# Uncertainty Inclusive Runway Balancing Using Convolutional Neural Network

Ryota Mori, Daniel Delahaye

# Uncertainty Inclusive Runway Balancing Using Convolutional Neural Network

Ryota Mori*
*Electronic Navigation Research Institute, Tokyo 1820012, Japan*
and
Daniel Dalahaye†
*ENAC, National School of Civil Aviation, 31400 Toulouse, France*

https://doi.org/10.2514/1.D0313

**This paper proposes a new optimization scheme using neural networks for runway balancing to minimize departure and arrival aircraft delay. The delay prediction for runway balancing optimization is obtained by a neural network, only without any additional simulations. Developing an accurate simulation model under an uncertain environment is difficult, but the proposed neural network model can estimate the average delay without modeling uncertainty explicitly. In this paper, the effectiveness of the proposed method is validated through numerical simulations. First, simulations are used to generate the data, which are then used to train the neural network. Next, the runway balancing problem is solved via simulated annealing using the delay predicted by the neural network. The simulation result shows that the proposed approach outperforms the simulation-based method under an uncertainty environment. Therefore, the neural network is shown to accurately estimate the delay under the uncertainty environment, which makes the proposed neural-network-based method applicable to objective function calculations for optimization.**



## I. Introduction

AIRPORT ground operations are a bottleneck of air traffic control. More efficient runway usage is necessary to maximize capacity and optimize traffic handling [1]. Apart from new runway constructions, there are two main approaches to reduce aircraft delay: takeoff/landing separation reduction and runway balancing. The former is a straightforward idea to reduce the delay, with various methods proposed such as optimal aircraft sequencing while considering the wake turbulence category, time-based separation applied under strong headwinds, and development of new separation standards (RECAT: Wake Turbulence Re-Categorization) [2]. The latter (runway balancing), on the other hand, aims at optimal runway balancing of takeoff and landing aircraft to multiple runways. This is a promising solution for major hub airports with multiple runways, and so this research focuses on runway balancing.

There have been many studies to optimize the runway balancing of departures and/or arrivals. To optimize the runway balancing, several types of objective functions can be minimized, depending on the problems such as aircraft delay [3–5], makespan [6], and the combination of makespan/aircraft delay as well as other factors such as noise, fuel consumption, and congestion level [7–13]. The runway balancing is a combinatorial optimization, where various optimization methods can be applied such as mixed integer linear programming [11], genetic algorithm [3,7], simulated annealing [4,10,12], ant colony optimization [9], swarm intelligence algorithm [6], bat algorithm [5], greedy method [8], and dynamic programming [13]. Furthermore, many researchers optimize the aircraft sequencing and runway balancing simultaneously. Operational constraints, such as conflict on air routes and taxiing, are also considered in some studies [12]. Most researchers focus deterministic environment only, and uncertainties are considered in relatively few past works. Such uncertainties are always present in the real world, with takeoff time uncertainties exceeding flight time, and thus arrival time uncertainties, due to difficulties in the boarding process prediction times [14]. 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 for the worst-case scenario. Runway sequencing has been optimized with robust optimization, as shown in Ref. [15]. With this method, the optimization is translated into a deterministic optimization problem. However, the optimal parameters found for the worst-case scenario are not necessarily optimal for various uncertain scenarios. Another method considers the stochastic process internally, and the expected value is optimized. The stochastic programming is a common method to consider the uncertain environment, and there have been many works on runway scheduling [16–18]. Because a large-scale problem is sometimes difficult to solve directly, a simulation-based method is often used. This approach requires a large number of simulation runs varying the uncertainty parameters, followed by some statistic evaluation of the obtained simulation results (sample average approximation [19]). This method is referred to as a simulation method here. There are some studies using the simulation method for runway balancing and sequencing [6,20]. This method can consider many uncertain scenarios, but it is computationally expensive.

However, both methods discussed earlier in this paper share a common issue: they require an accurate runway simulation model. In particular, the following parameters influence the results greatly: takeoff/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. Inaccurate simulation models decrease the fidelity of the solution. To tackle this issue, the current authors proposed a new optimization approach without using a simulation or parameter estimation of airport operations in their previous research [21]. The optimization requires an objective function, but the authors proposed that such an objective function can be obtained by a neural network (NN) trained with the actual operational data directly. Because the NN training process requires both inputs and outputs from actual data, the NN can learn the actual operational environment directly. In addition, the actual operation includes various types of uncertainties, which are difficult to handle in the optimization. From a NN perspective, uncertainties in the output resemble noisy output; so, with a sufficiently large dataset, a good enough NN model can be built despite the existing uncertainties. Therefore, with a large dataset, the appropriate input–output

Received 7 March 2022; revision received 14 August 2022; accepted for publication 19 August 2022; published online Open Access 13 September 2022. 

*Chief Researcher, Air Traffic Management Department. Member AIAA.
†Professor, Head of Optim Group.

mapping can be obtained. As a result, the NN can estimate the output of the expected value by considering uncertainty. By doing that, an optimization problem considering uncertainty can be converted to a simple deterministic problem, and the classical optimization method can be used.

In their previous work, the current authors demonstrated the initial potential of the proposed method [21]. This paper elaborates on the method by considering a more realistic environment, and it focuses on the detailed performance of the proposed method. A more realistic airport simulation model [22] is implemented based on real operational data, and the performance of the proposed method is first investigated under a deterministic environment; then, the performance of the proposed method is evaluated under an uncertainty environment. The data size sufficient for NN training is also investigated.

The rest of the paper is organized as follows: The problem is formulated in Sec. II, and the NN model is developed in Sec. III. Section IV shows the simulation environment, followed by Sec. V, which presents the simulation results. Section VII concludes this paper.

## II. Problem Formulation

### A. Airport Operation and Simulation Method

The target airport with a runway balancing problem that is considered in this paper is Tokyo International Airport (Haneda Airport). Figure 1 shows the airport layout and runway operations under north wind. Runway A is used for arrival only, and it is independent of traffic on other runways. On the other hand, runway C is used for both departure and arrival, i.e., a mixed-mode operation. Runway D is used for departure only, but the arrival to runway C also affects the departure from runway D. Runway B is not used under north wind operation due to airspace limitation, and it is used under south wind operation only. Under this condition, the runway balancing for arrivals between runways A and C is considered.

In this paper, the aforementioned runway operation is described in the following mathematical forms:

#### *1. Inputs*

The set of departure aircraft is defined as $\mathcal{D}$; $\mathcal{D} = \{1, \ldots, n_D\}$.

The set of departure runways is defined as $R_D$; $R_D = \{c_D, d\}$, where $c_D$ is runway C used by departures and $d$ is runway D.

The set of arrival aircraft is defined as $\mathcal{A}$; $\mathcal{A} = \{1, \ldots, n_A\}$.

The set of arrival runways is defined as $R_A$; $R_A = \{a, c_A\}$, where $a$ is runway A and $c_A$ is runway C used by arrivals.

$\mathrm{PTOT}_i$ is the earliest possible takeoff time (PTOT) for aircraft $i, \forall\ i \in \mathcal{D}$.

$\mathrm{PLDT}_i$ is the earliest possible landing time (PLDT) for aircraft $i, \forall\ i \in \mathcal{A}$.

$r_i^{\mathrm{nom}}$ is the nominal runway for departure and arrival; $r_i^{\mathrm{nom}} \in R_D\ \forall\ i \in \mathcal{D}$ and $r_i^{\mathrm{nom}} \in R_A\ \forall\ i \in \mathcal{A}$.

$r_i$ is the departure/arrival runway; $r_i = r_i^{\mathrm{nom}}\ \forall\ i \in \mathcal{D}$ and $r_i \in R_A\ \forall\ i \in \mathcal{A}$.

#### *2. Decision Variables*

The binary variable for arrival runway decision ($\delta_i\ \forall\ i \in \mathcal{A}$) is set as a decision variable:

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

#### *3. Variables Given by the Airport Operation*

The actual takeoff or landing time is defined as $t_i$; $\forall\ i \in \mathcal{A} \cup \mathcal{D}$.

#### *4. Constraints*

For runway separation,

$$t_j \geq t_i + S_{ij} \quad \forall\ i \neq j \in \mathcal{A} \cup \mathcal{D}$$

where $S_{ij}$ is the minimum separation between aircraft $i$ and $j$, where aircraft $i$ precedes aircraft $j$:

$$S_{ij} \begin{cases} = -\infty & \text{if } t_i < t_j \\ \sim N(110\ \mathrm{s}, u) & \text{if } r_i = r_j = a \text{ for } i, j \in \mathcal{A}(*) \\ \sim N(240\ \mathrm{s}, u) & \text{if } r_i = r_j = c_A \text{ for } i, j \in \mathcal{A}(*) \\ \sim N(65\ \mathrm{s}, u) & \text{if } r_i = c_A, r_j = c_D \text{ for } i \in \mathcal{A}, j \in \mathcal{D}(*) \\ \sim N(85\ \mathrm{s}, u) & \text{if } r_i = c_D, r_j = c_A \text{ for } i \in \mathcal{D}, j \in \mathcal{A}(*) \\ \sim N(100\ \mathrm{s}, u) & \text{if } r_i = r_j = d \text{ for } i, j \in \mathcal{D}(*) \\ \sim N(80\ \mathrm{s}, u) & \text{if } r_i = d, r_j = c_A \text{ for } i \in \mathcal{D}, j \in \mathcal{A}(*) \\ = 5\ \mathrm{s} & \text{if } r_i = c_A, r_j = d \text{ for } i \in \mathcal{A}, j \in \mathcal{D}(*) \\ = 0 & \text{otherwise} \end{cases}$$

For the earliest landing time,

$$t_i \geq \mathrm{PLDT} + w_{r_i, r_i^{\mathrm{nom}}} \quad \forall\ i \in \mathcal{A}$$

where $w_{r_1, r_2}$ indicates the additional flight time required when the nominal runway is $r_1$ and the landing runway is $r_2$:

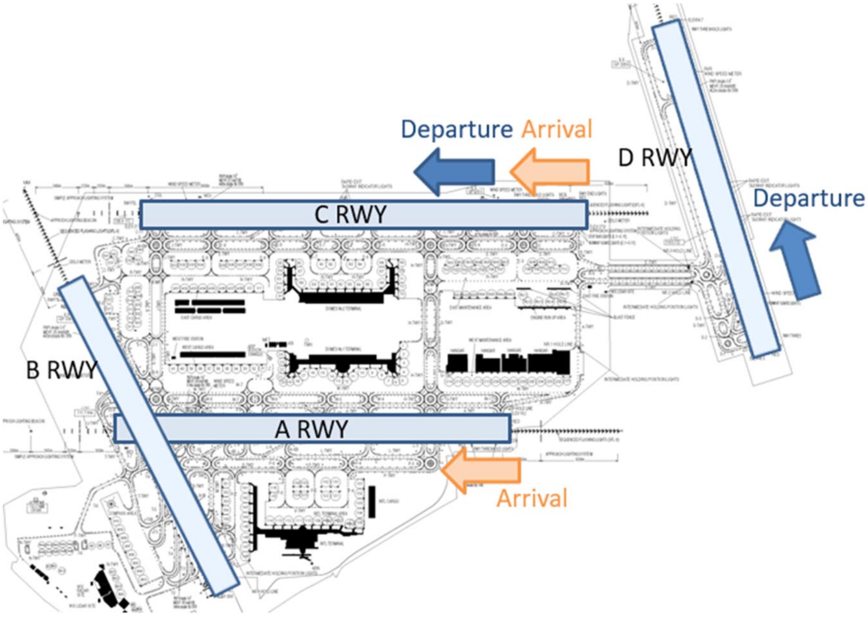


**Fig. 1 Airport layout and runway (RWY) configuration under north wind at Tokyo International Airport.**

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

For the earliest takeoff time,

$$t_i \geq \text{PTOT} \quad \forall\, i \in \mathcal{D}$$

Arrivals are always given priority over departures.

The separation and mutual interference between runways $S_{ij}$ are set according to previous research [22] using real operational data. $N(a, b)$ indicates the normal distribution with the average $a$ and standard deviation (SD)$b$, and $u$ is set to 15 s [22]. When $S_{ij}$ becomes negative after the calculation of $N(a, b)$, it is set to zero. The model can easily be adapted to consider any wake turbulence constraints. In the real world, an aircraft does not necessarily take off or land, even when the sufficient separation is established. These outliers are modeled by the following calculation, with the probability of 1% only when the condition $(*)$, which appears in the calculation of $S_{ij}$, is applied:

$$S_{ij} = \begin{cases} S_{ij} + 120 \text{ s} & 1\% \text{ probability when } (*) \text{ is applied} \\ S_{ij} & \text{otherwise} \end{cases} \tag{1}$$

This means that 120 s of additional separation is required, which models the outlier of the separation. Additionally, the characteristic operation is found in arrival aircraft landing to runway C. The standard 240 s separation is set to the landing aircraft to runway C so that the takeoff is possible from runway C and runway D between the two landings. This also means that arrival to runway C requires long separation, regardless of departure aircraft presence. Therefore, the separation is not constant but randomly distributed. Arrival is always given priority over departure. Arrivals are always given priority over departures, i.e., aircraft cannot depart from runways C and D unless sufficient separation with runway C arrivals is established. Once the inputs and decision variables are determined, the $t_i$ of the arrival aircraft are first calculated on the order of the PLDTs to satisfy the constraints. After that, the $t_i$ of the departure aircraft are calculated in the order of the PTOTs to satisfy the constraints.

The objective function to be minimized is the total delay of all aircraft. Here, the delay is defined as the difference of actual takeoff/landing time and the estimated takeoff/landing time when no other aircraft traffic exists. However, the landing runway reassignment increases air traffic controller (ATC) workload, and so the number of runway reassignments is also included in the objective function, which is defined as the following form where α is the weight factor. Here, the runway reassignment indicates the case where the nominal and the actual runways are different:

$$J = \sum_{i \in \mathcal{D}} (t_i - \text{ETOT}_i) + \sum_{i \in \mathcal{A}} (t_i - \text{ELDT}_i + \alpha \delta_i) \tag{2}$$

where ETOT denotes the estimated takeoff time, and ELDT denotes the estimated landing time. If the PLDTs and PTOTs 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 include a considerable uncertainty, but departure time uncertainties are larger due to delayed pushback, which can be initiated only once all passengers are on board, as discussed by other researchers [14]. The estimated times of the PLDTs and PTOTs are denoted by ELDT and ETOT, respectively. Note that the PLDT and PTOT are unknown in the optimization process, and only the ELDT and ETOT are available. Here, the following uncertainty value is assumed according to past research [23,24]:

$$\text{ETOT} - \text{PTOT} \sim \begin{cases} N(0, 300 \text{ s}) & \text{if } \Delta T > 15 \text{ min} \\ N(0, 120 \text{ s}) & \text{otherwise} \end{cases} \tag{3}$$

$$\text{ELDT} - \text{PLDT} \sim N(0, 0.02\Delta T) \tag{4}$$

where $\Delta T$ is the time difference between the ELDT/ETOT and the current time. This means that the uncertainty of the ELDT is linear to the estimated flight time to the runway with the SD of 2% flight time. The uncertainty of the ETOT is assumed to decrease 15 min before the ETOT because, at this time, the largest contributor to the uncertainty (i.e., the pushback time) is completed.

The decision variables are the landing runways of each arrival aircraft only. However, last-minute changes to the landing runway are not possible from both the ATC and pilot workload perspectives, and so it is assumed that the runway decision must be made 30 min before ELDT, i.e., the runway decision must be made under uncertainty.

Note that some constraints are given by random variables, but all these random variables are determined in advance, and so the problem itself is deterministic. However, the random variables are changed in each run, and so multiple runs of simulations are required to evaluate the uncertainty effects.

### B. Optimization Method

The optimized parameters are $\delta_i\ \forall\ i \in \mathcal{A}$. This paper assumes that each aircraft can land at the designated optimized runway; although, in practice, the optimized runway assignment is not necessarily realized due to the airspace or aircraft conflicts. However, not all $\delta_i$ need to be optimized simultaneously because aircraft due to depart further in the future have little impact on the runway decision of the aircraft 30 min before the landing. As in other studies, this time, the sliding windows approach is applied [12]. The sliding window approach optimizes the aircraft with a certain time window of the 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 there are $n$ aircraft to be optimized, the possible combinations are $2^n$. 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 infeasible to calculate all combinations. Therefore, simulated annealing is applied to find the best solution. The simulated annealing [25] is a metaheuristic optimization algorithm that imitates the annealing process in metallurgy. The simulated annealing is often used to solve combinatorial optimization problems. The general optimization process of the simulated annealing is shown as follows:

```
Initialization i := i_0, T := T_0
While T > T_final, do
  Generate solution j near solution i
  if f(i) > f(j), then i := j
  else i := j with probability of exp((f(i) − f(j))/T)
  end if
  Compute T
i becomes the final solution
```

where $T$ denotes the temperature. $T$ is large at the beginning, which means that the worst solution is likely to be chosen to find the larger solution space. As time proceeds, the solution improves until it converges to the optimal solution.

## III. NN Development for Objective Function Calculation

### A. Inputs and Outputs of the Network

In this research, the authors propose to calculate the objective function by data-trained NN. Therefore, 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 the ETOT/ELDT and runway information, the number of aircraft on each runway at each time slot is set as input. Figure 2 shows the representation 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 the ETOT/ELDT. The size of the time slot is set to 120 s.

Current time … Current time + 1 h

| 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 |

**Fig. 2 Representation of NN inputs.**

The time slot starts with the current time and ends 1 h later. Because the runway reassignment is considered within the sliding window only, the aircraft out of the sliding window do not affect the decision of the runway reassignment. In this example, a 1 h sliding window is assumed, and $4 \times 30$ inputs are made. This input size (1 h sliding window) is used for NN development.

As for the outputs, the objective function is the sum of the delay and the number of runway reassignments. NN calculates the delay only, and the number of runway reassignments is implicitly incorporated in the decision variable. This time, the following four output values are set as NN outputs:

$$o_1 = \sum_{j \in \{\mathcal{A} | r_j = a\}} (t_j - \mathrm{ELDT}_j) \tag{5}$$

$$o_2 = \sum_{j \in \{\mathcal{A} | r_j = c_A\}} (t_j - \mathrm{ELDT}_j) \tag{6}$$

$$o_3 = \sum_{j \in \{\mathcal{D} | r_j = c_D\}} (t_j - \mathrm{ETOT}_j) \tag{7}$$

$$o_4 = \sum_{j \in \{\mathcal{D} | r_j = d\}} (t_j - \mathrm{ETOT}_j) \tag{8}$$

Each output is affected by different inputs (aircraft queues). According to the authors' preliminary calculations, a single network with all necessary inputs and four outputs tends to cause overfitting. For example, $o_1$ is obviously affected by arrival on runway A only, but other unnecessary inputs are also connected when applying a single network. To avoid overfitting, more data are required, which is not preferable for real-world implementation. Therefore, four separate NNs are made in this paper. The sum of all four outputs corresponds to the delay of all aircraft, which matches the objective function given in Eq. (2) exactly, except for the number of runway reassignments. Each network requires only the necessary inputs that affect its 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 NNs (FFNNs) and convolutional NNs (CNNs). To estimate aircraft delay, the delays of nearby aircraft and the resulting queue length propagation should be considered. In the FFNN, all inputs are connected, but not all connections are actually needed for delay estimation. Unnecessary connections often cause overfitting and result in failed training of the network: an observation confirmed by trial and error. On the other hand, the CNN connects the neighborhood inputs only, and a smaller network can be created. A popular application of the CNN is image processing [26]. There are also many applications of CNN

**Table 1 Inputs and outputs used in each network**

| Network name | Inputs | Outputs |
|---|---|---|
| Network 1 | Arrival A (30 inputs) | $o_1$ |
| Network 2 | Arrival C (30 inputs) | $o_2$ |
| Network 3 | Arrival C, departure C ($30 \times 2$ inputs) | $o_3$ |
| Network 4 | Arrival C, departure D ($30 \times 2$ inputs) | $o_4$ |

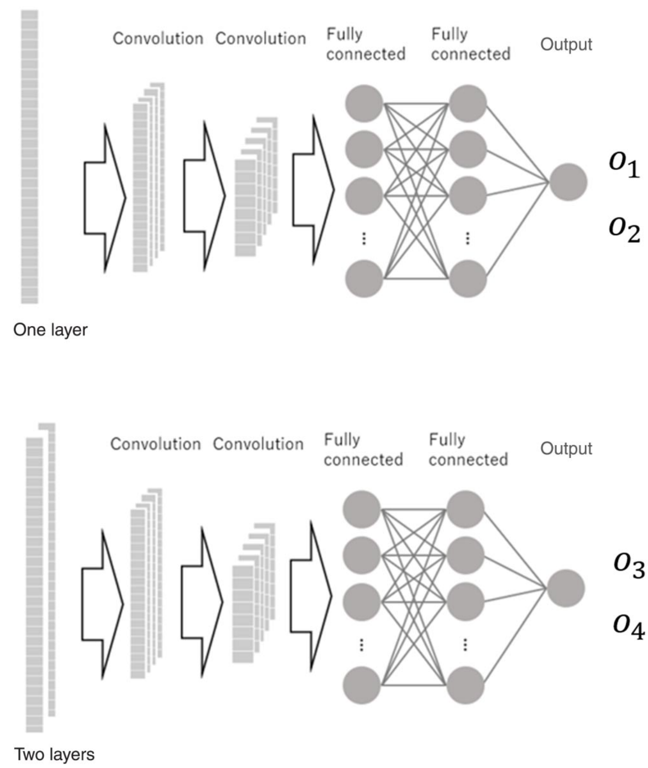


**Fig. 3 NN structures used in this research.**

classification for time series data [27], which are similar to the inputs in this research. Time series data are considered as one-dimensional data, and the CNN is applied.

Figure 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 parameters of the network [28]. Because network 3 and network 4 have more inputs than network 1 and network 2, the NN size is set larger in network 3 and network 4. However, the general NN structures are set the same for all networks. This structure is decided based on the trial and error. Although a better structure could exist, the proposed structure demonstrates a sufficient performance, as will be described in the following:

### C. Training Data and NN Training

Datasets are required to train the NN. In reality, actual past operational data can be used. However, to evaluate the proposed method, a simulation environment is required. Because it is difficult to develop a high-fidelity simulation model based on the actual operational data, in this paper, data are generated via simulations without explicit

**Table 2 The parameters in NN used in this research[a]**

| | | Values | |
|---|---|---|---|
| Layers | Variables | Networks 1 and 2 | Networks 3 and 4 |
| First (convolution) | Number of out channels | 128 | 256 |
| | Strides | 2 | 2 |
| | Kernel size | 2 | 2 |
| Second (convolution) | Number of out channels | 128 | 256 |
| | Strides | 3 | 3 |
| | Kernel size | 3 | 3 |
| Third (fully connected) | Number of hidden nodes | 128 | 256 |
| Fourth (fully connected) | Number of hidden nodes | 128 | 256 |
| First–fourth | Activation function | ReLU [28] | ReLU |
| Output | Activation function | Linear | Linear |

[a]ReLU denotes rectified linear unit.

modeling to confirm the effectiveness of the proposed method. The data should include the characteristics of the operational environment (i.e., cover various scenarios) and various patterns of runway balancing; otherwise, the data would be biased, and the appropriate NN could not be made. Therefore, through a data generation process, the NN is trained iteratively and its output is used for runway assignment of each arrival aircraft in the simulation. This NN is not intended to be used for the delay estimation explained in the previous subsection, but it is used for data generation only. Therefore, the NN used for delay estimation is trained with the generated data only. To simulate various situations, $\alpha$ is uniformly set between 0 and 900 s, and the runway assignment is randomly set with a probability of 0.05. In this way, the training data cover various input/output relationships, but we also need to confirm that the actual data reflect various relationships when actual data are used.

The simulation parameters will be discussed later in Sec. IV.A. The data are obtained every 1 min for 3 h, and so a single simulation can generate 180 datasets. By running simulations about 17,000 times, $3 \times 10^6$ (3 million) datasets are created.

Once the training data are obtained, the NN is trained. The well-known training algorithm, Adam [29], 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. The NN tries to minimize the loss function between model output and trained data output. Because both input and output data usually include noise, minimizing the error loses the generalization (called overfitting). There are various ways to avoid overfitting, but one method is to collect a sufficient number of datasets.

However, the possible number of datasets obtained is also limited if this process uses 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 min. For one month, $60 \times 16 \times 30 = 28{,}800$ is a reasonable number of inputs. However, more datasets are necessary, as about 30,000 datasets are not sufficient for most NN applications. Consequently, a data augmentation technique is used. Data augmentation is a technique to create data artificially based on existing data. This technique is often used in the image processing field [30]. In this field, the data augmentation is done in various ways such as rotating the image, reflecting the image, and changing the scaling.

In this research, data augmentation is done with the following process:

1) Copy the input/output from the original data.
2) Pick up one aircraft from the input data.
3) Move the aircraft to the next time slot.

This process is very easy, but it is plausible because the ETOT/ELDT includes uncertainty; so, similar output is expected even if the ETOT/ELDT is slightly changed. If each original dataset is augmented, in each original dataset, and the number of data becomes two times larger than the original number of datasets. Once the data are ready, the data are split into 70% training data and 30% validation data. The NN is trained with the training data, and the NN in which the loss function between the model output and the validation data output is minimum is used as the obtained NN for the stopping criterion. To investigate the impact of the original data volume, three cases are assumed in NN development, as shown in Table 3.

Case 1 uses only 30,000 original data, which are equivalent to one month of data. Forty-nine times the augmentation is done, and a total of 1.5 million data are used for NN training. Case 2 uses 300,000 original data, which are equivalent to one year of data. Case 3 uses 3 million original data, which cannot be obtained in the real world but are assumed as a benchmark. This augmentation technique can also be applied to the real-world data once the data are available.

Batch training is also applied (batch size is set to 2048). As for the loss function, the mean squared error (MSE) is used. The weight decay is also introduced to avoid the overfitting, where the decay parameter is set to 0.00005.

**Table 3 Several cases of data preparation**

| Parameters | Case 1 | Case 2 | Case 3 |
|---|---|---|---|
| Number of original data | 30,000 | 300,000 | 3,000,000 |
| Data augmentation | 49 times | 9 times | None |
| Total number of data including augmentation | 1,500,000 | 3,000,000 | 3,000,000 |

## IV. Simulation Environment

### A. Simulation Parameters for Evaluation

The performance of the developed NN model is evaluated via simulations. To conduct simulations, several simulation parameters need to be set. First, the traffic volume is determined as shown in Table 4. Because the maximum traffic is expected to be 40 aircraft/h for both departures and arrivals at the considered airport, this traffic volume is set as maximum, and the minimum is set to 75% of the maximum traffic. In each simulation, the hourly traffic of both departures and arrivals is randomly set based on a uniform distribution.

Without the optimization of runway reassignment, there is an optimal ratio of runway balance for both departures and arrivals. The nominal ratio should not be significantly different from the optimal ratio. Therefore, the nominal ratio of arrivals/departures is also set in the scenario. As for the departure, 0.35 is the best ratio of departure, and so it is randomly set between 0.25 and 0.45 in the simulation. As for the arrival, 0.7 is the best ratio of arrival for runway A, so it is randomly set between 0.6 and 0.8. These values are obtained via numerical simulations by assuming the conditions explained in Sec. II.A. This traffic volume is set for both the evaluation process and the data generation process for NN development. In evaluation, simulations were conducted 100 times, and the average is investigated.

In the simulation, first, the PTOT/PLDT are randomly distributed for 3 h; then, 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 min, and the runway decision is made 30 min before the ELDT. The time window for the optimization is set for the next 45 min here, and so the landing runway is optimized for arrival aircraft where the ELDT is between $t + 30$ and $t + 45$ min. The runway that is actually assigned is the one obtained in the latest optimization process.

Next, Table 5 shows the parameters of simulated annealing. As for the number of iterations, a maximum of 24 aircraft landing runways are optimized in the single optimization, which means that all combinations (224) cannot be calculated. Therefore, it is beneficial to use simulated annealing for optimization. The solution hardly changes, even if the number of iterations is set 10 times larger (2560); therefore, 256 iterations are used in this research. The authors also tried other heuristics (tabu search), but the simulated annealing worked better; so, the simulated annealing is chosen here. If the number of possible combinations is less than 256, the objective functions of all possible combinations are calculated and the best one is chosen, which translates into a deterministic method.

**Table 4 Traffic volumes in each calculation**

| Parameters | Values |
|---|---|
| Total arrivals per hour | 30–40 |
| Total departures per hour | 30–40 |
| Nominal ratio of arrival to runway A | 0.6–0.8 |
| Nominal ratio of departure from runway C | 0.25–0.45 |
| Total number of aircraft | 180–240 |

**Table 5 Parameters of simulated annealing**

| Parameters | Values |
|---|---|
| Number of iterations | 256 |
| Initial temperature | 15,000 |
| Terminal temperature | 7.5 |

The computational time is also a key factor in the real-world implementation. Compared to the simulation-based method, the NN output is much slower, which means that the number of iterations in the optimization is also limited. In the authors' environment (CPU with core i9-9900K), it takes about 15 ms to produce a single NN output; so, 256 iterations take about 4 s, which is sufficiently fast.

### B. Simulation-Based Method to Calculate the Objective Function

To proceed with the optimization process, the objective function must be calculated. In this research, the authors propose to calculate it by a 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 by assuming that

$$\mathrm{PLDT}_i = \mathrm{ELDT}_i \quad \forall\ i \in \mathcal{A},$$
$$\mathrm{PTOT}_i = \mathrm{ETOT}_i \quad \forall\ i \in \mathcal{D}$$

and $u = 0$ s for given conditions. This simulation method does not consider the uncertainty for runway balancing because the uncertainty is unknown when assigning the runway to each arrival aircraft. However, the simulation considers the uncertainty in the calculation of actual delay for the evaluation purposes. This discrepancy can cause inappropriate assignment of arrival runway for each aircraft. In the following, the case where the objective function is calculated by the simulation without considering uncertainty is denoted by the SIM method. The proposed NN-based method is denoted by the NN method.

### C. Simulation Flow

This subsection discusses the simulation flow of the current research. The baseline simulation is called the "evaluation simulation," as indicated in the following. After the initialization, calculations (updates) are conducted every 10 min. At each time step, the ELDT is updated randomly; but, the ETOT is updated only once when considering the actual data observation. The landing runway is optimized as explained before by either the NN method or the SIM method. The separation $S_{ij}$ is also updated when needed. First, the landing time is determined because landing aircraft are given priority. After that, the takeoff time is determined. This calculation is iterated until the takeoff/landing times of all aircraft are determined.

The following is the evaluation simulation flow:

Initialization $t \coloneqq t_0, S_{ij}, \mathrm{ETOT}_i, \mathrm{ELDT}_i$
While $t \le t_{\mathrm{end}}$,
  $\mathrm{ELDT}_i$ is updated,
  $\mathrm{ETOT}_i$ is updated only if $\Delta T < 15$ min,
  Optimize $\delta_i$ by NN method or SIM method.
  $S_{ij}$ is updated only if $r_i$ is changed after the last time step.
  Find earliest $t_i \forall\ i \in A(t_i \le t)$ ordered by $\mathrm{PLDT}_i$.
  Find earliest $t_i \forall\ i \in D(t_i \le t)$ ordered by $\mathrm{PTOT}_i$.
  $t = t + 10$ min.

The baseline simulation includes the optimization flow (optimize $\delta_i$), which is further explained here. Using the NN method, $\delta_i$ is optimized by simulated annealing. The objective function in the optimization is calculated by the NN, where the NN inputs come from the ETOT/ELDT in the evaluation simulation. As for the SIM method, $\delta_i$ is optimized by simulated annealing in the same way as the NN method. The objective function in the optimization is calculated as shown in the following. This calculation is performed independently from the evaluation simulation. First, all data are copied from the evaluation simulation. Second, $r_i$ is set based on decision variables. Third, the future PLDT/PTOT are set to the ELDT/ETOT because only the ELDT/ETOT data are available for optimization. The future $S_{ij}$ is also recalculated by assuming no randomness because no information about randomness is available. Finally, $t_i \forall\ i \in \mathcal{A}, \mathcal{D}$ are determined; and the objective function is calculated. The SIM method, in other words, calculates $t_i$ by assuming the PLDT/PTOT are set to the estimation and no randomness of separation is considered.

The following is the objective function calculation in the SIM method:

Initialization: copy all data from evaluation simulation.
Set $r_i$ based on the decision variables.
$\mathrm{PLDT}_i = \mathrm{ELDT}_i$ if $t_i = \varnothing \forall\ i \in \mathcal{A}$.
$\mathrm{PTOT}_i = \mathrm{ETOT}_i$ if $t_i = \varnothing \forall\ i \in \mathcal{D}$.
$S_{ij}$ is recalculated by assuming $u = 0$ and Eq. (1) is not applied if $t_i = \varnothing$ or $t_j = \varnothing$.
Find earliest $t_i \forall\ i \in \mathcal{A}$ ordered by $\mathrm{PLDT}_i$.
Find earliest $t_i \forall\ i \in \mathcal{D}$ ordered by $\mathrm{PTOT}_i$.
Calculate objective function using $t_i \forall\ i \in \mathcal{D}, \mathcal{A}$.

In the evaluation simulation, several parameters are initialized first. Therefore, the NN method and the SIM method can be compared if the same initial parameters are used in the evaluation simulation.

## V. Simulation Results

This section shows the simulation results using the proposed NN method and the SIM method. To evaluate the uncertainty environment, two conditions (i.e., deterministic environment and uncertainty environment) are considered. The uncertainty environment handles the uncertainty in the simulation in which the types of uncertainties were explained in the previous section. On the other hand, the deterministic environment does not consider uncertainties. The following conditions are considered: ETOT = PTOT, ELDT = PLDT, and $u = 0$. If the uncertainty is not considered, the SIM method can perfectly estimate the objective function. Therefore, the SIM method can be used as a benchmark to evaluate the proposed NN-based method. NNs are developed in each environment, and the result is evaluated independently.

### A. NN Estimation Performance

First, the NN prediction performance is evaluated and shown in Table 6. Thirty-thousand datasets are used for the evaluation, which are not used by either training or validation. Overall, the root-mean-squared error (RMSE) values improve as the number of training datasets increases (from case 1 to case 3). Under the deterministic environment, the SIM method becomes a perfect estimator, and so the RMSE is equal to zero. Under the uncertainty environment, the RMSE by the SIM method is the best for networks 1 and 2, and the RMSE is better than most NN cases for networks 3 and 4. If the performance of the runway balancing optimization depended on RMSE values only, the SIM method would work better under the uncertainty environment too. This is illustrated by the result shown in Table 6.

### B. Performance Evaluation of Individual Aircraft

To illustrate the obtained results, a single scenario is used and the delay of each aircraft is shown. Figure 4 shows the delay of each aircraft for two cases (no optimization, and optimized by the proposed NN method; $\alpha = 0$; case 3) under the uncertainty environment.

**Table 6 RMSE of each NN**

| Estimator | Network 1 | Network 2 | Network 3 | Network 4 |
|---|---|---|---|---|
| *Deterministic environment* | | | | |
| NN case 1 | 563 | 610 | 759 | 1016 |
| NN case 2 | 431 | 499 | 671 | 795 |
| NN case 3 | 429 | 490 | 595 | 739 |
| SIM | 0 | 0 | 0 | 0 |
| *Uncertainty environment* | | | | |
| NN case 1 | 859 | 831 | 1155 | 1822 |
| NN case 2 | 776 | 632 | 1053 | 1638 |
| NN case 3 | 759 | 605 | 919 | 1546 |
| SIM | 756 | 383 | 983 | 1670 |

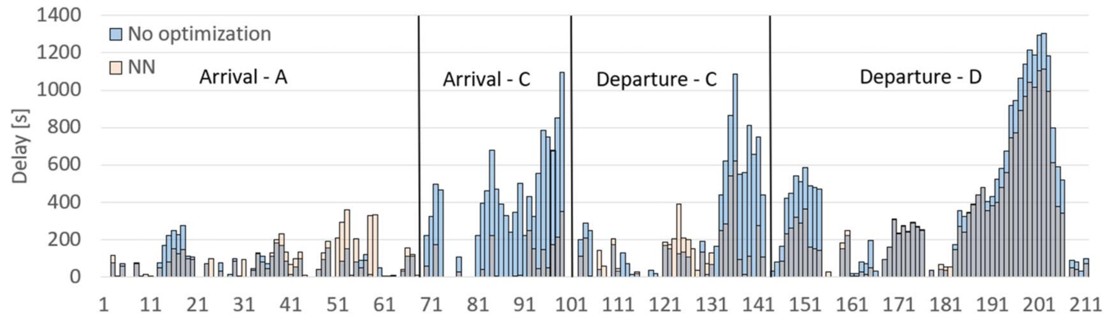


**Fig. 4 Delay of each aircraft for the sample scenario.**

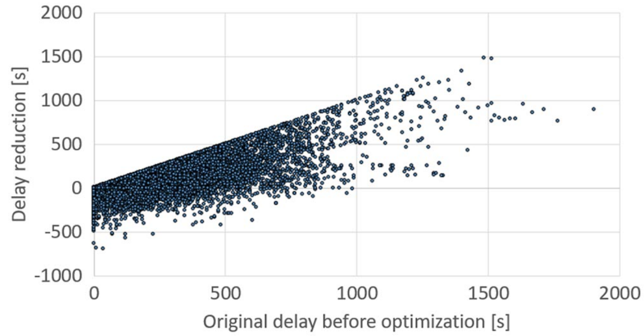


**Fig. 5 Original delay before optimization vs delay reduction.**

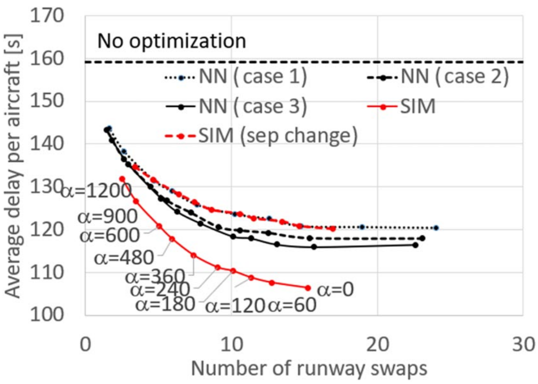


**Fig. 6 Average delay among NN and SIM methods under deterministic environment.**

The aircraft are ordered by the takeoff or landing time in each category. In this scenario, without optimization, the delay on the C arrival as well as the C and D departures is large, whereas the delay of the A arrival is small. Therefore, in general, the arrival traffic should be moved from runway C to runway A. In the NN optimization result, four aircraft are moved from runway A to runway C, 13 aircraft are moved from C to A, and the delay of 84 s per aircraft is reduced.

Next, the delay reduction of each aircraft is investigated. The proposed method minimizes the total delay, but some aircraft may experience larger delays, which is not preferred in the real world. A single case (optimized by the proposed NN method; $\alpha = 0$; case 3) is used, and 100 different scenarios are calculated. Figure 5 shows the delay reduction vs delay before the optimization of each aircraft. The negative delay reduction means the delay increase, which cannot be avoided in the runway balancing problem. However, in general, the delay increase is observed only when the original delay is relatively small. The maximum delay increase is 695 s, but the original delay of this aircraft is only 35 s. Here, only a sample result is shown, but a similar trend is observed for other cases too. The minimization of the total delay does not penalize any specific aircraft, and it tends to reduce the delay of each aircraft evenly.

### C. Evaluation Under Deterministic Environment

This section considers the deterministic environment first. Under the deterministic environment, the SIM method should outperform the proposed method because the SIM method is a perfect estimator. NN inputs are represented as shown in Fig. 2, but the detailed ETOT/ELDT information is lost; i.e., ELDT = 0 s and ELDT = 115 s are recognized as the same input. The key point is how the degraded NN method compares to the SIM method.

For both SIM and NN methods, the optimization is conducted in each parameter of α. The following 10 values of α are use:

$$[0, 60\ \mathrm{s}, 120\ \mathrm{s}, 180\ \mathrm{s}, 240\ \mathrm{s}, 360\ \mathrm{s}, 480\ \mathrm{s}, 600\ \mathrm{s}, 900\ \mathrm{s}, 1200\ \mathrm{s}]$$

As for the NN methods, the performance is expected to depend on the number of training data as well. To compare results, a total of 100 simulations using each method, but all under the same initial conditions, are conducted by setting completely the same initial conditions (traffic volume, and ELDT and ETOT of each aircraft in each scenario).

Figure 6 shows the average delay between NN and SIM methods. In the no-optimization case (all arrival aircraft land at original runway), a 159.1 s delay per aircraft is observed. Using the SIM method with $\alpha = 0$ s, this delay is reduced to 106.4 s, which corresponds to a 33.2% reduction. When α is set larger, the delay saving becomes small as the number of runway swaps becomes small. As expected, when using the NN method, the delay savings is not as much as the SIM method. However, the average delay becomes 115.9 s by the NN method's case 3, which achieves a 27.2% reduction as compared to the no-optimization case. As for NN cases 1 and 2, the result gets slightly worse than NN case 3, but no significant difference is observed.

The SIM method assumes a perfect simulation environment, which is not realistic in the real world. Therefore, we consider the simulation parameter estimation error, and the impact of the result is investigated. Here, the separation is assumed to be 5% larger; i.e., $S_{ij}$ is set 1.05 times larger. The result is also shown in Fig. 6 and denoted as "sep change." Because the separation includes a certain degree of prediction errors in the SIM method with the separation error, the performance gets worse. It is interesting that the result is similar to the NN method's case 1. Because the NN method uses the data only, the NN method does not affect the selection of simulation parameters. Although the performance degradation depends on the accuracy of the simulation parameters in the SIM method, the proposed NN method has the potential to outperform the SIM method even under a deterministic environment if the simulation parameter includes the error.

### D. Evaluation Under Uncertainty Environment

In this section, the same analysis is conducted under the uncertainty environment. The difference with the assumptions set in the previous section is the inclusion of uncertainty in the ELDT/ETOT and $S_{ij}$. Figure 7 shows the calculation result under the uncertainty environment. A 183.7 s average delay is observed for the nonoptimization case. This value is greater than that under the deterministic environment because of the uncertain separation. The SIM method can achieve the minimum delay of 144.9 s, and it can achieve a 38.8 s delay reduction via optimization by the SIM method. On the other

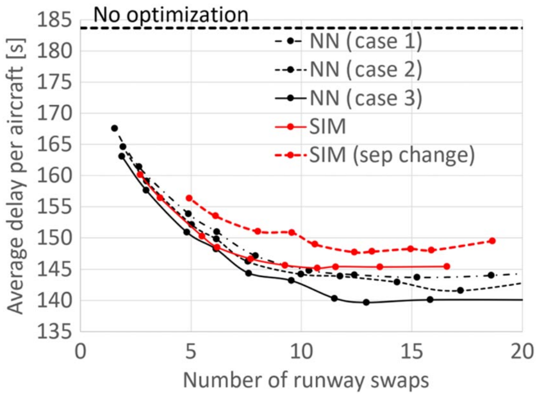


**Fig. 7 Average delay among NN and SIM methods under uncertainty environment.**

hand, as for the NN method's case 3, the minimum delay is 139.6 s, and a 44.1 s delay reduction is achieved, which is 13.7% greater than the SIM method. Case 3 uses 3 million original data, and cases 1 and 2 (with less original data) show the smaller delay reduction. However, even for case 1, the minimum delay is 143.7 s, which is slightly smaller than the SIM method. Even with 30,000 original datasets (equivalent to one month of data), the proposed method outperforms the SIM method.

Like the deterministic environment, the case where 5% longer separation (sep change) is assumed is also considered; and the result is shown in Fig. 7. The result of the SIM (sep change) becomes worse than the SIM method; the average delay per aircraft increases by about 5 s. The advantage of the proposed NN method is further boosted when the simulation environment includes the uncertainty of parameters.

One may argue that the difference of the results shown in Fig. 7 is not statistically significant, and so further investigation is performed. Because the average delay differs significantly among 100 scenarios, it is inappropriate to show the error bars in Fig. 7. Instead, the performance improvement of the NN method from the SIM method is discussed. Here, the following two methods are considered: the SIM method uses the setting of $\alpha = 0$, and the NN method uses the setting of $\alpha = 0$ and case 3. Figure 8 shows the delay of each method for each scenario. The delays of the two methods obviously show a similar trend because the best possible reduction of the delay depends on the scenario. The delay of the NN method is, in general, smaller than that of the SIM method, but this changes sometimes. This is because an "accurate" estimation sometimes works worse due to the uncertainty trend of each scenario. Therefore, the delay reduction of the NN method relative to the SIM method is calculated; the average is 5.37, and the SD is 10.32. The 95% confidence interval of the average delay reduction is calculated by the following equation:

$$\text{average} \pm \frac{1.96\text{SD}}{\sqrt{n}} \tag{9}$$

Note that

$$\frac{1.96\text{SD}}{\sqrt{n}} = 2.10 \text{ s}$$

So, the 95% confidence interval is calculated as [3.27, 7.47], which means the average delay reduction of the NN method is statistically significant. The confidence interval is calculated with any combination of the delay, but the value does not change significantly. Therefore, it is concluded that about a 2 s delay difference in Fig. 7 is statistically significant.

In Sec. V.B, the RMSE of the SIM method is, in general, better than that of the NN methods, but the optimization result by the NN methods tends to be better; so, this reason is considered here. Figure 9 shows the estimation result of four outputs (runway A arrival, runway C arrival, runway C departure, and runway D departure) for three NN cases and the SIM method under the deterministic environment. In this evaluation, about 30,000 data samples (which are used for neither training nor for validation) are used. The bar charts indicate the observed data ratios in each estimated delay range. In general, the estimation accuracy deteriorates as the estimated delay increases, which is easily understandable. Also, among the three NN cases, case 3 shows the best performance and case 1 shows the worst. This is also reasonable because of the original data volume used for NN training.

As for arrival runway A, all NN cases and the SIM method show similar results; so, no difference is found by using either of the estimators. Although the NN potentially improves the estimation accuracy under uncertainty, the NN input disregards the detailed information of the traffic, which worsens the estimation accuracy. Both effects seem to be similar for runway A, and the NN cases and the SIM method show similar accuracy. On the other hand, for arrival runway C, the SIM method estimates the flight delay most accurately. Compared to runway A, the landing separation is large (240 s) but the uncertainty (SD = 15 s) is the same, and the estimation accuracy is improved under uncertainty. The NN input information loss is the same; so, in total, the NN estimates the delay less accurately than the SIM method. Because arrival aircraft are always given priority, as mentioned before, the departure traffic does not affect the arrival traffic at all. In addition, the accuracy of the ELDT is better than that of the ETOT, and so it is difficult for the NN method to outperform the SIM method.

Departure traffic is affected by arrival traffic as well, and the accuracy of the ETOT is worse than that of the ELDT; so, the departure traffic is highly affected by the uncertainty. As for departure runway C, the NN estimation outperforms the SIM method when the estimated total delay is less than 40 min. However, the SIM method estimates better when the estimated total delay is more than 40 min.

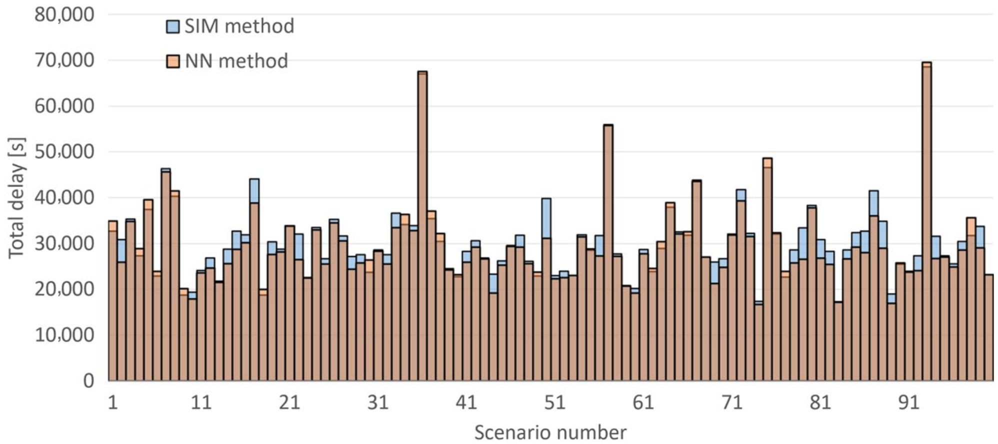


**Fig. 8 Total delay with SIM method and NN method in each scenario.**

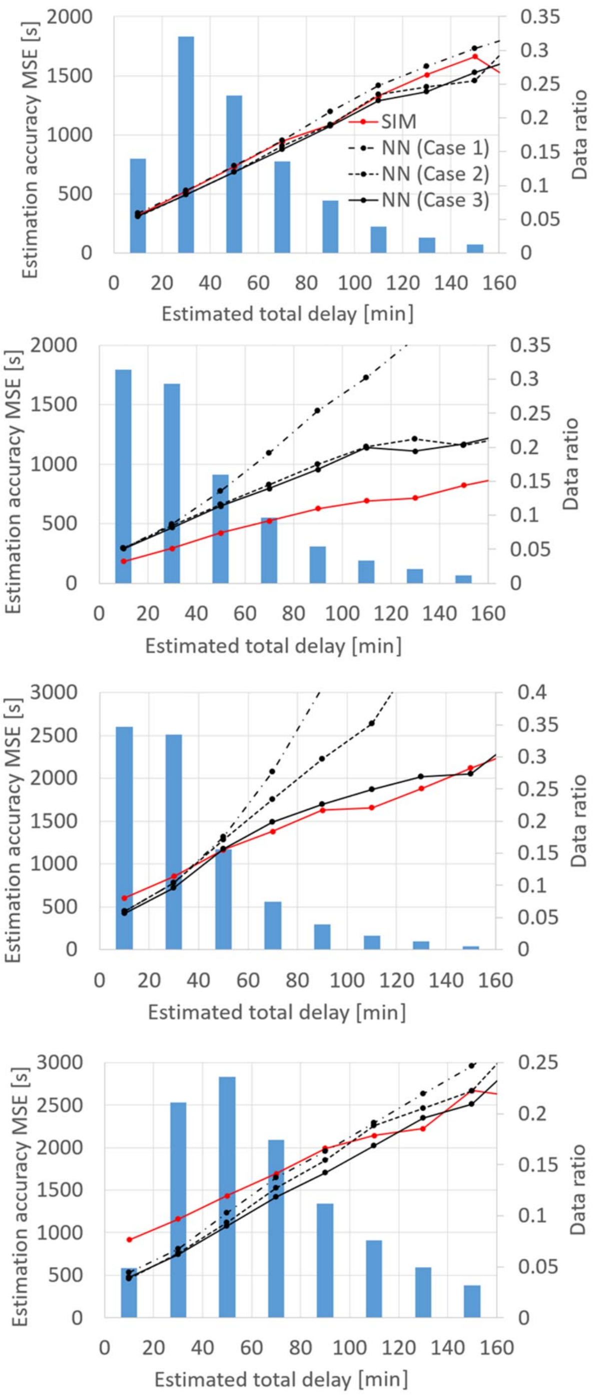


**Fig. 9 Delay accuracy by NN methods and SIM method under uncertainty: runway A arrival, runway C arrival, runway C departure, and runway D departure (from the top).**

This is due to the number of available data. About 70% of the training data fall into the estimated delay being less than 40 min, and so the estimation accuracy is good within this range. However, as the estimated delay increases, the data available decrease, and the relative accuracy worsens. As for departure runway D, the NN estimation is much better than the SIM method when the estimated total delay is less than 120 min, where many training data are available.

According to this data analysis, the NN method of estimation is not better than the SIM method for arrival traffic because of the smaller arrival traffic uncertainty. As for departure traffic, the NN method estimation excels over the SIM method when sufficient data are available for training. On the other hand, NN case 1 still outperforms the SIM method for runway balancing optimization according to Fig. 7, even though the delay estimation accuracy of the NN method is sometimes worse than that of the SIM method. The possible reason is that the NN estimation is better when sufficient training data are available. This training data are obtained based on the daily simulation, and so the optimization process encounters such data very frequently. Even if the NN estimation accuracy is bad when the estimated delay is large, the optimization process does not encounter such data so often, and so it makes a relatively small impact on the optimization.

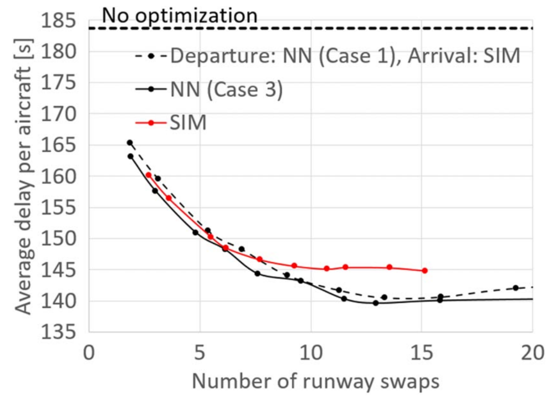


**Fig. 10 Average delay under uncertainty environment when estimation of departures is given by NN and estimation of arrivals is given by SIM method.**

To combine the advantages of both methods, the departure traffic delay should be estimated by the proposed NN method, and the delay estimation of the arrival traffic is given by the SIM method, which may maximize the benefit. Figure 10 shows the calculation results when the delay of the departure traffic is estimated by the proposed NN method (case 1) and that of the arrival traffic is estimated by the SIM method. Although case 1 uses only 30,000 data (one month of data), the optimization result becomes close to case 3. This will be the best use of the proposed NN method when considering the optimization performance and data availability.

### E. Summary of Results and Advantages of the Proposed Method

In summary, the proposed method can reduce the delay, especially under the uncertainty environment. Using 300,000 datasets (corresponding to one yearly dataset), the proposed method outperforms the SIM method. When the SIM method is combined with the proposed method, the proposed method outperforms the SIM method with 30,000 datasets only (corresponding to one monthly dataset).

In addition, the proposed method does not require a simulation model to optimize the runway balancing problem, and so the optimization can be simplified. The SIM method uses a simulation environment, and it is often difficult to develop an accurate simulation model. When the SIM method is assumed to include the uncertainty of separation, the benefit of the proposed method is further boosted.

This paper assumes the runway operation at Tokyo International Airport only, but the key point is to develop delay prediction NN models on each runway. Therefore, the proposed method will be applicable as long as proper inputs are selected for NN development.

The proposed method is expected to be used for runway balancing at the terminal airspace of the considered airport. All information needed for optimization can be obtained at the current en route and terminal ATC system in Japan. Only the minimum one-month operational data are needed, and so even if the operational environment is changed, the rapid implementation is possible to introduce the runway balancing optimization scheme to the changed environment. Also, the operational environment may change with time, but a continuous update is also possible by the proposed method. The proposed work will be effective to optimize the runway balancing in real time.

## VI. Conclusions

This paper proposed a new scheme to optimize a runway balancing problem using a NN. The NN modeling approach does not require any explicit operational models and simulations in theory, and it relies on actual operational data only. This means that simulation parameters such as departure/arrival separations and their interactive effect are not required because these characteristics are expected to be modeled by the NN. This paper showed the effectiveness of the proposed method via simulations. The uncertainty effect was modeled appropriately by the NN, and the solution of runway balancing was improved. However, because less training data are used, less delay reduction was observed. According to the NN estimation performance, departure delay was relatively better estimated by the NN than arrival delay. When only the departure delay was estimated by the NN and the arrival delay was estimated by the SIM method, better performance was expected with less training data. This paper suggests that the proposed NN approach could outperform the simulation-based method by estimating the objective function in the runway balancing optimization process, and it shows a bigger impact under a larger uncertainty environment.

## References

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

[2] "RECAT-EU, European Wake Turbulence Categorisation and Separation Minima on Approach and Departure," Ed. 1.2, EUROCONTROL, Brussels, 2018, https://www.eurocontrol.int/publication/european-wake-turbulence-categorisation-and-separation-minima-approach-and-departure.

[3] Cheng, V. H. L., Crawford, L. S., and Menon, P. K., "Air Traffic Control Using Genetic Search Techniques," *Proceedings of the 1999 IEEE International Conference on Control Applications*, Inst. of Electrical and Electronics Engineers, New York, 1999. https://doi.org/10.1109/CCA.1999.806209

[4] Salehipour, A., Modarres, M., and Naeni, L. M., "An Efficient Hybrid Meta-Heuristic for Aircraft Landing Problem," *Computers and Operations Research*, Vol. 40, No. 1, 2013, pp. 207–213. https://doi.org/10.1016/j.cor.2012.06.004

[5] Xie, J., Zhou, Y., and Zheng, H., "A Hybrid Metaheuristic for Multiple Runways Aircraft Landing Problem Based on Bat Algorithm," *Journal of Applied Mathematics*, Vol. 2013, Jan. 2013, Paper 742653. https://doi.org/10.1155/2013/742653

[6] Ng, K. K. H., Lee, C. K. M., Chan, F. T., and 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*, Vol. 106, Oct. 2017, pp. 115–136. https://doi.org/10.1016/j.tre.2017.08.006

[7] Hu, X.-B., and Di Paolo, E., "An Efficient Genetic Algorithm with Uniform Crossover for Air Traffic Control," *Computers and Operations Research*, Vol. 36, No. 1, 2009, pp. 245–259. https://doi.org/10.1016/j.cor.2007.09.005

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

[9] Bencheikh, G., Boukachour, J., and Alaoui, A. E. H., "Improved Ant Colony Algorithm to Solve the Aircraft Landing Problem," *International Journal of Computer Theory and Engineering*, Vol. 3, No. 2, 2011, pp. 224–233. https://doi.org/10.7763/IJCTE.2011.V3.309

[10] Liang, M., Delahaye, D., and Maréchal, P., "Integrated Sequencing and Merging Aircraft to Parallel Runways with Automated Conflict Resolution and Advanced Avionics Capabilities," *Transportation Research Part C: Emerging Technologies*, Vol. 85, Dec. 2017, pp. 268–291. https://doi.org/10.1016/j.trc.2017.09.012

[11] Roling, P. C., Delsen, J., and Curran, R., "Flexible Arrival & Departure Runway Allocation Using Mixed-Integer Linear Programming," *17th AIAA Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2017-4257, 2017. https://doi.org/10.2514/6.2017-4257

[12] Ma, J., Delahaye, D., Sbihi, M., Scala, P., and Mota, M. M., "Integrated Optimization of Terminal Maneuvering Area and Airport at the Macroscopic Level," *Transportation Research Part C: Emerging Technologies*, Vol. 98, Jan. 2019, pp. 338–357. https://doi.org/10.1016/j.trc.2018.12.006

[13] Malaek, S. M. B., and Naderi, E., "A New Scheduling Strategy For Aircraft Landings Under Dynamic Position Shifting," *2008 IEEE Aerospace Conference*, Inst. of Electrical and Electronics Engineers, New York, 2008. https://doi.org/10.1109/AERO.2008.4526378

[14] Mori, R., "Evaluation of Departure Pushback Time Assignment Considering Uncertainty Using Real Operational Data," *9th SESAR Innovation Days*, 2019.

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

[16] Solak, S., Solveling, G., Clarke, J. P. B., and Johnson, E. L., "Stochastic Runway Scheduling," *Transportation Science*, Vol. 52, No. 4, 2018, pp. 917–940. https://doi.org/10.1287/trsc.2017.0784

[17] Liu, M., Liang, B., Zheng, F., Chu, C., and Chu, F., "A Two-Stage Stochastic Programming Approach for Aircraft Landing Problem," *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, 2018. https://doi.org/10.1109/ICSSSM.2018.8465107

[18] Liu, M., Liang, B., Zhu, M., and Chu, C., "Stochastic Runway Scheduling Problem with Partial Distribution Information of Random Parameters," *IEEE Access*, Vol. 8, March 2020, pp. 68,460–68,473. https://doi.org/10.1109/ACCESS.2020.2984513

[19] Kleywegt, A. J., Shapiro, A., and Homem-de-Mello, T., "The Sample Average Approximation Method for Stochastic Discrete Optimization," *SIAM Journal on Optimization*, Vol. 12, No. 2, 2002, pp. 479–502. https://doi.org/10.1137/S1052623499363220

[20] Khassiba, A., Bastin, F., Gendron, B., Cafieri, S., and Mongeau, M., "Extended Aircraft Arrival Management under Uncertainty: A Computational Study," *Journal of Air Transportation*, Vol. 27, No. 3, 2019, pp. 131–143. https://doi.org/10.2514/1.D0135

[21] Mori, R., and Delahaye, D., "Simulation-Free Runway Balancing Optimization Under Uncertainty Using Neural Network," *9th SESAR Innovation Days*, 2019.

[22] Mori, R., "Development of Fast-Time Stochastic Airport Ground and Runway Simulation Model and Its Traffic Analysis," *Mathematical Problems in Engineering*, Vol. 2015, Jan. 2015, Paper 919736. https://doi.org/10.1155/2015/919736

[23] Mori, R., "Refined Collision Risk Model for Oceanic Flight Under Longitudinal Distance-Based Separation with ADS-C Environment," *Journal of Navigation,* Vol. 67, No. 5, 2014, pp. 845–868. https://doi.org/10.1017/S0373463314000162

[24] Mori, R., "Development of a Pushback Time Assignment Algorithm Considering Uncertainty," *Journal of Air Transportation,* Vol. 25, No. 2, 2017, pp. 51–60. https://doi.org/10.2514/1.D0069

[25] Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P., "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, 1983, pp. 671–680. https://doi.org/10.1126/science.220.4598.671

[26] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, MIT Press, 2012, pp. 1097–1105.

[27] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., and Muller, P. A., "Deep Learning for Time Series Classification: A Review," *Data Mining and Knowledge Discovery*, Vol. 33, No. 4, 2019, pp. 917–963. https://doi.org/10.1007/s10618-019-00619-1

[28] Nair, V., and Hinton, G. E., "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, Inst. of Electrical and Electronics Engineers, New York, 2010, pp. 807–814.

[29] Kingma, D. P., and Ba, J. L., "Adam: A Method for Stochastic Optimization," *International Conference on Learning Representations*, 2015, pp. 1–13.

[30] Shorten, C., and Khoshgoftaar, T. M., "A Survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, Vol. 6, No. 1, 2019, Paper 60. https://doi.org/10.1186/s40537-019-0197-0

J. Kuchar
*Associate Editor*