Simulation-Free Runway Balancing Optimization Under Uncertainty Using Neural Network

Ryota Mori, Daniel Delahaye

Simulation-Free Runway Balancing Optimization Under Uncertainty Using Neural Network

Ryota Mori
Air Traffic Management Department
ENRI
Tokyo, Japan
r-mori@mpat.go.jp

Daniel Delahaye
OPTIM Research Lab
ENAC
Toulouse, France
delahaye@recherche.enac.fr

Abstract— **This paper proposes a new optimization scheme using neural network for runway balancing to minimize departure and arrival aircraft delay. While other researchers have proposed solutions to the runway balancing problem using a simulation-based technique to calculate aircraft delay, the proposed method replaces the simulation by a neural network model-based estimation using the actual operational data, thus providing the following two advantages. First, accurate estimation of aircraft delay can improve the solution of the runway balancing problem. Second, the simulation process is not required in the optimization. Although it is difficult to develop an accurate simulation model especially under uncertain environment, the neural network model can estimate the average delay without explicitly modeling uncertainty. In this paper, as a first step, the effectiveness of the proposed method is validated through simulations. First, simulations considering uncertainty are used to generate the data, which are then used to train the neural network. The neural network predicts the delay under the current traffic and only this predicted delay is used for the runway balancing optimization with simulated annealing. The simulation result shows that the result by neural network outperforms the one by the simulation-based method under uncertainty. This means that the neural network can accurately estimate the delay under uncertainty environment, and is applicable in the optimization process.**

Keywords; simulated annealing, arrival manager, convolutional neural network

I. INTRODUCTION

Airport runway is a main bottleneck of air traffic control system, and thus more efficient runway usage is needed to maximize capacity and optimize traffic handling[1]. Apart from new runway construction, there are two main approaches to increase runway capacity: take-off/landing separation reduction and runway balancing. The former is a straightforward idea to increase the runway capacity, and various methods are proposed such as optimal aircraft sequencing considering wake turbulence category, time-based separation applied under strong head wind, and development of new separation standard (RECAT) [2]. The latter (runway balancing) aims at optimal runway allocation of take-off/landing aircraft to multiple runways. Although this method works only when there are multiple runways at the airport, it is one of the promising solutions for major airports in the world with multiple runways. This research focuses on runway balancing.

There have been many prior studies to optimize the runway allocation of departures and/or arrivals. To optimize the runway allocation, there are several types of the objective functions to be minimized depending on the problems, such as aircraft delay[3][8][13], makespan[7], and combination of makespan/aircraft delay and other factors such as noise, fuel consumption, and congestion level [4][5][6][9][10][11][12]. The runway allocation is a combinatorial optimization, and various optimization methods are applied such as mixed integer linear programming[10], genetic algorithm[3][4], simulated annealing[8][9][11], ant colony optimization[6], swarm intelligence algorithm[7], bat algorithm[13], greedy method[5], and dynamic programming[12]. Also, many researchers optimize the aircraft sequencing and runway allocation simultaneously. Operational constraints, such as conflict on air routes and taxiing, are also considered in some studies[11]. Most researchers consider deterministic environment only, and just a few researches are found to take uncertainties into account. However, uncertainty exists in real world, and the uncertainty of take-off time is usually greater than that of landing time because the boarding process includes large uncertainty. Therefore, uncertainty becomes critical when the runway handles departure aircraft.

To tackle uncertainty, several approaches are considered. One is robust optimization. The robust optimization assumes a certain time window of uncertainty, and the best strategy is found under the worst scenario. Runway sequencing has been optimized with robust optimization as shown in [14]. With this method, the optimization is converted to a deterministic optimization problem, so the deterministic optimization method can be applied. The optimal parameters found for the worst case scenario are not necessarily optimal for various uncertain scenarios, however. Another method is to run numerous simulations by changing uncertainty parameters, and sets of simulation results are evaluated by the expected value (sample average approximation) or its alternatives. This method is referred to as a simulation method here. There are some researches using the simulation method for runway balancing and sequencing[7][15]. This method can consider many uncertain scenarios, but requires large computational time.

However, both methods discussed above have a common problem: they require an accurate runway simulation model, and especially the following important parameters should be found: take-off/landing separation, interaction of departure and landing, and their uncertainty. The optimal runway balancing, for example, is usually obtained by minimizing the objective function that is estimated based on such simulation models. Intricate airport operations make it difficult to develop an accurate simulation model. If the simulation model is inaccurate, the obtained optimal solutions becomes less reliable. To tackle this issue, the authors propose a new optimization approach without using a simulation or parameter estimation of airport operations. The objective function must be calculated for optimization, but the authors propose that this be done by a neural network (NN) trained with the actual operational data directly. Since the NN training process requires both inputs and outputs from actual data, the NN can learn the actual operation environment directly by the actual data. In addition, the actual operation includes various types of uncertainties, which are difficult to be handled in the optimization. However, the NN is trained to minimize the error between the model output and data output, which means that the NN estimates output of the expected value considering uncertainty.

This approach should work in theory, but investigation is necessary to evaluate its performance in practice. Therefore, in this paper, the above simulation-free approach is formulated and the result is compared to the simulation-based approach. One of the advantages of the proposed method is the use of the actual operational data directly with no simulation environment required. However, to validate the proposed method, as a first step, the simulated data including uncertainty instead of real-world data is used for NN training, which makes it possible to evaluate the obtained NN performance.

This paper starts with the problem formulation in Sec. II, and the NN model is developed in Sec. III. Sec. IV shows the simulation environment and results, and Sec. V concludes this paper.

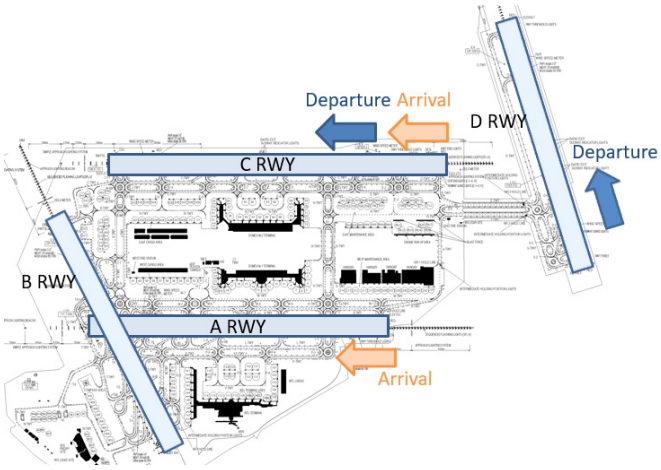

Figure 1. Airport layout and runway configuration under north wind at Tokyo International Airport.

II. Problem Formulation

A. Airport operation and simulation method

This time, Tokyo International Airport (Haneda Airport) is set as a target airport, and the runway balancing problem is considered. Fig. 1 shows the airport layout and runway operations under north wind. Runway A is used for arrival only, and is independent of traffic on other runways. On the other hand, Runway C is used for both departure and arrival i.e. mixed-mode operation. Runway D is used for departure only, but the arrival to Runway C also affects the departure from Runway D. Under this condition, the runway balancing for arrivals between Runways A and C is considered.

In this paper, as a first step, the above runway operation is simply described in the following mathematical forms.

1) Inputs

- D: Set of departure aircraft $D=\{1,...,n_D\}$
- R_D: Set of departure runways $R_D=\{c_D,d\}$, where c_D is Runway C used by departures and d is Runway D.
- A: Set of arrival aircraft $A=\{1,...,n_A\}$
- R_A: Set of arrival runway $R_A=\{a,c_A\}$, where a is Runway A and c_A is Runway C used by arrivals.
- $PTOT_i$: Earliest possible take-off time for aircraft i. $\forall i \in D$
- $PLDT_i$: Earliest possible landing time for aircraft i. $\forall i \in A$
- r_i^{nom} : Nominal runway for departure and arrival. $r_i^{nom} \in R_D \forall i \in D$ and $r_i^{nom} \in R_A \forall i \in A$
- r_i : Departure/Arrival runway. $r_i = r_i^{nom} \forall i \in D$ and $r_i \in R_A \forall i \in A$

2) Decision variables

- $\delta_i \forall i \in A$: binary variable for arrival runway decision.

$$\delta_i = \begin{cases} 1 & r_i \neq r_i^{nom} \\ 0 & otherwise \end{cases}.$$

3) Variables given by the airport operation

- t_i: Actual take-off or landing time. $\forall i \in A \cup D$

4) Contraints

- Runway separation

$t_j \geq t_i + S_{ij} \quad \forall i \neq j \in A \cup D$ where S_{ij}: Minimum separation between aircraft i and j where aircraft i precedes aircraft j.

$$S_{ij} = \begin{cases} -\infty & \text{if } t_i > t_j \\ & \text{if } r_i = r_j \text{ for } i, j \in A \cup D \\ N(120s, 15s) & \text{or } (r_i = c_A \,\&\, (r_j = c_D \text{ or } r_j = d) \text{ for } i \in A, j \in D \\ & \text{or } ((r_i = c_D \text{ or } r_i = d) \,\&\, r_j = c_A) \text{ for } i \in D, j \in A \\ 0 & \text{otherwise} \end{cases}$$

- Earliest landing time

  $t_i \geq PLDT_i + w_{r_i, r_i^{nom}} \quad \forall i \in A$ where $w_{r1,r2} \; \forall r_1, r_2 \in R_A$:

  Additional flight time required when the nominal runway is r_1 and the landing runway is r_2.

  $$w_{r_1, r_2} = \begin{cases} 0 & \text{if } r_1 = r_2 \\ 120 & \text{otherwise} \end{cases}$$

- Earliest take-off time

  $t_i \geq PTOT_i \quad \forall i \in D$

- Arrival is always given priority over departure.

To simplify the problem, the take-off and landing separations are set the same regardless of the wake turbulence category, and are equal to 120 s. The departure-arrival (C runway-C runway) or arrival-departure (C runway-C runway and C runway-D runway) separations are also set to 120 s. $N(a,b)$ indicates the normal distribution with average a and standard deviation b, so the separation is not constant but randomly distributed. Arrival is always given priority over departure, so the departure from both C runway and D runway cannot take off if arrival at C runway is approaching and enough separation cannot be established.

The objective function to be minimized is the total delay of all aircraft. However, the landing runway re-assignment in general requires additional ATC workload, so the number of runway re-assignment is also included in the objective function, defined as the following form where α is the weight factor. Here, the runway re-assignment indicates the case where the nominal and the actual runways are different.

$$J = \sum_{i \in D} (t_i - ETOT_i) + \sum_{i \in A} (t_i - ELDT_i + \alpha \delta_i) \qquad (1)$$

If PLDT and PTOT of all aircraft are known in advance, it is easy to optimize the runway balancing, which is not the case in the real world. Both have a considerable uncertainty, but departure aircraft tend to have larger uncertainty than arrivals, because the departure aircraft can start pushback only when all passengers are onboard, which causes the large uncertainty. The estimated time of PLDT and PTOT is denoted by ELDT (Estimated Landing Time) and ETOT (Estimated Take-off Time), respectively. This time, the following uncertainty value is assumed according to past researches[16][17].

$$ETOT - PTOT = \begin{cases} N(0, 300s) & \text{if } \Delta T > 15 \min \\ N(0, 120s) & \text{otherwise} \end{cases}$$

$$ELDT - PLDT = N(0, 0.02\Delta T)$$

where ΔT is the time difference between ELDT/ETOT and the current time. This means that the uncertainty of ELDT is linear to the estimated flight time to the runway with the standard deviation of 2% flight time. The uncertainty of ETOT is assumed to become small at 15 minutes before ETOT, because at this time the largest contributor to the uncertainty, i.e. the boarding process, is completed.

The decision variables are the landing runway of each arrival aircraft only. However, the last minutes change of landing runway is not possible considering both ATC and pilot workload perspectives, so it is assumed that the runway decision must be made 30 minutes before ELDT. Therefore, the runway decision must be made under uncertainty.

B. Optimization method

The optimized parameters are $\delta_i \forall i \in A$. However, all δ_i do not need to be optimized at the same time, because aircraft due to depart further in the future cause little impact on the runway decision of the aircraft 30 minutes before the landing. As in other researches, this time, the sliding windows approach is applied[11]. The sliding window approach optimizes the aircraft with a certain time window of ELDT only, and the target aircraft are changed as time proceeds. This optimization process is done repeatedly, with the latest result being used for decision making.

Assuming that there are n aircraft to be optimized, the possible combination is 2^n. If n is sufficiently small, the objective function for all combinations is calculated, and the best one is chosen. However, when n increases, it is impossible to calculate all combinations. Therefore, simulated annealing is applied to find the best solution. The simulated annealing is a metaheuristic optimization algorithm, and imitates the annealing process in metallurgy. The simulated annealing is often used to solve combinatorial optimization problems, and is therefore applicable here as well. The general optimization process of the simulated annealing is shown as follows.

Initialization $i := i_0$, $T := T_0$
While $T > T_{\text{final}}$ do
  Generate solution j near solution i
  if $f(i) > f(j)$ then $i := j$
  else $i := j$ with probability of $\exp\left(\frac{f(i) - f(j)}{T}\right)$
  end if
  Compute T
i becomes the final solution.

III. NN DEVELOPMENT FOR OBJECTIVE FUNCTION CALCULATION

A. Inputs and outputs of the network

In this research, the authors propose that the objective function is calculated by data-trained NN. Therefore, first the

NN inputs and outputs should be decided. First the inputs are considered. The possible inputs are the ETOT/ELDT of each aircraft and its runway information. This time, to represent both ETOT/ELDT and runway information, the number of aircraft on each runway at each time slot are set as inputs. Fig. 2 shows the image of the inputs. There are four different queues in this runway operation (runway A for arrival, runway C for arrival, runway C for departure, and runway D for departure). In each queue, the number of aircraft is set to each time slot of ETOT/ELDT. The size of time slot is set to 120 s. The time slot starts with the current time and ends at the end of sliding window. Since the runway re-assignment is considered within the sliding window only, the aircraft out of the sliding window do not affect the decision of the runway re-assignment. In this example, 1 hour sliding window is assumed, and 4*30 inputs are made. This input size (1 hour sliding window) is used for NN development.

	Current time							End of sliding window	
Queue	**00:00**	**00:02**	**00:04**	**00:06**	**00:08**	**00:10**	**…**	**00:56**	**00:58**
Arr-A	0	0	1	2	0	0	…	0	2
Arr-C	1	2	0	1	0	0	…	0	3
Dep-C	0	1	0	1	1	1	…	1	0
Dep-D	2	0	0	0	1	0	…	2	1

Figure 2. Representation of NN inputs.

As for the outputs, the objective function is the sum of the delay and the number of runway re-assignment. Only the delay should be calculated by NN, and the number of runway re-assignment is implicitly incorporated in the decision variable. This time, the following four output values are set as NN outputs.

$$o_1 = \sum_{j \in \{A | r_j = a\}} \left(t_j - ELDT_j \right) \tag{2}$$

$$o_2 = \sum_{j \in \{A | r_j = c_A\}} \left(t_j - ELDT_j \right) \tag{3}$$

$$o_3 = \sum_{i \in \{D | r_i = c_D\}} \left(t_i - ETOT_i \right) \tag{4}$$

$$o_4 = \sum_{i \in \{D | r_i = d\}} \left(t_i - ETOT_i \right) \tag{5}$$

Each output is affected by different inputs (aircraft). Since it is easier to develop several small networks than a single large network, four separate NNs are made. The sum of all four outputs corresponds to the delay of all aircraft, which exactly matches the objective function given in Eq. (1) except the number of runway re-assignment. Each network requires only the necessary inputs that affect the output. Table 1 summarizes the used input in each network.

B. NN structures

Next, NN structures are considered. There are various types of possible networks, such as feedforward fully-connected NN (FFNN) and convolutional NN (CNN). To estimate the delay of aircraft, a single aircraft usually affects the nearby aircraft, and queue length is propagated to the later aircraft. As for FFNN, all inputs are connected, but most connections are actually not needed for estimating the delay. The unnecessary connections often cause over-fitting and results in the failure of training a network. On the other hand, CNN connects the neighborhood inputs only, and smaller network can be created. A popular application of CNN is image processing. Although the inputs are 2 dimensional in image processing, time sequence data in this research is considered as 1 dimensional data, and CNN is applied.

TABLE I. INPUTS AND OUTPUTS USED IN EACH NETWORK.

Network name	Inputs	Outputs
Network1	Arr-A (30 inputs)	o_1
Network2	Arr-C (30 inputs)	o_2
Network3	Arr-C, Dep-C (30*2 inputs)	o_3
Network4	Arr-C, Dep-D (30*2 inputs)	o_4

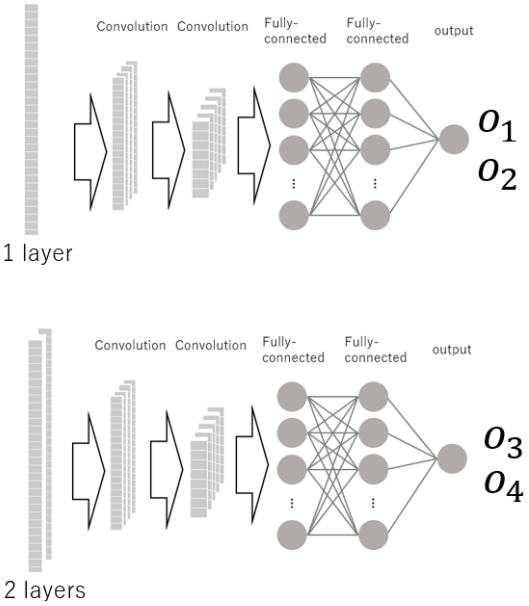

Figure 3. NN structures used in this research.

TABLE II. THE PARAMETERS IN NN USED IN THIS RESEARCH.

Layers	Variables	Values	
		Network1,2	***Network3,4***
1st (convolution)	Number of out channels	128	256
	Strides	3	3
	Kernel size	3	3
2nd (convolution)	Number of out channels	128	256
	Strides	2	2
	Kernel size	2	2
3rd (fully-connected)	Number of hidden nodes	128	256
4th (fully-connected)	Number of hidden nodes	128	256
1st-4th	Activation function	ReLu[18]	ReLu
Output	Activation function	Linear	Linear

Fig. 3 summarizes the NN structures used in this research. The first and second layers use a convolutional layer, and the third and fourth layers use a fully-connected layer. Table 2 summarizes the detailed values of the network. Since Network3 and Network4 have more inputs than Network1 and Network2, the NN size is set larger in Network3 and Network4. However, the general NN structures are set the same for all networks.

C. Training data and NN training

Training data sets are required to train the NN. In reality, the actual operational data is used, but this time data must be generated via simulations. The data should include various scenarios, and various patterns of runway balancing, otherwise the data would be biased, and the appropriate NN cannot be made. Therefore, through a data generation process, NN is trained iteratively and its output is used for runway assignment of each arrival aircraft in the simulation. In order to simulate various situations, α is randomly set with exponential distribution of scale parameter 600 s, and the runway assignment is randomly set with a probability of 0.05.

The simulation parameters are given later in Sec. IV A. The data is obtained every minute for 3 hours, so a single simulation can generate 180 data sets. By running about 1700 times of simulations, 300,000 data sets are created.

Once the training data is obtained, NN is trained. The well-known training algorithm Adam[19] is applied here. During a training process, the generalization capability is a big issue. Generalization refers to the ability of the NN to produce reasonable outputs for inputs that are not encountered during training. NN tries to minimize the loss function between model output and trained data output. Since both inputs and outputs data usually include noise, minimizing the error loses the generalization (called over-fitting). There are various ways to avoid over-fitting, but one method is to collect sufficient number of data.

However, the possible number of data obtained is also limited if this process is implemented using real operational data. As for the airport operational data, it is assumed that each day consists of 16 operational hours, with data obtained every 1 minute. For 1 year, 60*16*365=350,400 is the maximum possible number of data obtained. However, a good NN could not be made with 350,400 data sets according to the authors' pre-calculation. Consequently, data augmentation technique is used. The data augmentation is the technique where the data is artificially made based on the obtained data. This technique is often used in the image processing field. In this field, the data augmentation is done by various ways such as rotating the image, reflecting the image, and changing the scaling.

In this research, data augmentation is done by the following way. 1) copy the input/output from the original data. 2) pick up one aircraft from the input data. 3) the aircraft moves to the next time slot. This process is very easy, but it should work because ETOT/ELDT includes uncertainty, so similar output is expected even if the ETOT/ELDT is slightly changed. This time, it is assumed that 300,000 data are available in advance. First, the data is augmented by 9 times in each original data set, and the number of data becomes 3,000,000 (300,000*(9+1)). Second, the data is split into 70 % training data (2,100,000) and 30% test data (900,000). The NN is trained with the training data, and the NN where loss function between model output and test data output is the minimum is used as the obtained NN.

Also, batch training is applied, and the batch size is set to 2048. As for the loss function, the mean squared error (MSE) is used.

IV. Simulation Results

A. Simulation parameters

Before conducting a simulation, several simulation parameters need to be set. First, the traffic volume is decided as shown in Table 3. There are three scenarios considered. Scenario 1 is low traffic, Scenario 2 is heavy traffic, and Scenario 3 covers various traffic. Without the optimization of runway re-assignment, there is an optimal ratio of runway balance for both departures and arrivals, and the nominal ratio should not be significantly different from the optimal ratio. Therefore, the nominal ratio of arrival/departure is also set in the scenario. As for the departure, 0.5 is the best ratio of departure, so it is randomly set between 0.4 and 0.6 in the simulation. As for the arrival, 0.85 is the best ratio of arrival for A Runway, and distributed between 0.75 and 0.95. Scenario 3 is used for NN training as explained in Sec. III, and Scenarios 1 and 2 are used to investigate the simulation results.

In the simulation, first, PTOT/PLDT are randomly distributed for 3 hours, and the simulation is conducted until all departure and arrival aircraft take off or land. The optimization of runway assignment of each arrival aircraft is conducted every 10 minutes, and the runway decision is made 30 minutes before ELDT. Since the time window for the optimization is set for the next 60 minutes, the runway is optimized for arrival aircraft where ELDT is between t + 30 and t + 60. The runway actually assigned is the one obtained in the latest optimization process.

TABLE III. Traffic volumes in each scenario.

	Traffic [aircraft/hour]		
Scenario	1	2	3
Total arrival	20	32	15-34
Total departure	20	32	15-34
Nominal ratio of arrival for A Runway	0.75-0.95	0.75-0.95	0.75-0.95
Nominal ratio of departure for C Runway	0.4-0.6	0.4-0.6	0.4-0.6
Total number of aircraft	120	192	90-204

Next, Table 4 shows the parameters of simulated annealing. If the number of possible combinations is less than the number of iterations, the objective functions of all possible combinations are calculated, and the best one is chosen, which is actually a deterministic method.

TABLE IV. PARAMETERS OF SIMULATED ANNEALING.

Parameters	Values
Number of iterations	1024
Initial temperature	15000
Terminal temperature	7.5

B. *Calculation of objective function without considering uncertainty*

To proceed with the optimization process, the objective function must be calculated. In this research, the authors propose to calculate it by NN. However, the simulation method is also used here as a benchmark. To calculate the objective function, the simulation method developed in Sec. II A is used, but no uncertainty is considered, i.e. the simulation runs assuming that $PLDT_i = ELDT_i \forall i \in A$, $PTOT_i = ETOT_i \forall i \in D$, and $S_{ij} = 120s$ for given conditions. Since this simulation does not consider uncertainty, the objective function calculated by this simulation differs from that by the simulation including uncertainty. This discrepancy can cause the inappropriate assignment of arrival runway for each aircraft. In reality, the separation $S_{ij} = 120s$ is also sometimes difficult to be identified, but this time it is assumed that the "average" separation is already known in advance. From now, the case where the objective function is calculated by the simulation without considering uncertainty is denoted by SIM method.

C. *Comparison between SIM method and NN method*

To compare the results obtained by SIM method and NN method, 100 times simulations are conducted in each scenario (1 and 2). Since the same random seed is used between SIM method and NN method, completely the same initial values, the same randomized separations, and the same randomized ETOT/ELDT errors are used in both methods. The optimization calculation is conducted with various α in the objective function between 0 s and 2400 s, which models the cost of a single runway re-assignment.

First, the accuracy of NN performed estimations are discussed. Fig. 4 shows the delay estimation result of both methods by arrival on Runway A (which corresponds to o_1), while Fig. 5 shows the result by departure on Runway C (which corresponds to o_3). The other two values (o_2 and o_4) also show a similar trend. The total delay in this result indicates the sum of the delay for next 60 minutes time windows, and is used to calculate the objective function. Also, this is the result of Scenario 2 where α is 60 s. The root mean square error (RMSE) values are also included in the result. As expected, the RMSE by NN is smaller than that by SIM for all cases, which means that the NN can estimate the total delay more accurately than SIM method. The delay of arrival is estimated better than that of departure, because the uncertainty of arrival is smaller than that of departure. The SIM method estimates the relatively smaller delay than the actual delay. This can be understood by the queueing theory. According to the queueing theory, the arrival process on Runway A can be described as M/G/1 queue[20], and the average waiting time increases if there exists uncertainty of the service time (which corresponds to the

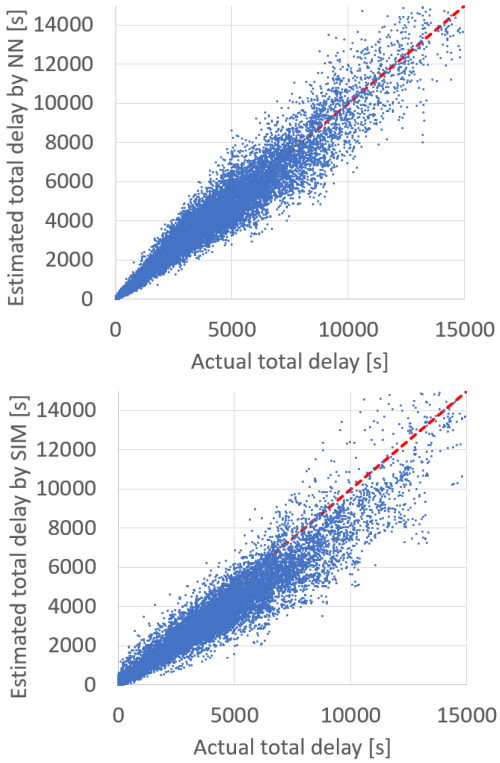

Figure 4. The estimated total delay by NN and SIM methods compared to the actual delay by arrivals on Runway A. (RMSE=809s by NN, RMSE=1122s by SIM)

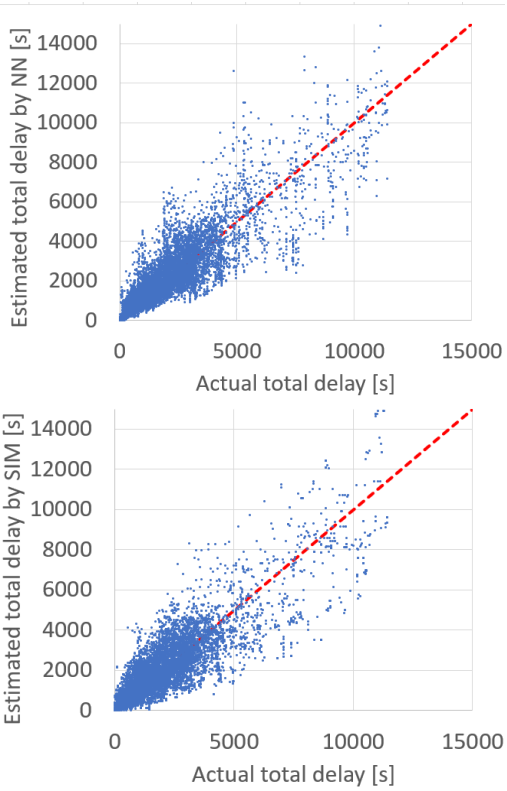

Figure 5. The estimated total delay by NN and SIM methods compared to the actual delay by departures on Runway C. (RMSE=927s by NN, RMSE=1009s by SIM)

landing separation here). SIM method does not consider the uncertainty, so its estimate of the waiting time tends to be lower than the actual one. On the other hand, the NN can estimate the delay better than the SIM method, because NN models the delay considering uncertainty. As for the departure, the delay is estimated less accurately than the departure, because the uncertainty of the departure is assumed to be much larger than that of the arrival. However, the NN can estimate the delay slightly better than SIM method when measured by RMSE.

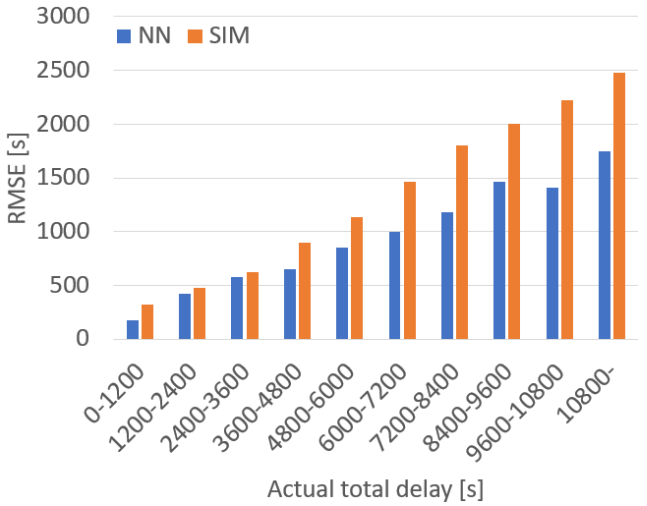

Figure 6. RMSE in each total delay range by arrivals on Runway A.

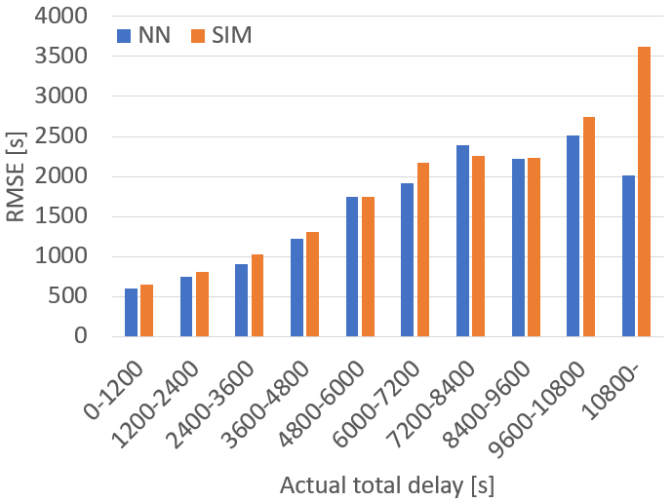

Figure 7. RMSE in each total delay range by departures on Runway C.

To analyze the estimation accuracy in more detail, Fig. 6 shows the RMSE of each method in each range of actual total delay by arrivals on Runway A, while Fig. 7 shows the result by departures on Runway C. In general, the RMSE tends to increase with the actual total delay. As for arrivals, NN always estimates the delay better than SIM method. However, as for departures, SIM method sometimes estimates the delay better than NN when the actual total delay is large. This can also be understood by Figs. 4 and 5. As for arrivals, the number of plots becomes small as the actual total delay increases, but there are enough number of plots even when the actual total delay is large. However, as for departures, the number of plots gets significantly small when the actual total delay is larger than 5000 s. Since NN is trained by the data, the training process does not work well if there is not enough training data. According to Fig. 7, NN can always estimate the delay better than the SIM method where there seems to be sufficient data. However, NN can estimate the delay better than SIM method in most cases, and the difference of the delay accuracy between two methods (SIM and NN) can affect the solution of the runway balancing.

Next, Fig. 8 shows the total delay after the optimization by both methods under scenario 1 (low traffic), while Fig. 9 shows the result under scenario 2 (heavy traffic). Here, the total delay means that the sum of delay for all aircraft for a single simulation. The dotted line indicates the “no optimization” result. Therefore, the results of both methods converge to “no optimization” when the number of runway re-assignment is 0. In each parameter of α, the optimization is conducted in each method, and the following 12 values of α are used. [0, 60s, 120s, 180s, 240s, 360s, 480s, 600s, 900s, 1200s, 1800s, 2400s].

When traffic volume is small (Fig. 8), the average flight delay is also small compared to that in the heavy traffic case. Also, the possible reduction of delay is also limited. Since there are 20*2(dep/arr)*3(hours)=120 aircraft in a single simulation, 1 s average flight delay corresponds to 2 minutes total delay. SIM method does not estimate the objective function very accurately, so the delay increases when α is small because unnecessary runway re-assignment can increase the delay due to the term w_{r_1,r_2}. However, that is not the case when using NN, and the minimum delay is observed when α is 0. While SIM method can reduce the waiting time by 8.7 s per aircraft via runway balancing, NN can reduce the waiting time by 11.8 s per aircraft, or additional 36% savings. On the other hand, when the number of runway re-assignment is small, the possible delay reduction is almost the same for both SIM and NN methods. That could be because “large benefit” runway re-assignment strategy is the same regardless of the consideration of uncertainty. If the number of runway re-assignment is limited, the advantage of the proposed method may be limited.

As for heavy traffic (Fig. 9), the delay per aircraft is large, and the possible delay reduction is also large. However, the difference between SIM and NN methods is not quite big. SIM method can reduce the delay by 64.0 s per aircraft, while the NN method can reduce the delay by 68.2 s per aircraft, about 7 % larger. This could be because many departure aircraft exist anyway under heavy traffic, so even if one departure aircraft has large uncertainty, another departure aircraft can compensate it, and the uncertainty effect could be reduced. However, NN method still outperforms the SIM method. These results infer that the proposed method using NN is very promising.

This time, simulation environment is developed in a simple manner, which is advantageous for SIM method. However, in reality, much more complicated airport operation is conducted,

so the advantage of the proposed data-driven NN method will be significant in the real world.

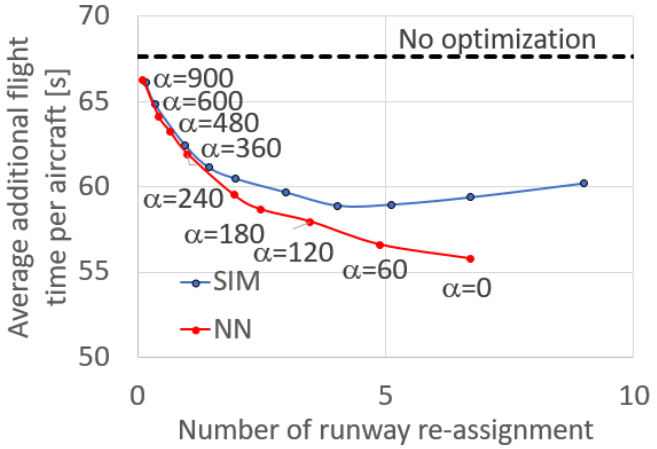

Figure 8. Average additional flight time (flight delay) per aircraft on Scenario 1.

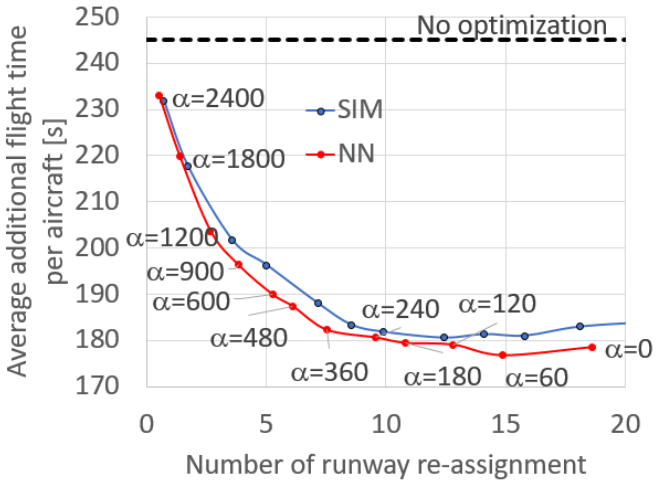

Figure 9. Average additional flight time (flight delay) per aircraft on Scenario 2.

V. Conclusions

This paper proposed a new scheme to optimize a runway balancing problem using NN. NN modeling approach does not require any explicit operational models and simulations, and relies on actual operational data only. This means that simulation parameters such as departure/arrival separations and its interactive effect are not required, because these characteristics are expected to be modeled by NN. This paper showed the effectiveness of the proposed method via simulations. The uncertainty effect was appropriately modeled by NN, and the solution of the runway balancing was improved. This time, relatively simplified airport operation was assumed, so it will be interesting to investigate how much NN can handle more complicated operation. Also, the proposed method will be applied using the actual operational data, and its ability will be investigated. These will be subjects of the future work.

References

[1] International Civil Aviation Organization, "2016-2030 Global Air Navigation Plan," Doc 9750, 2016.

[2] EUROCONTROL, "RECAT-EU, European Wake Turbulence Categorisation and Separation Minima on Approach and Departure," Edition 1.2, 2018.

[3] Cheng, V. H. L., Crawford, L. S., Menon, P. K., "Air traffic control using genetic search techniques," Proceedings of the 1999 IEEE International Conference on Control Applications, IEEE, 1999.

[4] Hu, X.-B., Di Paolo, E., "An efficient genetic algorithm with uniform crossover for air traffic control," Computers and Operations Research, 36(1), 245-259, 2009.

[5] Fritzsche, M., Günther, T., Fricke, H., "Potential of Dynamic Aircraft to Runway Allocation for Parallel Runways," 4th International Conference on Research in Air Transportation, 2010.

[6] Bencheikh, G., Boukachour, J., Alaoui, A. E. H., "Improved ant colony algorithm to solve the aircraft landing problem," International Journal of Computer Theory and Engineering, 3(2), 224-233, 2011.

[7] Ng, K. K. H., Lee, C. K. M., Chan, F. T., Qin, Y., "Robust aircraft sequencing and scheduling problem with arrival/departure delay using the min-max regret approach," Transportation Research Part E: Logistics and Transportation Review, 106, 115-136, 2017.

[8] Salehipour, A., Modarres, M., Naeni, L. M., "An efficient hybrid meta-heuristic for aircraft landing problem," Computers and Operations Research, 40(1), 207-213, 2013.

[9] Liang, M., Delahaye, D., Maréchal, P., "Integrated sequencing and merging aircraft to parallel runways with automated conflict resolution and advanced avionics capabilities," Transportation Research Part C: Emerging Technologies, 85, 268-291, 2017.

[10] Roling, P. C., Delsen, J., Curran, R.. "Flexible Arrival & Departure Runway Allocation Using Mixed-Integer Linear Programming," 17th AIAA Aviation Technology, Integration, and Operations Conference, AIAA 2017-4257, 2017.

[11] Ma, J., Delahaye, D., Sbihi, M., Scala, P., Mota, M. M., "Integrated optimization of terminal maneuvering area and airport at the macroscopic level," Transportation Research Part C: Emerging Technologies, 98, 338-357, 2019.

[12] Malaek, S. M. B., Naderi, E., "A new scheduling strategy for aircraft landings under dynamic position shifting," 2008 IEEE Aerospace Conference, IEEE, 2008.

[13] Xie, J., Zhou, Y., Zheng, H., "A hybrid metaheuristic for multiple runways aircraft landing problem based on bat algorithm," Journal of Applied Mathematics, 742653, 2013.

[14] Heidt, A., Helmke, H., Liers, F., and Martin, A., "Robust Runway Scheduling Using a Time-Indexed Model," Fouth SESAR Innovation Days, SESAR JU, 2014.

[15] Khassiba, A., Bastin, F., Gendron, B., Cafieri, S., Mongeau, M., "Extended Aircraft Arrival Management under Uncertainty: A Computational Study," Research Report CIRRELT, 2018.

[16] Mori, R., "Refined Collision Risk Model for Oceanic Flight Under Longitudinal Distance-Based Separation with ADS-C Environment," Journal of Navigation, Royal Institute of Navigation, Vol. 67, No. 5, pp.845-868, 2014.

[17] Mori, R., "Development of a Pushback Time Assignment Algorithm Considering Uncertainty," Journal of Air Transportation, AIAA, Vol. 25, No. 2, pp. 51-60, 2017.

[18] Nair, V., Hinton, G. E., (2010), "Rectified linear units improve restricted boltzmann machines," In Proceedings of the 27th international conference on machine learning (ICML-10), 807-814, 2010.

[19] Kingma, D. P., Ba, J. L., "Adam: a Method for Stochastic Optimization," International Conference on Learning Representations, 1–13, 2015.

[20] Bhat, U. N., An Introduction to Queuing Theory, Birkhäuser, Boston, 2008.