Bus Loop Scheduling with Dueling Double Deep Q Network

Andri Pradana[1][0000−0002−9749−8882] and Lock Yue Chew[1][0000−0003−1366−8205]

Nanyang Technological University, 21 Nanyang Link, Singapore 637371, Singapore
andri.pradana@ntu.edu.sg, lockyue@ntu.edu.sg

Abstract. In this paper, we investigate the application of a reinforcement learning algorithm known as the Dueling Double Deep Q-Network to discover bus scheduling strategies and compare them against conventional approaches. In particular, we look into real-time control strategies where buses may choose to stay or leave at bus stops. We explore both waiting time and travel time as the optimization objectives. The results for uniform bus frequency show that average waiting time can be reduced by allowing buses to stay longer at stops with higher passengers' arrival rate but at the cost of increased average travel time. This is also supported by our analytical calculation on a theoretical bus loop model. We then apply our method to a model based on a real world bus loop in Nanyang Technological University. The results highlight the potential benefit of reinforcement learning methods to find novel strategies that can be better than conventional approaches.

Keywords: Bus scheduling · Reinforcement Learning · Complex Systems.

1 Introduction

Recently, the machine learning technique of reinforcement learning (RL) has been successful in tackling computational problems that are NP-hard. For example, self-RL had become a demonstrably state-of-the-art approach in discovering novel solutions in board games like Chess and Go [1, 2], which are NP-hard. In fact, DeepMind's self-RL algorithm known as AlphaZero [3] contributed new theoretical insights into chess playing after only four hours of self-play, even though humans developed extensive theories and principles in chess over centuries. Another class of NP-hard problem where RL finds success is the protein folding problem. Here, AlphaFold uncovered millions of intricate 3D protein structures which closely match laboratory determined experimental structures, leading to the team developing AlphaGO to win the Nobel Prize in Chemistry in 2024.

Bus scheduling is also a NP-hard problem. A network of buses picking-up and delivering commuters at bus-stops is in fact a complex system whose dynamics have been analyzed with its bunching behavior being recognized as a synchronization phenomenon [4]. Bus bunching is a form of operational inefficiency as it increases the waiting and travel time of the commuters, leading to a drop in the

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

quality of service of the buses. Bus operators had tried to address the bunching problem with different strategies, such as holding [5], stop-skipping [6], deadheading [7], limiting boarding [8], and dispatching buses with wide doors [9]. In addition, models are also created as a test bed to simulate intervention strategies to overcome the inefficiencies of bus bunching [10]. Recently, research exploring the use of self-RL to overcome the bunching problem interestingly uncovered two new strategies: the no-boarding strategy [11–13], and the semi-express bus configuration [14, 15]. For the no-boarding strategy, the bus may leave the bus stop even though there is somebody wishing to board. RL found that a combination of no-boarding and holding strategies creates a staggered bus configuration that avoids bus bunching and minimizes the average waiting time of commuters. The semi-express bus configuration, on the other hand, consists of a combination of normal buses (which pick up and deliver commuters at all bus stops) and express buses (which pick up and deliver commuters at selected bus stops). Such semi-express bus configuration has already been observed in bus networks and it is found to lower the average waiting time of commuters relative to the operation of purely normal buses. It is surmised that the efficiency of the semi-express bus configuration results from its intrinsic chaotic behavior.

In this paper, we investigate into an advanced RL algorithm known as the Dueling Double Deep Q Network (D3QN) to yield bus scheduling strategies that go beyond our previous approaches [11–13]. In particular, we base our evaluation by minimizing the average travel time of the commuters. This differs from our earlier approach and other works where optimization is performed according to bus headway. We compare our results with the case where the buses adopt the holding strategy. The reason for using the holding strategy as a basis of comparison results from its robustness and consistency in granting bus schedules that nearly achieve the optimal efficiency in bus operations.

2 Literature Review

To schedule buses, bus system operators perform planning, control, and operation on the buses after processing diverse sources of historical and real-time information. Their aim is to determine an optimal schedule for buses, where travel time are minimized, fleet utilization is maximized, and commuters face minimal waiting time. To achieve this goal, there is a need for accurate demand and traffic prediction, optimization of route and timetable, and a fleet management system that is efficient.

Our research focuses on the active real-time decision aspect of this optimization process, instead of the passive decisions which are only relevant over a longer time horizon, such as frequency of bus operations, timetabling of buses, and drivers' scheduling and rostering. Specifically, we look into real-time control strategies where buses may choose to stay or leave at bus stops. This is made possible recently with real-time data from Automated Passenger Counter System (APC), Automated Vehicle Location System (AVL), Geographical Positioning System (GPS), and Automated Fare Collection System (AFC) [16]. Although

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

techniques such as Integer Programming, Mixed-Integer Linear Programming, Genetic Algorithms, Simulated Annealing, Particle Swarm Optimization, Ant Colony Optimization, etc have been used to yield the optimal schedules, the advent of novel machine learning techniques have given rise to a more flexible approach to derive these optimal schedule in real-time [17]. In particular, we desire to determine the best bus scheduling strategies that would minimize either the waiting time or the travel time of the commuters.

In the literature, the machine learning techniques that relate to bus scheduling in real-time performs travel time prediction of the buses. Understanding the travel time of the buses would allow the inference of the travel time of the commuters and their waiting time. For instance, travel time predictions were performed using the k-nearest neighbor and random forest methods with GPS data when the bus travels between consecutive bus stops with traffic intersections in between [18]. When the road traffic exhibits high variability conditions, such as adverse weather, traffic junctions, diverse vehicular types, a Kalman filter cum support vector regression approach becomes applicable [19]. In another approach that predicts bus travel time, projection pursuit regression, support vector regression, random forest, together with dynamic weighting and dynamic weighting with selection in the integration step, are employed with data from AVL [20]. From another perspective, prediction of bus arrival time is also relevant. In this context, the techniques of artificial neural network and linear regression are utilized with a traffic density matrix to perform the prediction [21]. Alternatively, multilayer perceptron and a deep neural network implemented with PyTorch and TensorFlow could be used to predict the arrival time of buses at bus stops [22]. Finally, models capable of estimating travel and arrival time of buses can be built through linear regression, artificial neural network, and long short term memory network model with the usage of historical data from AVL systems, bus routes, and bus stop information as inputs [23].

One of the most successful machine learning techniques for bus scheduling is reinforcement learning. The decision outcome from it invariably relieves bus bunching and gives a staggered configuration that is optimal [24]. It is also applicable for real-time bus scheduling through the dynamical optimization of online schedule which leads to shorter commuters' waiting time and lower operating cost [25, 26]. RL has also been used to optimize the holding durations of each bus by means of a multi-agent framework when the bus system adopts the holding strategy [27]. A model that incorporates reinforcement learning in this sense was built to carry out dynamic holding control in a noisy environment, where the buses are modeled as agents that minimize headway deviations. The use of multi-agent reinforcement learning improves real-time operations compared with previous works which are focused on centralized control. This model adopts a hierarchical approach such that on top of the bus agents, other agents are needed to coordinate, manage and interface the agents with the environment [28]. In another piece of work, deep RL was applied to maintain bus headway at a prescribed value through holding the bus. The authors used Double Deep Q-learning to minimize deviations from a target headway, bus travel time, and

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

holding time by combining these quantities in a customized cost function. The action is holding the bus at a bus stop dynamically [29]. A recent improvement in dynamic bus control employs proximal policy optimization (PPO) [30] and a joint action tracker to exploit the multi-agent nature of the problem [31]. The technique was found to compare favorably against simpler methods.

A further development in Deep Q-learning techniques comes in the form of D3QN which integrates the Dueling Network Architecture with Double Deep Q-Learning [32]. The combination reduces overestimation bias and improves learning efficiency. Whilst D3QN has been explored in bus scheduling by deciding on the departure time of buses in the timetable [26], it has yet to be investigated on its efficacy in the determination of optimal dynamic bus control strategies, which is of principal interest in this paper.

3 Approaches

3.1 Dueling Double Deep Q-Network

The Dueling Double Deep Q-Network (D3QN) is an advanced RL architecture designed to improve the performance and stability of the standard Deep Q-Network (DQN) [32]. It combines three key techniques: Dueling Network Architecture, Double Q-Learning, and Deep Q-Learning.

Deep Q-Learning uses the DQN algorithm which is a RL algorithm implemented with neural networks to approximate the Q-value function $Q(s, a)$. $Q(s, a)$ gives the expected return for taking an action a in a state s. The network learns to predict the Q-values by minimizing the error between the predicted and target Q-values.

Because the same Q-network is used to select and evaluate actions, Deep Q-Learning can be plagued by overestimation bias. To address this issue, Double Q-Learning employs two separate networks where one is used for selection and the other evaluation. Specifically, Double Q-Learning uses a *Policy Network* as the primary network to choose the action, while it uses a *Target Network* to evaluate the Q-value of the chosen action as follows:

$$y = r + \gamma Q_{\text{target}}\left(s', \arg\max_a Q_{\text{policy}}(s', a)\right) . \tag{1}$$

Furthermore, the Q-value is computed for every action in the output layer in DQN. This, however, may not lead to good decision making. The Dueling Network Architecture improves this by splitting the evaluation of the Q-value function into two separate streams in the network. One stream estimates the *Value Function* $V(s)$, which denotes how good it is to be in state s, independent of the action. The second stream estimates the *Advantage Function* $A(s, a)$, which denotes the relative benefit of each action a in state s, compared to other actions. The two calculations are combined as follows to give the Q-value function of D3QN:

$$Q(s, a) = V(s) + \left(A(s, a) - \frac{1}{|A|} \sum_{a'} A(s, a') \right) . \tag{2}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

3.2 The Holding Strategy

The holding strategy is a bus scheduling technique employed to improve service reliability and manage the adherence of headway, i.e., the spacing between buses. The idea is to have the buses wait at certain control points (usually the bus stops) so as to prevent them from running too early or too close to the preceding bus. Thus, the strategy minimizes bus bunching and serves to maintain a consistent schedule for the commuters.

To implement the holding strategy, holding points need to be defined along the route where buses are instructed to wait. In addition, bus locations and headway are monitored in real-time using GPS or other tracking systems. Information from these devices allows the bus to know the gap ahead with the previous bus, and the gap behind with the following bus. The bus would then decide how long to wait by avoiding (a) being too close to the previous bus and (b) leaving a large gap with the following bus. This decision depends on a target interval between buses (which could be computed in real-time) the system aims to maintain. The consequence is a minimization of waiting time for the commuters at the bus stops.

The advantage of the holding strategy is a reduction of irregular service by preventing bus bunching. It improves service reliability by distributing the buses evenly along the route. Its key disadvantage is an increase in in-vehicle travel time for commuters who are already onboard. To be effective, it requires accurate real-time data and communication systems. It may become ineffective when subjected to delays due to traffic congestion.

4 Methods

4.1 Modeling and Simulation

We model a bus loop system as an isometric (distance-preserving) map on a unit circle, where the location along the loop can be denoted by a phase angle from $0°$ to $360°$. The arrival of passengers at each bus stop is assumed to follow a Poisson process corresponding to a specified arrival rate. Each bus has a natural frequency (rev/s) with which it travels the loop. After dropping all alighting passengers at a bus stop, each bus can choose an action: either *stay* or *leave*. The alighting and boarding rate is assumed to be 1 passenger per second. The simulation advances in discrete time steps with a 1 second simulation time step.

In a naive approach, after all the passengers alight at the bus stop, the bus picks up all the passengers wanting to board the bus and then simply leaves if there are no more passengers wanting to board. This is the approach in which bus bunching commonly occurs. In the holding strategy, after dropping all alighting passengers and picking up all boarding passengers, the bus has to stay or hold if its phase headway with the bus behind it is greater than the perfectly staggered phase, $360°/N_{bus}$, where N_{bus} is the number of buses in the loop. Otherwise, the bus just leaves. In our simulation, once a bus decides to stay/hold, we implement a 5 s holding duration before the bus checks its headway again. If a

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

passenger arrives within this duration, the passenger can board the bus. In the RL approach, the buses still have to drop all alighting passengers but can choose to stay or leave afterward, even when there are passengers wishing to board. However, once a bus decides to stay, we impose the condition that it must pick up all passengers wishing to board before making the decision to stay or leave again. Here, we also implement a 5 s holding duration.

4.2 Reward Function

Suppose that a decision for an action is required at time t and, after the decision is made, the next time a decision is required again is at time t'. The reward corresponding to an action a_t given the state s_t at time t is calculated according to a continuous-time discounted future reward function [33] given by

$$R(s_t, a_t) = \int_t^{t'} e^{-\beta(\tau - t)} r_\tau \, d\tau, \tag{3}$$

where r_τ is the instantaneous reward at simulation time τ and β is the decay rate of the discount factor $e^{-\beta(\tau-t)}$. The instantaneous reward r_τ is a function of the elapsed time from each passenger's perspective, counted from the time of arrival, which is given by

$$r_\tau = -\sum_p \left(\tau - t_p^{\text{arrival}}\right)^k \tag{4}$$

where the sum is taken over all passengers in the bus loop system. There is a negative sign in the reward because the optimization goal is to minimize, not maximize, either the waiting time or travel time. $k \geq 0$ is an exponent that controls the scaling of the reward with respect to passengers' elapsed time. $k = 0$ gives uniform reward regardless of the length of each passenger's elapsed time, whereas $k > 0$ penalizes longer passenger's elapsed time. We obtain the best results using $k = 0$ for optimization of average waiting time and $k = 1$ for optimization of average travel time. For waiting time optimization, the counting of a passenger's elapsed time ends when the passenger boards, whereas for travel time optimization, the counting ends when the passenger alights at the destination stop.

4.3 State Representation

The state of the system encompasses information about the bus stops and buses. For optimization of average waiting time, bus stop information includes the elapsed time of the earliest passenger arriving and still waiting at each stop. For optimization of average travel time, bus stop information includes the number of passengers waiting at each stop. There are N_{stop} entries for bus stop information, where N_{stop} is the number of bus stops.

For each bus, three types of information are included, and each of these information has N_{stop} entries: whether the bus is going to a bus stop or currently

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

at a bus stop (binary), its phase headway, and the number of passengers in the bus going to each bus stop. Although the headway for each bus is a single value, we find that the RL algorithm learns better when it is paired with the information of whether the bus is going to or currently at a bus stop. So, for the N_{stop} headway entries (each entry representing a bus stop), a headway entry can only be nonzero if the bus is going to or currently at a bus stop corresponding to that entry.

The ordering of the buses in the state description follows certain rules depending on the scenarios discussed in the following. The first scenario is one in which the buses have approximately identical natural frequencies. In this scenario, the ordering of the buses in the state description follows the actual ordering of the buses in the loop. The first bus in the state description is always the bus requiring the decision. The second one is the bus in front of the first bus in the loop, the third one is the bus in front of the second one, and so on. There are only two nodes in the output layer of the neural network, corresponding to the actions that this first bus can take. This implies that these buses follow a collective strategy, i.e., given similar situation or circumstances, the different buses make the same decision.

The second scenario is one in which there is *frequency detuning*, where the buses have different natural frequencies. In this scenario, the ordering of the buses in the state description is fixed, regardless of the actual ordering of the buses in the loop. The buses individually follow their own strategies, instead of a single collective strategy as in the previous scenario. Therefore, in the output layer of the neural network, there should be N_{bus} pairs of nodes, where each pair corresponds to each bus's set of actions. Thus, for the bus requiring the decision, the relevant Q-values are contained in the pair of nodes associated with it.

5 Results and Discussion

5.1 Uniform Bus Frequency

Reinforcement Learning Results We applied our reinforcement learning methods to a simple case of 4 bus stops and 3 buses with identical frequencies. Two bus stops are located at 0° and 30°, each with passenger arrival rate of 0.04/s, and the other two are located at 180° and 210°, each with passenger arrival rate of 0.02/s. It is assumed that all passengers are traveling to the stop antipodal to their origin stop. The frequency for all the buses is 1 mHz, so that without stopping, each bus can complete the loop in about 17 minutes. The results are summarized in Table 1. The results for the naive approach (where bus bunching occurs) and the holding strategy are also shown for comparison.

The averages are obtained from 100 simulation realizations with random initial condition. For each realization, the simulation is run for an initial period of 10,000 s first to smooth out any transient behavior before measurement is taken within the next 10,000 s. The different approaches considered include the naive approach (in which bus bunching occurs), the holding strategy, RL

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

Table 1. The average waiting time (AWT), average time spent traveling in a bus (ABT), and average travel time (ATT) for various approaches over 100 simulation realizations. RL-WT and RL-TT are reinforcement learning methods minimizing the waiting time and travel time, respectively. RL-WT-extended 1 and 2 allow the buses to learn to hold for a longer period of time. RL-WT-extended 2 allows the buses to hold even longer than RL-WT-extended 1, with a maximum holding duration of 200 s.

Approach	AWT (s)	ABT (s)	ATT (s)
Naive (bus bunching)	527 ± 24	540 ± 1	1073 ± 27
Holding strategy	171 ± 3	569 ± 3	741 ± 5
RL-WT	169 ± 3	569 ± 3	739 ± 4
RL-WT-extended 1	150 ± 4	693 ± 5	842 ± 4
RL-WT-extended 2	143 ± 4	755 ± 7	899 ± 7
RL-TT	174 ± 3	541 ± 2	717 ± 4

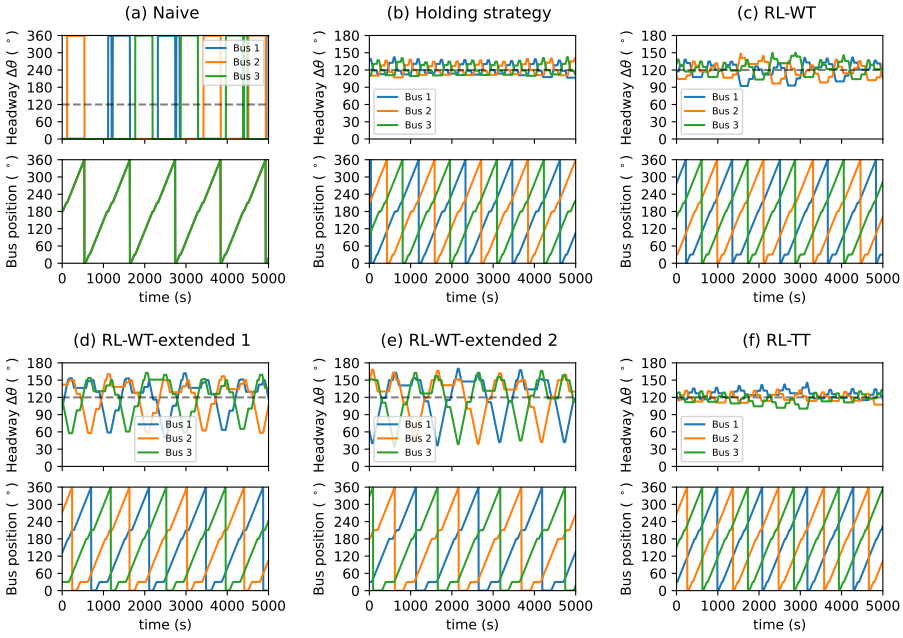

Fig. 1. Plots of phase headway (top) and position (bottom) vs time of the buses for the (a) naive approach, (b) holding strategy, (c) RL-WT, (d) RL-WT-extended 1, (e) RL-WT-extended 1, and (f) RL-TT. Dashed line on the top figures indicates the perfectly staggered phase headway of 120°. For bunched buses in (a), phase headway may 'jump' between 0° and 360° due to buses overtaking each other.

methods minimizing waiting time (RL-WT, RL-WT-extended 1 and 2), and RL method minimizing travel time (RL-TT). In RL-WT-extended 1 and 2, when the action to stay/hold is selected during the ε-greedy exploration, it also has a chance to execute this action multiple number of times consecutively (this

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

number of times is uniformly distributed with a specified maximum number of times). This allows for the buses to learn to hold for a longer period of time. RL-WT-extended 2 allows the buses to hold even longer than RL-WT-extended 1, with a maximum holding duration of 200 s. It is observed that the RL-WT-extended methods result in shorter average waiting time compared to the holding strategy, but at the cost of much longer average travel time. On the other hand, RL-TT approach has similar performance to the holding strategy in terms of both the average waiting and travel times. The holding strategy and the RL methods clearly perform better than the naive approach.

The plots of the phase headway and position against time of the buses are shown in Fig. 1. As expected, the phase headway for the holding strategy is very close to the perfectly staggered phase of 120°. The RL-WT and RL-TT methods also exhibit phase headway near 120°. The plots for position for the RL-WT-extended methods show that the buses execute longer holding at the bus stops with higher passengers' arrival rate (located at 0° and 30°). By doing so, the buses are able to reduce the average waiting time for the passengers at these stops considerably and, as a consequence, reduce the overall global average waiting time. Another consequence of this behavior is that the phase headway can deviate significantly from the perfectly staggered phase, as shown in Fig. 1(d) and (e). However, we find that staying longer at particular stops does not help in reducing the average travel time and may in fact be detrimental to it. We provide an analytical explanation for this in the next section.

Analytical Results Here, we consider a simple theoretical model of a bus loop system in which the buses have identical frequencies, the arrival of passengers is assumed to be continuous instead of discrete, and there is no fluctuation in the arrival rate. We also consider the staggered time-headway among the buses. At a bus stop, each bus drops all alighting passengers first and then picks up all waiting passengers until the bus stop is empty. Afterward, the bus can stay at the bus stop for an additional amount of time to pick up arriving passengers. However, the bus may only stay until the next bus arrives, such that only one bus can be at a bus stop at any point in time. Let the total time spent by a bus at stop i be τ_i and the time required for the bus from leaving stop i to reaching stop j in one loop be $\mathcal{T}_{i,j}$. The period of a bus completing a loop is then

$$T = \sum_{i=1}^{N_{\text{stop}}} (\tau_i + \mathcal{T}_{i,i+1}) = \left(\sum_{i=1}^{N_{\text{stop}}} \tau_i \right) + T^*, \tag{5}$$

where $T^* = \sum_i \mathcal{T}_{i,i+1}$ is the time spent by a bus moving on the road and the index $i \in \{1, 2, \ldots, N_{\text{stop}}\}$ denoting the i-th bus-stop is periodically bounded (so that $i = N_{\text{stop}} + 1 = 1$).

If there are N_{bus} buses in the loop following the staggered time-headway strategy, then the time interval between a bus leaving a bus stop and the next bus leaving the same stop is T/N_{bus}. All passengers arriving within this time interval will be picked up by the next bus. Let the passenger arrival rate at stop

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

i be r_i. Within the time interval from 0 to T/N_{bus}, a small passenger element, $dn_i = r_i\,dt$, arrives at stop i at time t and subsequently boards the next arriving bus. Since the bus leaves at time T/N_{bus}, the time interval between dn_i arriving and the bus leaving is given by $T/N_{\text{bus}} - t$. Therefore, the sum of elapsed time for all passengers for this stage is

$$\Delta t_1 = \sum_i \int_0^{\frac{T}{N_{\text{bus}}}} \left(\frac{T}{N_{\text{bus}}} - t\right) r_i\,dt = \frac{1}{2}\left(\frac{T}{N_{\text{bus}}}\right)^2 \sum_i r_i. \tag{6}$$

Next, the passengers travel in the bus to their destination stops. Let the passenger origin-destination probability from stop i to stop j be c_{ij} ($c_{ii} = 0$ and $\sum_j c_{ij} = 1$). Then the total number of passengers arriving at stop i within time interval T/N_{bus} who want to travel to stop j is $N_{ij} = r_i c_{ij} T/N_{\text{bus}}$. Noting that the travel time from origin stop i to destination stop j is precisely $\mathcal{T}_{i,j}$ defined earlier, the sum of elapsed time for all passengers for this stage is therefore

$$\Delta t_2 = \sum_{ij} N_{ij}\mathcal{T}_{i,j} = \frac{T}{N_{\text{bus}}} \sum_{ij} r_i c_{ij} \mathcal{T}_{i,j}. \tag{7}$$

Lastly, at a destination stop j, the total number of passengers alighting is $\sum_i N_{ij}$. Assuming an alighting/boarding rate of l, the amount of time it takes to drop all alighting passengers is $\sum_i N_{ij}/l$ and the small alighting passenger element is $dn = l\,dt$. Therefore, the sum of elapsed time for all passengers for this stage is

$$\Delta t_3 = \sum_j \int_0^{\sum_i N_{ij}/l} t\,l\,dt = \sum_j \frac{l}{2}\left(\frac{\sum_i N_{ij}}{l}\right)^2 = \frac{1}{2l}\left(\frac{T}{N_{\text{bus}}}\right)^2 \sum_j \left(\sum_i r_i c_{ij}\right)^2. \tag{8}$$

Finally, the average travel time for all the passengers can be calculated from $\text{ATT} = (\Delta t_1 + \Delta t_2 + \Delta t_3)/N$, where $N = \sum_{ij} N_{ij} = (T/N_{\text{bus}}) \sum_i r_i$ is the total number of passengers picked up by a bus in one loop. After substituting Δt_1, Δt_2, and Δt_3, and some algebraic manipulation, ATT can be expressed as

$$\text{ATT} = \frac{T}{2N_{\text{bus}}}\left[1 + \frac{\sum_j \left(\sum_i r_i c_{ij}\right)^2}{l \sum_i r_i}\right] + \frac{\sum_{ij} r_i c_{ij} \mathcal{T}_{i,j}}{\sum_i r_i}. \tag{9}$$

The task now is to find the appropriate values of T and $\mathcal{T}_{i,j}$'s in Eq. (9) which minimize ATT. These quantities are linear functions of τ_i's, which are adjustable. The relation between T and τ_i is given in Eq. (5). As for $\mathcal{T}_{i,j}$, since its definition is the travel time of a bus from stop i to stop j, it also includes the total time spent at bus stops in between i and j. For example, $\mathcal{T}_{1,4} = \mathcal{T}_{1,2} + \tau_2 + \mathcal{T}_{2,3} + \tau_3 + \mathcal{T}_{3,4}$. Here, $\mathcal{T}_{1,2}$, $\mathcal{T}_{2,3}$, and $\mathcal{T}_{3,4}$ cannot be decomposed any further as they are simply times spent purely on the road. So, the values of τ_i's need to be analyzed next.

The total time a bus spends at stop i, τ_i, can be broken down into three parts. First, once the bus arrives at the stop, it drops all $\sum_j N_{ji}$ alighting passengers with a rate of l. The total time spent for this part is

$$\tau_i^{(1)} = \frac{1}{l}\sum_j N_{ji} = \frac{T}{lN_{\text{bus}}} \sum_j r_j c_{ji}. \tag{10}$$

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

Next, the bus starts emptying the bus stop by picking up passengers. Recall that the time interval between a bus leaving the stop and the next bus leaving the same stop is T/N_{bus}. Therefore, the time interval between a bus leaving and the next bus arriving is given by $T/N_{\text{bus}} - \tau_i$. During this time interval plus an additional time $\tau_i^{(1)}$ defined above, passengers do not board yet and bus stop i keeps receiving passengers with a rate of r_i. The number of passengers accumulated at the stop after this time is then $r_i \times (T/N_{\text{bus}} - \tau_i + \tau_i^{(1)})$. Once the bus starts picking up passenger with the rate of l (with the stop still receiving passengers with a rate of r_i), the number of passengers at the stop reduces by a rate of $l - r_i$. The amount of time it takes to empty the bus stop is then

$$\tau_i^{(2)} = \frac{r_i}{l - r_i}\left(\frac{T}{N_{\text{bus}}} - \tau_i + \tau_i^{(1)}\right) = \frac{r_i}{l - r_i}\left[\frac{T}{N_{\text{bus}}}\left(1 + \frac{1}{l}\sum_j r_j c_{ji}\right) - \tau_i\right]. \tag{11}$$

After stop i is empty, the bus may stay at the stop for an additional amount of time s_i. We can now calculate the total time a bus spends at stop i as $\tau_i = \tau_i^{(1)} + \tau_i^{(2)} + s_i$ which, after substitution and some algebraic manipulation, yields

$$\tau_i = \left(1 - \frac{r_i}{l}\right) s_i + \frac{T}{l N_{\text{bus}}}\left(r_i + \sum_j r_j c_{ji}\right). \tag{12}$$

Taking the sum of τ_i over all stops and noting that $\sum_i \tau_i = T - T^*$ from Eq. (5) and $\sum_i c_{ji} = 1$ yields, after simplification, the expression

$$\frac{T}{l N_{\text{bus}}} = \frac{T^* + \sum_i \left(1 - \frac{r_i}{l}\right) s_i}{l N_{\text{bus}} - 2\sum_i r_i}. \tag{13}$$

Substituting this back into Eq. (12) yields

$$\tau_i = \left(1 - \frac{r_i}{l}\right) s_i + \frac{T^* + \sum_j \left(1 - \frac{r_j}{l}\right) s_j}{l N_{\text{bus}} - 2\sum_j r_j}\left(r_i + \sum_j r_j c_{ji}\right). \tag{14}$$

This is τ_i expressed in terms of s_i and other constants. More specifically, τ_i is a linear function of s_i's. Setting $s_i = 0$ for all i simultaneously minimizes τ_i for all i, which in turn minimizes T, $\mathcal{T}_{i,j}$'s, and, consequently, ATT as well. From Eqs. (13) and (14), the minimum values of τ_i and T are

$$\tau_{i,\min} = \frac{T^*}{l N_{\text{bus}} - 2\sum_j r_j}\left(r_i + \sum_j r_j c_{ji}\right), \tag{15}$$

$$T_{\min} = \frac{l N_{\text{bus}} T^*}{l N_{\text{bus}} - 2\sum_i r_i}. \tag{16}$$

The optimal condition of $s_i = 0$ implies that the buses should not stay unnecessarily longer at the bus stops. In fact, staying longer can be detrimental to the average travel time, which also agrees with the results in the previous section.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

We can use the parameters from the scenario of 3 buses serving 4 bus stops in the previous section to estimate the optimal average travel time. Inserting these parameters into Eqs. (15) and (16) yields $\tau_{1,\min} = \tau_{2,\min} = \tau_{3,\min} = \tau_{4,\min} \approx 21.74$ s and $T_{\min} \approx 1087$ s. Using Eq. (9), we obtain ATT ≈ 709 s. Despite the relative simplicity of this idealized model, its result is remarkably close to the holding strategy and the RL methods minimizing the travel time presented in Table 1 in the previous section. This is because in the scenario analyzed here, the time spent by each bus at the bus stops are much shorter than the time spent moving on the road. So, stopping at the bus stops should not appreciably change the phase difference between the buses. As a result, maintaining staggered time headway becomes similar to maintaining staggered phase headway.

5.2 Detuned Bus Frequency

We also applied our RL-TT method to a model based on a real world bus loop in Nanyang Technological University (NTU): the shuttle bus Blue route which consists of 12 reasonably staggered bus stops. Its measured parameters can be found in [4, 14]. In particular, here we consider the lull period of the afternoon on the weekdays with two detuned buses serving the loop with frequencies of 0.93 mHz and 1.39 mHz. The passengers' arrival rates for the 12 stops are 0.001,

Table 2. The average waiting time (AWT), average time spent traveling in a bus (ABT), and average travel time (ATT) for various approaches over 100 simulation realizations. RL-TT is reinforcement learning method minimizing the travel time.

Approach	AWT (s)	ABT (s)	ATT (s)
Naive (bus bunching)	449 ± 67	584 ± 5	1026 ± 67
Holding strategy	333 ± 5	704 ± 7	1041 ± 11
RL-TT	359 ± 10	581 ± 7	944 ± 15

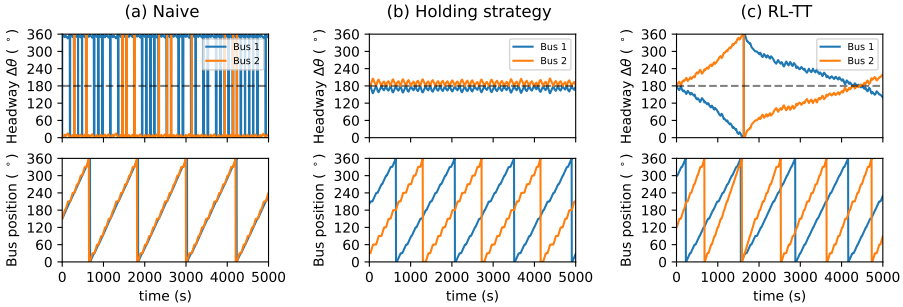

Fig. 2. Plots of phase headway (top) and position (bottom) vs time of the buses for the (a) naive approach, (b) holding strategy, and (c) RL-TT. Dashed line on the top figures indicates the perfectly staggered phase headway of 120°. Bus 1 is the slower bus. For bunched buses in (a), phase headway may 'jump' between 0° and 360° due to buses overtaking each other.

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

0.023, 0.015, 0.005, 0.016, 0.040, 0.018, 0.035, 0.024, 0.030, 0.007, and 0.010. All passengers are assumed to be traveling to the stop antipodal to their origin stop.

The results using the naive approach, holding strategy, and RL-TT method are shown in Table 2 and Fig. 2. The holding strategy yields a better average waiting time but surprisingly no improvement in the average travel time compared to the naive approach. On the other hand, the RL-TT method improves average travel time by around 8% over the naive approach and around 10% over the holding strategy. It also improves the average waiting time by around 22% over the naive approach. Interestingly, using the RL-TT method, it can be observed from Fig. 2(c) that the buses may become temporarily bunched before they quickly unbunch. Obviously, in this case the buses do not maintain a perfectly staggered phase headway of 180°. Perhaps unintuitively, allowing this temporary bunching turns out to yield a better average travel time.

6 Conclusion

We have explored the application of RL with the D3QN architecture to the bus loop system. For identical buses, waiting time optimization may result in better average waiting time compared to the holding strategy by allowing the buses to stay longer at stops with higher passengers' arrival rate, but at the cost of much longer average travel time. Using a theoretical model of a bus loop system, we also analytically proved that staying longer at any particular stops does not help in reducing the average travel time. This may not be a worthy trade-off considering the purpose of a transportation system is to move commuters to their destinations, preferably as quickly as possible. On the other hand, with travel time optimization, the buses managed to learn a close approximation of the holding strategy, maintaining headway near the perfectly staggered phase.

For detuned buses optimizing travel time, using the conditions of the NTU shuttle bus Blue route which consists of 12 reasonably staggered bus stops served by two buses, the buses surprisingly learned to allow temporary bunching to achieve better average travel time than both the holding and naive strategies, and also better average waiting time than the naive approach. The results highlights the potential benefit of RL methods to find novel strategies beyond just maintaining a target headway that are better than conventional approaches.

Finally, we found in our case that if the commuter inflow rate into the system is higher than the maximum delivery rate of the buses, unbounded growth of waiting commuters will occur which can only be solved by adding more buses. Therefore, we considered the scenario in which the Poissonian commuter inflow rate is comfortably lower than the maximum delivery rate. In this case, the buses may never be full and the bus capacity can effectively be treated as unlimited.

The weakness of our RL approach is the scaling problem. For a system with large number of buses and bus stops, we observe slow computation time and it becomes harder to converge to an optimal strategy. We intend to find ways to tackle this issue as our future work. In addition, since our approach is only based on the decision of each bus to stay or leave at bus stops, we believe our approach

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

is general and thus also applicable for non-loop services. We will look into its application in bus line services and expanding to the city-scale bus network.

Acknowledgments. This work was supported by the Singapore Ministry of Education (MOE) Academic Research Fund (AcRF) Tier 2 Grant No. MOE-T2EP20222-0004.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Silver, D., Hubert, T., Schrittwieser J. et al.: A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. Science, **362**(6419), 1140–1144 (2018).
2. Silver, D., Huang, A., Maddison, C. et al.: Mastering the game of Go with deep neural networks and tree search. Nature **529**, 484–489 (2016).
3. Kasparov, G.:Chess, a drosophila of reasoning. Science, **362**(6419), 1087 (2018).
4. Saw, V-L., Chung, N. N., Quek, W. L., Pang, Y. E. I., Chew, L. Y.: Bus bunching as a synchronisation phenomenon. Scientific Reports **9**, 6887 (2019).
5. Cats, O., Larijani, A. N., Koutsopoulos, H. N., Burghout, W.: Impacts of holding control strategies on transit performance. Transportation Research Record: Journal of the Transportation Research Board **2216**(1), 51—58 (2011).
6. Sun, A., Hickman, M.: The real-time stop-skipping problem. Journal of Intelligent Transportation Systems **9**(2), 91—109 (2005).
7. Furth, P. G.: Alternating deadheading in bus route operations. Transportation Science **19**(1), 13—28 (1985).
8. Delgado, F., Munoz, J. C., Giesen, R.: How much can holding and/or limiting boarding improve transit performance? Transportation Research Part B: Methodological **46**(9), 1202—1217 (2012).
9. Stewart, C., El-Geneidy, A.: All aboard at all Doors. Transportation Research Record: Journal of the Transportation Research Board **2418**(1), 39—48 (2014).
10. Quek, W. L., Chung, N. N., Saw, V-L., Chew, L. Y.: Analysis and simulation of intervention strategies against bus bunching by means of an empirical agent-based model. Complexity **2021**, 2606191 (2021).
11. Saw, V.-L., Chew, L. Y.: No-boarding buses: Synchronisation for efficiency. PLoS One **15**(3), e0230377 (2020).
12. Saw, V.-L., Chew, L. Y.: No-boarding buses: agents allowed to cooperate or defect. Journal of Physics: Complexity **1**(1), 015005 (2020).
13. Saw, V.-L., Vismara, L., Chew, L. Y.: Intelligent buses in a loop service: emergence of no-boarding and holding strategies. Complexity **2020**, 7274254 (2020).
14. Vismara, L., Chew, L. Y., Saw, V.-L.: Optimal assignment of buses to bus stops in a loop by reinforcement learning. Physica A **583**, 126268 (2021).
15. Saw, V.-L., Vismara, L., Chew, L. Y.: Chaotic semi-express buses in a loop. Chaos **31**, 023122 (2021).
16. Ibarra-Rojas, O., Delgado, F., Giesen, R., Muñoz, J.: Planning, operation, and control of bus transport systems: a literature review. Transportation Research Part B: Methodological **77**, 38—75 (2015).
17. Saw, V-L., Vismara, L., Suryadi, Yang, B., Johansson, M., Chew, L. Y.: Inferring origin-destination distribution of agent transfer in a complex network using deep gated recurrent units. Scientific Reports **13**, 8287 (2023).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3

18. Bahuleyan, H., Vanajakshi, L. D.: Arterial path-level travel time estimation using machine-learning techniques. Journal of Computing in Civil Engineering **31**(3), 04016070 (2017).
19. Reddy, K. K., Kumar, B. A., Vanajakshi, L.: Bus travel time prediction under high variability conditions. Current Science **111**(4), 700 (2016).
20. Mendes-Moreira, J., Jorge, A. M., de Sousa, J. F., Soares, C.: Improving the accuracy of long-term travel time prediction using heterogeneous ensembles. Neurocomputing **150**, 428—439 (2015).
21. Panovski, D., Scurtu, V., Zaharia, T.: A neural network- based approach for public transportation prediction with traffic density matrix. Proc., 7th European Workshop on Visual Information Processing (EUVIP), pp. 1-–6. IEEE, New York, Tampere, Finland, (2018).
22. Heghedus, C., Chakravorty, A., Rong, C.: Neural network frameworks. Comparison on public transportation prediction. Proc., International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 842—849. IEEE, New York, Rio de Janeiro, Brazil, (2019).
23. Taparia, A., Brady, M.: Bus journey and arrival time prediction based on archived AVL/GPS data using machine learning. Proc., 7th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS), pp. 1—6. IEEE, New York, Heraklion, Greece, (2021).
24. Xiao, M., Xiahou, J., Ge, M.: A reinforcement-learning-based bus scheduling model. 2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), pp. 923–927. IEEE, New York, Chongqing, China, (2022).
25. Ai, G., Zuo, X., Chen, G., Wua, B.: Deep reinforcement learning based dynamic optimization of bus timetable. Applied Soft Computing **131**, 109752 (2022).
26. Liu, Y., Zuo, X., Ai, G., Liu, Y.: A reinforcement learning-based approach for online bus scheduling. Knowledge-Based Systems **271**, 110584 (2023).
27. Chen, W., Zhou, K., Chen, C.: Real-time bus holding control on a transit corridorv based on multi-agent reinforcement learning. Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pp. 100–106. IEEE, New York, Rio de Janeiro, Brazil, (2016).
28. Chen, C.X., Chen, W.Y., Chen, Z.Y.: A multi-agent reinforcement learning approach for bus holding control strategies. Advances in Transportation Studies, 41–54 (2015).
29. Alesiani, F., Gkiotsalitis, K.: Reinforcement learning-based bus holding for high-frequency services. in: BT - 21st International Conference on Intelligent Transportation Systems, ITSC 2018, pp. 3162–3168. Maui, HI, USA, November 4-7, (2018).
30. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv:1707.06347, (2017).
31. Wang, J., Sun, L.: Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework, Transportation Research Part C: Emerging Technologies **116**, 102661 (2020).
32. Wang, Z., Schaul, T., Hessel, M., Hasselt, H. v., Lanctot, M., Freitas, N. d.: Dueling Network Architectures for Deep Reinforcement Learning. arXiv:1511.06581, (2015).
33. Bradtke, S.J., Duff, M.O.: Reinforcement learning methods for continuous-time Markov decision problems. In: Tesauro, G., Touretzky, D., Leen, T. (eds.) Advances in Neural Information Processing Systems, NIPS 1994, vol. 7, MIT Press (1994).

ICCS Camera Ready Version 2025
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-97626-1_3