

Research Article

Intelligent Buses in a Loop Service: Emergence of No-Boarding and Holding Strategies

Vee-Liem Saw ^{1,2}, Luca Vismara ^{1,3} and Lock Yue Chew ^{1,2,4}

¹Division of Physics and Applied Physics, School of Physical and Mathematical Sciences, 21 Nanyang Link, Nanyang Technological University, Singapore 637371

²Data Science and Artificial Intelligence Research Centre, Block N4 #02a-32, Nanyang Avenue, Nanyang Technological University, Singapore 639798

³Complexity Institute, Interdisciplinary Graduate Programme, 61 Nanyang Drive, Nanyang Technological University, Singapore 637335

⁴Complexity Institute, 61 Nanyang Drive, Nanyang Technological University, Singapore 637335

Correspondence should be addressed to Lock Yue Chew; lockyue@ntu.edu.sg

Received 11 April 2020; Revised 23 May 2020; Accepted 7 July 2020; Published 26 August 2020

Guest Editor: Felix Chan

Copyright © 2020 Vee-Liem Saw et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We study how N intelligent buses serving a loop of M bus stops learn a *no-boarding strategy* and a *holding strategy* by reinforcement learning. The no-boarding and holding strategies emerge from the actions of *stay* or *leave* when a bus is at a bus stop and everyone who wishes to alight has done so. A reward that encourages the buses to strive towards a staggered phase difference amongst them whilst picking up passengers allows the reinforcement learning process to converge to an optimal Q-table within a reasonable amount of simulation time. It is remarkable that this emergent behaviour of intelligent buses turns out to minimise the average waiting time of commuters, in various setups where buses move with the same speed or different speeds, during busy as well as lull periods. Cooperative actions are also observed, e.g., the buses learn to *unbunch*.

1. Introduction

In the urban environment, a bus network system is a complex system. It is not possible to study a part of the system to understand its whole. In order to characterise the dynamics within the system, a holistic approach has to be taken. Buses move through space and pick up commuters who arrive at bus stops at random moments. This leads to a spatiotemporal complexity within the bus system which has been shown to behave chaotically [1, 2].

Bus systems are notorious for being highly susceptible towards buses ending up bunching, leading to unnecessarily long waiting times for commuters. Buses may end up bunching due to uncertainty in travelling time, as well as the fact that a preceding bus would have a proclivity of picking up most of the passengers at a bus stop whilst the succeeding bus picks up relatively fewer passengers when it then arrives. This speeds up the latter as it spends less time stopping,

allowing it to catch up with the former—leading to them bunching.

A common goal of bus operators is to schedule the fleet to minimise such bunching occurrence. Alternatively, some approaches consider dynamical corrections to maintain equal headway between buses. The popular method to stabilise the buses' headways is to implement a holding strategy [3–21] which slows down a fast bus if its headway with respect to the bus ahead of it is deemed to be too small. Other approaches include stop-skipping [11, 22–26] or deadheading (i.e., sending an empty bus directly to a set of designated bus stops) [23, 26–29], limiting boarding [30–35], as well as despatching buses with wide doors [36–38].

1.1. Literature Review. Many studies in the literature have focused on the *holding strategy* [3–21]: if a bus is too fast, it would exercise an extended stoppage duration to correct for

the headway from the bus in front of it—otherwise, it would bunch with it. Holding back buses, however, may tend to slow down the system and would require that some slack in the schedule has been allocated beforehand. Some of these investigated how bus systems can improve their scheduling using some form of reinforcement learning [8, 9, 16–18, 20, 21]. For example, Chen et al. [17] implemented a multiagent reinforcement learning framework solely to optimise holding durations for each bus, where they run around a loop corridor of bus stops. This simple setup let all buses move at the same speed in one loop, and overtaking is not allowed. Furthermore, they implemented their reinforcement learning scheme on a relatively small discrete set of states and actions (about 900 states in total, with each bus having 4 actions to choose from corresponding to the duration of holding), which is not readily or directly scalable to larger and more complicated bus route networks without a change in architecture. Nevertheless, Menda et al. [21] introduced a deep reinforcement learning framework, whereby a deep neural network serves as a function approximator to extend the state space from discrete to continuous and illustrated that it works reasonably well in the holding strategy carried out in [17]. Even a recent work by Alesiani and Gkiotsalitis [20] that applied deep reinforcement learning methods to model a bus loop service around Chua Chu Kang-Yew Tee in Singapore only focused on “how long to hold a bus at a bus stop,” with the aim of maintaining the headway close to some prescribed headway, but balanced by not excessively adding to total travel time due to holding. Similar to the work in [17, 21], Alesiani and Gkiotsalitis [20] also assumed that all buses are identical and do not overtake, which we have found to be not generally true in real human-driven buses [39].

In contrast to the majority of studies focusing on holding back buses, relatively fewer studies explored the opposite, *viz.* a *no-boarding strategy* [30–35]: a slow bus would always allow passengers to alight at a bus stop, but would disallow boarding and leave the bus stop if it is *too slow* in order to speed it up. Preliminary work by Delgado et al. [30] argued that this is a viable strategy to improve the bus system, mainly by carrying out extensive simulations. The bus system decides on when to implement no-boarding by numerically optimising an objective function subjected to a number of constraints. All buses are assumed to have an identical speed, no overtaking is allowed, and they introduced a “dummy bus” as a form of boundary condition such that all remaining passengers must eventually be picked up. Further simulation work in [31] extended the work of Delgado et al. [30] to more complicated setups with more constraints and dropping the use of the “dummy bus,” though as before they considered buses to move in a loop. Similar improvements due to the no-boarding strategy are also observed. In contrast to references [30, 31], Zhao et al. [32] constructed a theoretical analysis that describes how this strategy is beneficial, but appeared not to consider alighting (passengers only board the bus). More comprehensive theoretical calculations are given by Saw and Chew [34], where commuters first alight at a bus stop before new passengers can board. They also considered simulations on

buses with different speeds and can overtake. Sun and Schmöcker [33] considered the scenarios where buses can overtake and cannot overtake, where they allow only a certain percentage of passengers to board the front bus of a platoon of bunched buses.

Apart from references [30–35], we are not aware of any other papers on a no-boarding strategy, and none of them involved a fully reinforcement learning approach. The relatively fewer papers on no-boarding compared to holding may be due to the perception that a bus which is not full but not allowing passengers to board is not going to go well with the public. To account for commuters who have urgent needs to board the bus, Saw and Chew [35] considered a no-boarding strategy that allows the commuters to cooperate or defect the no-boarding instruction. Of course, there are pros and cons to cooperating or defecting. A defector may occasionally get away if the number of defectors is small, but may be penalised with a hefty fine otherwise. It turns out that such an arrangement is viable as the overall system of buses-commuters would settle down in a dynamical equilibrium whereby there would always be some proportion of cooperators and defectors, and the buses are able to implement no-boarding to prevent bus bunching.

Generally, the no-boarding strategy works well for a bus system with buses moving with the same average speed. This strategy maintains the buses’ headways close to being ideally staggered, *i.e.*, buses are reasonably spread out throughout the route. If the bus system is such that different buses move with different average speeds, the no-boarding strategy is also successful during the busy period since there is enough demand to slow down the “faster bus.” Surprisingly, this strategy backfires during the lull period, as the slow bus has been sped up to the maximum by picking up nobody whilst there is insufficient demand to slow down the fast bus enough [31, 34, 40]. Consequently, the system is effectively operating with one less bus (since the slowest bus is almost always disallowing boarding).

1.2. Bus Bunching as a Synchronisation Phenomenon. Bus systems which serve a loop of bus stops are common, where buses start from a terminal and go around a route of bus stops, before returning to the same terminal. Several previous works on studying how bus systems can be optimised also considered a loop service [17, 20, 21, 30–32, 39]. In particular, our work here is based on a real university loop shuttle bus service where buses continually serve a loop of 12 bus stops [39] (see Figure 1).

Work in [39] described buses serving a loop of bus stops as coupled oscillators which interact with one another due to picking up passengers from bus stops. Given a generic loop where buses travel around, we can map such a loop onto a unit circle and preserve distances between neighbouring points. Such a mathematical representation allows us to define angles on the unit circle to represent the locations of the buses. The distance between buses is then represented as the phase difference between them on the unit circle. Furthermore, the notion of the speed of a bus on the route becomes equivalent to angular frequency (or just frequency)

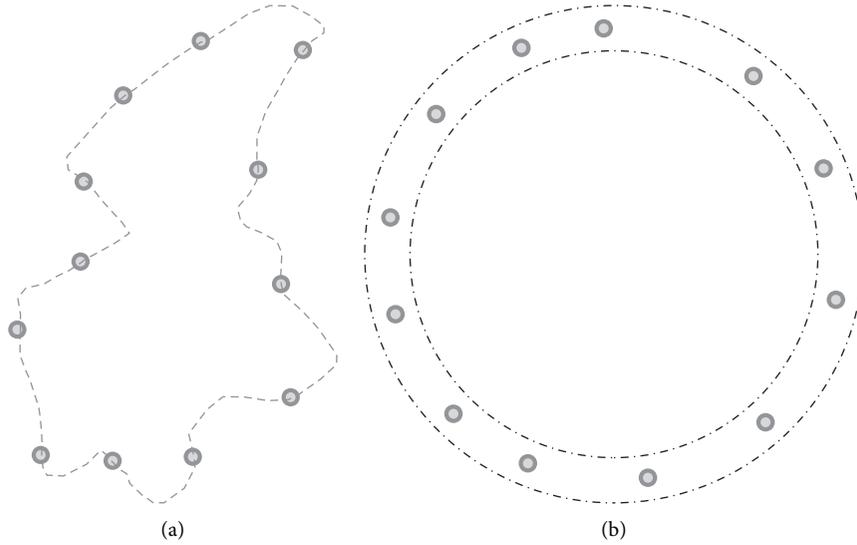


FIGURE 1: (a) A map of the *Blue* route in the Nanyang Technological University, Singapore (NTU), campus shuttle loop service. Buses go clockwise. (b) The corresponding map of the loop on a unit circle, preserving distances between neighbouring points on the actual loop. Grey circular markers denote each of the 12 bus stops in this loop. With the unit circle representing a generic bus loop service, the speed of a bus can be described by its angular frequency (or frequency) of motion around the unit circle. Different human-driven buses with different intrinsic speeds correspond to oscillators moving with different natural frequencies on the unit circle. Oscillators with different natural frequencies are said to have *frequency detuning*.

on the unit circle. The notion of “frequency detuning” [41] denotes the fact that different human-driven buses generally move at their own natural speeds (one driver has a tendency to drive faster, and another prefers to drive a bit slower).

Consider N buses serving M bus stops in a loop. These M bus stops are staggered around the loop, each having a people arrival rate of s people per second. Each of the N buses has natural (angular) frequencies $\omega_1 > \omega_2 > \dots > \omega_N$, respectively (excluding any time stopped at bus stops). When a bus arrives at a bus stop, it first allows passengers who wish to alight to do so and then allows passengers to board. The alighting and boarding rate is l people per second. Overall, the quantity $k := s/l$ is a parameter that describes the level of demand for service. We assume that passengers from a bus stop would like to travel to the bus stop antipodally opposite to it, or the one just before travelling half a loop if M is odd. By an analytical calculation, work described in [39] has shown that there is a critical $k_c(N)$:

$$k_c(N) = \frac{1}{M} \sum_{i=1}^{N-1} \left(1 - \frac{\omega_N}{\omega_i} \right), \quad (1)$$

where all N buses would be completely bunched into a single unit (i.e., completely synchronised) if $k > k_c(N)$. Also, we have the relations $\omega_i = 2\pi f_i = 2\pi/T_i$ between angular frequency, frequency, and period.

This antipodal destination is by no means an unrealistic simplification. One may argue that in a realistic bus system, passengers from one bus stop have some probability distribution to travel to some other bus stops. On average, when a bus arrives at a bus stop, there would be some proportion of passengers who wish to alight. The antipodal destination effectively also leads to some proportion of the passengers on

the bus who wish to alight—the only difference is that they originated from the same bus stop which is located antipodally away, instead of from several different bus stops. This does not affect the overall dynamics of how long a bus generally stops at a bus stop. We have considered the situation where people want to alight at any bus stop (excluding the bus stop of origin) with uniform probability distribution. Our simulations show that the average waiting time is essentially identical to the setup where people want to head to an antipodal bus stop. The average time spent on the bus, on the other hand, would depend on the probability distribution of where people want to go, since if more people want to head to a further bus stop, then they would spend more time on the bus to travel further.

(Note: Saw et al. [39] assumed that boarding and alighting occur *simultaneously* via different doors. Here, we assume that these occur *sequentially* through one door, alighting followed by boarding. Thus, we should include an overall factor of $1/2$ into equation (1) for this paper since the processes being sequential via one door instead of two different doors would double the time a bus dwells at a bus stop. Generally, if multiple doors are available such that people could board simultaneously, we also find via simulations that the dwell time is proportionately reduced by such a factor (i.e., if there are two doors, then the dwell time is halved). However, there are instances where this is not exactly the case, for example, when there is an odd number of people, then a bus still needs to spend an extra time step to allow for that extra person to board via one of the two doors).

On top of that, numerical simulations with parameters based on values measured from a real university shuttle bus loop service showed that the buses are not persistently

bunched (no phase-locking, completely unsynchronised) if $k < \bar{k}$, where $\bar{k} := k_c(2)$ is the corresponding system with $N = 2$ buses having natural frequencies ω_1 and ω_N . The bus system would have some buses persistently bunched (partial synchronisation) if $\bar{k} < k < k_c(N)$. In the case where all buses have an identical natural frequency, all buses would typically end up bunching into a single unit unless k is sufficiently low such that each bus only spends the minimum amount of time stopping at each bus stop. Therefore, bus bunching is a perennial phenomenon, and it is of great interest to employ strategies such that the buses are able to maintain a regular headway between them, always remaining staggered.

1.3. The Problem: A Loop Shuttle Bus Service Modelled after Our NTU Campus Loop Shuttle Bus Service. The purpose of this paper is to explore if there are ways beyond what was analytically studied in [34], such that a no-boarding policy may actually be salutary, especially in the lull period for a bus system with frequency detuning (where no-boarding backfires). Investigating a bus system with frequency detuning is crucial because human-driven buses tend to move with different natural frequencies due to differing driving styles as measured in our Nanyang Technological University (NTU) campus loop shuttle bus service [39]. This is in fact the key result of bus bunching being viewed as a synchronisation phenomenon of coupled oscillators with different natural frequencies, as elaborated in the previous section. The opposite of bunching, viz. to remain staggered, is also a synchronisation of the oscillators such that they remain phase-locked with phase difference between adjacent oscillators being $\sim 2\pi/N$, where N is the number of buses (oscillators) [39, 40].

Instead of implementing a human-thought-out or human-defined idea, we let buses figure out an ideal strategy via *reinforcement learning* [42]: buses are given two actions whenever they are at a bus stop, viz. *stay* or *leave*, with no prejudice nor human input, other than a feedback on whether the average waiting time of the commuters is minimised. This feedback may be the actual waiting time of the commuters, or the headway between the buses where a stable staggered headway signifies stable minimal average waiting time. We will consider the latter here and the former in a separate paper elsewhere. A so-called normal bus would *stay* if there is somebody who wants to board and *leave* if there is nobody there. A no-boarding strategy would correspond to the bus deciding to *leave* even if there is somebody who wants to board. Apart from the no-boarding strategy, the framework that is developed here would also be well suited to study the holding strategy: if there is nobody at the bus stop, a normal bus would *leave*, but a holding strategy would correspond to *stay*. By reinforcement learning (we shall employ Q-learning in this paper), the buses are initialised with random choices to execute. They would then progressively explore and converge to an optimal strategy, as is implied by the theory of Q-learning based on a Markov decision process [42].

Our consideration here with a single loop of $M = 12$ bus stops is modelled after a university (NTU) campus shuttle

bus service that serves tens of thousands of students, staff, and faculty members [43, 44]. This is thus a realistic system which also exists in many bus systems worldwide with loop services, just like those studied in [17, 20, 21, 30–32]. Note that unlike references [17, 20, 21] which applied reinforcement learning on the holding strategy, our approach does not endow the buses with a conscience of holding (or no-boarding); rather, we will show that the holding (as well as no-boarding) strategy emerges from low-level actions of *stay* or *leave*.

Thus, this paper will show how buses with no knowledge of the no-boarding and holding strategies would turn out to collectively learn them via reinforcement learning, with only two actions to execute whenever they are at a bus stop. This will be achieved by the buses observing local distances of the bus immediately behind them and striving to keep such distances close to the prescribed idealisation whereby the buses would be staggered throughout the loop. In doing so, the bus system collectively turns out to learn cooperative actions such as unbunching themselves, if they end up being in unfavourable bunched configurations. Such an approach whereby no explicit human instructions are given to the system has the potential of uncovering novel solutions in more complicated situations to be further explored in subsequent work.

2. Setup for Reinforcement Learning of the Bus Loop System

2.1. Reward for Actions of Stay or Leave Taken at Every Time Step at Bus Stop. With the goal of minimising the average waiting time of commuters for a bus to arrive at a bus stop (i.e., this excludes any time spent on the bus), each time a bus is at a bus stop (and passengers who want to alight have done so), it executes either *stay* or *leave* and then receives the waiting time of the person ahead of the queue to board the bus (or who is supposed to board, but denied boarding if the bus leaves). However, this feedback is problematic: the waiting time of each person has high variance. For instance, the luckiest person who arrives at the bus stop when a bus is there has zero waiting time, whilst the unluckiest person who arrives when a bus has just left would have maximum waiting time, with the other people's waiting times distributed between these two extremes. This high variance is difficult for our low-level setup where a bus takes an action at every time step when it is at a bus stop and receives such a fluctuating feedback. Such a setup is low level in the sense that the decision-making process occurs at the level of *microscopic time scale*. Nevertheless, a different formulation where a bus only takes one single action when it arrives at a bus stop (viz. to decide to hold, implement no-boarding, or behave like a normal bus), instead of deciding whether to stay or leave at every time step, would be able to converge to a solution. The high-level formulation (making decision at a *macroscopic time scale* with the specified behaviour of holding, no-boarding, and normal bus) will be reported elsewhere in a separate paper.

Alternatively, we note the following property for a bus loop system [34]: if the buses are staggered, then the average

waiting time of commuters for a bus to arrive at the bus stop is minimised. The loop can be isometrically (i.e., preserving distance) mapped to a unit circle, which has well-defined phase angles from 0° to 360° . Thus, we consider the feedback for the buses' actions as *the phase difference between itself and the bus immediately behind it*, $\Delta\theta$. A recent work from [34] found that backwards-looking angle turns out to be superior to forward-looking angles. Whilst the inclusion of both angles (or even more information like number of passengers waiting at the bus stops) may lead to further improved performance, the additional quantities would augment the state space and slow down the optimisation for the reinforcement learning process. Moreover, we find that the performance solely using the angle from behind turns out to be close to the ideal situation where buses are nearly perfectly staggered. This phase difference experiences a more gradual change, thereby eliminating the high variance of measuring individual commuters' waiting times. The goal would be to keep this phase difference close to the staggered value. In other words, buses would be *rewarded for being close to the staggered configuration*. For example, if there are two buses, then $\Delta\theta = 0^\circ$ gives 0 reward, whilst $\Delta\theta = 180^\circ$ gives 1 point, with values in between scaling linearly. If $\Delta\theta > 180^\circ$, then the reward linearly decreases to 0 at $\Delta\theta = 360^\circ$. In fact, striving to keep headway staggered is also applicable to nonloop services as it amounts to keeping buses spaced out from the start of the route to its end.

Note, however, this alone would lead to the buses striving to achieve the perfectly staggered configuration without any regard to the commuters, possibly even at the expense of not boarding anybody just to keep $\Delta\theta = 180^\circ$. Therefore, we incentivise a reward of 1 point for each passenger who is picked up. A weighting hyperparameter can be selected such that the system is in a balanced region between closeness to staggered configuration and picking up passengers, where a bus aims to both pick up passengers and maintain a configuration that is nearly staggered. Of course, the choice and structure of the reward are arbitrary—as long as it results in the intended minimisation of the average waiting time. Different reward functions can lead to the same eventual result of keeping the angle between buses staggered to minimise the average waiting time. For example, the shape of the reward function that we use in equation (4) is linear, but other shapes can lead to the same result. This will be discussed in more detail in Section 4.

Nevertheless, there are other kinds of rewards that aim to minimise other quantities such as the worst waiting time of a commuter for a bus to arrive at the bus stop or the total travelling time. For example, if the reward function aims to minimise time spent on bus or total travelling time, then the holding strategy in the lull period with frequency detuning would be different since it increases these quantities (see Section 4.2.3). Hence, the eventual outcome does depend on the intended reward functions and what the bus operator strives to optimise.

2.2. Situations of Interest. The setup for the bus system undergoing Q-learning is as follows. Each bus has its own

Q-table containing 72 states where they represent the phase difference as measured from the bus immediately behind it. This number of states is arbitrary, chosen to balance between not being too coarse and not taking too long for the simulations to run. Moreover, for subsequent future applications on real-time nonstationary environments, it would be desirable for these buses to respond fast enough to adapt appropriately. Independent Q-tables allow different buses to possibly learn different strategies, where one bus may occasionally perform a “sacrificial action” for the system as a whole to benefit.

The 72 states would coarse-grain the phase difference into bins of 5° . In each of these states, it records the two Q-values representing the expected total rewards for the two actions *stay* or *leave*, respectively. The buses typically move around on the road, where it must proceed with moving forward. When it reaches a bus stop, it must allow passengers who wish to alight to do so, i.e., we do not allow the possibility of stop-skipping. This is because we find it to be not beneficial to speed up the bus at the expense of another round of time spent on the bus for these passengers, or asking them to alight one stop earlier and “walk their last mile” to their intended destinations. Furthermore, this allows the reinforcement learning process to have better chances of converging to an optimal Q-table for every bus within a reasonable amount of simulation time.

The only time a bus is allowed to consider whether to execute *stay* or *leave* is when it is at a bus stop and there is nobody on the bus who wishes to alight. Here are the following situations that we would explore. A bus is allowed to consider its action when it is at a bus stop, and:

- (1) There is nobody to alight, but there is somebody at the bus stop who wishes to board.
- (2) There is nobody to alight as well as nobody at the bus stop who wishes to board.
- (3) There is nobody to alight.

The first situation is intended to create a possibility where the buses may learn to implement the no-boarding strategy, since it may learn to leave the bus stop even though there is somebody who wishes to board. The second situation is intended to create a possibility where the buses may learn to implement the holding strategy, since it may learn to stay at the bus stop even when there is nobody to pick up. Finally, the third situation allows the possibility for the buses to learn some combination of the no-boarding and holding strategies. In the first two situations, each bus has a Q-table with 72 states and each state contains two values—one for *stay* and one for *leave*. For the third situation, there are 144 states because 72 states are when there is somebody who wants to board and another 72 states are when there is nobody who wants to board.

2.3. Updating the Q-Table. In Q-learning [42], when a bus is at a bus stop and has to pick an action A of either *stay* or *leave*, it has to first determine what state S it is presently in. To do so, it measures its phase difference $\Delta\theta$ with respect to the bus behind it. In this state, there are two actions and it

chooses the one which has the highest Q-value—unless it is in the ϵ -greedy exploration phase where there is a probability ϵ of randomly selecting an action. According to the theory, after executing the action A and receiving a reward of R , it should then subsequently measure again its phase difference from the bus immediately behind it to determine its future state S' for the purpose of updating its Q-table with a future expected reward:

$$Q(S, A) \leftarrow Q(S, A) + \alpha(R + \gamma \max_a (Q(S', a)) - Q(S, A)). \quad (2)$$

The hyperparameters α is the learning rate and γ is the discount factor. The former determines how sensitively the Q-values would adjust due to new feedback, whilst the latter determines how seriously to believe an estimated future expected reward from its own Q-table.

However, since the bus only gets to execute an action at a bus stop when nobody wants to alight, if its action is *leave*, then its future state would only occur quite some time later when it reaches a bus stop. We find that this has the effect of affecting reliable convergence as other things may happen during the time when this bus leaves the bus stop and reaches another bus stop: for example, other buses would have picked up passengers at other bus stops and affect the overall waiting time of the commuters, as well as leading to a widely different future $\Delta\theta$. To circumnavigate this issue in obtaining an estimated future expected reward for updating the Q-table, we impose that when a bus executes *stay* in a state S with phase difference $\Delta\theta$, then its future state S' is defined as the same state with phase difference $\Delta\theta' := \Delta\theta$ since it remained; whilst if instead it executes *leave*, then its future state S' is defined as the state with phase difference $\Delta\theta' := \Delta\theta + 5^\circ$, i.e., the phase difference has increased, since it moves to increase the phase difference from the bus behind it. Recall that this 5° is the size of each state, since we use 72 states to discretise the angles from 0° to 360° .

2.4. Bus System Environment Simulation Parameters and Reinforcement Learning Hyperparameters. In all our simulations for the bus system environment, the parameters used are based on values measured from a real university (NTU) shuttle bus loop service with $M = 12$ bus stops [39]. The value for the rate of passengers boarding/alighting is $l = 1$ person per second. In the lull period, a representative average value for the people arrival rate at each bus stop is about $s = 0.020$ people per second, whilst that in the busy period it could be as high as $s = 0.065$ people per second. The natural frequencies of the buses are measured to be in the range of 0.93 mHz to 1.39 mHz, or a natural period of 12 minutes to 18 minutes excluding time stopped at bus stops. We adapt these values accordingly in our simulations for the bus system environment. Each simulation time step corresponds to 1 second.

For reinforcement learning, we carry out 1000 episodes, where each episode is 150 revolutions long. At the start of each new episode, the buses are randomly placed on the loop. The performance of the bus system in each episode is measured from the last 30 revolutions, where most of the

transient part due to random initial conditions would have been weeded out. The system undergoes ϵ -greedy learning (i.e., there is a probability of ϵ that a random action is taken), where ϵ decays linearly from 1 to 0.1 in the first 200 episodes, after which it remains at 0.1 until the 700th episode. The learning rate α is kept at 0.2 for the first 700 episodes. In the last 300 episodes, we let the system fully exploit what they have learnt, with $\epsilon = 0$, and α toned down to 0.1. The discount factor is always fixed at $\gamma = 0.9$.

The first 200 episodes represent an *exploration phase*, where the buses carry out many random actions due to the high value of ϵ . This is crucial to allow for the buses to avoid getting stuck in near-sighted local minima which may lead to missing out potentially better long-term strategies. The next 500 episodes form a mix of *exploration and exploitation*, where the buses take advantage of their learned Q-tables but maintain some degree of exploration just in case they get stuck in some local minima. Finally, the last 300 episodes denote a *fully exploitation phase*. Here, the buses still fine-tune their Q-tables since $\alpha = 0.1$. The difference from previous episodes is that they now always take their best-perceived action, never taking a random action anymore.

For each particular setup throughout this paper, we carry out (at least) five independent runs. Generally, we obtain essentially identical qualitative results when the same setup is repeated even though the learning process involves random initial conditions in each new episode and stochasticity in the ϵ -greedy exploration. This therefore assures robustness in our results.

Before diving into these interesting situations involving N buses serving M bus stops, we first consider the simplest or trivial situation of $N = 1$ bus serving $M = 12$ bus stops in the next section. With only one bus, there is no nontrivial phase difference with respect to another bus. Therefore, this bus must eventually *learn to be a normal bus*, i.e., it *stays* to pick up passengers when there is somebody who wishes to board and *leaves* otherwise. In Section 4, we study the case of $N = 2$ buses serving $M = 12$ bus stops for each of the three situations described in Section 2.2, followed by more buses in Section 5.

3. $N = 1$ Bus Learns to Be a Bus

With $N = 1$ bus serving $M = 12$ staggered bus stops in a loop, we aim to let this bus learn to be a bus, i.e., learn to *stay* at the bus stop when somebody wants to board and *leave* when nobody is at the bus stop. Recall that by default, it must allow anybody who wishes to alight to do so. Figure 2 shows the average waiting time of commuters at the bus stop for a bus to arrive, average time spent on bus, average total travel time (which is the sum of waiting time and time spent on bus), and average number of passengers on the bus, all as functions of number of episodes. Since there is only one bus, its phase difference as measured from the bus behind it (itself) is always 360° (or 0°). Hence, the reward is purely 1 point for each person picked up.

This reinforcement learning scenario corresponds to the third one listed in Section 2.2 where the bus decides on an action when it is at a bus stop and nobody wants to alight.

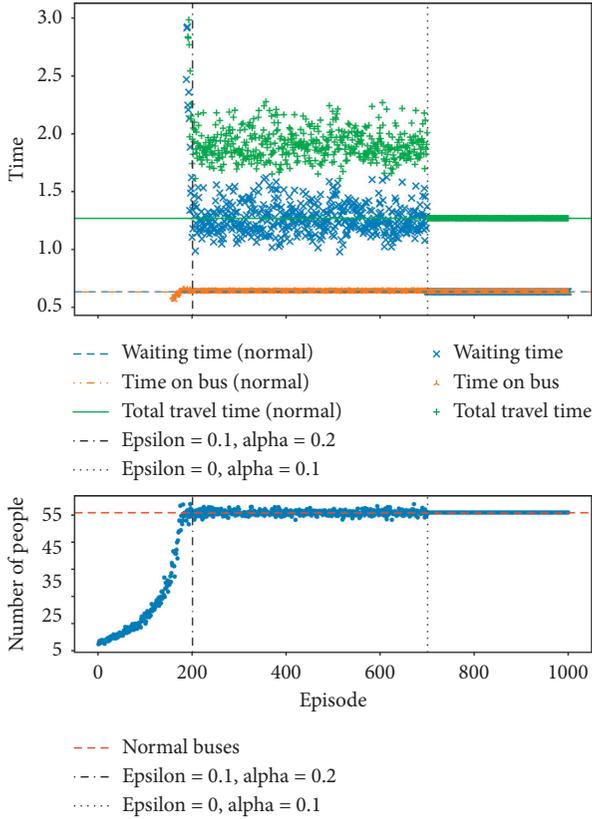


FIGURE 2: A bus serving a loop of bus stops learns to be a bus by reinforcement learning.

The bus has a Q-table with two states, one when somebody wants to board and the other when nobody wants to board. Each state has two Q-values, one for *stay* and one for *leave*. This Q-table therefore has only four numbers. For the bus system environment, we set the natural period of the bus (excluding time stopped at bus stops) to be $T = 12$ minutes, and the rate of people arriving at each bus stop $s = 0.010$ people per second. The unit of time for the top graph is $T = 12$ minutes (this is the unit of time for all graphs in this paper, unless otherwise stated).

As Figure 2 shows, the bus successfully learns to behave like a normal bus (*stays* when there is somebody to board and *leaves* when there is nobody to board), where it matches the performance of a hard-coded normal bus when it acts greedily in the last 300 episodes based on the Q-values that it has learnt. In the 201st to 700th episode, since $\epsilon = 0.1$, it makes a random action once in every ten times, on average. A wrong action has ramifications on the waiting time of the commuters, since the bus leaves and they have to wait for one additional revolution. Only when the bus acts greedily does the performance match that of a hard-coded normal bus. Nevertheless, the time spent on the bus is not too affected during the phase where $\epsilon = 0.1$, since passengers who want to alight must be allowed to do so. Large variance in the average number of passengers on the bus is observed before the 701st episode due to the ϵ -greedy action selection. This variance vanishes in the last 300 episodes when the bus acts greedily.

4. $N = 2$ Buses Learn No-Boarding and Holding

Let us now study the interesting situations with $N = 2$ buses serving a loop of $M = 12$ staggered bus stops. We consider bus system environments with the following three setups throughout this section which are based on the measured values from the NTU campus shuttle buses [39]:

- Identical natural frequency, taking $T = 12$ minutes to complete a loop (excluding time stopped at bus stops). The rate of people arriving at each bus stop is set at $s = 0.010$ people per second.
- Frequency detuning, with $T_1 = 12$ minutes and $T_2 = 18$ minutes to complete a loop (excluding time stopped at bus stops), respectively. The first bus is the faster one, whilst the second bus is the slower one. We consider a busy period where $s = 0.040$ people per second. A *busy period* is defined by $k > \bar{k}$ in equation (1), where at least a pair of buses is persistently bunched. With these T_1 and T_2 , we have the critical $k_c = \bar{k} = 0.014$ ($\bar{k} = k_c$ since $N = 2$). Note that we have included an overall factor of $1/2$ in equation (1) since alighting and boarding occur sequentially.
- Frequency detuning in (b), during a lull period where $s = 0.010$ people per second. A *lull period* is defined by $k < \bar{k}$ in equation (1), where no buses are permanently bunched.

Note that it suffices to consider one value of $s = 0.010$ in (a) where the buses have an identical natural frequency, since the behaviour of the bus system is the same for any fixed s . The different phases of lull and busy become distinct only when the system has frequency detuning [34].

For each of the three situations 1, 2, and 3 as stated in the Introduction (Section 2.2), we consider these setups (a), (b), and (c).

4.1. No-Boarding. The first situation is where buses are given the choices to *stay* or *leave* whenever they are at a bus stop, everybody who wishes to alight has done so, and there are passengers who would like to board. The reward R_{NB} for each action (applicable to a system with any N number of buses) is given by

$$R_{NB} := P + wf(\Delta\theta), \quad (3)$$

where P is 1 if a person is picked up and 0 otherwise. Note that since the rate of people loading is $l = 1$ person per second, either somebody boards or nobody boards at any time step of the simulation so this quantity is well defined.

The phase difference of the bus from the bus immediately behind it, $\Delta\theta$, gives a reward defined by

$$f(\Delta\theta) = \begin{cases} \frac{\Delta\theta}{360^\circ/N}, & \text{if } \Delta\theta \leq \frac{360^\circ}{N}, \\ 1, & \text{otherwise.} \end{cases} \quad (4)$$

This function $f(\Delta\theta)$ which remains at 1 beyond $360^\circ/N$ implies that the bus is doing fine and is not too slow, but is only receiving linearly diminishing reward if $\Delta\theta$ is smaller than $360^\circ/N$ which implies that it is too slow. The rationale for $f(\Delta\theta)$ staying flat instead of decreasing beyond $360^\circ/N$ is due to the fact that once there is nobody at the bus stop, it must leave, i.e., there is no option for it to lengthen its stay or try holding back. It is only when it is too slow ($\Delta\theta < 360^\circ/N$) that it gets a lower reward.

A weight w balances between P (which encourages *stay*) and $f(\Delta\theta)$ (which encourages *leave* when $\Delta\theta < 360^\circ/N$). Generally, a small $w \ll 1$ leads to the buses eventually learning to behave like normal buses, where they would always *stay* since they always find somebody at the bus stop who wants to board. On the other hand, $w \gg 1$ leads to the buses eventually learning to always *leave* and maintain their perfectly staggered configuration of $\Delta\theta \approx 360^\circ/N$. There is a finite range of $w \sim 1$ where the bus system eventually learns to both pick up passengers and attain a reasonably staggered configuration such that the *average waiting time* of the commuters at the bus stop is minimised. This precise range for w depends on the particular conditions of the simulation environment like s , l , T_i , N , and M . In each of our reinforcement learning runs, we set an appropriate w in the balanced range. It appears that as long as w is within this range, essentially identical qualitative results are obtained. In other words, the actual value of w is unimportant as long as it is within that balanced range (we will see later that the corresponding setup is not quite true for situation 2 on holding).

4.1.1. Identical Natural Frequency. Figure 3 shows the results of these two buses with identical natural frequency undergoing reinforcement learning, comprising the average waiting time at the bus stop for a bus to arrive, average time spent on the bus, average total travel time (sum of the previous two quantities), and average number of passengers on the bus. In addition, it also shows the average phase difference from one bus as measured from the bus behind it, as well as the Q-tables for each bus.

The performance of the $N = 2$ system is comparable to the analytical results in [34] where no-boarding is hard-coded, with an average waiting time of ~ 0.30 units of T during the last 300 episodes where the buses act greedily with respect to their learned Q-tables. Normal buses would typically end up bunching and the average waiting time is ~ 0.55 units of T , so we see a nearly 50% improvement.

Generally, a bus would implement no-boarding, i.e., *leave* if $\Delta\theta < 360^\circ/N$, and *stay* otherwise. Remarkably, they also discover the following known result from [34]: there is

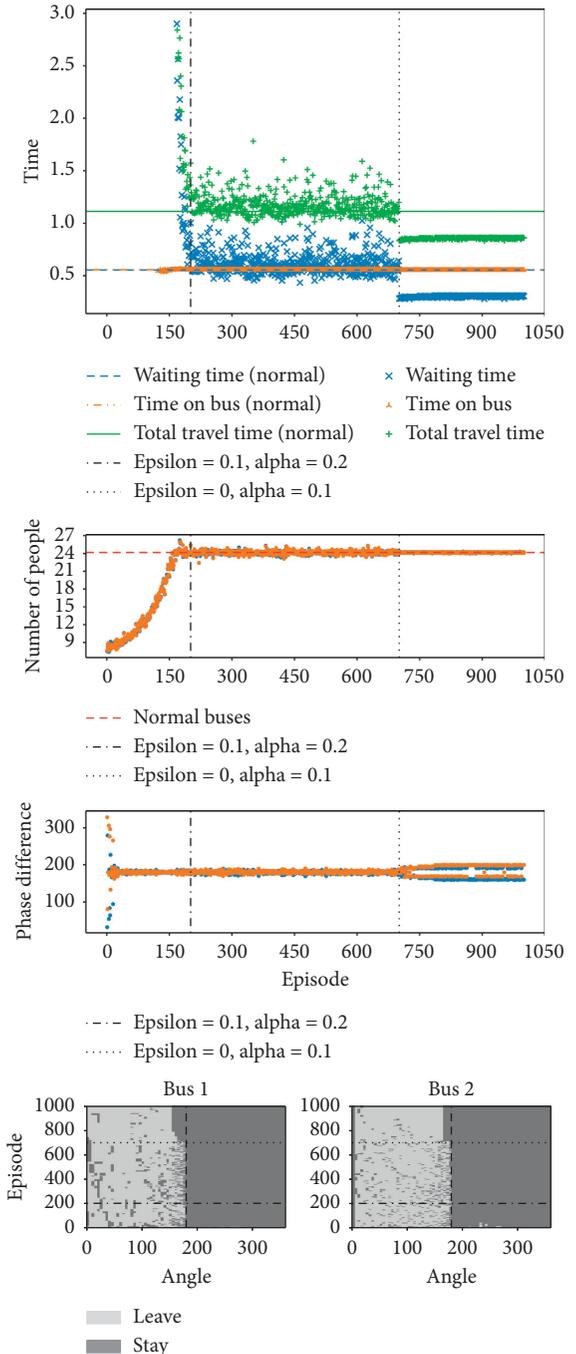


FIGURE 3: Two buses with identical natural frequency serving a loop of bus stops learn the no-boarding strategy by reinforcement learning. Notably, they learn about the *upper bound* at some angle *strictly* less than 180° to implement no-boarding, where if they exceed, then they would inadvertently implement no-boarding too frequently such that they are not meeting the level of demand for service. Also, they learn to *unbunch*, where one learns to *stay* whilst the other learns to *leave*, if their phase difference is 0° .

an *upper bound* on the angle to implement the no-boarding strategy, *strictly* below the perfectly staggered angle of $360^\circ/N$. This upper bound arises because if the angle is too close to the staggered configuration, then the buses would end up implementing no-boarding too frequently at a rate

where the passengers picked up is lower than the demand for service. The correspondence to the results from [34] is seen when the buses act greedily. On the other hand, in the ϵ -greedy phase before the 701st episode, the upper bound is $\sim 360^\circ/N$ because there is a probability of *staying* which picks up passengers instead of stringently not boarding them. This is noted in the graph for the average phase difference and the Q-tables for the two buses: the buses are typically fluctuating around $\Delta\theta = 180^\circ$ when exploration is involved, but this value is strictly less than 180° in the fully exploitation phase. The performance before the 701st episode is about those of normal buses with an average waiting time of ~ 0.55 units of T and sometimes even 1 unit of T , instead of ~ 0.30 units of T in the last 300 episodes.

Since the buses are randomly placed on the loop at the start of every episode, they may occasionally end up bunching. Can they unbunch? The answer is *affirmative*. Since the buses are endowed with independent Q-tables, they learn *opposite actions* if $\Delta\theta = 0^\circ$: one bus *stays* and the other *leaves*. The system as a whole discovers a cooperative mechanism to correct itself when bunched. These results are consistently obtained in all five independent runs.

The effect of this no-boarding strategy is to reduce the average time a commuter has to wait at the bus stop for a bus to arrive (i.e., the average waiting time). It does not increase the average time spent on bus, since it only leaves a bus stop earlier but never dwells at a bus stop longer than necessary. Therefore, a reduction in average waiting time with no increase in average time spent on bus would imply a corresponding reduction in average total travel time.

4.1.2. Frequency Detuning during Busy Period. Figure 4 shows the corresponding results of these two buses with frequency detuning undergoing reinforcement learning during a busy period. The results here are essentially the same as the case with identical natural frequency, where the buses are able to learn the no-boarding policy. Bus 1 is the faster bus (in all frequency detuning cases for $N = 2$, bus 1 is always the faster bus) and tends to pick up more passengers since the slower bus implements no-boarding and leave, leaving more passengers to the former to slow down its higher natural frequency. The average waiting time is also comparable to the results found in [34]. Incidentally, the absence of data points before the 701st episode in the first graph is because the quantities are way too large due to the great number of passengers demanding service but not quite met by these two buses, such that they are beyond the range of the graph shown here.

Similar to the case with identical natural frequency, the two buses learn opposite actions when $\Delta\theta = 0^\circ$ which would enable them to unbunch, and they also discover some upper bound strictly less than 180° where no-boarding is implemented. The slower bus (bus 2) seems to find a lower value for the upper bound to implement no-boarding than the faster bus (bus 1), since it is the one which is usually slower and has to implement no-boarding. The slow bus should spend enough time at the bus stop to actually allow passengers to board, even though its phase difference $\Delta\theta$ may be

less than 180° ; otherwise, it would not be picking up passengers and loses some reward for that. Therefore, the upper bound for it to implement no-boarding is lower to let this happen.

4.1.3. Frequency Detuning during Lull Period. Figure 5 shows the corresponding results of these two buses with frequency detuning undergoing reinforcement learning during a lull period. Here, the buses do not quite end up with the expected no-boarding strategy. This is in accordance with the observation noted in [34] where the no-boarding strategy backfires during the lull period because the slow bus has been sped up to the maximum by not picking up anybody! A hard-coded no-boarding policy would lead to the system effectively serving with one less bus because the slow bus almost always implements the no-boarding policy. Here, the buses found that perhaps it is better to just behave (almost) like normal buses, with performance that eventually matches closely to those of hard-coded normal buses. Incidentally, they do not necessarily need opposite actions when $\Delta\theta = 0^\circ$ because their different natural frequencies allow them to unbunch.

Astonishingly, the optimal strategies for these two buses appear to defy what a human may intuitively conceive (at least initially), upon examining the Q-tables of the buses (bottom plots in Figure 5). When they begin to act greedily from the 701st episode onwards (whilst still maintaining a learning rate of $\alpha = 0.1$ so that they do continuously fine-tune their Q-tables), the slow bus (bus 2) quickly changes to always behaving like a normal bus, with the *fast bus* (bus 1) implementing the no-boarding policy when it is “too slow.” Eventually, by the 1000th episode, it implements the no-boarding policy if $\Delta\theta \sim 60^\circ$ from the bus behind it.

Perhaps the slow bus realises that there is no point for it to implement no-boarding as it simply cannot be sped up fast enough to overcome its lower relative velocity, and if it keeps leaving, then it loses reward from not picking up passengers. On the other hand, the fast bus seems to think that when it is too slow, it should just *leave* so that it can quickly attain $\Delta\theta \sim 180^\circ$ which offers greater reward compared to getting stuck near $\Delta\theta \sim 60^\circ$. If $\Delta\theta \ll 60^\circ$, it probably would lose too much from not picking up passengers if it *leaves*, before it can make $\Delta\theta$ grow up to $\sim 180^\circ$ so that it would rather behave normally and just *stay*, whilst if $\Delta\theta \gg 60^\circ$, then the deficit in reward is not too high compared to $\Delta\theta \sim 180^\circ$, such that it is fine with behaving normally and just *stay* to earn the reward from picking up passengers. Hence, in the lull period, we find that instead of trying in vain to keep the two buses staggered, they effectively *increase the frequency detuning*.

With only the no-boarding strategy being studied in [34], could the holding strategy or a holding + no-boarding strategy work to somehow provide some form of improvement for the bus system during the lull period? This is one primary aim of the framework in this paper, where we investigate reinforcement learning of the bus system to learn holding and holding + no-boarding strategies in the following sections with $N = 2$ buses serving a loop of bus stops.

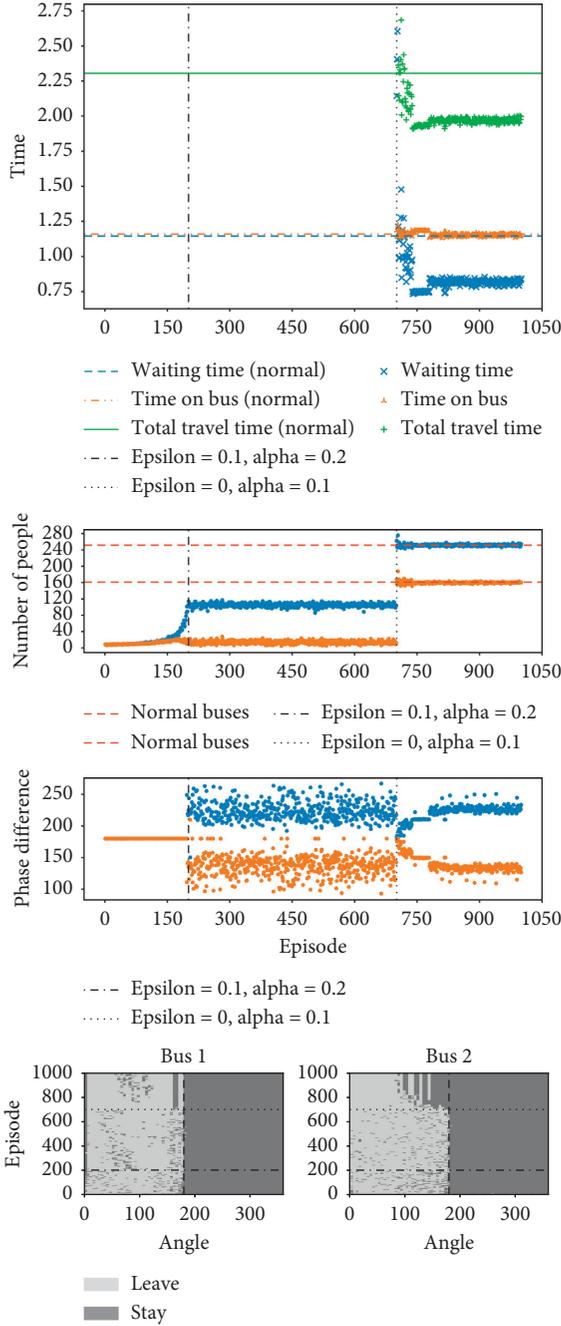


FIGURE 4: Two buses with frequency detuning serving a loop of bus stops during a busy period learn the no-boarding strategy by reinforcement learning. The results are qualitatively similar to those in Figure 3.

4.2. *Holding.* The second situation is where buses are given the choices to *stay* or *leave* whenever they are at a bus stop, everybody who wishes to alight has done so, and there is nobody at the bus stop. The reward R_H for each action (applicable to a system with any N number of buses) is as follows:

$$R_H := g(\Delta\theta), \tag{5}$$

where

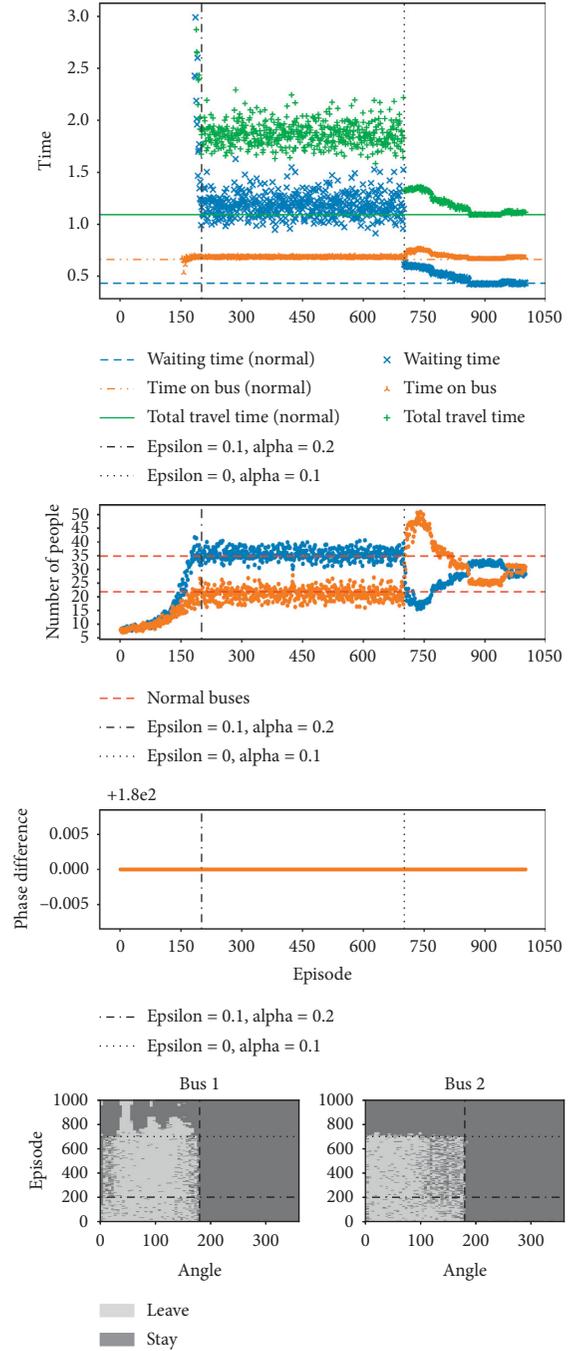


FIGURE 5: Two buses with frequency detuning serving a loop of bus stops during a lull period *do not* learn the expected no-boarding strategy by reinforcement learning.

$$g(\Delta\theta) = \begin{cases} \frac{1 - \Delta\theta/360^\circ}{1 - 1/N}, & \text{if } \Delta\theta > \frac{360^\circ}{N}, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

This function $g(\Delta\theta)$ is discontinuous at $\Delta\theta = 360^\circ/N$. It is 0 at and less than $360^\circ/N$ since it is regarded as “slow” with respect to the bus behind it. On the other hand, it approaches 1 from the right if $\Delta\theta > 360^\circ/N$ since this is the ideal phase

difference that it should strive for when it is “too fast.” From a reward of 1 just over $360^\circ/N$, it then linearly decreases to 0 as $\Delta\theta$ grows towards 360° since larger phase difference is getting away from ideal.

This reward R_H does not say anything about how long it will repeatedly *stay* at a bus stop. One option is to include a negative reward so that buses do not simply remain at a bus stop indefinitely, i.e., -1 for each *stay* action or if a certain number of consecutive *stay* actions are executed (recall that in this situation, everyone who wishes to alight has done so, and there is nobody at the bus stop hence a negative reward discourages “time wasting”). Then, a weight w_H could be introduced between this negative reward and $g(\Delta\theta)$, analogous to the no-boarding reward R_{NB} in equation (3). The hope with this is that there is some balanced region for w_H such that the bus system does not excessively remain at a bus stop. It turns out that equation (5) works well and a bus does not indefinitely remain at a bus stop because it will get 0 reward if $\Delta\theta \leq 360^\circ/N$ which would prompt it to *leave*. Furthermore, unlike the no-boarding case where the actual value of w does not change the outcome as long as w is in the balanced region, here different values of w_H would lead to different durations a bus may hold at a bus stop. We find this to be equivalent to just imposing a limit on how long a bus can hold.

Note also that this situation is different from the no-boarding situation in the following sense: in the no-boarding situation, if a bus chooses the “unconventional” action of *leave* when there is somebody to pick up, then that is the end for this round at this current bus stop. It leaves the bus stop and moves on. However, for the holding situation here, if a bus chooses the “unconventional” action of *stay* when there is nobody to pick up, then it gets to choose its action again at this current bus stop. This is why the nature of the rewards and their emergent behaviours (as we will see below for holding) are not directly analogous.

4.2.1. Identical Natural Frequency. Figure 6 shows the results of these two buses with identical natural frequency undergoing reinforcement learning, corresponding to the graphs in the previous figures. The holding strategy looks impressively effective, where it achieves sub-0.3 units of T for the average waiting time of the commuters, even before the 201st episode where ϵ decays to 0.1 and $\alpha = 0.2$. The way the holding strategy works is that if $\Delta\theta$ is not reasonably close to $360^\circ/N$, then the faster bus would remain at the bus stop until $\Delta\theta \sim 360^\circ/N$. Since there is not much noise in the simulation environment (no traffic, rate of people arrival at bus stops is constant), the two buses would just remain fairly staggered thereafter.

The buses learn that there is a *lower bound* to implement the holding strategy, which is strictly larger than $360^\circ/N$. This is the consequence of the discontinuity at $360^\circ/N$ in the reward R_H in equation (5) where its value at $360^\circ/N$ itself is 0, which discourages *staying*. The buses also learn to never *stay* for any phase difference $\Delta\theta \leq 360^\circ/N$ as that gives 0 reward. This is important because the two buses may be exactly staggered with $\Delta\theta = 180^\circ$, and if they both learn to *stay*, then they would just stay forever.

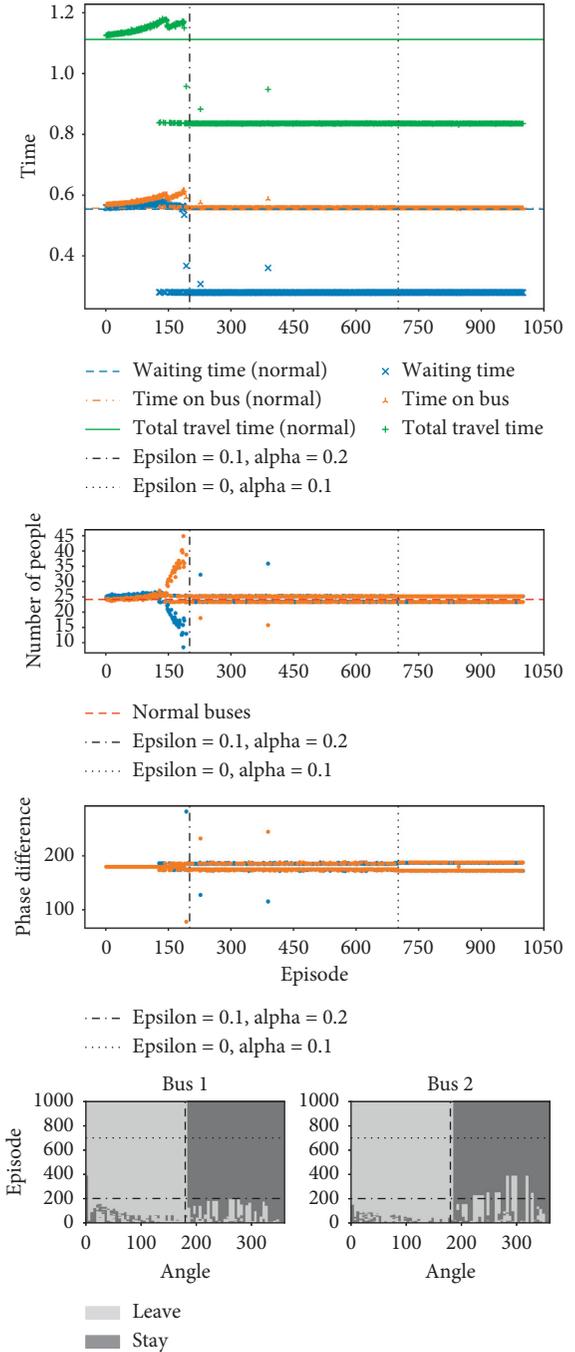


FIGURE 6: Two buses with identical natural frequency serving a loop of bus stops learn the holding strategy by reinforcement learning.

Curiously, the buses ostensibly learn opposite actions when they bunch, i.e., $\Delta\theta = 0^\circ$, during some earlier episodes, but these opposite actions disappear in subsequent episodes and both end up learning to *leave* if $\Delta\theta = 0^\circ$. How do they unbunch then, if they both take the same action? Since their positions on the loop are randomised at the start of each episode, they would inevitably end up bunching at some point. Upon closer inspection, there is actually a natural mechanism for normal buses to *momentarily* unbunch: if there is an odd number of passengers at the bus stop (or the

number of passengers modulo N is not zero, for N bunched buses in general), then at that instant one bus stays there to pick up that last person whilst the other bus sees nobody and leaves (note: in the simulation, a loop over all buses is carried out at that instant. The first bus sees somebody and stays to pick up, whilst the second bus then sees nobody and leaves). This is how normal buses can momentarily unbunch from $\Delta\theta = 0^\circ$ to be $\Delta\theta = 5^\circ > 0^\circ$ (recall that we discretise the angles by 72 bins of 5°). Of course, at the next bus stop, they would swiftly bunch again since they have to allow passengers to alight. So normal buses remain bunched. Even buses implementing the no-boarding strategy would remain bunched, which is why they learn opposite strategies via the process of reinforcement learning over the 1000 episodes.

However, the holding strategy differs in this unique manner: when a pair of bunched buses naturally unbunch due to one bus staying to pick up somebody and the other bus leaving as it has nobody to pick up, then the bus that stayed would see its phase difference as measured from the bus behind it (which is the bus that has just left, in front of it) to be $\Delta\theta = 355^\circ$, implying that it is *way too fast!* Therefore, it would implement *stay* all the way based on its Q-table, until $\Delta\theta = 185^\circ$. This is why there is no need for these buses implementing the holding strategy to have to learn opposite strategies.

4.2.2. Frequency Detuning during Busy Period. Figure 7 shows the results of these two buses with frequency detuning undergoing reinforcement learning, corresponding to the graphs in the previous figures. Here, demand for service is high in a busy period. The holding strategy slows down the fast bus, effectively slowing down the entire bus system. Since the reward for the system is purely to keep $\Delta\theta$ staggered, it does not care about staying too long at bus stops and the average waiting time suffers. This is indicated by the average number of passengers on the bus blowing up into thousands! Since there are only $N = 2$ buses trying to meet a high demand during the busy period, mistakes made when the buses explore other actions would lead to many of the other $M = 12$ bus stops rapidly accumulating passengers waiting for service. We will see in Section 5 that with $N = 6$ buses in the busy period, there are sufficient buses going around such that they are able to reasonably learn the holding strategy. With more buses, mistakes made by one bus when it explores are covered by other buses such that the number of passengers waiting at the $M = 12$ bus stops does not blow up.

The overall performance here is generally worse than normal buses. The average time that commuters spend on the bus also suffers since the buses expend more time at each bus stop before they get off at their respective destinations. This is one drawback of the holding strategy where the system gets slowed down, which is why a no-boarding strategy is arguably superior in a busy period.

Oddly enough for the holding strategy, this time in the busy period, the buses learn opposite actions when they bunch with $\Delta\theta = 0^\circ$. Here, they have to learn to unbunch deliberately because the busy period would otherwise keep them persistently bunched.

4.2.3. Frequency Detuning during Lull Period. Figure 8 shows the results of these two buses with frequency detuning undergoing reinforcement learning, corresponding to the graphs in the previous figures. For the first time in a lull period, we find a way to improve the average waiting time of commuters, by means of a holding strategy. However, the cost involved is that commuters would spend more time on the bus, on average, since the way the holding strategy works in keeping the buses staggered is by delaying the fast bus to the extent of being as slow as the slow bus. The average number of passengers on the fast bus is closer to that on the slow bus when the holding strategy is implemented, as compared to the normal buses where one bus consistently picks up more passengers than the other.

In spite of increasing the average time spent on bus and the average total travel time, perhaps the holding strategy may be viewed as viable since it is arguably less of a pain point to be on the bus enjoying the air conditioner compared to being out at the open bus stop where it may be hot under the blazing sun, wet during a thunderstorm, or even chilly during winter (in countries with four seasons).

We summarise the qualitative performance of the no-boarding and holding strategies for each of the setups that we have discussed in Table 1. Quantitative percentage improvement (or worsening) will be given in the next section where we consider a system with $N = 6$ buses serving $M = 12$ bus stops in a loop.

4.3. Combined No-Boarding and Holding Strategies. The third situation is where buses are given the choices to *stay* or *leave* whenever they are at a bus stop and everybody who wishes to alight has done so. Here, “somebody wants to board” and “nobody wants to board” are distinct. Therefore, we take these situations as a combination of the first two situations where situation 1 occurs when there is somebody who wants to board and situation 2 occurs when nobody wants to board.

Note that since situation 2 can only occur after everybody at the bus stop has been picked up, if the bus leaves when somebody is still there, then that is the end for this round at the bus stop and situation 2 is completely side-stepped. After 1000 episodes of training, we find that the buses’ Q-tables for situation 1 are trained but those for situation 2 are not. To allow for a fair amount of training for the latter Q-tables, we implement the following additional exploration possibility: in the first 200 episodes, if a bus chooses to *leave* when there is somebody to pick up, then there is a probability of $Y = 0.9$ that it switches to *stay*. From the 201st to 500th episode, Y is linearly decayed from 0.9 to 0. A reasonably high value of Y is necessary to expose the buses to situation 2 for training, because, for example, if there are 10 passengers at the bus stop, then a bus needs to choose 10 consecutive *stay* actions before it has the chance to encounter and train for situation 2.

Figure 9 summarises the results for the three setups (a), (b), and (c) listed right at the beginning of this section (before the start of Section 4.1) corresponding to having an identical natural frequency, frequency detuning in the busy

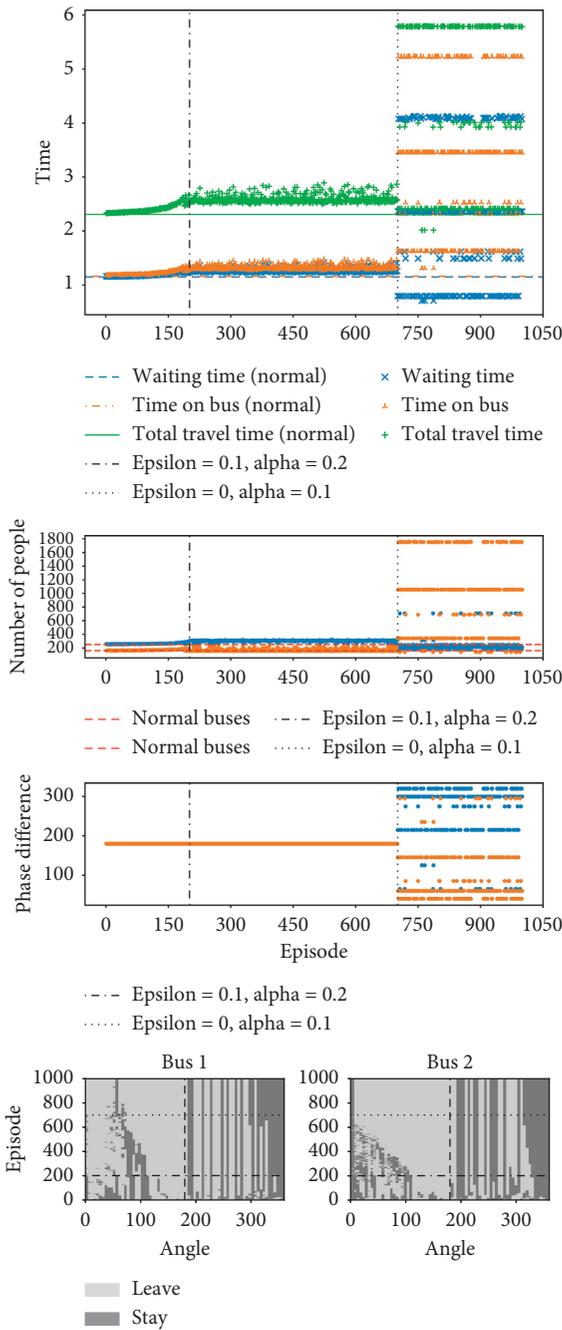


FIGURE 7: Two buses with frequency detuning serving a loop of bus stops during a busy period learn the holding strategy by reinforcement learning. The entire system is slowed down greatly and generally performs worse than normal buses.

as well as in the lull periods, respectively. We find that the buses are indeed able to learn both Q-tables such that each Q-table resembles that in the corresponding situations in Sections 4.1 and 4.2, with some minor differences that account for the fact that the buses can decide on *stay* or *leave* in two different situations 1 and 2. In terms of the performance, the graphs near episode ~ 200 are primarily dominated by the holding strategy (recall that here, the performance is as good as fully exploiting even though $\epsilon = 0.1$), whilst those

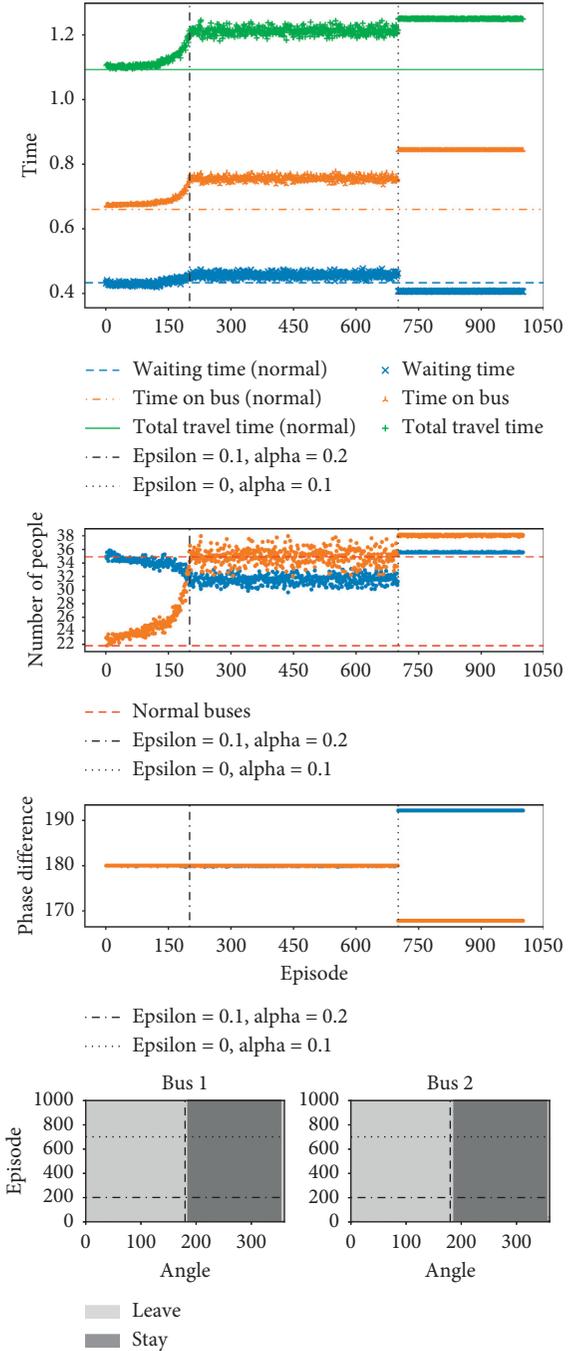


FIGURE 8: Two buses with frequency detuning serving a loop of bus stops during a lull period learn the holding strategy by reinforcement learning. It is able to slightly improve the average waiting time, at the expense of slowing down the fast bus such that the average time spent on bus and average total travel time are increased.

approaching episode ~ 500 are primarily dominated by the no-boarding strategy (recall that here, the performance is not as good as fully exploiting since $\epsilon = 0.1$ induces the bus to *leave* when it should not, leaving the passengers behind to unnecessarily wait for the next bus). The performance transitions between that of holding to no-boarding somewhere between episodes 200 to 500 as γ decays from 0.9 to 0.

TABLE 1: Qualitative performance of the no-boarding and holding strategies, respectively, in various setups of $N = 2$ buses serving $M = 12$ bus stops in a loop.

Strategy	No-boarding	Holding
Identical frequency	Positive	Positive
Frequency detuning, busy	Positive	Negative, Q-tables unable to completely train
Frequency detuning, lull	Negative	Improves waiting time, but adds time spent on bus

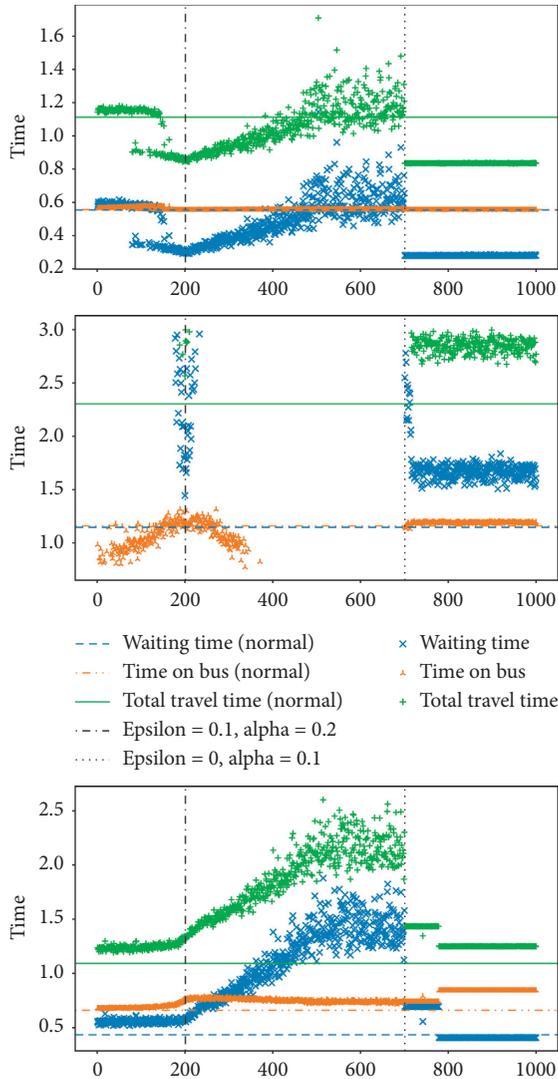


FIGURE 9: $N = 2$ buses serving $M = 12$ bus stops in a loop, where a bus takes an action at a bus stop when nobody wants to alight. Top: identical natural frequency, middle: frequency detuning during the busy period, and bottom: frequency detuning during the lull period. All three graphs share the same legend.

Finally, in the last 300 episodes, where there is no longer any exploration $\epsilon = 0$, the bus system settles into a most optimal strategy that they have acquired from the possible combinations.

In the case with identical natural frequency, the buses harness both the no-boarding and holding strategies where a bus would implement no-boarding if it is “too slow” ($\Delta\theta < 360^\circ/N$) and implement holding if it is “too fast” ($\Delta\theta > 360^\circ/N$). For the busy period with frequency detuning,

however, since the buses are not able to train the Q-tables corresponding to holding, it does not actually execute the holding strategy properly. This results in the extended waiting times, similar to what happened with the holding strategy alone during a busy period in Section 4.2.2.

For the two buses with frequency detuning during the lull period, it turns out that the buses perform as good as the holding strategy in terms of improving the average waiting time of commuters at the bus stop for a bus to arrive. The buses are able to learn that the no-boarding strategy, when applied by the slow bus (bus 2) in an attempt to speed it up, would result in it not picking up sufficient passengers and nullify its whole purpose of serving the loop. Since the frequency detuning is too large compared to the demand level, the no-boarding policy alone cannot speed it up, as we have seen in Section 4.1.3. Here, with both the options to *stay* and *leave* when there is somebody as well as nobody who wants to board, the buses try to harness both no-boarding and holding strategies, such that a bus would implement no-boarding if it is “too slow” ($\Delta\theta < 360^\circ/N$) and implement holding if it is “too fast” ($\Delta\theta > 360^\circ/N$). However, the buses realise that the combination of no-boarding and holding is not the most optimal way to go, since bus 2 would be leaving with few passengers when it implements no-boarding and relatively little gain in speeding up. Eventually somewhere close to the 800th episode, the slow bus decides that it is better to forget about no-boarding even if $\Delta\theta < 360^\circ/N$, and the bus system relies entirely on the fast bus (bus 1) to implement the holding strategy. The performance then matches with the purely holding strategy presented in Section 4.2.3, with identical improvement in average waiting time, and identical increases in average time spent on bus as well as average total travel time.

In summary, given both possibilities of situations 1 and 2, the bus system is able to find the relevant most optimal strategy depending on the particular conditions. For example, with identical natural frequency, they harness both the no-boarding and holding strategies to the fullest. On the other hand, with frequency detuning in the lull period, they revert to the holding strategy and ditch the no-boarding policy.

5. Any N Buses Serving M Bus Stops in a Loop

This framework is directly generalisable to any N buses serving M bus stops in a loop. We have carried out more simulations with $N = 3$ buses and even $N = 6$ buses, respectively. A system with many buses generally produces qualitatively similar results to those already discussed for the case with $N = 2$ buses.

Table 2 summarises the quantitative performance (in terms of average waiting time for a bus to arrive at a bus stop) for these $N = 6$ buses serving $M = 12$ bus stops under various conditions and setups. The setups with identical natural frequency are given $k = 0.010$, busy with frequency detuning is given $k = 0.063$, and lull with frequency detuning is given $k = 0.010$. When there is frequency detuning, the six buses are prescribed different natural frequencies, selected within the range of 12 minutes to 18 minutes (excluding time stopped at bus stops).

Here are some noteworthy points in the *lull period* where the buses have *frequency detuning*:

- (1) Holding positively improves the average waiting time by 27.6% whereas no-boarding backfires and lengthens by 66.7%. Nevertheless, holding would increase the average time spent on bus by 24.4% (from $0.624T$ to $0.776T$) such that the average total travel time would increase by 13.0% (from $0.798T$ to $0.902T$).
- (2) A combination of no-boarding and holding positively improves the average waiting time by 27.6%, matching the savings due to pure holding. Nevertheless, this combination would increase the average time spent on bus by 20.4% (from $0.624T$ to $0.751T$) such that the average total travel time would increase by 9.9% (from $0.798T$ to $0.877T$).

With more buses, the ability to keep buses staggered over the loop significantly divides off the average waiting time at the bus stop for a bus to arrive. The expense incurred with the increase in average time spent on bus and also the average total travel time by implementing the holding strategy during the lull period seems to be well worth it, when described in terms of percentages. Apart from that, the combined strategies save as much time as holding on the average waiting time whilst incurring *less cost on average time spent on bus (and average total travel time)*. Hence, we see a further improvement thanks to a combination of no-boarding and holding strategies during the lull period for this system with $N = 6$ buses.

6. Discussion and Concluding Remarks

The use of reinforcement learning for a bus system serving a loop of bus stops has shown the potential of discovering strategies to optimise the performance of the system. The framework employed in this paper takes advantage of the phase difference between buses in a loop (also applicable to nonloop routes) where maintaining a staggered configuration translates to minimising the average waiting time of commuters at the bus stops for a bus to arrive. This provides a way to deal with the high variance of the individual waiting times that causes convergence of the Q-table to be essentially impossible within reasonable simulation time (or perhaps not even possible in some cases [45]).

The system has learnt that no-boarding and holding strategies are indeed both useful strategies when the buses have an identical natural frequency. No-boarding speeds up the slower bus whilst holding slows down the faster bus. The

former is also useful when buses have frequency detuning during the busy period, but the latter may slow down the system too much. Nevertheless, the holding strategy is salutary in the lull period where the fast bus is slowed down to match the slow bus in order to maintain a reasonably staggered configuration, at the expense of increasing time spent on bus and total travel time. This offers a solution during the lull period, where the no-boarding strategy simply does not work at all. Incidentally, whilst the no-boarding strategy is arguably disruptive to passengers who need service urgently, Reference [35] has shown that allowing passengers the options of cooperating or defecting turns out to be a viable way of implementing such a no-boarding strategy in real bus systems.

It is interesting to note that although the buses are given *low-level actions* of *stay* or *leave* at the bus stop when nobody wants to alight—essentially *knowing only the rules of the game*, reinforcement learning leads to the discovery of *high-level strategies* of no-boarding [34, 35] and holding [3–8, 10–15, 18, 19]. This illustrates the utility of a reinforcement learning framework where the system is able to arrive at high-level strategies without human presumptions and priors, like how the *AlphaZero* programme [46] can come up with and even validate known human strategies and tactics (e.g., the Berlin defence against the Ruy Lopez in Chess), discrediting some of them (e.g., the French Defence in Chess, apparently) and even revealing new possibilities (e.g., sacrificing multiple pawns and pieces in favour of long-term subtle activity in Chess—highly impressing many Chess Grandmasters, including a former World Champion [47, 48]).

In particular, these intelligent buses are able to behave cooperatively to unbunch in unique and interesting ways like learning opposite actions in the case of no-boarding, as well as one bus just holding to allow the other to correct their phase difference. They also discover useful strategies with the appropriate bounds where no-boarding and holding are implemented. These emergent behaviours arise from the ability of the buses to learn and improve from their interactions, eventually settling into some collectively optimal strategies. On top of that, the system also makes use of combining the options appropriately in various setups. This is important when we move on to nonstationary environments where the system *must* encounter various situations and be able to act with an optimal strategy.

Being low level, however, implies that the bus system does not actually “know” that it can “choose to implement a no-boarding strategy or a holding strategy” at will. All that it cares is a bus is at a bus stop, nobody wants to alight. Is there anybody who wants to board? If yes, then should it *stay* or *leave*? If not, then should it *stay* or *leave*? Since it typically encounters somebody who wants to board, if it *leaves*, then it will not encounter the latter situation where nobody wants to board—thus sidestepping the holding option. In order to allow for a balanced combination between a no-boarding strategy and a holding strategy, we have augmented the exploration phase with a new hyperparameter Υ . Alternatively, perhaps a different approach with this framework to be *higher level* would allow for faster convergence of the

TABLE 2: Quantitative performance of the no-boarding strategy, holding strategy, and a combination of them, respectively, in various setups of $N = 6$ buses serving $M = 12$ bus stops in a loop.

	Average waiting time ($T = 12$ minutes)	Percentage change
No-boarding		
Identical frequency	0.511 (normal) to 0.180	Improves by 64.8%
Frequency detuning, busy	0.793 (normal) to 0.340	Improves by 57.1%
Frequency detuning, lull	0.174 (normal) to 0.290	Aggravates by 66.7%
Holding		
Identical frequency	0.511 (normal) to 0.086	Improves by 83.2%
Frequency detuning, busy	0.793 (normal) to 0.160	Improves by 79.8%
Frequency detuning, lull	0.174 (normal) to 0.126	Improves by 27.6%
Combined		
Identical frequency	0.511 (normal) to 0.180	Improves by 64.8%
Frequency detuning, busy	0.793 (normal) to 0.300	Improves by 62.2%
Frequency detuning, lull	0.174 (normal) to 0.126	Improves by 27.6%

Q-table. In other words, when a bus is at a bus stop, it is “conscious” about the options on (a) behaving like a normal bus; (b) *leave*—implement no-boarding; or (c) *stay longer*—implement holding. This therefore places the options for no-boarding and holding on an equal footing, alongside behaving normally, and alleviates the bias towards implementing no-boarding over holding. Nevertheless, we have shown here that these low-level actions do lead to the high-level no-boarding and holding strategies, in the situations where somebody wants to board and nobody wants to board, respectively. This establishes the mechanisms on how low-level actions lead to the emergence of high-level coordinated strategies of the buses. With higher-level actions, the faster convergence becomes a crucial utility for being adaptive in nonstationary environments of the real world.

Thus far, this paper assumes that all $M = 12$ bus stops are perfectly staggered around the loop and all have the same rate of people arrival, s . Whilst seemingly simplified, this represents an important first step in a series of increasingly complex progression for our research on the bus system undergoing reinforcement learning. In particular, we have established and clarified the behaviour of the bus system with identical natural frequency, as well as with frequency detuning in the busy and lull periods. Each setup has distinct characteristics of its own and the appropriate strategy should be applied especially if there is frequency detuning, viz. no-boarding during busy and holding during lull.

A step forward would be to generalise the environment based on real data that we have collected in [39], to investigate how the bus system may arrive at novel and even adaptive strategies to deal with nonstationary environments where passengers may wish to head towards some *hubs* at certain times of the day, with some bus stops having higher rates of people arrival, i.e., s_i for each $i = 1, 2, \dots, M$. The framework in this paper serves as a good platform for greater layers of complexity to be piled up on the environment. Eventually, we could then implement such strategies to our Nanyang Technological University campus shuttle bus service upon where this environment is modelled after [39, 44, 45], and subsequently even adapt to more complex bus routes.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the Joint WASP/NTU Programme (Project no. M4082189) and the DSAIR@NTU Grant (Project no. M4082418).

References

- [1] P. Ball, “There’s chaos on the buses,” *Nature*, 2003.
- [2] T. Nagatani, “Fluctuation of riding passengers induced by chaotic motions of shuttle buses,” *Physical Review E*, vol. 68, no. 3, Article ID 036107, 2003.
- [3] M. Abkowitz and I. Engelstein, “Methods for maintaining transit service regularity,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 961, p. 1, 1984.
- [4] M. D. Rossetti and T. Turitto, “Comparing static and dynamic threshold based control strategies,” *Transportation Research Part A: Policy and Practice*, vol. 32, no. 8, pp. 607–620, 1998.
- [5] X. J. Eberlein, N. H. M. Wilson, and D. Bernstein, “The holding problem with real-time information available,” *Transportation Science*, vol. 35, no. 1, pp. 1–18, 2001.
- [6] M. D. Hickman, “An analytic stochastic model for the transit vehicle holding problem,” *Transportation Science*, vol. 35, no. 3, pp. 215–237, 2001.
- [7] L. Fu and X. Yang, “Design and implementation of bus-holding control strategies with real-time information,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1791, no. 1, pp. 6–12, 2002.
- [8] Y. Bin, Y. Zhongzhen, and Y. Baozhen, “Bus arrival time prediction using support vector machines,” *Journal of Intelligent Transportation Systems*, vol. 10, no. 4, pp. 151–158, 2006.
- [9] N. Mukai, T. Watanabe, and J. Feng, *Route Optimisation Using Q-Learning for On-Demand Bus Systems*, I. Lovrek, R. Howlett, and L. Jain, Eds., Springer, Berlin, Germany, 2008.
- [10] C. F. Daganzo, “A headway-based approach to eliminate bus bunching: systematic analysis and comparisons,”

- Transportation Research Part B: Methodological*, vol. 43, no. 10, pp. 913–921, 2009.
- [11] C. E. Cortés, D. Sáez, F. Milla, A. Núñez, and M. Riquelme, “Applications of Advanced technologies in transportation: selected papers from the 10th AATT conference,” *Transportation Research Part C: Emerging Technologies*, vol. 18, p. 757, 2010.
- [12] O. Cats, A. N. Larijani, H. N. Koutsopoulos, and W. Burghout, “Impacts of holding control strategies on transit performance,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2216, no. 1, pp. 51–58, 2011.
- [13] C. Gershenson, “Self-organization leads to supraoptimal performance in public transportation systems,” *PLoS One*, vol. 6, p. 1, 2011.
- [14] J. J. Bartholdi and D. D. Eisenstein, “A self-coordinating bus route to resist bus bunching,” *Transportation Research Part B: Methodological*, vol. 46, no. 4, pp. 481–491, 2012.
- [15] C. Chen, W. Chen, and Z. Chen, “A multi-agent reinforcement learning approach for bus holding control strategies,” *Advances in Transportation Studies, Special Issue*, vol. 2, p. 41, 2015.
- [16] O. Ibarra-Rojas, F. Delgado, R. Giesen, and J. Munoz, “Planning, operation, and control of bus transport systems: a literature review,” *Transportation Research Part B: Methodological*, vol. 77, pp. 38–75, 2015.
- [17] W. Chen, K. Zhou, and C. Chen in *Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 100 (2016).
- [18] L. Moreira-Matias, O. Cats, J. Gama, J. Mendes-Moreira, and J. F. de Sousa, “An online learning approach to eliminate bus bunching in real-time,” *Applied Soft Computing*, vol. 47, pp. 460–482, 2016.
- [19] P. Wang, X. Chen, W. Chen, L. Cheng, and D. Lei, “Provision of bus real-time information: turning passengers from being contributors of headway irregularity to controllers,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2672, no. 8, pp. 143–151, 2018.
- [20] F. Alesiani and K. Gkiotsalitis, “Reinforcement learning-based bus holding for high-frequency services,” in *Proceedings of the 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018.
- [21] K. Menda, Y.-C. Chen, J. Grana et al., “Deep reinforcement learning for event-driven multi-agent decision processes,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1259–1268, 2019.
- [22] Y. Li, J.-M. Rousseau, and M. Gendreau, “Real-time scheduling on a transit bus route: a 0-1 stochastic programming mode,” in *Proceedings of the thirty-third annual meeting, Transportation Research Forum*, 157 (1991).
- [23] X. J. Eberlein, “Real-time control strategies in transit operations: models and analysis,” Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA, Ph.D. dissertation, 1995.
- [24] L. Fu, Q. Liu, and P. Calamai, “Real-time optimization model for dynamic scheduling of transit operations,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1857, no. 1, pp. 48–55, 2003.
- [25] A. Sun and M. Hickman, “The real-time stop-skipping problem,” *Journal of Intelligent Transportation Systems*, vol. 9, no. 2, pp. 91–109, 2005.
- [26] L. Li, K. Yang, Z. Li, and Z. Zhang, “The optimality condition of the multiple-cycle smoothed curve signal timing model,” *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 46–57, 2013.
- [27] P. G. Furth, “Alternating deadheading in bus route operations,” *Transportation Science*, vol. 19, no. 1, pp. 13–28, 1985.
- [28] P. Furth and F. Day, “Transit routing and scheduling strategies for heavy demand corridors,” *Transportation Research Record*, vol. 1011, p. 23, 1985.
- [29] X. J. Eberlein, N. H. M. Wilson, C. Barnhart, and D. Bernstein, “The real-time deadheading problem in transit operations control,” *Transportation Research Part B: Methodological*, vol. 32, no. 2, pp. 77–100, 1998.
- [30] F. Delgado, J. C. Muñoz, R. Giesen, and A. Cipriano, “Real-time control of buses in a transit corridor based on vehicle holding and boarding limits,” *Transportation Research Record*, vol. 2090, pp. 59–67, 2009.
- [31] F. Delgado, J. C. Munoz, and R. Giesen, “How much can holding and/or limiting boarding improve transit performance?” *Transportation Research Part B: Methodological*, vol. 46, no. 9, pp. 1202–1217, 2012.
- [32] S. Z. Zhao, C. X. Lu, S. D. Liang, and H. S. Liu, “A self-adjusting method to resist bus bunching based on boarding limits,” *Mathematical Problems in Engineering*, no. 1, , 2016.
- [33] W. Sun and J.-D. Schmöcker, “Considering passenger choices and overtaking in the bus bunching problem,” *Transportmetrica B: Transport Dynamics*, vol. 6, no. 2, pp. 151–168, 2018.
- [34] V.-L. Saw and L. Y. Chew, “No-boarding buses: Synchronisation for efficiency,” *PLoS One*, vol. 15, no. 3, Article ID e0230377, 2020.
- [35] V.-L. Saw and L. Y. Chew, “No-boarding buses: agents allowed to cooperate or defect,” *Journal of Physics: Complexity*, vol. 1, no. 1, Article ID 015005, 2020.
- [36] S. Jara-Díaz and A. Tirachini, “Urban bus transport: open all doors for boarding,” *Journal of Transport Economics and Policy*, vol. 47, p. 91, 2013.
- [37] C. Stewart and A. El-Geneidy, “All aboard at all Doors,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2418, no. 1, pp. 39–48, 2014.
- [38] A. El-Geneidy, D. van Lierop, E. Grisé et al., “Get on board: assessing an all-door boarding pilot project in Montreal, Canada,” *Transportation Research Part A: Policy and Practice*, vol. 99, pp. 114–124, 2017.
- [39] V.-L. Saw, N. N. Chung, W. L. Quek, Y. E. I. Pang, and L. Y. Chew, “Bus bunching as a synchronisation phenomenon,” *Scientific Reports*, vol. 9, p. 6887, 2019.
- [40] L. Y. Chew, V.-L. Saw, and Y. E. I. Pang, “Stability of anti-bunched buses and local unidirectional kuramoto oscillators (to appear in a book chapter from the 15th international conference on dynamical systems-theory and application in Łódź,” 2020, <https://arxiv.org/abs/1912.06470>.
- [41] A. Pikovsky, M. Rosenblum, and J. Kurths, *Synchronization: A Universal Concept in Nonlinear Sciences, Cambridge Nonlinear Science Series*, Cambridge University Press, Cambridge, UK, 2003.
- [42] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, A Bradford Book, Cambridge, MA, USA, 2018.
- [43] S. Begum, “Need for higher shuttle bus frequency in the evenings: survey,” *The Nanyang Chronicle*, vol. 24, no. 9, p. 2, 2018.
- [44] *Nanyang Technological University Media Releases, NTU Singapore and Volvo Unveil World’s First Full Size, Autonomous Electric Bus*, Nanyang Technological University Media Releases, Singapore, 2019, <https://media.ntu.edu.sg/NewsReleases/Pages/newsdetail.aspx?news=42801270-1948-431e-9502-1c725d513c83>.
- [45] Y. Sato, E. Akiyama, and J. D. Farmer, “Chaos in learning a simple two-person game,” *Proceedings of the National Academy of Sciences*, vol. 99, no. 7, pp. 4748–4751, 2002.

- [46] D. Silver, T. Hubert, J. Schrittwieser et al., “A general reinforcement learning algorithm that masters chess, shogi, and go through self-play,” *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.
- [47] G. Kasparov, “Chess, a drosophila of reasoning,” *Science*, vol. 362, no. 6419, p. 1087, 2018.
- [48] M. Sadler and N. Regan, *Game Changer: AlphaZero’s Groundbreaking Chess Strategies and the Promise of AI*, New in Chess, Amsterdam, Netherlands, 2019.