Modeling of Bus Holding Strategy in Public Transit Systems with Multi-Agent Reinforcement Learning

Chunxiao Chen*

School of Rail Transit transportation, Hunan Railway Professional Technology College, Zhuzhou 412001, China

**Corresponding Author.*

Abstract:

Excessive fluctuations in travel time between stops and demand at bus stops during bus operations can lead to operational instability in bus systems, such as bus bunching. To tackle this issue, this paper presents a dynamic bus holding control strategy leveraging multi-agent reinforcement learning to stabilize bus system operations and prevent bus bunching. First, bus motion system is constructed, and the rules for bus operation and passenger behavior are defined. Then, agent-based transit operation management is established, the elements of the multi-agent reinforcement learning framework are outlined, and a centralized training and decentralized execution method is proposed. Additionally, an event-driven simulation environment is developed for training and testing the agents. Finally, extensive numerical simulations are conducted to evaluate the proposed method against baseline approaches using various performance metrics. The results demonstrate that the proposed method effectively captures the dynamics of the bus system and accounts for the long-term impacts of current decisions, resulting in the most balanced bus trajectories, optimal passenger load distribution, and minimal total holding time.

Keywords: bus bunching, bus holding, multi-agent reinforcement learning, hysteretic Q-learning

INTRODUCTION

As a vital component of urban public transportation, bus systems offer significant convenience for residents' daily travel while serving as effective tools for energy conservation, emission reduction, traffic congestion alleviation, and air quality improvement. However, compared to rail-based transportation systems like subways that operate underground, bus systems are more vulnerable to surface-level environmental factors such as road congestion, adverse weather, and unforeseen incidents. These challenges contribute to operational instability, leading to issues like bus bunching. Unstable bus operations not only increase passengers' waiting times, diminishing the appeal of public transportation, but also escalate operating costs for bus service providers. To address these issues, implementing effective control strategies in bus systems is essential to optimize operations and enhance their attractiveness as a sustainable and green travel option. Significant research efforts have been devoted to developing flexible, real-time control strategies, including holding strategies, stop-skipping approaches [1-5], short-turning [6-9], speed adjustments [10-12], boarding limits to manage dwell times [13,14], traffic signal priority [15,16], and bus substitution methods [17,18]. However, some of these strategies face practical challenges that limit their widespread adoption and application. For example, stop skipping increases waiting times for passengers intending to board or alight at skipped stations, whereas short-turning forces passengers to disembark and wait for the next bus. Speed adjustments can be ineffective when the required speeds exceed safety limits, and traffic signal priority strategies often negatively impact general traffic flow. Despite the rapid development of new solutions, bus holding has emerged as the most widely studied control strategy in recent years. This approach involves holding buses at specific stops for a designated period, which not only reduces passenger frustration but also enhances the maneuverability of transit operations [19].

Bus holding control strategies have been extensively studied and are generally classified into two categories: static control, which is relies on a fixed headway or schedule, and dynamic control, which functions without a predefined headway or schedule. The static approach is typically used in transit services with low passenger demand and longer headways to align operations with planned schedules. In contrast, dynamic control is more commonly applied to high-frequency bus or transit lines. This paper primarily reviews the literature related to the dynamic control approach. Early research primarily focused on the establishment and verification of bus holding control models aimed at simple transit systems [20-22]. Multi-source public transportation big data, derived from bus IC cards, GPS/Beidou positioning systems, smartphone apps, web platforms, sensors, and video detection, possesses high levels of continuity, completeness, and timeliness. They has facilitated the incorporation of real-time transit system data into academic research by numerous scholars. An earlier study on real-time strategies for the holding problem was conducted by Eberlein et al., who formulated the issue as a quadratic programming problem and proposed an iterative heuristic to identify the optimal set of headways [19]. An adaptive control scheme to mitigate the bus bunching problem was analyzed by Daganzo, who proposed a control strategy utilizing dynamic holding times informed by real-time headway data. [23]. A model was employed that utilized multiple control points to maintain headways as close as possible to the desired values, thereby minimizing the need for significant adjustments. However, because the buses in this study were designed to respond only to disturbances occurring ahead and not those behind, the model was found to be ineffective in scenarios involving substantial disruptions. A model that adjusts the bus cruising speed in real-time based on the forward and backward headways to alleviate the large disturbance problem was proposed by Daganzo and Pilachowski [11]. Building on Daganzo's (2009) work, Xuan et al. proposed a family of dynamic holding strategies incorporating a virtual schedule into the model, offering the advantage of applicability to high and low frequency transit lines. [24]. The above study was supplemented from forward headway control which the every consecutive buses are dynamically self-equalizing [25]. Matthias Andres extended current bus holding control frameworks to include prediction data [26]. The majority of methods proposed in previous research predominantly rely on centralized schemes. In these approaches, the actions of all buses are determined by a single central unit using the available information. This setup allows the system to coordinate and optimize the actions of multiple buses simultaneously, facilitating their synchronization. However, this approach requires potentially heavy computational resources to solve a centralized optimization problem, which can limit the number of buses that can be managed effectively. Additionally, Public transit systems are particularly susceptible to external factors such as road congestion, weather conditions, and unexpected incidents, resulting in significant instability of passenger demand and travel time, which makes the optimization of bus holding problems using real-time information progressively more complex and challenging. Moreover, real-time data, such as bus position, speed, dwell time, and passenger demand—is often influenced by external factors like weather, the terrain of the transit route, or the performance of the equipment itself. In this context, conventional computing models exhibit significant limitations, including poor scalability, low fault tolerance, and vulnerability to single points of failure when alternative system components are unavailable. Furthermore, maintaining robust transmission links to monitor all buses in real time is not only costly but also difficult, adding to the challenges of achieving a reliable system.

With recent advances in artificial intelligence, particular in deep reinforcement learning, have provided new insights into addressing complex real-time control problems. By integrating the power of deep learning and reinforcement learning, these advancements have shown great potential for solving various transportation operation challenges [27]. Clearly, it extends single-agent systems to multi-agent frameworks, enabling decentralized decision-making and learning, which makes them highly applicable for addressing complex real-time problems with real-life data. [28]. Chen et al. proposed bus holding model based on MARL [29, 30]. Wang Jiawei systematically studied bus fleet control with MARL framework [31–34]. However, there are still some limitations to be addressed. First, in the model, the backward headway data must be predicted before it can be utilized, which affects the accuracy of the control method. Second, the coordination between long-term and short-term profit objectives has not been adequately resolved. In recent years, there has been a surge in research on multi-agent reinforcement learning. New algorithms and coordination mechanisms have shown promise in addressing these challenges. This paper proposes a MARL framework for designing a dynamic bus holding control system that leverages a virtual schedule to account for both immediate and long-term rewards. The primary contributions of this study are outlined as follows:

- A MARL framework is presented to implement dynamic bus holding control strategies, featuring carefully designed elements such as states, action sets, reward functions, a multi-agent system, the physical and operational constraints governing bus motion, coordination mechanism.

- The virtue schedule is introduced into model as implicit coordination mechanism among agents, and also the hysteretic Q-learning algorithm is designed to train by joint agent efficiently and execute by each independent learner.

- To demonstrate the advantage of the proposed MARL framework, transit simulator is developed for the comparative experiment.

PRELIMINARIES

Agent-based Transit Operation Management System

The system modeled in this study represents a one-way loop route consisting M homogenous buses and K stops (see Figure 1). The buses depart from the starting stop (referred to as Stop 1) at fixed headways of H, and proceed sequentially through all downstream stop $(2,3,\cdots,K)$. It is stipulated that all passengers on board must alight at stop k+1, after which a new bus service trip will commence from Stop 1. All operating buses run sequentially in the order of 1 to n, with overtaking strictly prohibited. The variables and parameters used in the model are detailed in Table 1.

Table 1. Notation

Notation	Meaning
i	Bus index, $i=(1,2,\cdots M)$
k	Stop index, $k=(1,2,\cdots K)$
$t_{i,k}$	Scheduled arrival time bus i at stop k
$ta_{i,k}$	Actual arrival time of bus i at stop k
$th_{i,k}$	Actual holding time of bus i at stop k
$td_{i,k}$	Actual departure time of bus i from stop k
$L_{i,k-1}$	Number of onboard passengers for bus i upon departing from stop k−1.
d_k	Amount of slack time in the virtual schedule at stop k
$A_{i,k}$	Number of passengers alighting from bus i at stop k
$B_{i,k}$	Number of passengers boarding bus i at stop k
α_0,β_0	Parameter defining the cost of passengers alighting and boarding
λ_k	Passenger arrival rate at stop k
$w_{i,k}$	Passenger waiting time of bus i for boarding and alight at stop k, $w_{i,k}=\max(\alpha_0\cdot A_{i,k},\beta_0\cdot B_{i,k})$
H	Scheduled headway to meet the transit system's demand during a specific period
$h_{i,k}$	Actual headway of bus i and bus i-1 when bus i arrives at stop k
c_k	Average travel time between stop k-1 and stop k
$v_{i,k+1}$	Stochastic variability in the travel time of bus i between stop k and stop k+1

The common control architectures of agent-based systems are generally classified into three types: hierarchical, heterarchical, and hybrid [35]. The hierarchical approach divides the overall system into smaller subsystems with minimal interaction between them. In contrast, the heterarchical approach adopts a fully decentralized structure. The hybrid method integrates the advantages of both. This study employs a partial dynamic hierarchical control architecture, where multiple agents are dynamically organized based on task decomposition. To facilitate this dynamic organization, heterogeneous agents are grouped into virtual clusters as needed. The proposed agent-based transit operation management system is structured into three layers, as shown in Figure 2: (1) Lowest Layer: Contains data agents and user interface agents. (2) Middle Layer: Comprises bus agents. (3) Highest Layer: Hosts agents operating within the transit control center server, including the coordination agent, reinforcement learning agent, and agent management system. The system employs six distinct types of agents, which work collaboratively to manage operations effectively.

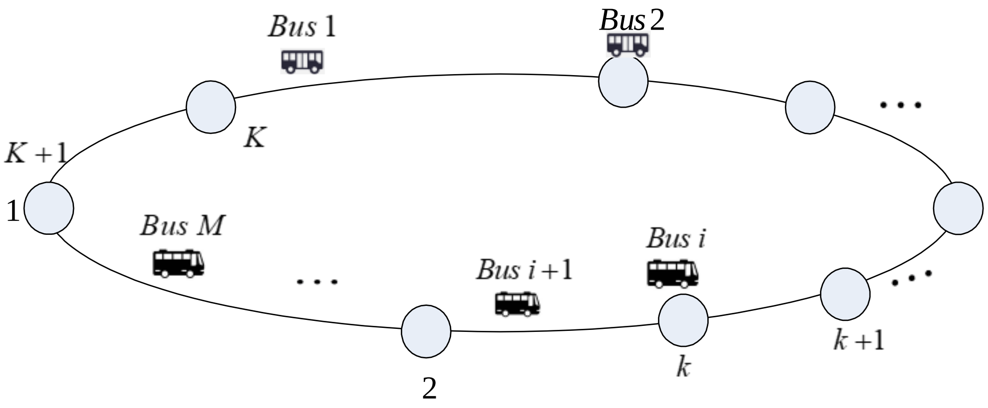


Figure 1. Representation of transit system

- Bus agent

Agents: one agent for each bus operated on the single fixed transit route.

Roles: sensing state variables from the transit operation environment; making autonomous decisions according to the base; using actions that affect the environment to control the bus

Interactions: communicating and cooperating with other bus agents, coordination agent, reinforcement learning agent, user interface agent, and data agent.

- Agent management system

Agents: single agent.

Roles: Overseeing the life cycle of agents within the system, including their creation, registration, retirement, migration, and maintenance.

- Coordination agent

Agents: single agent serving in the transit control centre server.

Roles: coordinating the bus agents’ control actions.

Interaction: cooperating with bus agents whose events are triggered at the same time.

- Reinforcement learning agent

Agents: single agent for serving in the transit control centre server.

Roles: providing reinforcement learning algorithm for the bus agents.

Interaction: communicating with bus agents.

- User interface agent

Agents: one agent for interacting with users.

Roles: showing information to the users, such as decision makers and drivers; participating in interaction.

Interaction: communicating with the environment and users.

- Data agent

Agents: one agent for each device.

Roles: collecting the bus and stop state information.

Interaction: communicating with bus agents.

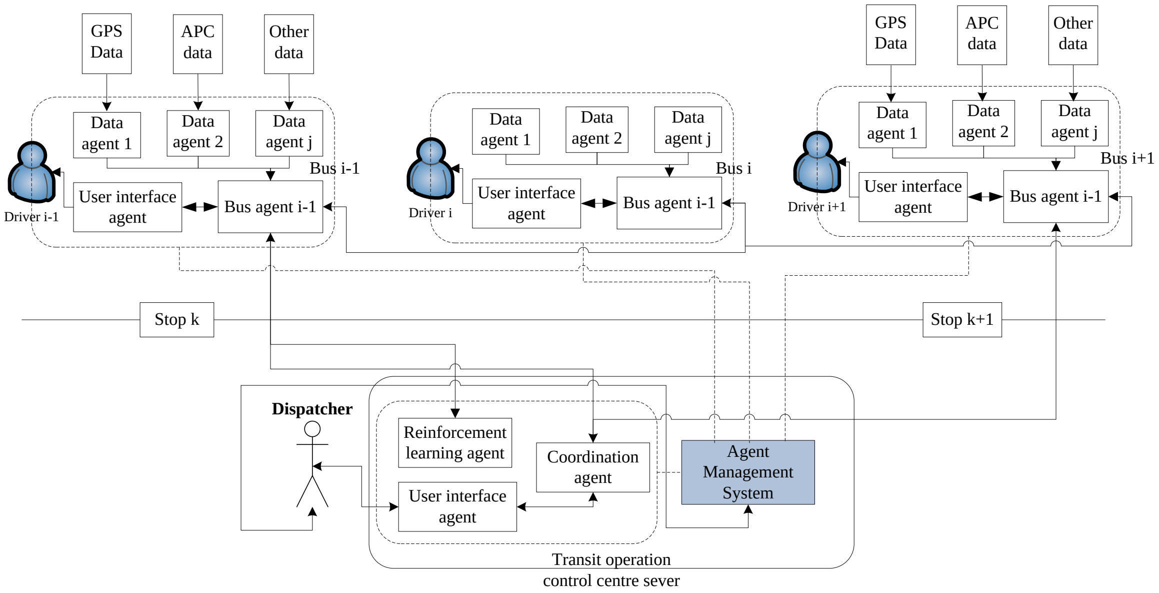


Figure 2. Architecture of the agent-based transit operation management system

Bus Motion

The bus motion system is characterized by the sequence of bus arrival times at successive control points. These arrival times are interconnected through the following equation, which accounts for the slack:

$$t_{i,k+1} = t_{i,k} + \lambda_k H + d_k + r_k \tag{1}$$

$$t_{i,k} - t_{i-1,k} = H \tag{2}$$

From the above equation, it can be seen that the virtue schedule is constructed by the assumed dwell time and the slack time, which is suitable for both high-frequency short and low-frequency long headway. The actual arrival times obey:

$$ta_{i,k+1} = ta_{i,k} + w_{i,k} + th_{i,k} + c_k + v_{i,k+1} \tag{3}$$

After the bus arrives at a stop, since most buses in cities have two doors for passengers to board and alight, the process of passengers alighting and the process of passengers boarding occur simultaneously. The dwell time of the bus is determined by the maximum of the time required for passengers to alight and the time required for passengers to board, as follows:

$$w_{i,k} = \max(\alpha_0 \cdot A_{i,k}, \beta_0 \cdot B_{i,k}) \tag{4}$$

When bus i departs from the starting stop k-1 and travels to a downstream stop k to serve passengers, its arrival time at the stop is given by:

$$ta_{i,k} = td_{i,k-1} + r_k \tag{5}$$

After bus i arrives at stop k, it will dwell a certain period to complete passenger boarding and alighting. Then, based on the agent's decision, the bus remains at the station for a specified duration before departing. The departure time is the sum of the dwell time and the bus holding time, as follows:

$$td_{i,k} = ta_{i,k} + w_{i,k} + th_{i,k} \tag{6}$$

The real-time control model for the bus motion system must comply with both physical and operational constraints, which are outlined as follows:

$$L_{i,k} = L_{i,k-1} + B_{i,k} - A_{i,k} \tag{7}$$

$$B_{i,k} = \lambda_k \cdot (td_{i,k} - td_{i-1,k}) \tag{8}$$

$$A_{i,k} = L_{i,k-1} \cdot q_k \tag{9}$$

$$td_{i,k} \le ta_{i+1,k} \tag{10}$$

$$th_{i,k} \geq 0 \tag{11}$$

$$td_{i,k} \geq 0 \tag{12}$$

$$ta_{i,k} \geq 0 \tag{13}$$

$$L_{i,k} \geq 0 \tag{14}$$

Constraint (7) establishes the relationship between the total numbers of onboard passengers and the number of passengers boarding and alighting. This balances the passenger flow and accounts for real-time updates. Constraints (8) and (9) outline the estimation methods for calculating the number of passengers alighting and boarding at a specific stop. Constraint (10) ensures that overtaking does not occur between consecutive buses. Constraints (11)–(14) ensure that all variables (such as passenger counts, times, and other parameters) are non-negative. This guarantees that the model operates within realistic and feasible conditions. These constraints collectively govern the bus transit system's operations, ensuring consistency, efficiency, and adherence to logical and practical requirements.

METHODOLOGY

Definition of Marl Elements

System states

The state of the transit system on a fixed route should reflect the time headway information on buses. The state of bus motion is thus defined as the degree of deviation between the planned arrival time in the virtue schedule and the actual arrival time at each stop. While bus arrive at control point k, the level of deviation from scheduled arrival time is given by

$$VSAR_{i,k} = \frac{(ta_{i,k} - t_{i,k})}{H} \times 100\% \tag{15}$$

In order to meet the requirement that the state space is finite in MARL model, value of the level of deviations should be discretized. The binning technique is used to group the virtual schedule adherence ratio values into bins with equal-width. Let $\phi_{\min}$ and $\phi_{\max}$ denote corresponding state of the lower and upper bound value of the level of deviation. Let G be the number of bins, which are numbered, 1 through G. Also the bin width is given by

$$\Delta_{\phi} = \frac{\phi_{\max} - \phi_{\min}}{G} \tag{16}$$

Then, the range of the gth bin is as shown in (17).

$$U_g = (\phi_{\min} + (g-2)\cdot\Delta_{\phi}, \phi_{\min} + (g-1)\cdot\Delta_{\phi}], g \geq 2 \tag{17}$$

Action set

Once a bus arrives at a station and completes passenger boarding and alighting, the bus holding time is determined based on the bus holding control strategy. The action set is defined such that the holding periods are represented as multiples of a fixed interval. Analytically,

$$A_i(t) = \theta_i \Delta_h, \theta_i \in Z^+, \Delta_h > 0 \tag{18}$$

This assumption simplifies both the mathematical formulation and the implementation of the solution algorithm. In the simulation experiments (Section 4), $\Delta_h = 20\,\mathrm{s}$ and $\theta_i \in \{0,1,2,3\}$. When $\theta_i = 0$, it signifies the immediate departure of the bus once passengers have completed boarding and alighting. Discrete holding intervals are employed primarily for operational practicality, simplifying the process for bus drivers to follow instructions from central dispatcher.

Reward function

The objective of control is to reducing bus bunching, which is regulating headway and achieve a long-term goal of reducing variance of headway. For sake of immediate reward definition in the MARL model, construction of the utility state function is proposed by making use of the normal probability density function. Once the $g(s_j)$

values are obtained, each state utility values is found from

$$f(g(s_j)) = \frac{1}{\sqrt{2\pi}} e^{-\frac{g^2(s_j)}{2}}, g(s_j) = \phi_{\min} + (j-1)\Delta_\phi + \frac{\Delta_\phi}{2} \tag{19}$$

Where g(sj) is mid-point value of interval corresponding to pseudo state .The higher values of this function indicate the nearer distance from the optimal state. Conversely, the lower values indicate further distance from the optimal state. Therefore, the immediate reward function is to measure the changes in deviation distance from current state si,k to next state si,k+1 under bus holding action ai, which is shown in Equation (20)

$$R_i(s_{i,k}, a_i, s_{i,k+1}) = M_1(\left\| f(g(s_{i,k+1}) - f(g(s^*)) \right\|_2 - \left\| f(g(s_{i,k}) - f(g(s^*)) \right\|_2) - M_2 e^{th_{i,k}} \tag{20}$$

Where M1 and M2 are equilibrium normalization coefficients represented by a larger positive integer, s^* is the optimal state, $\| \|_2$ is Euclidean norm of each state. The details of the reward computation process are provided in Algorithm 1.

Algorithm 1: Reward computing algorithm
1 Input $ta_{i,k}, t_{i,k}, ta_{i,k+1}, \Delta_\phi, \phi_{\min}, M_1, M_2\ H, S$;
2 Initialize: Coding with integral number according to the position of partition interval corresponding to pseudo state on the number axis. The coding integral number begin with $-\lfloor G/2 \rfloor$,a hash table to store the serial number of state *j*;
3 Calculate by Equation (15), query serial number of state of $VSAR_{i,k}$;
4 Calculate mid-value $g(s_j) = \phi_{\min} + (j-1)\Delta_\phi + \frac{\Delta_\phi}{2}$;
5 Calculate utility state values of the utility state by Equation (19)
6 Repeat step3-step5, compute utility state values $VSAR_{i,k}$ of bus *i* at bus station *k* + 1;
7 Calculate immediate reward value $R_i(s_{i,k},a_i,s_{i,k+1})$ by Equation (20)
8 Return $R_i(s_{i,k}, a_i, s_{i,k+1})$.

Coordination problem between agents

Accord to the reward function, the virtue schedule including in system state is implicit coordination mechanism from the macroscopic view. Learning agent can be broadly categorized into two fundamental types: independent learners and joint-action learners. This study adopts centralized training by joint agent and decentralized execution by the independent learners [28]. This paradigm is well-suited for real-world transit operations, as it allows extensive offline data to be utilized during the train phase, while significantly reducing execution time. During execution, each agent independently choose an action that maximize its local Q-function. Consequently, the general update equation for agent *i* is:

$$Q_i(s, a_i) = (1-\alpha)Q_i + \alpha(r + \max_{u \in A_i} Q_i(s', u)) \tag{21}$$

However, the association of all agents' individual actions fails to achieve Pareto optimality, posing the first challenge for the independent learner to overcome. In this paper, the coordination issue exists only if two or more buses are at stops during the same period. The basic idea of coordination between agents is to define a joint action for the agents, as πu(s) = (π1(s), · · · πm(s)), m=|P|. After they execute a joint action, the reward of each bus agent is the same as the joint reward. This is formulated as the equation:

$$r(s, a^u) = r^i(s, a^i) = r^j(s, a^j), \forall i, j \in P, a^u = (a^1, \cdots a^i, \cdots, a^j \cdots a^m), m = |P| \tag{22}$$

Solution Based on MARL

Markov games serve as the foundation for a significant portion of MARL research, so the following definitions of key elements of MARL including the state, action, and reward function are given combined with the holding control model. Readers can refer to Busoniu et al. [36], Zepeng, Ning [28], Hu et al.(2024) for an excellent overview of various algorithms for multi-agent reinforcement learning. These include decentralized Q-learning, distributed Q-learning, hysteretic Q-learning, recursive Frequency Maximum Q-Value (FMQ), and Win or Learn

Fast Policy Hill Climbing (WoLF PHC) [37]. The hysteretic Q-learning framework is applied to seek a satisfactory solution.

The primary distinction between the hysteretic Q-learning algorithm, the distributed Q-learning algorithm, and the decentralized Q-learning algorithm lies in their update equations, as shown below:

$$\delta \leftarrow r + \gamma \max_{a'} Q_i(s', a') - Q_i(s, a_i)$$
$$Q_i(s, a_i) \leftarrow \begin{cases} Q_i(s, a_i) + \alpha_1 \delta & if \quad \delta \geq 0 \\ Q_i(s, a_i) + \beta_1 \delta & else \end{cases} \tag{23}$$

Where α_1 and β_1 represent the rates of increase and decrease of Q-values, respectively. The hysteretic Q-learning algorithm functions in a decentralized manner, where each independent learner maintains its own Q-table. The size of the Q-table is independent of the total number of agents and scales linearly with the learner's individual action set. The hysteretic Q-learning algorithm applied to the holding problem is outlined as follows:

Algorithm 2: The hysteretic Q-learning algorithm for the holding problem
1Initialization: the number of states and actions, learning rates α_1 and β_1, discount rate γ , $Q(s_i, a_i)$;
2 Iteration: at iteration p for agent i, to observe the state s_i ,choose an action a_i according to $\pi(s_i) = \arg\max_{a_i} \hat{Q}(s_i, a_i)$ If more than two agents are at stops during this iteration, go to step 7;
3 Perform action and observe reward r along with subsequent state s_i' ;
4 Update the Q-value using Equation(23);
5 Transit to a new state $s_i = s_i'$;
6 $p = p + 1$,and repeat Step 1 until the value function $Q(s_i, a_i)$ converges;
7 Take action a^u , and observe rewards $r(s, a^u)$, $r^i(s, a^i)$, $r^i(s, a^j)$ and the subsequent state s_i' , s_j' ;modify the reward of agent $i, j, \cdots$ according to Equation(22) and go to Step 4.

SIMULATION EXPERIMENTS

Experiment Description

The simulation experiments were conducted using Matlab to build the simulation environment and train the algorithm. An overview of the simulation process is provided in Figure 3.Offline independent learning within the cooperative multi-agent system was executed at the start of transit operation control simulation. The MARL algorithm required approximately 30 minutes to converge in Matlab, running on a 1.6 GHz Intel Core i5-8250U CPU. This setup involved 10500 states and four actions. The proposed model is applied to a 4km transit corridor with ten evenly spaced bus stops at 0.4km intervals. The bus fleet comprised of six buses, each with a passenger capacity of 72. It was assumed that all buses operated at a constant speed of 25km/h, with fixed travel times between adjacent stops. The parameters for boarding and alighting were set to be $\alpha_0 = 3\text{s pax}^{-1}$ and $\beta_0 = 5\text{s pax}^{-1}$, respectively. The planned headway was set to, and passenger arrivals followed a Poisson process, with average arrival rates at each stop illustrated in Figure 4. The parameters for the hysteretic Q-learning algorithm were set as follows: the discount factor $\gamma = 0.9$, while the learning rates $\alpha_1 = 0.3$ and $\beta_1 = 0.03$. The total simulation duration was two hours, including a 15 minutes warm-up time at the beginning.

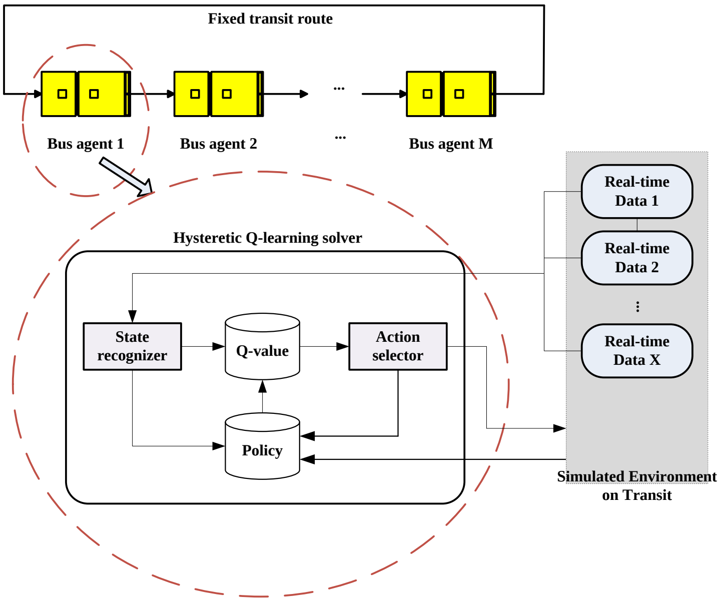


Figure 3. Schematic representation of the MARL approach for solving the holding problem

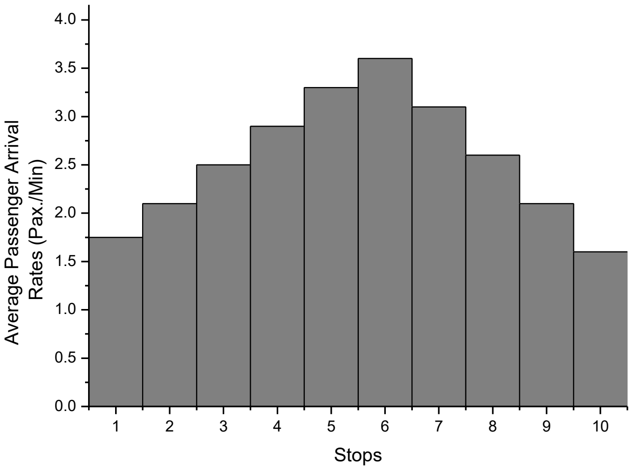


Figure 4. Average passenger arrival rates at stops

The following strategies are tested and compared:

1. No-control strategy: in this case, the holding action are not applied whatever.

2. Threshold control: in this holding control model, the H_0 represents the target headway. The holding action is initiated if the headway to the preceding bus falls below the threshold headway (φH_0), whereas the bus is dispatched immediately otherwise. The holding control strength parameter φ ranges from 0.0 to 1.0.

3. This holding control is implemented using fuzzy rules, as defined by the fuzzy knowledge framework proposed by Milla et al. [38]

4. Proposed control strategy in this paper: holding control based on MARL.

Simulation Analysis and Results

Twenty replications were run for each bus holding control model discussed in the previous sections. Table 2 presents mean and standard deviation of passenger waiting Time, in-vehicle ride time and total travel time per passenger across four scenarios. The in-vehicle ride times were similar across all four scenarios. The waiting time at stops and total travel time, when using only the threshold control strategy, were reduced by 26.55% and 6.81% respectively, and compared to the no control strategy. These results are comparable to the bus holding control strategy based on fuzzy rules, which resulted in a 35.99% reduction in waiting times at stops and a 14.73% reduction in total travel time. The slight improvement in performance observed with the fuzzy rule based strategy applies holding actions at four specific stop (stops 2, 3, 8, and 9), whereas the proposed strategy allow holding actions at all stops. From the Table 1, it is clear that the reduction in total time is primarily due to the savings in waiting time at stops, while in-vehicle ride time remains largely unchanged, as we assume that constant travel time between stops.

Table 2. Mean and standard deviation of passenger waiting time

	Waiting Time (min)			In-vehicle ride time(min)	Total time (min)	
	Mean	Std.	Variation to no control strategy: B (%)	Mean	Mean	Variation to no control strategy:C (%)
1	3.39	1.29	-	4.69	8.08	-
2	2.49	0.58	26.55%	5.04	7.53	6.81%
3	2.17	0.48	35.99%	4.72	6.89	14.73%
4	2.19	0.56	35.4%	4.74	6.93	14.23%

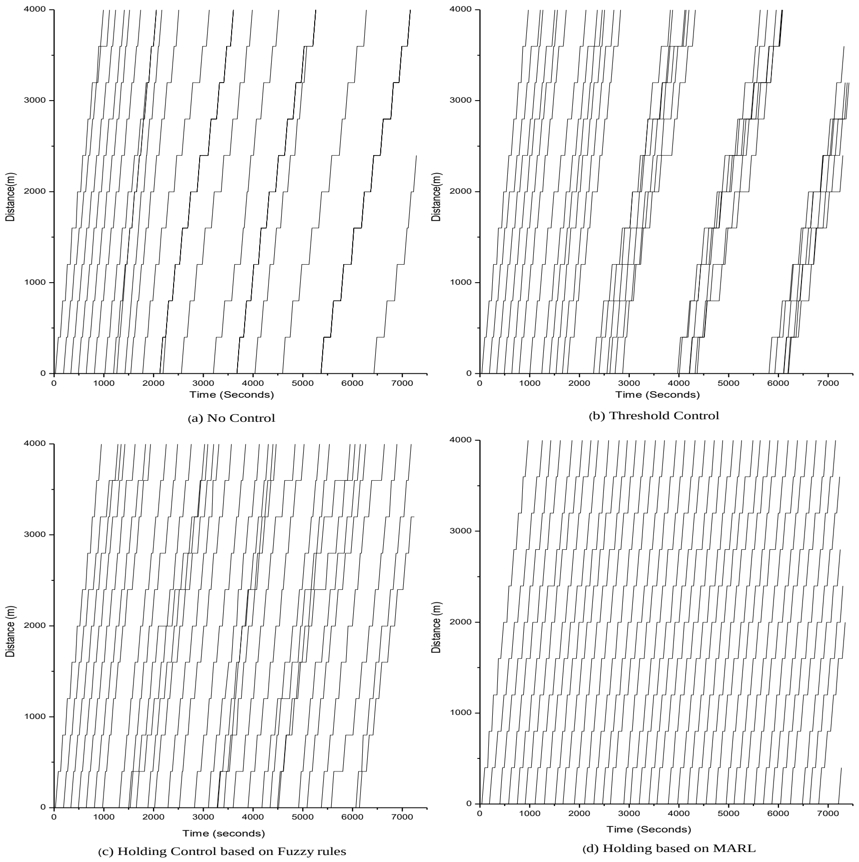
Notes: % variation= $\frac{(case - no\ control)}{no\ control} \cdot \%$

Figure 5 illustrates the bus trajectories for four different control strategies. As shown in Figure 5a, the no-control strategy results in uneven headways, causing buses to bunch together early on, which leads to long periods where no buses pass a given stop. In Figure 5b, the threshold control strategy reduces some of the bunching, allowing buses to maintain more consistent headways. However, long holding times in this strategy can cause delays to propagate to subsequent buses, affecting their headways. In Figure 5c, the application of holding control based on fuzzy rules significantly reduces bunching, but coordinating the objectives of all buses proves challenging.

Finally, the trajectories shown in Figure 5d for the proposed strategy exhibit more consistent headways between buses compared to other control strategies. This improvement is largely attributed to the reward function, which is specifically designed to regularize bus headways, along with the coordination mechanism among bus agents.

Figure 6 illustrates the passenger load on each bus as it travels from departure at stop 1 to arrival at stop 10 across the four different holding control strategies. In the Figure 6a, Under the no-control strategy, significant variability in bus loads is observed at each stop due to bus bunching. Some buses reach full capacity, while others carry considerably fewer passengers. In Figure 6b, the bus loads improve compared to the no-control scenario. Figure 6c and Figure 6d show that the proposed holding control strategy results in less variability and a more uniform distribution of passengers across buses. These findings indicate that the proposed strategy enhances passenger comfort relative to the other control strategies, enabling buses to operate with reduced crowding and offering a more reliable transit service.

As shown in Figure 7, the frequency of no-holding actions for the six buses is approximately 50%, while the frequencies for the other three holding actions are 24.44%, 13.78%, and 12.89%, respectively. In terms of total holding time, it was reduced by 21.2% and 14.9%, respectively, compared to the other two control strategies (Threshold and Fuzzy rules). Table 3 demonstrates that the proposed control strategy effectively reduces the variance in headways and average waiting times. Although holding actions increase the travel time for passengers onboard, the benefits in terms of improved service efficiency make this trade-off worthwhile.



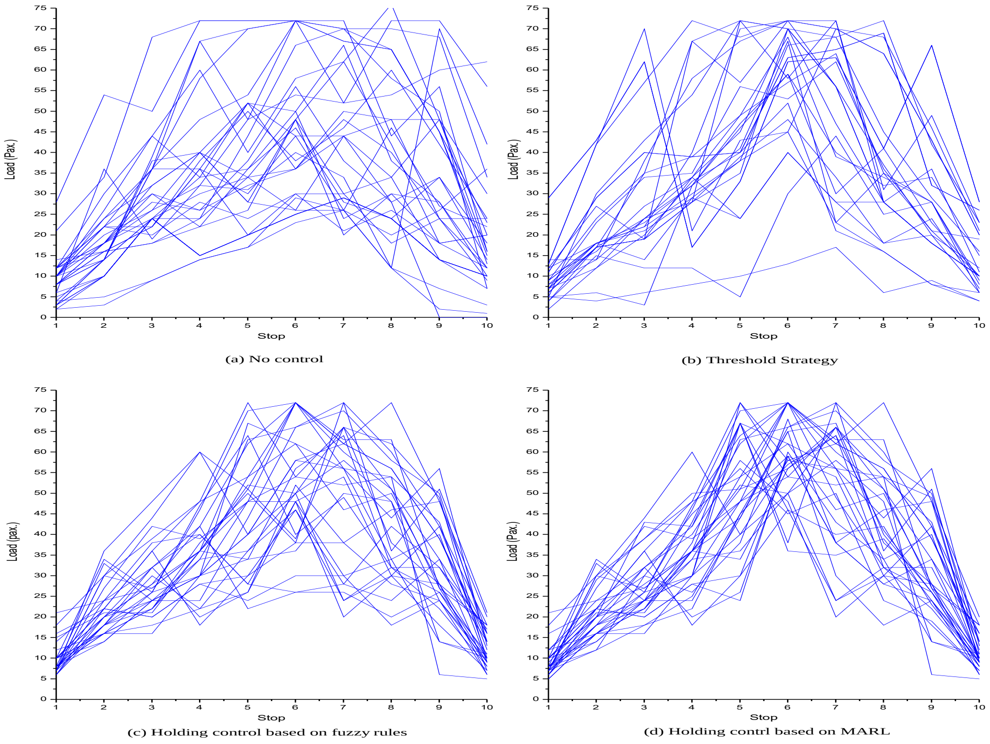
(a) No Control

(b) Threshold Control

(c) Holding Control based on Fuzzy rules

(d) Holding based on MARL

Figure 5. Bus trajectories for the different four strategies



(a) No control

(b) Threshold Strategy

(c) Holding control based on fuzzy rules

(d) Holding contrl based on MARL

Figure 6. Passenger Loads at various stops under four control strategies

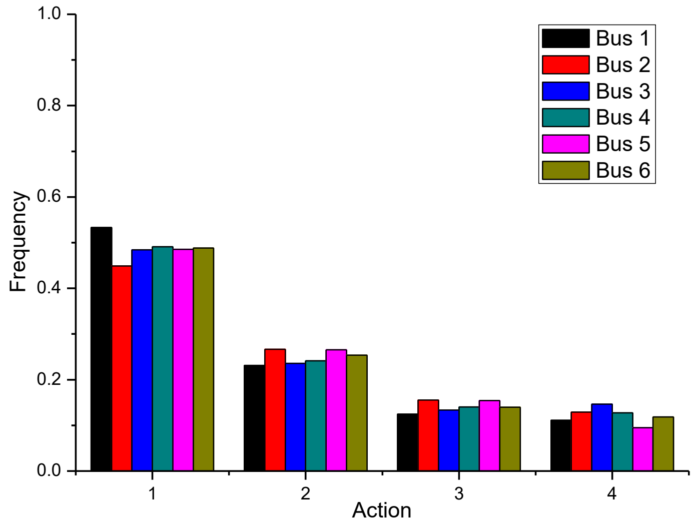


Figure 7. Frequency of use of each bus’s holding control actions

CONCLUSIONS

This paper proposes a dynamic control method based on multi-agent reinforcement learning to implement real-time bus holding control, prevent bus bunching, and ensure the stability of the bus system. By carefully designing key elements of multi-agent reinforcement learning, such as the system state, action set, reward function, multi-agent system structure, and the physical and operation constraints of bus motion. Each bus agent can effectively implement dynamic bus holding strategies to maintain regular headway. To facilitate coordination among all bus agents, the state is defined using a virtue schedule that ensures regular headway. A centralized training approach with decentralized execution is introduced, with virtue schedule playing a crucial role in coordinating bus agents from a macro perspective. To demonstrate the advantage of the proposed multi-agent reinforcement learning framework, the transit simulator is developed for comparative experiments. Meanwhile, the computing time of the hysteretic Q-learning algorithm is suitable for online implementation and real-time decision-making. Simulation results comparing the proposed method with baseline methods reveal the following findings: (1) Compared to the baseline method, the proposed method effective captures the dynamics of the bus system and the long-term effects of current decisions, leading to the most balanced bus trajectories, optimal passenger load distribution, and minimal total holding time; (2) The proposed control strategy outperforms others, with a 35.4 % reduction in waiting time at stops and a 14.23% reduction in total travel time; (3) In terms of total holding time, the proposed method reduces it by 21.2% and 14.9%, respectively, compared to the threshold and fuzzy rules strategies.

For future work, we plan to extend this framework to account for the uncertainties in the bus operation, such as the restricted bus load capacity, traffic signals at intersections. Additionally, in practice scenarios, not all state attribute variables may be fully available due to unexpected events like communication interruptions or data loss. Future efforts will focus on developing models to handle such situations. The integration of mixed control strategies, including skip-stop, deadheading, changes in speed, and leapfrogging operations with the bus ahead should also be explored. Therefore, future research work could make use of real-world traffic data from transit agencies to develop stable and intelligent bus operation and management system.

ACKNOWLEDGMENTS

The authors would like to sincerely thank the editor and the anonymous reviewers for their insightful comments on an earlier draft of this manuscript. Their suggestions have greatly enhanced the quality of this paper. This work was supported by the Hunan Provincial Department of Education Scientific Research Project (Grant No. 23C0814).

REFERENCES

[1] Aichong Sun and Mark Hickman. The real-time stop skipping problem. Journal of Intelligent Transportation Systems, 9(2):91–109, 2005.

[2] Zhiyuan Liu, Yadan Yan, Xiaobo Qu, and Yong Zhang. Bus stop-skipping scheme with random travel time. Transportation Research Part C: Emerging Technologies, 35:46–56, 2013.

[3] Xumei Chen, Bruce Hellinga, Chengzhi Chang, and Liping Fu. Optimization of headways with stop-skipping control: a case study of bus rapid transit system. Journal of Advanced Transportation, 49(3):385–401, 2015.

[4] Konstantinos Gkiotsalitis. Robust stop-skipping at the tactical planning stage with evolutionary optimization. Transportation research record, 2673(3):611–623, 2019.

[5] Konstantinos Gkiotsalitis. Stop-skipping in rolling horizons. Transportmetrica A: Transport Science, 17(4):492–520, 2021.

[6] Avishai Ceder. Optimal design of transit short-turn trips. Transportation Research Record, 1221(557):8–22, 1989.

[7] Shengnan Tian. A short-turning strategy for the management of bus bunching considering variable spatial-temporal running time. Journal of Uncertain Systems, 14(03):2150020, 2021.

[8] Tommaso Schettini, Ola Jabali, and Federico Malucelli. Demand-driven timetabling for a metro corridor using a short-turning acceleration strategy. Transportation Science, 56(4):919–937, 2022.

[9] Seda Yanık and Salim Yılmaz. Optimal design of a bus route with short-turn services. Public Transport, 15(1):169–197, 2023.

[10] P. Chandrasekar, Ruey Long Cheu, and Hoong Chor Chin. Simulation evaluation of route-based control of bus operations. Journal of Transportation Engineering, 128 (6):519–527, 2002.

[11] Carlos F. Daganzo and Josh Pilachowski. Reducing bunching with bus-to-bus cooperation. Transportation Research Part B: Methodological, 45(1):267–277, 2011.

[12] Weiya Chen, Hengpeng Zhang, Chunxiao Chen, and Xiaofan Wei. An integrated bus holding and speed adjusting strategy considering passenger’s waiting time perceptions. Sustainability, 13(10):5529, 2021.

[13] Felipe Delgado, Juan Carlos Munoz, and Ricardo Giesen. How much can holding and/or limiting boarding improve transit performance? Transportation Research Part B: Methodological, 46(9):1202–1217, 2012.

[14] Shuzhi Zhao, Chunxiu Lu, Shidong Liang, and Huasheng Liu. A self-adjusting method to resist bus bunching based on boarding limits. Mathematical Problems in Engineering, 2016(pt.5):1–7, 2016.

[15] Paul Anderson and Carlos F. Daganzo. Effect of transit signal priority on bus service reliability. Transportation Research Procedia, 38:2–19, 2019. Journal of Transportation and Traffic Theory.

[16] Jia Hu, Zihan Zhang, Yongwei Feng, Zhongxiao Sun, Xin Li, and Xianfeng Yang. Transit signal priority enabling connected and automated buses to cut through traffic. IEEE Transactions on Intelligent Transportation Systems, 23(7):8782–8792, 2021.

[17] Antoine Petit, Yanfeng Ouyang, and Chao Lei. Dynamic bus substitution strategy for bunching intervention. Transportation Research Part B: Methodological, 115:1–16, 2018.

[18] Antoine Petit, Chao Lei and Yanfeng Ouyang. Multiline bus bunching control via vehicle substitution. Transportation Research Part B: Methodological, 126:68–86, 2019.

[19] Xu Jun Eberlein, Nigel H. M. Wilson, and David Bernstein. The holding problem with real–time information available. Transportation Science, 35(1):1–18, 2001.

[20] ARNOLD BARNETT. On controlling randomness in transit operations. Transportation Science, 8(2):102–116, 1974.

[21] WARREN B. POWELL. Analysis of vehicle holding and cancellation strategies in bulk arrival, bulk service queues. Transportation Science, 19(4):352–377, 1985.

[22] Mark Abkowitz, Amir Eiger, and Israel Engelstein. Optimal control of headway variation on transit routes. Journal of Advanced Transportation, 20(1):73–88, 1986.

[23] Carlos F Daganzo. A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons. Transportation Research Part B: Methodological, 43(10):913–921, 2009.

[24] Yiguang Xuan, Juan Argote, and Carlos F Daganzo. Dynamic bus holding strategies for schedule reliability: Optimal linear control and performance analysis. Transportation Research Part B: Methodological, 45(10):1831–1845, 2011.

[25] John J. Bartholdi and Donald D. Eisenstein. A self-coordinating bus route to resist bus bunching. Transportation Research Part B: Methodological, 46(4):481–491, 2012.

[26] Matthias Andres and Rahul Nair. A predictive-control framework to address bus bunching. Transportation Research Part B: Methodological, 104:123–148, 2017.

[27] Nahid Parvez Farazi, Bo Zou, Tanvir Ahamed, and Limon Barua. Deep reinforcement learning in transportation research: A review.Transportation Research Interdisciplinary Perspectives, 11:100425, 2021.

[28] Zepeng Ning and Lihua Xie. A survey on multi-agent reinforcement learning and its application. Journal of Automation and Intelligence, 3(2):73–91, 2024.

[29] Chunxiao Chen, Weiya Chen, and Zhiya Chen. A multi-agent reinforcement learning approach for bus holding control strategies. Advances in Transportation Studies, 2:41–54, 12 2015.

[30] Weiya Chen, Kunlin Zhou, and Chunxiao Chen. Real-time bus holding control on a transit corridor based on multi-agent reinforcement learning. In 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), pages100–106, Nov 2016.

[31] Jiawei Wang and Lijun Sun. Dynamic holding control to avoid bus bunching: A multi-agent deep reinforcement learning framework. Transportation Research Part C: Emerging Technologies, 116:102661, 2020.

[32] Jiawei Wang and Lijun Sun. Robust dynamic bus control: a distributional multiagent reinforcement learning approach. IEEE Transactions on Intelligent Transportation Systems, 24(4):4075–4088, 2022.

[33] Jiawei Wang and Lijun Sun. Multi-objective multi-agent deep reinforcement learning to reduce bus bunching for multiline services with a shared corridor. Transportation Research Part C: Emerging Technologies, 155:104309, 2023.

[34] Mengdi Yu, Tao Yang, Chunxiao Li, Yaohui Jin, and Yanyan Xu. Mitigating bus bunching via hierarchical multi-agent reinforcement learning. IEEE Transactions on Intelligent Transportation Systems, 2:1–18, 2024

[35] S.S. Heragu, R.J. Graves, Byung-In Kim, and A. St Onge. Intelligent agent based framework for manufacturing systems control. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 32(5):560–573, 2002.

[36] Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38(2):156–172, 2008.

[37] Kai Hu, Mingyang Li, Zhiqiang Song, Keer Xu, Qingfeng Xia, Ning Sun, Peng Zhou, and Min Xia. A review of research on reinforcement learning algorithms for multi-agents. Neurocomputing, 599: 128068, 2024.

[38] Shih-Che Lo and Wei-Jea Chang. Design of real-time fuzzy bus holding system for the mass rapid transit transfer system. Expert Systems with Applications, 39(2):1718–1724, 2012.