



# Deep Reinforcement Learning for Pedestrian Guidance

Hitoshi Shimizu<sup>1,2(✉)</sup>, Takanori Hara<sup>2</sup>, and Tomoharu Iwata<sup>1</sup>

<sup>1</sup> NTT Communication Science Laboratories, Kyoto, Japan  
{hitoshi.shimizu.kg,tomoharu.iwata.gy}@hco.ntt.co.jp

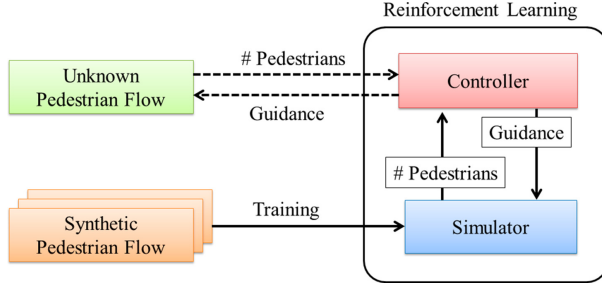
<sup>2</sup> Nara Institute of Science and Technology, Nara, Japan  
hara.takanori.hm8@is.naist.jp

**Abstract.** In large-scale events where many people gather, providing them with appropriate, efficient, and safe guidance about where to proceed is critical to ease congestion. We can evaluate guidance candidates using a pedestrian flow simulator to find appropriate guidance. However, evaluating many candidates by simulation requires high computational cost, which prohibits real-time guidance. We propose a method that finds appropriate guidance in real-time for observed situations based on deep reinforcement learning. Our proposed method learns a function that outputs appropriate guidance given the observed situation to minimize the average travel time of pedestrians. The difficulty here is that the real-world measurements of pedestrian travel time are limited due to privacy issues since it tracks individuals. Though our method uses only the observation obtained without locating specific individuals: the number of pedestrians who are moving on roads, it is guaranteed by Little's law to be equivalent to minimizing the average travel time. Our experimental results for unknown pedestrian flow show that our proposed method outperforms rule-based controls, and its guidance is as effective as one selected from many candidates by repeated simulations with massive computational cost.

**Keywords:** Crowd simulation · Reinforcement learning · Pedestrian guidance

## 1 Introduction

At large-scale events where thousands of people gather, appropriate, safe, and efficient guidance must be provided to ease congestion. To find appropriate guidance, we can evaluate guidance candidates on a pedestrian flow simulator. Yamashita et al. [9] developed a technique that simulated all candidates exhaustively. To search for better guidance with fewer simulations, Otsuka et al. [5] proposed to use Bayesian optimization (BO), and Shigenaka et al. [6] proposed to use Covariance Matrix Adaptation Evolution Strategy (CMA-ES). Although both BO and CMA-ES methods require fewer simulations than an exhaustive



**Fig. 1.** Our proposed scheme achieves pedestrian flow control using deep reinforcement learning and simulator. Using the observed number of pedestrians on the roads as a reward and observations, the Controller learns with various kinds of simulated pedestrian flow data. After training, it can output appropriate guidance for unknown pedestrian flows.

search, many evaluations with simulators are unavoidable and prohibit real-time guidance for unknown pedestrian flow.

Therefore we proposed a new scheme shown in Fig. 1. Our method uses a crowd simulation and reinforcement learning [7], which maximizes the *reward* obtained by selecting the *action* based on the *state* observed by the agent. By learning with various kinds of simulated pedestrian flow data (shown as *Synthetic Pedestrian Flow* in the Fig. 1), our proposed method outputs guidance for unknown pedestrian flows (shown as *Unknown Pedestrian Flow* in the Fig. 1). We experimentally demonstrate the effectiveness of our proposed method using a pedestrian flow simulator and consider an example problem that identifies which roads to block and encourages detours when the number of pedestrians on each road is observed as input.

We evaluate the guidance by the average travel time of pedestrians, where shorter average travel time is better guidance. However, since pedestrian travel times must track individuals, such measurements are often not provided due to privacy concerns. Aggregated data are more readily available because it does not locate specific individuals. As shown in the Fig. 1 as # Pedestrians between *Controller* and *Simulator*, our method uses the observed number of pedestrians on the roads as a reward and a state, which is one type of aggregated data. Minimizing the number of pedestrians is guaranteed by Little’s law to be equivalent to minimizing the average travel times.

Kato et al. [2] proposed a method to guide pedestrians from the fireworks event venue to the station. Their method also uses a crowd simulation and reinforcement learning. However, their proposed method depends on the road network, which makes it difficult to adjust the parameters. Because the reward of our proposed scheme is normalized, it has the advantage of being independent of the road network.

Our contributions are the followings: (1) To handle such congestion situations in real-time, we propose a method that learns a function with a deep RL that

outputs appropriate guidance based on observations. (2) The proposed reward based on the number of pedestrians has no privacy issues, and is guaranteed to be equivalent to the average travel time by Little’s law. (3) Experiment results show that its performance exceeds a rule-based guidance policy and comes close to one selected from many candidates by repeated simulations.

## 2 Problem Settings

We consider a situation where many people start walking at different times from different beginning points to different end points by roads. The controller agent selects a guidance (action) from a set of actions at each time step. The task is to find the sequence of guidance that minimizes the average travel times of people  $\frac{1}{I} \sum_{i=1}^I \tau_i$ , where  $\tau_i$  is the travel time of pedestrian  $i$  and  $I$  is the number of pedestrians. The definitions of each symbol in the paper are summarized in Table 1.

## 3 Proposed Method

The total travel time of pedestrians is equivalent to the time integral of the number of them moving at each time. This relationship, which is called Little’s law [3], is shown in Fig. 2. Gray area  $S$  enclosed by the red line that indicates the cumulative number of departures and the blue line that indicates the cumulative number of arrivals at each time can be expressed by two types of expressions:  $S = \sum_{i=1}^I \tau_i = \int_{t=0}^T N_t dt \approx \sum_{t=1}^T N_t \Delta$ , where  $N_t$  is the number of moving pedestrians at time  $t$  and  $\Delta$  is the interval between adjacent time steps.  $\sum_{t=1}^T N_t \Delta$  is the summation for the time direction, and  $\sum_{i=1}^I \tau_i$  is the summation for each pedestrian. Approximation is acceptable when  $\Delta$  is small enough for fluctuation in  $N_t$ . Therefore, average travel time  $\frac{1}{I} \sum_{i=1}^I \tau_i = \frac{S}{I}$  can be minimized by taking actions that minimize the total number of pedestrians traveling at each time  $\sum_{t=1}^T N_t = \frac{S}{\Delta}$  because  $I$  and  $\Delta$  are constants.

Little’s law holds even for a single pedestrian. The tasks of minimizing the time for a moving object to reach its goal have frequently been addressed in the history of reinforcement learning [7]. The Little’s law discussed here clarifies that a small negative reward to each step usually leads to the shortest travel time<sup>1</sup>. Our proposed method will be useful for tasks where a moving object must reach its goal in the shortest time.

In addition, if the absolute values of the rewards widely vary, adjusting the other RL parameters is difficult. Therefore, the rewards must be normalized, for example, into a range of  $-1$  to  $1$  (see Footnote 1). It is very difficult to assess how effective the currently selected strategy is without any evaluation criteria. Therefore, we propose a method to evaluate the relative effectiveness of the currently selected strategy by comparing it with the strategy that does

<sup>1</sup> <https://github.com/Unity-Technologies/ml-agents/blob/master/docs/Learning-Environment-Best-Practices.md>.

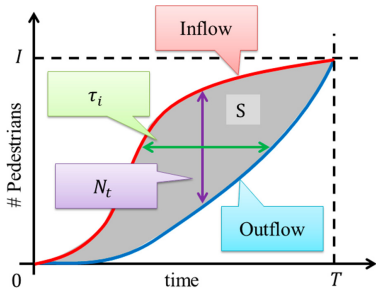
not do anything (no strategy). Thus we propose the reward EDGE/OPEN shown in Table 4. This reward satisfies  $-1 \leq r_t \leq 1$ , and  $r_t = 1$  when  $N_t = 0$ , and it satisfies  $r_t = 0$  when  $N_t = N_t^o$  if  $N_t^o > 0$ .

In the case that the number of pedestrians is observed for the reward, using the observation as the state is more convenient and efficient. To measure the number of pedestrians, just measuring their total does not identify where the congestion is occurring. Also, observing the number of people only at one time step does not tell whether their number is increasing or decreasing. For example, we can use the number of pedestrians on each road of multiple time steps as the state.

### 4 Experiments

We evaluated our proposed method on a task as an example that finds guidance to ease congestion around the entrance at the start of a big event. We used an in-house crowd simulator [5], where pedestrians move on the road network. Figure 3 shows the road network around Japan National Stadium in Tokyo, which is the stage of the simulation. Pedestrians start to walk from six stations to the stadium’s six gates, and are crowded on the roads in front of the gates. Pedestrians pass through 317 roads. For a state, we used the number of pedestrians on these roads for the past four steps, which give a 1268-dimensional vector.

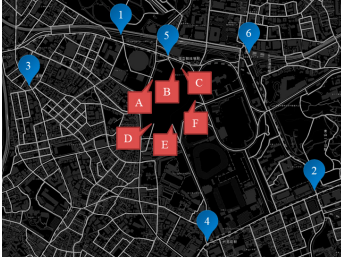
The number of pedestrians in one scenario ranged from 10,000 to 90,000 in 10,000 increments. In each scenario, the proportion of stations where pedestrians appear was varied using random numbers from a Dirichlet distribution. The expected value was set as the ratio of Table 2 by referring to the actual number



**Fig. 2.** Little’s law: red line represents cumulative number of departures, and blue line represents cumulative number of arrivals. Red and blue lines eventually meet at  $(T, I)$ , where let  $T$  be the time when the last person arrives.  $S$  is the gray area surrounded by red and blue lines. (Color figure online)

**Table 1.** Notation

Symbol	Description
$I$	number of pedestrians in system: $i \in \{1, \dots, I\}$
$J$	number of roads: $j \in \{1, \dots, J\}$
$T$	number of time steps: $t \in \{1, \dots, T\}$
$\Delta$	intervals between adjacent time steps
$N_t$	number of moving pedestrians at time $t$
$N_t^o$	the total number of pedestrians on the roads with the strategy that does not do anything
$v_t^i$	velocity of pedestrian $i$ at time $t$
$\rho_t^i$	density of a road in front of pedestrian $i$ at time $t$
$\rho_t^j$	averaged density of a road $j$ at time $t$
$x_t^j$	number of pedestrians on road $j$ at time $t$
$\tau_i$	travel time of pedestrian $i$



**Fig. 3.** Ratio of pedestrians emerging from each station.

**Table 2.** Ratio of pedestrians emerging from each station.

Station ID	Usage ratio of pedestrians
1	29%
2	11%
3	6%
4	11%
5	20%
6	22%

**Table 3.** Maximum number of people who pass each second at each gate.

Gate ID	Throughput [person/sec]
A	3
B	8
C	3
D	3
E	5
F	3

of station users. The timing of the pedestrians emerging from the station was distributed, so that they peaked 30 min after the start of simulation. At its entrance, assuming that the number of security staff varies depending on the gate, the maximum number of people who pass through it per second were set (Table 3).

We consider a guidance that temporarily closes the gate to avoid congestion at it. When a gate is closed, we assumed that pedestrians head to the nearest open gate. Since there are six gates, there are  $2^6 = 64$  open and closed combinations. However, we added a constraint that no more than two adjacent gates can be closed simultaneously to avoid long detours. Then we have 39 guidance candidates. Guidance lasts at least ten minutes, and a different guidance can be selected every ten minutes. The simulation time is set to 250 min to allow all pedestrians to enter the stadium regardless of which guidance to choose. Guidances are selected 25 times per episode. In the proposed method, a strategy of doing nothing (no strategy) corresponds to open all the gates always.

We compared the proposed method with OPEN as the baseline, where all gates are always open and no guidance is applied. We also prepared a rule-based guidance shown as RULE, where all gates are open if the population densities (number of people/road area) of all roads in front of the gates are less than a threshold, and the gate with the highest density road is closed if there is a road above the threshold. The threshold was set to 1.0 person/square meter.

GREEDY shows the guidance obtained by repeated simulations for comparison. With 25 time steps and 39 actions, there are  $39^{25} \sim 10^{40}$  guidance combinations. Since the computation time to execute every simulation combination is too long, GREEDY starts from OPEN and tries all the actions at each time step, and then adopts the best action sequentially in chronological order. FIX randomly selects the guidance policy obtained by GREEDY for test scenarios, regardless of the actual scenario. We also prepared the comparing methods with various rewards shown in Table 4, referring to the study of RL in traffic signal control. Note that there is privacy issues if its expression contains  $\tau_i$ .

As a learning model, we used a state-of-the-art RL method called Advantage Actor-Critic (A2C) [4, 8], which learns based on the experiences gained after

every episode is completed. The value function ( $V(x)$ ) and the action-value function ( $Q(x, a)$ ) were approximated by a common neural network with two hidden layers, each of which has 100 units. We used the ReLU function [1] to make each layer output nonlinear, and actions were sampled by softmax function of Q-value during training.

## 5 Results

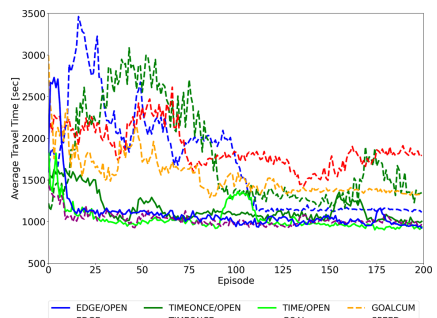
Figure 4 shows the average travel time for each episode when training with the rewards in Table 4. We used 16 training scenarios, which consist of eight different amounts of pedestrians ranging from 10,000 to 80,000, each with two different station use ratios. We performed 200 episodes  $\times$  16 simulations scenarios for training: 3,200 times for each deep RL. Within 200 episodes, the average travel time of EDGE/OPEN, SPEED, TIME/OPEN, and TIMEONCE/OPEN converge stably to smaller values than others.

We created 90 test scenarios, consisting of nine groups whose number of pedestrians ranged from 10,000 to 90,000 in 10,000 increments, which is not included in the training data. Table 5 shows the result of applying the guidances to the test scenarios. Figure 5 shows the breakdown of the average travel time by the number of pedestrians. Both Table 5 and Fig. 5 are evaluated as a ratio of OPEN. Although the average travel time of FIX resembled that of RULE, its effect was less effective than GREEDY. Note that the GREEDY and FIX methods need iterative evaluations ( $39 \times 25 = 975$  times of simulations) for the target scenario. These results required about 25 min to execute 39 parallel simulations 25 times.

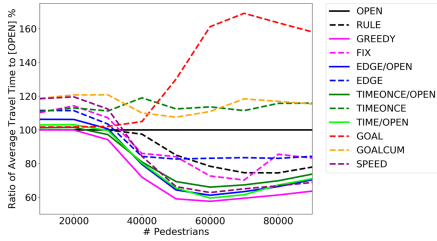
Although TIME/OPEN was the best RL results in Table 5, it is problematic due to privacy issues. SPEED also gives good results when  $I$  is large; its performance

**Table 4.** (left) Rewards for deep RL. Rewards with /OPEN use the result of OPEN for normalizing.

Name	Reward
EDGE/OPEN (proposed method)	$\max\left(-1, \frac{N_t^o - N_t}{N_t^o}\right)$ if $N_t^o > 0$ 0 if $N_t^o = 0$ and $N_t = 0$ -1 if $N_t^o = 0$ and $N_t > 0$
EDGE	$(I - N_t)/I$
TIMEONCE/OPEN	$(\sum_i \tau_i^o - \tau_i) / \sum_i \tau_i^o$ if $t = T$ 0 if $t \neq T$
TIMEONCE	$-\sum_i \tau_i / TI$ if $t = T$ 0 if $t \neq T$
TIME/OPEN	$\sum_i \frac{\tau_i^o - \tau_i}{\tau_i^o} \mathbb{1}((t-1)\Delta < \tau_i \leq t\Delta)$
GOAL	$\frac{1}{T} \sum_i^T \mathbb{1}((t-1)\Delta < \tau_i \leq t\Delta)$
GOALCUM	$\frac{1}{T} \sum_t^T \mathbb{1}(\tau_i \leq t\Delta)$
SPEED	$(\bar{v}^{\max} - v_t) / \bar{v}^{\max}$ , where $\bar{v}^{\max} = \frac{1}{T} \sum_i^T v_i^{\max}$ and $v_t = \frac{1}{N_t} \sum_j x_t^j \times v(\rho_j)$



**Fig. 4.** (right) Evaluation values in episodes during training of reinforcement learning. Horizontal axis is number of episodes. Vertical axis is average travel time.



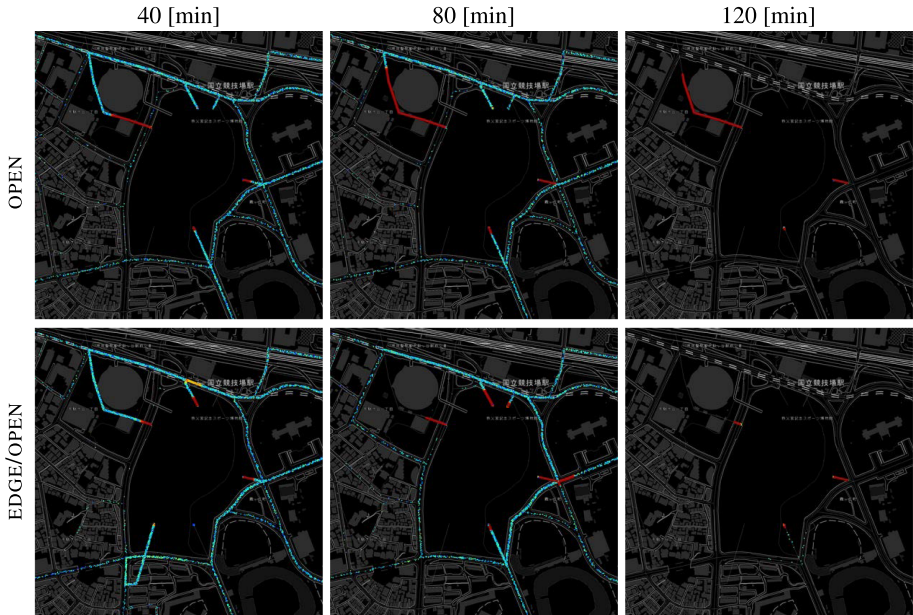
**Fig. 5.** (left) Evaluation of each method against test data. Horizontal axis is number of pedestrians. Vertical axis is ratio of average travel time to OPEN. Each point is the average of the results of 10 test data.

**Table 5.** (right) Average ratio of travel time to OPEN for each method for 90 scenarios. Ref. represents reference methods for comparison. OPEN took 1493.2 [s] on average. Bold indicates results that are not significantly different from best result (TIME/OPEN) except for GREEDY in paired t-test ( $p < 0.05$ ).

	Method	Ratio to OPEN %
Ref.	RULE	87.5
	GREEDY	74.1
	FIX	90.4
RL: reward	EDGE/OPEN	<b>79.8</b>
	EDGE	91.9
	TIMEONCE/OPEN	80.8
	TIMEONCE	113.7
	TIME/OPEN	<b>79.0</b>
	GOAL	132.5
	GOALCUM	115.5
	SPEED	85.0

is poor when  $I$  is small (Fig. 5). This method increases the moving speed by increasing users of the detours, which may cause extra travel time. Therefore, our proposed EDGE/OPEN yields the best result as the RL reward. The time required for the method to make a decision was about 5 ms each time, which was much smaller than GREEDY (25 min), and satisfies the demand for real-time use.

In Figs. 4 and 5, we can compare the solid line (with /OPEN) and dashed lines (without /OPEN) of the same color. These results show that normalization with /OPEN is effective. Figure 6 shows road conditions in the same simulations of EDGE/OPEN and OPEN. 40 min after the start, the pedestrians did not select gate D in OPEN, but EDGE/OPEN guides them to it by closing other gates. At 80 min, EDGE/OPEN has lines at five gates with better balance than OPEN. At 120 min, although OPEN has a long line at gate A, most pedestrians of EDGE/OPEN have already entered the stadium.



**Fig. 6.**  $I = 80000$ . Average travel times of OPEN and EDGE/OPEN were 2481.0 and 1658.3 [s], respectively. Dot colors represent pedestrian speeds: blue is fast and red is slow. Red lines in front of gates are pedestrian lines for entry. (Color figure online)

## References

1. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 315–323 (2011)
2. Kato, Y., Shigenaka, S., Nishida, R., Onishi, M.: Real-time pedestrian control by reinforcement learning. In: Proceedings of the 64th Annual Conference of the Institute of Systems, Control and Information Engineers (ISCIE), pp. 312–316 (2020)
3. Little, J.D., Graves, S.C.: Little’s law. In: Building Intuition, pp. 81–100. Springer (2008)
4. Mnih, V., et al.: Asynchronous methods for deep reinforcement learning. In: Proceedings of International Conference on Machine Learning (ICML), pp. 1928–1937 (2016)
5. Otsuka, T., Shimizu, H., Iwata, T., Naya, F., Sawada, H., Ueda, N.: Bayesian optimization for crowd traffic control using multi-agent simulation. In: Proceedings of the 22st International Conference on Intelligent Transportation Systems (ITSC), IEEE (2019)
6. Shigenaka, S., Takami, S., Ozaki, Y., Onishi, M., Yamashita, T., Noda, I.: Evaluation of optimization for pedestrian route guidance in real-world crowded scene. In: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS), pp. 2192–2194. IFAAMAS (2019)
7. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT press, Cambridge (2018)



8. Xu, Z., van Hasselt, H.P., Silver, D.: Meta-gradient reinforcement learning. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 2396–2407 (2018)
9. Yamashita, T., Okada, T., Noda, I.: Implementation of simulation environment for exhaustive analysis of huge-scale pedestrian flow. *SICE J. Control Meas. Syst. Integr.* **6**(2), 137–146 (2013). <https://doi.org/10.9746/jcmsi.6.137>