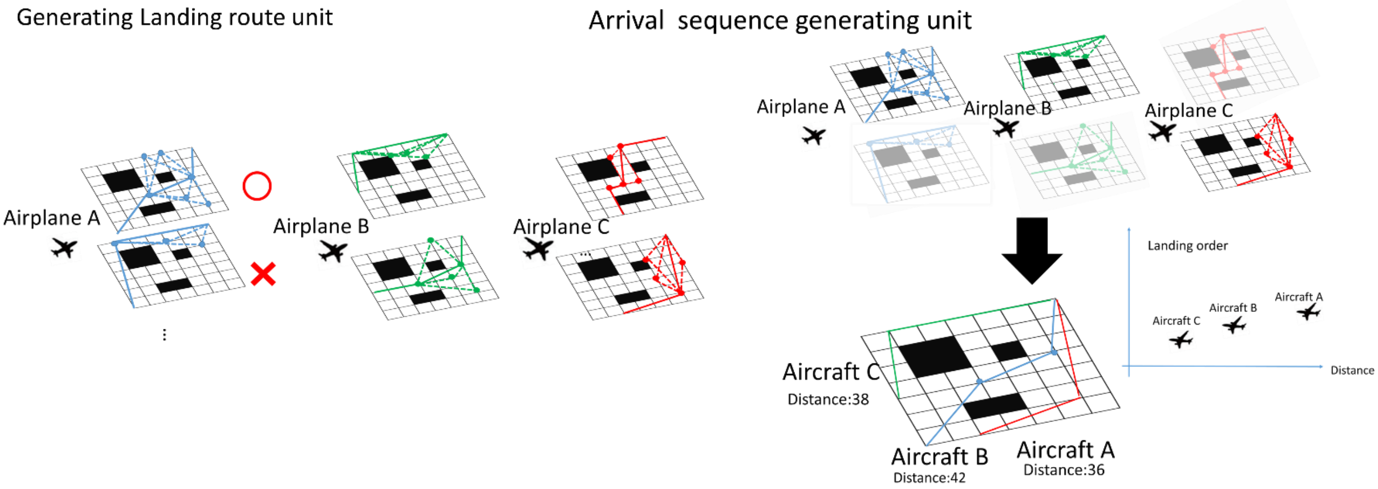


Figure 10 Architecture.

3.1 Mechanism

3.1.1 Generating landing route unit

As shown in Figure 11, this unit first generates candidates of main route and then, adds sub-routes to each of the generated main landing route. Then the candidates are evaluated in terms of two aspects; 1) the total distance of main route from the current position to the destination airport and 2) the robustness of the route can be quantified as the number of possible sub routes included in the main route. Thus, this unit eventually produces the landing paths which indicates small distance to the destination airport and can be customized for adapting as many situations as possible. Note that the routes that pass on the prohibit area is identified as infeasible solution.

Each candidate of main landing route (i.e., the gene) is generated by NSGA-II. Then, if the generated route is feasible solution, sub-routes are added to the generated main landing route (see Figure 12) by the following steps;

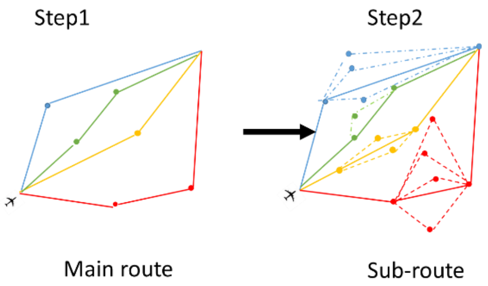


Figure 11 step of generating main route and sub-routes.

Sub-route algorithm

Step1: For one of main route, calculating each the distance between the waypoints.

Step2: Creating a square of distance r that center is a midpoint between the waypoints and a point on the vertical bisector randomly is determined. This point becomes new waypoint.

Step3: New route that the new waypoint created in step2 is added is calculated the evaluation value.

Step4: This route is saved as one of sub-route if the limitations of this route are met.

Step5: Repeating from step2 to step4 until the termination condition is satisfied

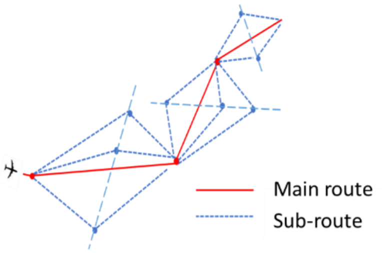


Figure 12 the main landing route with the sub-routes.

After adding the sub-routes to each landing route, they are evaluated with the fitness function and applied to genetic operators (i.e., the crossover and mutation) to find better solutions with a high fitness which have small distances to the destination and many sub-routes. Overall procedure can be described as follows;

Calculating fitness of each route in terms of the distance to the destination airport and the number of sub-routes. Selecting routes as parents and copying them as the offspring. Then, the crossover and the mutation are applied to generated offspring. Producing a next generation population by the non-dominant sorting and degree of congestion tournament selection.

3.1.2 Arrival sequence generating unit

The generating arrival sequence unit picks out the top N candidates by the non-dominant sorting and degree of congestion tournament selection and searches the best combination of landing routes for some aircrafts by GA. Each solution is represented in the integer type of gene and each locus indicates the selected route with each aircraft. This unit takes the following steps;

Step1: Selecting one route for each aircraft (Figure 14(a)-1).

Step2: Checking on interval of aircraft whether meet constraint condition or not.

Step3: Calculating evaluation value that meets limitation (Figure 14-3)

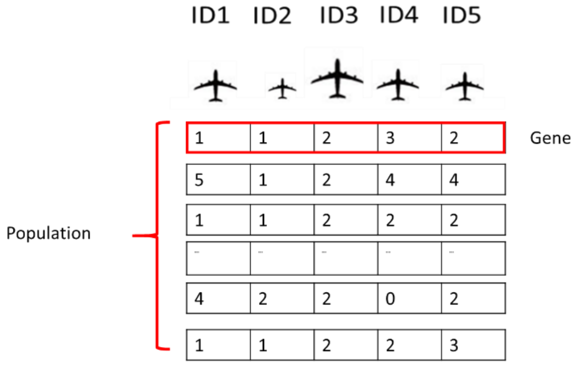
Step4: Selecting parents from among solutions that meet limitation by using tournament selection

Step5: Crossing between parents and mutation

Step6: Repeating from step2 to step5 until the termination condition is satisfied

The following is described the limitation condition that a solution meet, how to calculate evaluation value of the solution (Step2 and Step3) and how to mutate and crossover (Step5) in detail.

After this step, main route of each aircraft is determined. If turbulence occurs on main route for a little time, the aircraft which determined main route takes an opportunity to select sub-route Figure 14(b)-1). In case of this, arrival sequence generating unit replace sub-routes for main routes and reserch the best combination of landing routes by GA(Figure 14(b)-2) (Figure 14(b)-3).



ID1	ID2	ID3	ID4	ID5
1	1	2	3	2
5	1	2	4	4
1	1	2	2	2
-	-	-	-	-
4	2	2	0	2
1	1	2	2	3

Figure 13 locus and population of arrival sequence generating unit

Gene

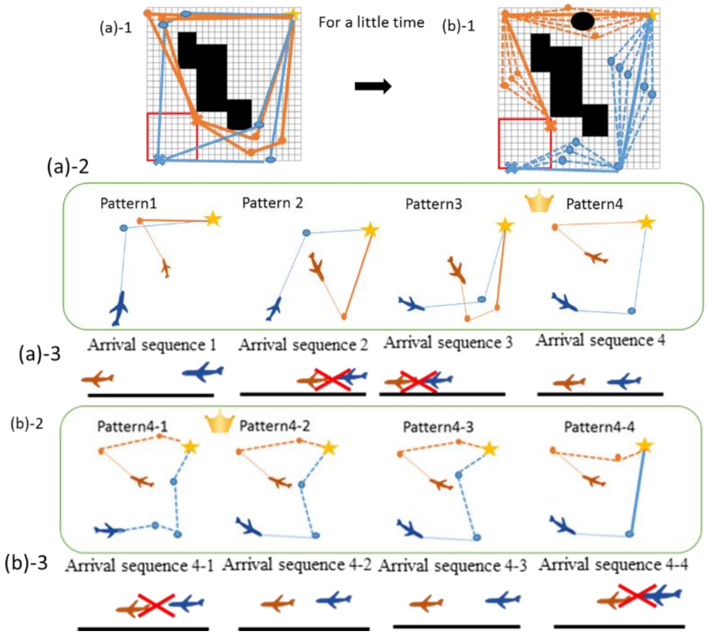


Figure 14 Overview of arrival sequence generating unit

Constraints and evaluation value

After generating of the combination of routes for some aircrafts, the arrival sequence generating unit determines the landing order of aircrafts. Here, when the landing order indicates that the interval between each aircraft's landing is small, its landing order can be identified as infeasible since for safe landing, an enough interval is required. Thus, to identify the generated landing order either one of feasible solution or infeasible one, we add the following constraint condition for landing order;

$$d_n - d_p > r$$

Symbol d_n and d_p represent the distances to the destination of two aircrafts n and p; thus $d_n - d_p$ represents the interval between the two aircrafts. Then, if the interval is larger than a threshold r for all possible route combination of two aircrafts, its landing order is identified as a feasible solution.

Additionally, to evaluate the generated landing order we introduce the following equation as calculation of fitness;

$$\text{Fitness} = \sum_{i=1}^{n} d_i$$

The fitness of landing order is simply calculated as the summation of distance of each aircraft. It is the distance of main route. Thus, the small value of fitness means that its landing order can reduce the occupancy time of the destination airport.

Crossover and mutation

This employ the uniform crossover. This crossover operation evaluates each it in the parent strings for exchange with a probability of 0.5. The mutation changes each gene locus to a random value with a certain probability. Mutation rate is $\frac{1}{l}$ with a bit. *l* is gene length. For example, this unit optimizes landing order for 10aircrafts and mutation rate is 0.1. In arrival sequence generating unit, top N is 20.

4. EXPERIMENT

4.1 Experiment Setting

We conduct experiments on the grid map as introduced in [4]. As shown in Figure 15, the grid map we use here is a 25 × 25 grid map and includes the prohibit area denoted by black square. We define six aircrafts (denoted ID1-ID6 respectively) are staying at the starting area denoted by the blue-rectangle in the figure.

To investigate whether our method successfully produces the robust landing routes and optimizes its order that reduces the occupancy time of the destination airport, we conduct the following two experimental cases; case1 is an ideal situation without any wake turbulences but for the case2 the turbulence is added to case1. There are some cases which landing order are changed in aircraft landing. For example, appearance of new aircrafts, changeable situation of the weather and an emergency situation for aircraft. This paper tackles on two case, appearance of aircrafts and changeable situation of the weather because two case often occurs than the other one.

We use the following parameter setting for NSGA-II and simple GA; the population size is set to 200, the maximal generation is 1500, the crossover rate is 1.0 and mutation rate is 0.5. Additionally r is set to 3.

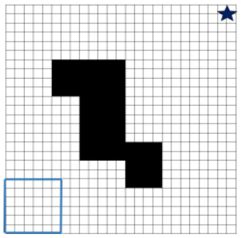


Figure 15 Test map

4.2 Results

Figures show the routes of each aircraft (Figure20-(a)), (Figure20-(b)). Figure16 indicates that each aircraft could obtain branches appropriately. Shows the landing order. The vertical axis represent order and the horizontal axis is the distance for destination. In Figure17 the blue graph is the landing order fitness of each case and the orange graph shows the fitness which no aircrafts would encounter a turbulence.

It is shows that routes and the landing order was generated appropriately in case1 Figure18 and Figure19 .In case2, the leading aircraft chooses sub-route because the landing aircraft that determined one route would have encountered turbulence. ID1 took alternative path and interval of distances changed. ID1 selected alternative path and interval of distance between ID1 and ID4 was so closed that it broke limitation. Consequently, ID4 changed path to maintain the interval. And ID2 also took the detour route to meet the limitation. Figure17 shows that the landing order fitness is almost unchanged between the orange and blue graph in case2.

However it occurs a turbulence on the map, it is possible to maintain a better fitness value little by proposed method.

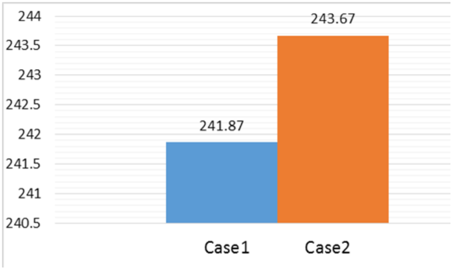


Figure 17 landing order fitness case1, case2

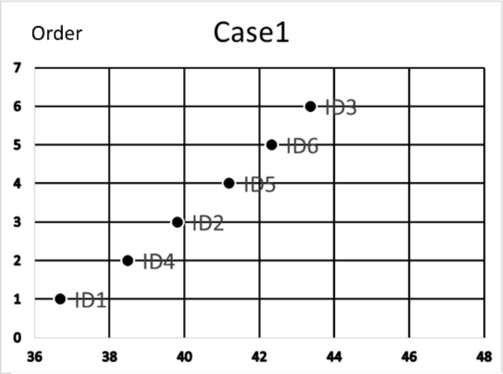


Figure 18 Case1 landing order

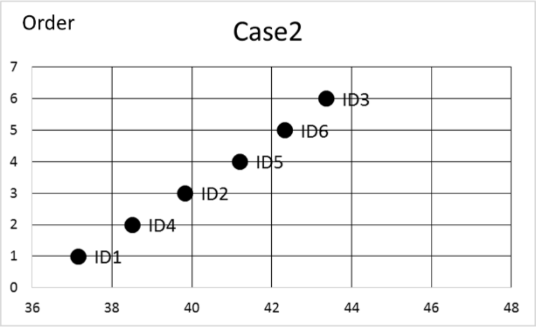
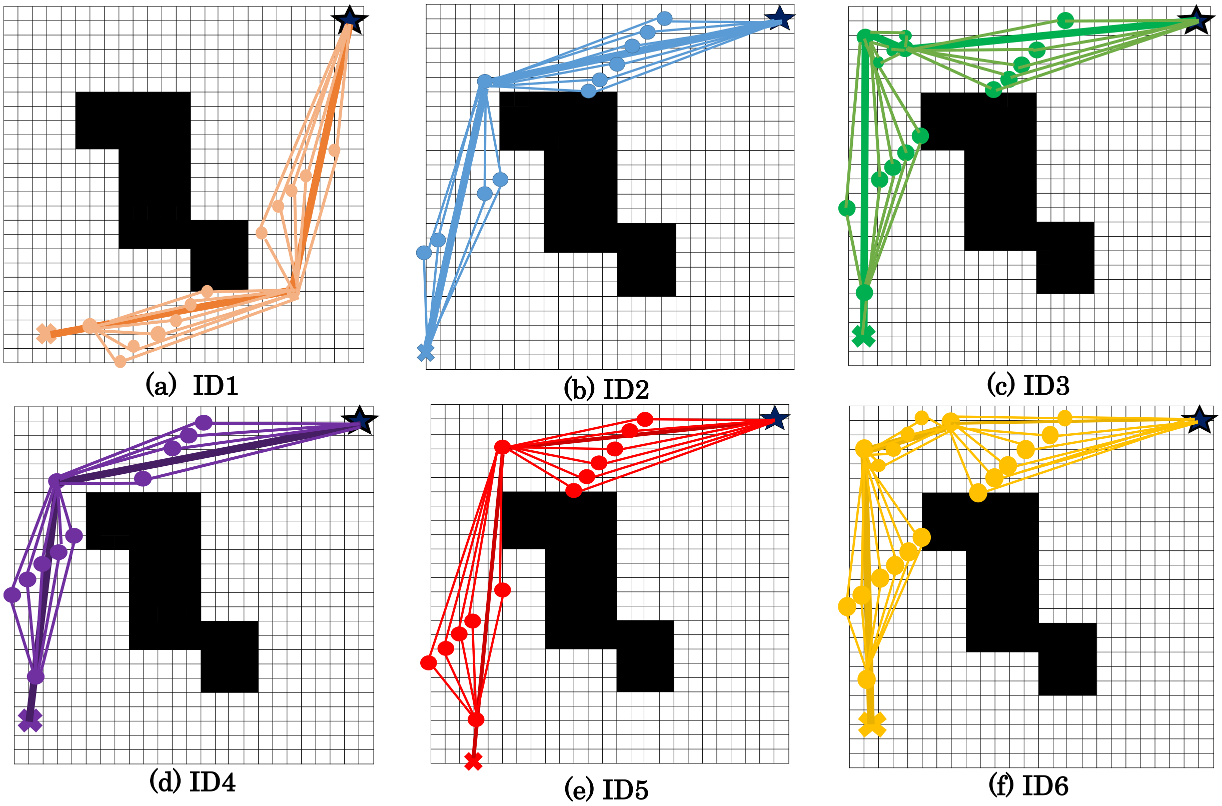
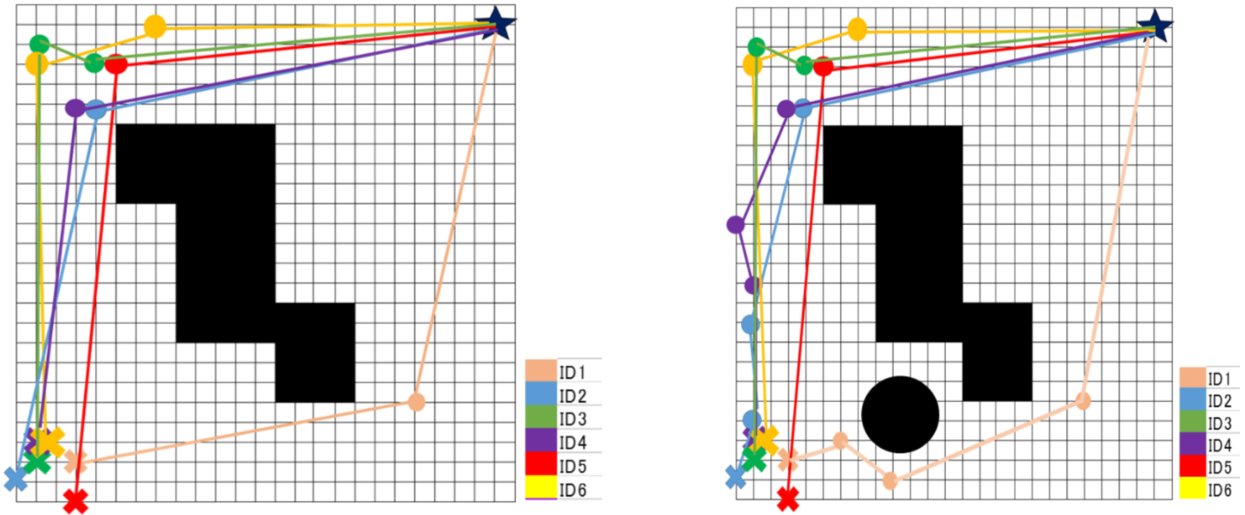


Figure 19 Case2 landing order


(a) ID1 (b) ID2 (c) ID3

(d) ID4 (e) ID5 (f) ID6

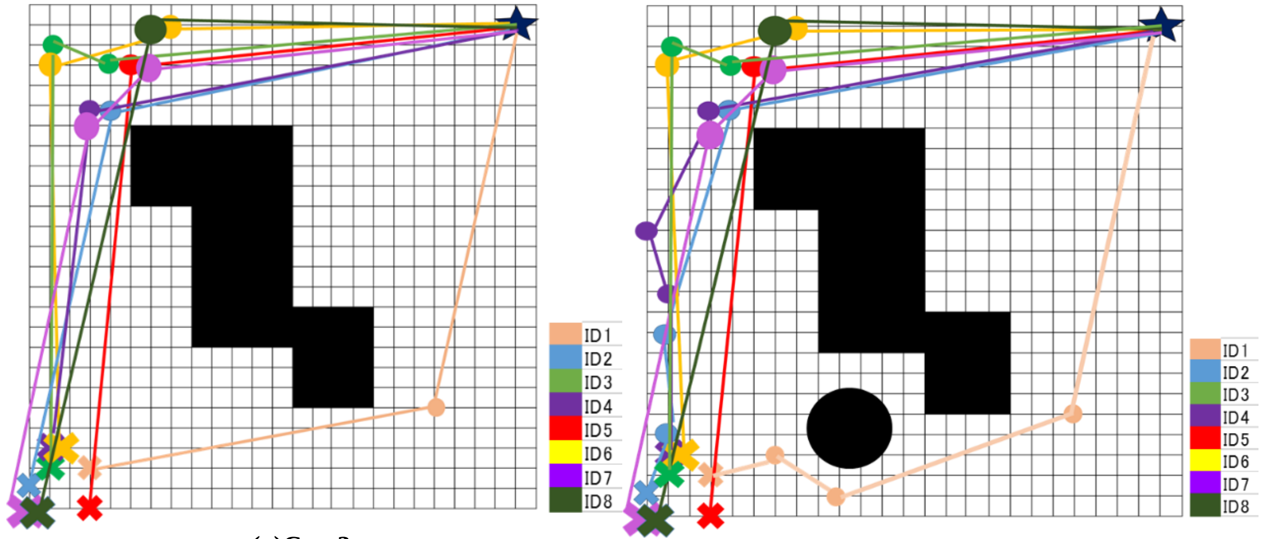
Figure 16 Route and branches



(a)Case1

(b)Case2

Figure 20 Route of aircrafts



(a)Case3

(b)Case4

Figure 21 Route of aircrafts

5.DISCUSSION

We conduct additional experiment in the same map. We simulate the following two experimental cases; case3 is an ideal situation without any wake turbulences and add next aircrafts after a certain time. In the case4, basically it is the same as Case1, and it occurs the turbulence and add next aircraft after a certain period of time. We define two next aircrafts (denoted ID7, ID8 respectively) which are placed at bottom left corner in the figure.

Figure indicates the routes in case3 and case4. Figure 22 and Figure 23 show the landing order. Figure24 shows landing fitness in case3 and 4.

In case3, basically following aircrafts would not use sub-routes because turbulence did not occurred. It may have caused to change the route of the other aircrafts by adding aircrafts. In this experiment. But it did not happened (Figure 21-a) (Figure 22). If more initial aircrafts are added, some aircrafts may change the route. In case4, both appearance of following aircrafts and generating turbulence on the map occurred. For this reason, it is difficult to optimize landing order (Figure 21-b) (Figure 23). ID1 would have encountered turbulence the same way as case2. Nevertheless, the result of landing order was the same as cese3.

In Figure24 shows that the landing order fitness is almost unchanged between the orange and blue graph in case4. It shows that using sub-routes does not give bad influence to follower aircraft.

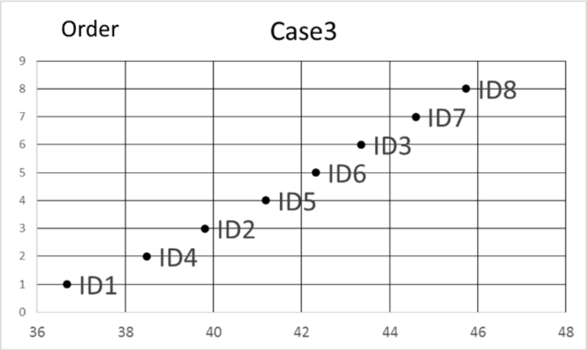


Figure 22 Case3 landing order

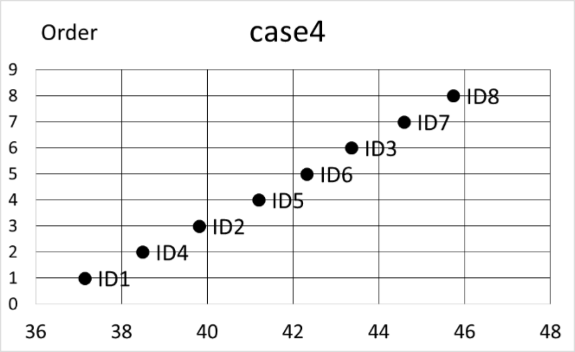


Figure 23 Case4 landing order

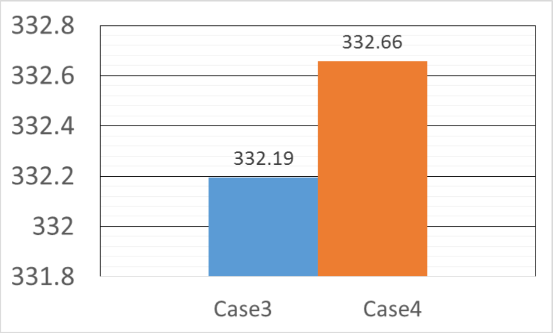


Figure 24 landing order fitness case3, case4

6. CONCLUSIONS

This paper focused on the aircraft landing optimization problem which addresses both the landing routes and order of aircrafts to minimize an occupancy time of airport (*i.e.*, the total distance of aircrafts to the airport), and proposed its optimization method which is robust to dynamical situations such as weather condition change and other aircrafts' landing routes change. Concretely, our proposed method based on the hierarchical evolutionary computation generates both the landing routes and order of the aircrafts which can be flexibly customized depending on the situations. In particular, the proposed method firstly generates candidates of the landing routes of the aircrafts with their own additional sub-routes by using NSGA-II (note that the generated routes have a short distance from the current position to the destination airport with many sub-routs for robust landing routes). Secondly, our method evolves the good combination of the landing routes (including their sub-routes) of all aircrafts to minimize the occupancy time of airport by changing the landing routes of the aircrafts for an appropriate their order. To investigate effectiveness of the proposed method, we tested our method on the grid map which simulates the flying area of the aircrafts. The experimental results revealed that our proposed method can optimize the landing routes and order of the aircrafts that reduces the occupancy time even in the case where the other aircraft should land for an emergency. These results suggested that (1) our method successfully generates the robust landing routes including some sub-routes and (2) our method finds an adequate landing order which contributes to reducing the occupancy time.

What should be noticed here is that the obtained results have only been obtained from one simple gird map which simulates the flying area of the aircrafts. Therefore, further careful qualifications and justifications, such as other maps or different number of aircrafts, are needed to generalize our results. Such important directions must be pursued in the near future in addition to the following future research: (1) an improvement of generating efficient sub-routes and (2) conducting simulations which consider real environment (*e.g.*,wind caused the delay).

REFERENCES

[1] ICAO, *Annual Report of the Council*, 2013..

[2] J. B. A., M. Mohammad and P. N. Chris, "Airport Runway Scheduling," *4OR(Volume 9: Issue 2) pp 115-138* 2011.

[3] T.tajima;K.Nakano;M.Ichikawa;H.maeda, "A Real-Time Path Planning Using Genetic Algorithms," The jouunal of the Japanese Society for Artificial Inttelligence (Volume:10 ,Issue 14)pp94-104 1995.

[4] H. Xiao-Bing and P. D. Ezequiel, Binary-Representation-Based Genetic Algorithm for Aircraft Arrival Sequencing and Scheduling,"Intelligent Transportation Systems, IEEE Transactions on (Volume:9 ,Issue: 2) pp 301-310 2008.

[5] L. Bianco, P. Dell'Olmo and S. Giordani, "Scheduling models and algorithms for TMA traffic management," Modelling and Simulation in Air Traffic Management pp139-167 1997.

[6] .Y. Logan, A. Adrian and T. Kagan, "Evolutionary Agent-Based Simulation of the Introduction of New Technologies in Air Traffic Management," Proceedings of the 2014 conference on Genetic and evolutionary computation 2014.

[7] T. Kyle and M. Zhi-Hong, "Conflict Resolution and Traffic Complexity ofMultiple Intersecting Flows of Aircraft," Intelligent Transportation Systems, IEEE Transactions on (Volume:9 ,Issue: 4) pp633-643 2008.

[8] ChaariImen,KoubaaAnis,BennaceurHachemi,aharTrigui, Al-ShalfanKhaled. "smartPATH: A Hybrid ACO-GA Algorithm for Robot Path Planning."Evolutionary Computation (CEC), 2012 IEEE Congress on, 2012.

[9] K. Deb, S. Agrawal, A. Pratap and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," Parallel Problem Solving from Nature PPSN VI,Volume 1917 of the series Lecture Notes in Computer Science pp 849-858 2000.