Alleviating Bus Bunching via Modular Vehicles

Yuhao Liu\textsuperscript{a,b,c}, Zhibin Chen\textsuperscript{a,b,*}, and Xiaolei Wang\textsuperscript{d}

\textsuperscript{a}Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning, NYU Shanghai, Shanghai, China
\textsuperscript{b}Shanghai Key Laboratory of Urban Design and Urban Science, NYU Shanghai, Shanghai, China
\textsuperscript{c}Department of Civil and Urban Engineering, New York University, New York, United States
\textsuperscript{d}School of Economy and Management, Tongji University, China

Keywords: Bus bunching, Modular vehicle, Dynamic system, Deep reinforcement learning

The notorious phenomenon of bus bunching prevailing in uncontrolled bus systems produces irregular headways and downgrades the level of service by increasing passengers’ expected waiting time. Modular autonomous vehicles (MAVs), due to their ability to split and merge en route, have the potential to help both late and early buses recover from schedule deviation while providing continuous service. In this paper, we propose a novel bus bunching alleviation strategy for MAV-aided transit systems. We first consider a soft vehicle capacity constraint and establish a continuum approximation (CA) model (Model I) to capture the system dynamics intertwined with the MAV splitting and merging operations, and then establish an infinite-horizon stochastic optimization model to determine the optimal splitting and merging strategy. To capture the reality that passengers may fail to board an overcrowded bus, we propose a second model (Model II) by extending Model I to accommodate a hard vehicle capacity constraint. Based on the characteristics of the problem, we develop a customized deep Q-network (DQN) algorithm with multiple relay buffers and a penalized ruin state applicable for both models to optimize the strategy for each MAV. Numerical results show that the strategy obtained via the DQN algorithm is an effective bunch-proof strategy and has a better performance than the myopic strategy for MAV-aided systems and the two-way-looking strategy for conventional bus systems. Sensitivity analyses are also conducted to examine the effectiveness and benefits of the proposed strategy across different operation scenarios.

1 Introduction

Bus bunching is an intrinsic problem for uncontrolled bus systems and can cause unreliable service and extended waiting time for passengers (Daganzo, 1997). The mechanism of bus bunching has been well...
recognized in the field of transit operation (Osuna and Newell, 1972). Studies on bus bunching prediction and identification also contribute to a better comprehension of the phenomenon (Yu et al., 2016; Gong et al., 2020; Iliopoulou et al., 2020). With uncertainty due to road traffic and passenger arrivals, etc., prevailing in transit service, a bus may fall behind or run ahead of its planned schedule. When a bus falls behind schedule, it faces larger headway resulting in more passengers accumulating at the following stop. The bus thus has to dwell for a longer time to serve the passengers, which escalates the delay. With this vicious cycle, it is difficult and will be more and more difficult for a late bus to catch up with its planned schedule without exogenous intervention. Moreover, as the spacing between a late bus and its successive bus becomes smaller, the successive bus will serve fewer passengers than expected. It hence tends to proceed faster due to reduced dwelling time. As a result, the successive bus catches up with the late bus, eventually causing them to bunch together. On the contrary, an early bus may make its successive bus a late bus as the spacing between them enlarges. Theoretical analyses have revealed that such a process leading to bus bunching is bound to happen if there is no intervention (Daganzo, 2009; Lin et al., 2024).

The key to counteracting bus bunching is to help late buses catch up while bringing early buses back to their planned schedules. Many bunching alleviation strategies have been proposed to address one or both of these aspects at the cost of higher operational costs and/or degraded passenger experience. Bus holding is the primary strategy for mitigating the impact of early buses (Osuna and Newell, 1972; Newell, 1974). By holding a bus when it runs ahead of schedule, the holding strategy maintains a desirable headway but also imposes extra waiting time on the onboard passengers. Over the years, plenty of variants of holding strategies have been proposed to improve the performance or alleviate the impact of the level of service (Eberlein et al., 2001; Bartheodl and Eisenstein, 2012; Berrebi et al., 2018; Liu et al., 2023). On the other hand, to help late buses catch up, researchers have proposed a diverse set of strategies such as stop-skipping (Sun and Hickman, 2005; Liu et al., 2013) in which a late bus is allowed to skip and provide limited service at one or more stops so that it could speed up. Apparently, the drawback of stop-skipping is that some passengers may be skipped and hence have to wait for delayed service. A strategy similar to stop-skipping or limited boarding is to provide in-vehicle crowding information about the upcoming buses to induce the passengers to give up boarding the current bus and wait for the next one (Drabicki et al., 2023). In addition, some researchers have considered giving signal priority to late buses to help them speed up and proposed different signal control strategies (Liu et al., 2003; Estrada et al., 2016). The above strategies address only one side of the problem, which may not be sufficient to completely prevent bus bunching. Therefore, many strategies focus on tackling both early and late buses simultaneously. For example, Daganzo and Pilachowski (2011) proposed a two-way-looking speed adjustment scheme in which each bus cooperates with both its preceding and successive buses to perform an optimal speed adjustment so as to maintain the ideal headway. Petit et al. (2018) proposed to invest in maintaining a fleet of stand-by buses used to substitute a servicing bus at its scheduled position when it is experiencing a certain level of schedule deviation. In this bus substitution scheme, passengers may still suffer from delayed service, because after the substitution of a late bus, the late bus switches to drop-off-only mode and the demand arising within the segment between the late bus and the stand-by bus has to wait for the next bus.

Recently, the modular autonomous vehicle (MAV) technology has caught extensive attention from the transportation industry and academia for its great potential in reshaping the future public transit service. An MAV consists of separable modular units that can decouple and merge en route, thereby enabling in-motion transfers of passengers. The value of such flexibility is worth examining and leveraging in the context of bus bunching alleviation because it enables an early or late MAV bus to adjust itself to its planned schedule by splitting and merging operations while overcoming the weakness of service disruption appearing in existing strategies for conventional buses such as stop-skipping and holding. Fig. 1 illustrates how MAVs can be
Figure 1: Illustration of (a) a catch-up split for a late bus and (b) a holding split for an early bus.
applied to alleviate bus bunching via splitting and merging. Here we adopt the continuum approximation (CA) model presentation where there are no discrete stops and the buses pick up passengers continuously along the route. Both Fig. 1(a) and Fig. 1(b) depict bus trajectories of three consecutive buses (i.e., \( n - 1 \), \( n \) and \( n + 1 \)) among which bus \( n \) decouples at time \( t_0 \). We term a split launched by a late bus a catch-up split (Fig. 1(a)) since the splitting operation aims to catch up with the scheduled trajectory. On the other hand, a split launched by an early bus is termed a holding split (Fig. 1(b)) since in such a split some modular units have to dwell to wait for the planned schedule. When a bus decouples into two independent platoons of units, the preceding platoon is referred to as a leading platoon while the following platoon is called a trailing platoon. In Fig. 1(a), when bus \( n \) finds itself falling behind schedule by \( \epsilon_n(t_0) \) at time \( t_0 \), it launches a catch-up split. Before splitting, all passengers on the bus are instructed to move to the trailing platoon which will continue full service (i.e., providing both pick-up and drop-off services) for a while. After splitting, the leading platoon (which is empty) will then switch to the deadheading mode and move at cruising speed to catch up with the scheduled trajectory. There are 3 critical time points (i.e., \( t_1 \), \( t_2 \), \( t_3 \)) for the operation of the split bus. Once the leading platoon reaches a scheduled point (at time \( t_1 \) in Fig. 1(a)), it resumes the full-service mode operating at commercial speed. In the meantime, the trailing platoon continues to provide full service until time \( t_2 \) when it arrives where the leading platoon switches to the full-service mode. Since then, the trailing bus switches to the drop-off-only mode in which it only drops off passengers. With no passenger boarding, the commercial speed of the trailing platoon will go up so that it can catch up with the leading platoon. At time \( t_3 \), when the two platoons are close enough, they recouple and provide full service together. During the catch-up split, the two platoons provide continuous service with no demand skipped while after the split the deviation is greatly reduced. In Fig. 1(b), when bus \( n \) goes ahead of the planned schedule by \( \epsilon_n(t_0) \) at time \( t_0 \), the bus operator decides to launch a holding split and instruct the onboard passengers to transfer to the leading platoon. The leading platoon then switches to drop-off-only mode to drop off the onboard passengers and stops running at time \( t_1 \) since all passengers have alighted. On the other hand, the trailing platoon will dwell at where they decouple until the planned departure time (\( t_2 \)) when it starts full service and catch up with the leading platoon at time \( t_3 \). The holding split differs from conventional holding strategies in that when a holding split is launched, the onboard passengers need not wait on the platoon on hold or are not forced to alight to walk to their destination. For both catch-up and holding splits, we term the time period from \( t_0 \) to \( t_3 \) a decoupling epoch. It represents the time period during which a bus is running in the decoupled state.

Since the debut of MAV technology, plenty of studies have been conducted to evaluate and optimize the MAV-based transit systems (Wu et al., 2021; Shi and Li, 2021; Liu et al., 2021; Pei et al., 2021; Dakic et al., 2021). For instance, different approaches have been proposed to investigate the optimal operations of MAVs on transit corridors with different system characteristics (Chen et al., 2019, 2020; Shi and Li, 2021; Chen et al., 2022; Pei et al., 2023). The optimal application of MAVs on transit networks was also extensively studied (Dakic et al., 2021; Pei et al., 2021; Liu et al., 2013). To solve the routing problem for MAV-aided on-demand mobility services, Fu and Chow (2022) and Fu and Chow (2023) developed mixed-integer programs and efficient heuristic algorithms. Despite the abundance of literature investigating the application of MAVs in transit systems, very few of them have been dedicated to quantifying the capability of MAVs to alleviate the bus bunching, let alone exploring optimal strategies in this regard. Among a few exceptions, Khan et al. (2023) made the first attempt to leverage the catch-up split capability of MAVs to mitigate bus bunching, and conducted simulation experiments to evaluate their performance. Later, they combined bus splitting with bus holding and proposed a “bus splitting + bus holding” strategy (Khan and Menendez, 2023). However, their holding strategy did not take full advantage of the splittable nature of MAVs and still imposed extra waiting time on the passengers onboard. In addition, they considered only two units for each bus, and each split is only allowed to skip one stop. To the best of our knowledge,
researchers have yet to explore the optimal operation of MAVs providing both catch-up and holding splits as illustrated in Fig. 1 to mitigate bus bunching.

In this paper, we aim to develop a CA mathematical framework to capture the dynamics of an MAV-aided transit service system and optimize each MAV’s operation decisions in a way to minimize the impact of bus bunching in the modular bus system. The decision will specify when each bus should decouple and recouple, the number of units in each platoon when decoupling, as well as when the leading platoon should resume service and when the trailing platoon should switch to drop-off-only mode. Compared with the CA model for the bus substitution strategy proposed in Petit et al. (2018), there are a few challenges in modeling the system dynamics with modular bus operation. First, the bus substitution strategy assumes a predetermined and fixed scheduling time for inserting the standby bus, while the time of mode switching in modular bus operation is subject to uncertainty, which significantly complicates the system dynamic modeling. Second, the changing of platoon capacity due to splitting and merging has to be subtly captured in the modular bus operation. Moreover, due to the randomness of the duration of decoupling epochs, decisions have to be made asynchronously across all buses. In light of this property, a deep reinforcement learning algorithm considering each bus as a decision agent is proposed to optimize the operation strategy.

The contribution of this work is fourfold. First, we develop a mathematical modeling framework to delineate the system dynamics of an MAV-aided transit system in which the interactions between the splitting/merging operations and the vehicle trajectories are carefully deliberated. The proposed models can be readily used to evaluate the performance of transit systems with both MAVs and conventional buses in terms of metrics such as the bus bunching cost and bus operational cost, etc. Second, we extend the CA model widely used for investigating bus bunching (Pilachowski, 2009; Daganzo and Pilachowski, 2011; Petit et al., 2018) by incorporating the hard capacity constraint without strengthening the required conditions. Specifically, a novel approach for moving speed estimation considering the hard capacity constraint is proposed and theoretically validated. To the best of our knowledge, we are the first to propose a CA model with hard capacity constraint for bus bunching study and use it to capture the self-regulating effect that mediates the trajectory deviation. Third, we propose a novel bus bunching alleviation strategy taking advantage of MAVs’ ability to split and merge en route to help both late and early buses recover from schedule deviation, and establish an infinite-horizon stochastic optimization model based on the proposed CA model to determine the optimal operation strategy. Fourth, we design a deep Q network (DQN) reinforcement learning algorithm to solve for the optimal operation strategy for each bus. Numerical results suggest that the solution obtained from DQN algorithm provides bunch-proof operation and outperforms the myopic operation strategy for MAV-aided systems and the two-way-looking strategy for conventional transit systems (Daganzo and Pilachowski, 2011).

The remainder of this paper is organized as follows. Section 2 develops the two models capturing the system dynamics with MAVs’ decoupling and recoupling decisions, and proposes the optimization objective based on the cost formulation. Section 3 provides a detailed description of the proposed DQN reinforcement learning algorithm. In Section 4, we present the numerical results illustrating the models and algorithm performance, and discuss the findings based on the numerical analysis. Finally, Section 5 concludes the paper and points out potential extensions for this work.

2 Model Formulation

In the same spirit of Petit et al. (2018), we consider a generic bus route providing bidirectional transit service, which can be represented by a closed loop as shown in Fig. 2. Due to symmetry, it suffices to focus
analysis on one direction. The loop length is denoted by \( l \), and the set of buses operating on the route is denoted by \( \mathcal{N} = \{1, 2, ..., N\} \). For ease of modeling, we adopt the assumption in Daganzo and Pilachowski (2011) and Petit et al. (2018) that passengers are distributed evenly along the route instead of concentrated at stops. A list of important notations used in our model is provided in the appendix.

2.1 Bus operation and system dynamics with conventional buses

Let \( x_n(t) \) denote the position of bus \( n \) along the route at time \( t \). Note that \( x_n(t) \in [0, l) \); i.e., it represents the absolute distance from bus \( n \) to the origin \( x = 0 \) along the considered direction along the loop. The trajectory of bus \( n \) is described by a series of non-decreasing variables \( \{y_n(t)\}_{t \in \{0, 1, \ldots\}} \), which keep track of the cumulative distance traveled by time \( t \). Using the trajectory profile \( \{y_n(t)\}_{t \in \{0, 1, \ldots\}} \), we can calculate the position of bus \( n \) at any time \( t \) by the following expression:

\[
x_n(t) = y_n(t) - \left\lfloor \frac{y_n(t)}{l} \right\rfloor l, \quad \forall n \in \mathcal{N}, t \in \{0, 1, \ldots\}
\]  

where \( \lfloor \cdot \rfloor \) is the round-down operator that rounds down \( \cdot \) to the maximum integer smaller than or equal to \( \cdot \). Given the loop length and number of buses, the scheduled spacing is \( H = l/N \). The actual spacing of bus \( n \) at time \( t \), defined as the distance between bus \( n \) and its immediately preceding bus (i.e., bus \( n - 1 \)), is calculated as

\[
s_n(t) = \begin{cases} 
  x_{n-1}(t) - x_n(t), & \text{if } x_{n-1}(t) - x_n(t) \geq 0 \\
  x_n(t) - x_{n-1}(t) + l, & \text{otherwise}
\end{cases}
\]  

which is illustrated in Fig. 2.
According to Petit et al. (2018), when all buses are evenly spaced along the route, the commercial speed (i.e., the long-term average speed) of each bus can be expressed as

\[ V = E(1 - B\lambda H) \]  

where \( E \) is the bus cruising speed, \( B \) is the unit boarding time per passenger, and \( \lambda \) is the constant arrival rate over time and space. Note that we assume deterministic demand in this study. The randomness originates from the trajectory noise, which in return will cause a fluctuating boarding demand. At equilibrium, there will be comparable numbers of boarding and alighting passengers. Here, only the boarding time is considered because boarding and alighting are assumed to happen at different doors, and according to Highway Capacity Manual (2000), the unit boarding time is usually longer than the unit alighting time. In the idealized setting, the scheduled trajectories \( \{Y_n(t)\}_{t \in \{0,1,\ldots\}} \) for bus \( n \in \mathcal{N} \) is given by the following difference equations

\[ Y_n(0) = (N - n)H \]  
\[ Y_n(t + 1) = Y_n(t) + V \]  

In reality, however, the bus trajectories may deviate from the schedule since they are susceptible to randomness due to factors such as traffic congestion. When the trajectories are perturbed, the bus spacing is no longer even and the commercial speed will not remain constant. By substituting the scheduled spacing \( H \) with the instantaneous spacing \( s_n(t) \) in Eq. (3), the instantaneous commercial speed can be approximated as

\[ v_n(t) \approx E(1 - B\lambda s_n(t)), \forall n \in \mathcal{N}, t \in \{0,1,2,\ldots\} \]  

With the instantaneous commercial speed \( v_n(t) \), the bus trajectories subject to random noises are defined as

\[ y_n(t + 1) = y_n(t) + v_n(t) + \xi_n(t), \forall n \in \mathcal{N}, t \in \{0,1,2,\ldots\} \]  

where the random noise terms \( \{\xi_n(t)\}_{n \in \mathcal{N}, t \in \{0,1,\ldots\}} \) are independent and identically distributed (i.i.d.) normal random variables with zero mean and variance \( \sigma^2 \). Then the deviation of bus \( n \) from schedule at time \( t \) can be defined as \( \epsilon_n(t) = Y_n(t) - y_n(t) \).

Based on Petit et al. (2018), the occupancy dynamics of bus \( n \) is given by

\[ O_n(t + 1) \approx O_n(t) \left[ 1 - \frac{v_n(t)}{l/2} \left( 2 - \frac{v_n(t)}{l/2} \right) \right] \]  
\[ + \lambda s_n(t) \]  

where \( 1 - \frac{v_n(t)}{l/2} \left( 2 - \frac{v_n(t)}{l/2} \right) \) represents the proportion of onboard passengers at time \( t \) remaining onboard at time \( t + 1 \), and the second term denotes the number of passengers boarding bus \( n \) between time \( t \) and \( t + 1 \).

### 2.2 CA models for MAV-aided bus systems

Now we assume all buses in \( \mathcal{N} \) are modular vehicles consisting of \( m \geq 2 \) modular units, which can be split en route. As such, there are in total \( mN \) modular units running on the route. Let \( K \) be the nominal capacity of a modular unit. Then a bus consisting of \( m \) units will have a nominal capacity of \( mK \). We assume that recoupling only applies to units split from the same bus such that the transit service remains predictable to both bus operators and passengers.
In what follows, two models are proposed to capture the dynamics of the modular bus system. The two models differ in how we approach the impact of capacity on modular bus operations. In the first model, a soft capacity constraint is imposed on the number of boarding passengers, via the introduction of a penalty term called in-vehicle crowding cost in the cost (or objective) function, to guide the decision away from causing high passenger load. In the second model, a hard capacity constraint is imposed; i.e., each platoon can only serve a passenger load no more than the capacity and the passengers who failed to board the platoon due to capacity constraint will have to wait for the next platoon. It is worth noting that both models have their values. Many studies in bus bunching intervention focus on the movement of the buses in the system by assuming unlimited capacity or ignoring the impact of vehicle capacity. Our first model can be aligned with these studies for fair comparison. Moreover, it can to some extent reflect the reality that the actual passenger load may be higher than the nominal capacity, especially during peak hours. On the other hand, our second model effectively extends the CA framework widely used in investigating bus bunching by incorporating the vehicle capacity constraint, endowing the CA framework with the capability of capturing the phenomenon of passengers failing to board a full bus.

2.2.1 Model I: soft capacity constraint

We first develop the model with soft capacity constraint and then propose the model with hard capacity constraint by making proper adaptations to the former one.

To model the decoupling and recoupling decisions in a modular bus system, we introduce the following decision variables for all \( n \in \mathcal{N} \) and \( t \in \{0, 1, \ldots\} \).

\[
z'^{l}_n(t) = \begin{cases} 1, & \text{if bus } n \text{ launches a catch-up split at time } t \\ 0, & \text{otherwise} \end{cases} \quad (9)
\]

\[
z'^{r}_n(t) = \begin{cases} 1, & \text{if bus } n \text{ launches a holding split at time } t \\ 0, & \text{otherwise} \end{cases} \quad (10)
\]

\[k_n(t) : \text{number of units in the leading platoon if bus } n \text{ decouples at time } t \quad (11)\]

\[
r'^{r}_n(t) = \begin{cases} 1, & \text{if bus } n \text{ recouples from a catch-up split at time } t \\ 0, & \text{otherwise} \end{cases} \quad (12)
\]

\[
r'^{c}_n(t) = \begin{cases} 1, & \text{if bus } n \text{ recouples from a holding split at time } t \\ 0, & \text{otherwise} \end{cases} \quad (13)
\]

\[
h'^{l}_n(t) = \begin{cases} 1, & \text{if bus } n\text{'s leading platoon switches to full-service mode in a catch-up split at time } t \\ 0, & \text{otherwise} \end{cases} \quad (14)
\]

\[
h'^{T}_n(t) = \begin{cases} 1, & \text{if bus } n\text{'s trailing platoon switches to drop-off-only mode in a catch-up split at time } t \\ 0, & \text{otherwise} \end{cases} \quad (15)
\]

\[
g'^{l}_n(t) = \begin{cases} 1, & \text{if bus } n\text{'s leading platoon switches to dwelling mode at time } t \\ 0, & \text{otherwise} \end{cases} \quad (16)
\]
Figure 3: Illustration of decision variables
These decision variables are illustrated in Fig. 3. When a split is launched at time $t$, the passengers will be instructed to move to the designated platoon within the following time step such that the bus is decoupled at time $t+1$. The transfer between modular units is expected to be more smooth and efficient in the future with the help of advanced vehicular technology such as movable seats.

Aside from the above decision variables indicating whether an action or mode switch occurs at time $t$, we need additional indicator variables to describe the platoons’ operating mode at time $t$. During an operation horizon, a leading platoon may switch among four modes: full-service, drop-off-only, deadheading, and dwelling, while a trailing platoon among three: full-service, drop-off-only and dwelling. For bus $n$, we use binary variable $u_n(t)$ to indicate whether it is in coupled state, and introduce binary variables $w_L^L(t)$ and $q_L^L(t)$ ($w_T^T(t)$ and $q_T^T(t)$) to indicate its leading (trailing) platoon’s mode at time $t$. The specific definition of these variables is given as follows.

Given the value of these indicator variables, we can uniquely determine the platoons’ operating modes. For example, when $w_L^L(t) = 1$ and $q_L^L(t) = 0$, we can tell that bus $n$’s leading platoon is in full-service mode at time $t$, because $w_L^L(t) = 1$ implies that the leading platoon is in full-service or drop-off-only mode while $q_L^L(t) = 0$ implies that it is not in drop-off-only mode. Based on the combinations of its leading and trailing platoons’ modes, a bus may exhibit eight phases. We summarize the eight possible phases in Table 1 and visualize them in Figs. 4 and 5 (an additional phase for the model with hard capacity constraint). For example, phase 1 corresponds to the case where both bus $n$’s leading and trailing platoons are in full-service mode and the bus is in the coupled state ($u_n(t) = 0$). This simply means bus $n$ is running in the coupled state. Phase 2 differs from phase 1 in that bus $n$ is not in coupled state. This could only happen during a catch-up split (see Fig. 4(a)). Note that scenario 2 for catch-up splits (Fig. 4(b)) is a rare situation where the trailing platoon catches up with the leading platoon while the leading platoon is still deadheading to the scheduled trajectory. It is rare because the deadheading leading platoon should be fast enough such that it

---

1 For completeness, we extend the usage of the term “platoon” to occasions when a bus is in coupled state. In this case, both “bus $n$’s leading platoon” and “bus $n$’s trailing platoon” refer to the whole bus $n$ in the coupled state.

2 A trailing platoon will actually switch to deadheading mode when all passengers have alighted during the drop-off-only period. However, it turns out not necessary to distinguish it from the drop-off-only mode in our model. We hence regard it as part of the drop-off-only mode (but with no passenger onboard).
Figure 4: Two scenarios of a catch-up split

Figure 5: Three scenarios of a holding split
will not be caught up with by a trailing platoon in full service. However, due to the presence of randomness, this situation could happen even with low probability. For the sake of completeness, we present it here and allow this to happen in our model.

We use \( \{y_n^L(t)\}_{t \in \{0,1,...\}} \) and \( \{y_n^T(t)\}_{t \in \{0,1,...\}} \) to keep track of the trajectories of the leading and trailing platoons of bus \( n \). The superscript \( L \) represents a leading platoon while \( T \) represents a trailing platoon. When the two platoons are coupled, they should have the same trajectory. Given this, we can calculate the position of the platoons along the route, which is similar to that in the previous section:

\[
x_n^j(t) = y_n^j(t) - \left( \frac{y_n^j(t)}{l} \right) l, \forall n \in \mathcal{N}, t \in \{0,1,...\}, i \in \{L,T\}
\] (23)

Different from Petit et al. (2018) and Khan et al. (2023), we allow the platoons of different buses to leapfrog. The spacing faced by a specific (leading or trailing) platoon of bus \( n \) is the distance between the platoon and its immediately preceding platoon of another bus (not necessarily bus \( n - 1 \)) in full-service or dwelling mode. Suppose we need to calculate the spacing faced by a platoon of bus \( n \) at location \( x_n^j(t) \) at time \( t \), the location of its immediately preceding platoon which is in full service can be calculated by

\[
x_{n-}^j(t) = \min_{i' \in \{L,T\}, n' \in \mathcal{N} \setminus \{n\}} \left\{ \tilde{x}_{n'^{j'}}(t) : \left| w_{n'^{j'}}(t) - q_{n'^{j'}}(t) \right| \cdot \tilde{x}_{n'^{j'}}(t) > x_n^j(t) \right\}
\] (24)

where

\[
\tilde{x}_{n'^{j'}}(t) = \begin{cases} x_{n'^{j'}}(t), & \text{if } x_{n'^{j'}}(t) > x_n^j(t) \\ x_{n'^{j'}}(t) + l, & \text{otherwise} \end{cases}, \forall i' \in \{L,T\}, n' \in \mathcal{N} \setminus \{n\}
\] (25)

Based on Eq. (25), \( \tilde{x}_{n'^{j'}}(t) - x_n^j(t) \) is the distance of the platoon \( i' \) of bus \( n' \) ahead of platoon \( i \) of bus \( n \) along the bus route. Then by Eq. (24) we select the location of the platoon in full service immediately preceding platoon \( i \) of bus \( n \). The spacing faced by platoons of bus \( n \) is then given by

\[
s_n^j(t) = x_{n-}^j(t) - x_n^j(t), \forall t \in \{0,1,...\}, i \in \{L,T\}
\] (26)
We use \( \dot{v}_n^i(t) \) and \( \ddot{v}_n^i(t) \) to denote the moving speed of platoon \( i \) of bus \( n \) at time \( t \) when it is in full-service mode and drop-off-only mode, respectively. According to Petit et al. (2018), we have

\[
\dot{v}_n^i(t) = E \cdot [1 - \lambda BS_n^i(t)], \forall n \in \mathcal{N}, i \in \{L, T\}, t \in \{0, 1, \ldots\}
\]

and when platoon \( i \) of bus \( n \) is in drop-off-only mode, its moving speed is given by

\[
\dot{v}_n^i(t) \approx \left[ E^{-1} + \max \left\{ \frac{A}{l/2 + \Delta^i_n(t)} \left( 2 - \frac{1}{l/2 + \Delta^i_n(t)} \right) \cdot O^i_n(t), 0 \right\} \right]^{-1}
\]

where \( O^i_n(t) \) is the occupancy of platoon \( i \) of bus \( n \) at time \( t \) and

\[
\Delta^i_n(t) = \dot{y}_n^i(t) - \dot{z}_n^i(t), \forall n \in \mathcal{N}, i \in \{L, T\}, t \in \{0, 1, \ldots\}
\]

Here, \( \dot{y}_n^L(t) \) and \( \dot{y}_n^T(t) \) are respectively calculated by,

\[
\dot{y}_n^L(t + 1) = \dot{y}_n^L(t) + z_n^L(t + 1) \cdot y_n^L(t + 1) - q_n^L(t + 1) \cdot \dot{y}_n^L(t), \forall n \in \mathcal{N}, t \in \{0, 1, \ldots\}
\]

\[
\dot{y}_n^T(t + 1) = \dot{y}_n^T(t) + h_n^T(t + 1) \cdot y_n^T(t + 1) - r_n^T(t + 1) \cdot \dot{y}_n^T(t), \forall n \in \mathcal{N}, t \in \{0, 1, \ldots\}
\]

\( \dot{y}_n^L(t + 1) \) records the platoon’s trajectory at the instant it switches to drop-off-only mode (i.e., when \( z_n^L(t) = 1 \)) and \( \dot{y}_n^T(t) \) records the platoon’s trajectory (\( y_n^T(t) \)) at that time until the catch-up split ends (\( r_n^T(t) = 1 \)) when \( \dot{y}_n^T(t) \) will be reset as zero.

At time \( t \in \{0, 1, \ldots\} \), the moving speeds of bus \( n \)’s platoons (\( n \in \mathcal{N} \)) are then respectively given by

\[
\dot{v}_n^L(t) \approx w_n^L(t) [1 - q_n^L(t)] \cdot \dot{y}_n^L(t) + w_n^L(t) q_n^L(t) \cdot \ddot{y}_n^L(t) + [1 - w_n^L(t)] [1 - q_n^L(t)] \cdot E
\]

\[
\dot{v}_n^T(t) \approx w_n^T(t) [1 - q_n^T(t)] \cdot \dot{y}_n^T(t) + [1 - w_n^T(t)] [1 - q_n^T(t)] \cdot \ddot{y}_n^T(t)
\]

since a platoon moves a) at speed \( \dot{v}_n^i(t) \) when it is in drop-off-only model, b) at commercial speed \( \dot{v}_n^i(t) \) when in full-service mode and c) at cruising speed \( E \) when in deadheading mode.

When recoupling does not happen, given the moving speed at time \( t \), the trajectory dynamics of the leading and trailing platoons are calculated by

\[
y_n^L(t + 1) = y_n^L(t) + \dot{v}_n^L(t) + w_n^L(t) q_n^L(t) \cdot \ddot{y}_n^L(t) + w_n^L(t) [1 - q_n^L(t)] \cdot [u_n(t) \tilde{\xi}_n^L(t)] + (1 - u_n(t)) \tilde{\xi}_n(t) + [1 - w_n^L(t)] [1 - q_n^L(t)] \cdot \tilde{\xi}_n(t), \forall n \in \mathcal{N}, t \in \{0, 1, \ldots\}
\]

\[
y_n^T(t + 1) = y_n^T(t) + \dot{v}_n^T(t) + w_n^T(t) [1 - q_n^T(t)] \cdot [u_n(t) \tilde{\xi}_n^T(t) + (1 - u_n(t)) \tilde{\xi}] + [1 - w_n^T(t)] [1 - q_n^T(t)] \cdot \tilde{\xi}_n(t), \forall n \in \mathcal{N}, t \in \{0, 1, \ldots\}
\]

where \( \tilde{\xi}_n(t) \) is a normal random variable with zero mean and variance \( \sigma^2 \) which represents the random disturbance on trajectories when a platoon is in drop-off-only mode. It is assumed that \( \sigma^2 \leq \sigma^2 \) because a drop-off-only platoon does not need to pick up passengers and thus endures less uncertainty. \( \tilde{\xi}_n^L(t) \) and \( \tilde{\xi}_n^T(t) \) are random variables following the same distribution with \( \xi_n(t) \). They are introduced to distinguish the random disturbances faced by the leading and trailing platoons when they operate separately. \( \tilde{\xi}_n(t) \) also a zero-mean normal random variable representing the random disturbance when a platoon is in deadheading.
mode. Its variance \( \hat{\sigma}^2 \) is assumed to be smaller than \( \sigma^2 \) since a deadheading platoon does not even drop off passengers.

When recoupling is considered, a trailing platoon should merge with its corresponding leading platoon when they are close enough; i.e., the trajectory of a trailing platoon at time \( t + 1 \) should equal that of its leading platoon at time \( t + 1 \) if \( y_n^L(t + 1) - y_n^T(t + 1) \leq \rho \). This enables recoupling. With this consideration, the trajectories of bus \( n \)'s trailing platoon should be determined by

\[
y_n^T(t + 1) = \begin{cases} y_n^L(t + 1), & \text{if } y_n^L(t + 1) - y_n^T(t + 1) \leq \rho, \forall n \in \mathcal{N}, t \in \{0, 1, \ldots\} \\ y_n^T(t + 1), & \text{otherwise} \end{cases}
\]  

With \( y_n^L(t) \) and \( y_n^T(t) \), we can define the schedule deviation for each platoon:

\[
\delta_n^i(t) = Y_n(t) - y_n^i(t), \forall n \in \mathcal{N}, t \in \{0, 1, \ldots\}, i \in \{L, T\}
\]

The occupancies of the leading and trailing platoons of bus \( n \) before recoupling are respectively given by

\[
\begin{align*}
\hat{O}_n^L(t + 1) & \approx w_n^L(t)[1 - q_n^L(t)] \cdot \left\{ O_n^L(t) \left[ 1 - \frac{v_n^L(t)}{l/2} \left( 2 - \frac{v_n^L(t)}{l/2} \right) \right] + \lambda s_n^L(t) \right\} \\
& \quad + w_n^L(t)q_n^L(t) \cdot O_n^L(t) \left[ 1 - \frac{v_n^L(t)}{l/2 + \Delta_n^L(t)} \left( 2 - \frac{v_n^L(t)}{l/2 + \Delta_n^L(t)} \right) \right]
\end{align*}
\]  

\[
\begin{align*}
\hat{O}_n^T(t + 1) & \approx [1 - w_n^T(t)][1 - q_n^T(t)] \cdot O_n^T(t) \left[ 1 - \frac{v_n^T(t)}{l/2 + \Delta_n^T(t)} \left( 2 - \frac{v_n^T(t)}{l/2 + \Delta_n^T(t)} \right) \right] \\
& \quad + w_n^T(t) \cdot \left\{ O_n^T(t) \left[ 1 - \frac{v_n^T(t)}{l/2} \left( 2 - \frac{v_n^T(t)}{l/2} \right) \right] + \lambda s_n^T(t) \right\}
\end{align*}
\]

When the leading or trailing platoon is in full-service mode, the occupancy dynamics is calculated similarly to Eq. (8). When the leading or trailing platoon is in drop-off-only mode, the proportion of onboard passengers alighting during time \( t \) and time \( t + 1 \) is given by \( \frac{v_n^L(t)}{l/2 + \Delta_n^L(t)} \left( 2 - \frac{v_n^L(t)}{l/2 + \Delta_n^L(t)} \right), i \in \{L/T\} \) (Petit et al., 2018). Note that after a non-empty platoon switches from drop-off-only mode to full-service mode, the proportion of alighting passengers may slightly deviate from that given by (38) or (39) as the distribution of the origins of the onboard passengers has changed. However, a valid extension can be readily made by introducing additional variables to keep track of the number of remaining passengers onboard (if any) when the platoon switches to full-service mode and calculate the alighting proportion separately. Here, we use (38) and (39) as an approximation for computational simplicity since the aforementioned situation is not expected to happen frequently.

When recoupling happens \( (r_n(t) = 1) \), the two platoons couple as one. From then on, the occupancy of the leading platoon, denoted by \( \hat{O}_n^L(t) \), and that of the trailing platoon, denoted by \( \hat{O}_n^T(t) \), should both equal the total occupancy of the coupled bus, i.e., \( \hat{O}_n^L(t) + \hat{O}_n^T(t) \). This gives

\[
\hat{O}_n^i(t) = [1 - r_n^i(t) - r_n^L(t)] \cdot \hat{O}_n^i(t) + [r_n^i(t) + r_n^L(t)] \cdot [\hat{O}_n^L(t) + \hat{O}_n^T(t)], \quad \forall n \in \mathcal{N}, t \in \{0, 1, \ldots\}, i \in \{L, T\}
\]

Now we proceed to capture the relationship between the decisions and system states. For conciseness, we introduce \( r_n(t) = r_n^L(t) + r_n^T(t) \) and \( z_n(t) = z_n^L(t) + z_n^T(t) \). Thus \( z_n(t) = 1 \) means bus \( n \) decouples at time \( t \)
and \( r_n(t) = 1 \) means bus \( n \) recouples at time \( t \). Given this, a set of constraints is introduced to specify how the decoupling and recoupling actions and the state of a bus (i.e., \( u_n(t) \)) affect each other.

\[
\begin{align*}
  z_n(t+1) &\leq 1 - u_n(t), \forall n \in \mathcal{N}, t \in \{0, 1, 2, \ldots\} \\
  u_n(t+1) &= u_n(t) - r_n(t+1) + z_n(t+1), \forall n \in \mathcal{N}, t \in \{0, 1, 2, \ldots\}
\end{align*}
\]

where constraint (41) states that decoupling bus \( n \) at time \( t+1 \) is allowed only when the platoons of bus \( n \) are in coupled state at time \( t \). Eq. (42) is the dynamics of \( u_n(t) \); i.e., \( u_n(t) \) turns from 0 to 1 when decoupling happens until the next recoupling when it turns back to 0.

As we have indicated when deriving Eq. (36), the recoupling decision is based on the trajectories of the leading and trailing platoons. This is guaranteed via the following set of constraints.

\[
\begin{align*}
  M \cdot [1 - r_n(t)] &\geq y_n^L(t) - y_n^T(t), \forall n \in \mathcal{N}, t \in \{0, 1, 2, \ldots\} \\
  M \cdot [y_n^L(t) - y_n^T(t)] &\geq [1 - r_n(t)] \cdot u_n(t-1), \forall n \in \mathcal{N}, t \in \{0, 1, 2, \ldots\} 
\end{align*}
\]

where \( M \) is a sufficiently large number. When the trajectories of the leading and trailing platoons do not match at time \( t \), constraint (43) requires \( r_n(t) = 0 \) implying that merging does not happen at time \( t \). When the leading and trailing platoons have the same trajectory at time \( t \) and bus \( n \) is in decoupled state at time \( t - 1 \) \((u_n(t-1) = 1)\), constraint (44) will enforce \( r_n(t) = 1 \) since this implies that the bus platoons recouple at time \( t \) according to Eq. (36).

The third set of constraints captures the dynamics of state variables \( w_n^L(t) \), \( q_n^T(t) \), and their relationship with the decision variables. For all \( n \in \mathcal{N} \) and \( t \in \{1, 2, \ldots\} \),

\[
\begin{align*}
  M[p_n^1(t) - 1] &\leq \delta_n^L(t) \leq Mp_n^1(t) \\
  M[p_n^2(t) - 1] &\leq \Delta_n^T(t) \leq Mp_n^2(t) \\
  M[p_n^3(t) - 1] &\leq \eta - O_n^L(t) \leq Mp_n^3(t) \\
  M[p_n^4(t) - 1] &\leq \delta_n^T(t) \leq Mp_n^4(t) \\
  h_n^L(t) &= [1 - p_n^1(t)] \cdot [1 - q_n^L(t) - w_n^L(t) - 1] \cdot [1 - r_n^L(t)] \\
  h_n^T(t) &= [1 - p_n^2(t)] \cdot [1 - w_n^L(t) - 1] \cdot [1 - r_n^L(t)] \cdot w_n^T(t - 1) \\
  g_n^L(t) &= p_n^3(t) \cdot w_n^L(t) - 1 - 1 - 1 \cdot u_n(t) \\
  g_n^T(t) &= p_n^4(t) \cdot q_n^T(t - 1) \\
  w_n^L(t) &= w_n^L(t - 1) - \left[ z_n^L(t) + g_n^L(t) \right] + h_n^L(t) + r_n^L(t) \cdot \left[ 1 - w_n^L(t - 1) \right] \\
  q_n^T(t) &= q_n^T(t - 1) - r_n^T(t) + z_n^T(t) \\
  q_n^L(t) &= q_n^L(t - 1) + z_n^L(t) - g_n^L(t) \\
  w_n^L(t) &= w_n^L(t - 1) - \left[ z_n^L(t) + h_n^L(t) \right] + g_n^T(t) + r_n^T(t) \cdot \left[ 1 - w_n^T(t - 1) \right] 
\end{align*}
\]

In constraints (45)-(48), \( p_n^1(t) \), \( p_n^2(t) \), \( p_n^3(t) \) and \( p_n^4(t) \) are binary variables related to the sign of the corresponding continuous variables. Take Eq. (45) for example, \( p_n^1(t) = 1 \) when \( \delta_n^L(t) > 0 \) (the leading platoon of bus \( n \) falls behind its schedule) and \( p_n^1(t) = 0 \) when \( \delta_n^L(t) \) is negative. In (47), \( \eta \) is a threshold parameter (e.g., \( \eta = 0.5 \)) used to determine when the leading platoon can be considered empty (and should dwell) in a holding split. This parameter is introduced since theoretically the leading platoon has to travel
to have all passengers alighting and the trailing platoon may not be able to catch up with the leading platoon within a reasonable travel distance. When the occupancy becomes lower than this value, we regard the platoon as empty.

Eqs. (49) and (50) specify the conditions for making decisions \( h^i_n(t) = 1, i \in \{L, T\} \) at time \( t \) during a catch-up split. Eq. (49) specifies when bus \( n \)'s leading platoon should resume full-service mode (\( h^L_n(t) = 1 \)) in a catch-up split. If at time \( t - 1 \), the leading platoon is in deadheading mode (\( w^L_n(t - 1) = 0 \)), and at time \( t \), it is still in the catch-up split (\( r^L_n(t) = 0 \)) but has caught up with the planned schedule (i.e., \( \delta^L_n(t) \leq 0 \) or \( p^L_n(t) = 0 \)), then it resumes to full-service mode. By (50), the trailing platoon switches to drop-off-only mode (\( h^T_n(t) = 1 \)) at time \( t \) if a) bus \( n \)'s leading platoon is in full-service during a catch-up split at time \( t \) (i.e., \( u_n(t)w^L_n(t) \cdot [1 - q^L_n(t)] = 1 \)), b) its trailing platoon is in full-service mode at time \( t - 1 \) (i.e., \( w^T_n(t - 1) = 1 \)), and c) the trailing platoon has reached the point where the leading platoon switches to full-service mode at time \( t \) (i.e., \( \Delta^T_n(t) \leq 0 \) or \( p^T_n(t) = 0 \)).

Eqs. (51) and (52) specify the conditions for making decisions \( g^i_n(t) = 1, i \in \{L, T\} \) at time \( t \) during a holding split. Eq. (51) ensures that bus \( n \)'s leading platoon switches to dwelling mode at time \( t \) when it a) is in drop-off-only mode at time \( t - 1 \) (i.e., \( w^L_n(t)q^L_n(t) = 1 \)), b) does not recouple with the trailing platoon at time \( t \) (i.e., \( r^L_n(t) = 0 \)) and c) has a significantly low occupancy (i.e., \( O^L_n(t + 1) < \eta \) or \( p^L_n(t) = 1 \)). Eq. (52) requires that bus \( n \)'s trailing platoon switches to full-service mode (\( g^T_n(t) = 1 \)) from dwelling mode (\( q^T_n(t - 1) = 1 \)) when it is no longer ahead of schedule (\( \delta^T_n(t) > 0 \) or \( p^T_n(t) \)).

Eqs. (53) and (54) are the transition equations for state variables \( w^L_n(t) \) and \( q^L_n(t) \), respectively. A switch from \( w^L_n(t - 1) = 1 \) to \( w^L_n(t) = 0 \) happens when bus \( n \) launches a catch-up split (\( z^L_n(t) = 1 \)) or the leading platoon switch to dwelling mode (\( \delta^L_n(t) = 1 \)); and the reverse switch happens when the leading platoon resumes full-service mode (\( h^L_n(t) = 1 \)) or the dwelling platoon recouples with the trailing platoon (\( r^L_n(t) = 1 \)). Eq. (54) ensures that \( q^T_n(t - 1) = 1 \) only when bus \( n \) is in the decoupled state for a holding split.

Eq. (55) and (56) are the transition function for state variables \( q^T_n(t) \) and \( w^T_n(t) \), respectively. Eq. (55) states that the 0-1 switch of \( q^T_n(t) \) is dictated by decisions \( z^T_n(t) \) and \( g^T_n(t) \). By Eq. (56), \( w^T_n(t) \) switches from 1 to 0 when bus \( n \) launches a holding split (\( z^T_n(t) = 1 \)) or the trailing platoon switches from full-service mode to drop-off-only mode (\( h^T_n(t) = 1 \)), and from 0 to 1 when the trailing platoon switches from dwelling mode to full-service mode (\( g^T_n(t) = 1 \)) or the trailing platoon recouples with the leading platoon at the end of a catch-up split (\( r^T_n(t) = 1 \)).

Finally, the capacity varies with the decoupling and recoupling operations. The following equations describe the capacity dynamics. For all \( n \in \mathcal{N} \) and \( t \in \{0, 1, \ldots\} \):

\[
\begin{align*}
z_n(t) &\leq k_n(t) \leq (m - 1) \cdot z_n(t) \\
K^L_n(t + 1) &= K^L_n(t) - z_n(t + 1)K[m - k_n(t + 1)] + r_n(t + 1)[mk - K^L_n(t)] \\
K^T_n(t + 1) &= K^T_n(t) - z_n(t + 1)Kk_n(t + 1) + r_n(t + 1)[mk - K^T_n(t)]
\end{align*}
\]

where \( K^L_n \) denotes the capacity of bus \( n \)'s leading (if \( i = L \)) or trailing (if \( i = T \)) platoon at time \( t \). Eq. (57) states when bus \( n \) decouples at time \( t \), the number of units in the leading platoon has to be greater than 1 and no more than \( m - 1 \). Eqs. (58) and (59) ensure when bus \( n \) runs in coupled state, the leading and trailing platoons have the same capacity which equals the total capacity (i.e., \( mk \)), and that when bus \( n \) runs in decoupled state, each platoon will have capacity equal to \( K \) multiplied by the number of units it contains.

Now we proceed to the cost formulation for the MAV-aided system. There are three cost components for operating such a bus system. The first component is passengers’ cost which consists of the cost due to the delayed schedule and the waiting time cost. Specifically, the passengers’ cost for bus \( n \) at time \( t \) is
calculated as
\[
C_{n}^{\text{pass}}(t) = C_{n}^{\text{pass,L}}(t) + u_{n}(t) \cdot C_{n}^{\text{pass,T}}(t)
\]  
(60)
where \(C_{n}^{\text{pass,L}}(t)\) is the passengers’ cost for bus \(n\)’s leading platoon at time \(t\). When bus \(n\) is in the coupled state \((u_{n}(t) = 0)\), this is the passengers’ cost for the whole bus \(n\). While bus \(n\) is in decoupled state \((u_{n}(t) = 1)\), the additional term \(C_{n}^{\text{pass,T}}(t)\) denotes the passengers’ cost for the trailing platoon. They are calculated as follows.

\[
C_{n}^{\text{pass,L}}(t) = w_{n}^{L}(t)[1 - q_{n}^{L}(t)]C_{n}^{\text{pass,L,1}}(t) + w_{n}^{L}(t)q_{n}^{L}(t)C_{n}^{\text{pass,L,2}}(t)
\]  
(61)
\[
C_{n}^{\text{pass,T}}(t) = w_{n}^{T}(t)[1 - q_{n}^{T}(t)]C_{n}^{\text{pass,T,1}}(t) + [1 - w_{n}^{T}(t)][1 - q_{n}^{T}(t)]C_{n}^{\text{pass,T,2}}(t)
\]  
(62)
\[
C_{n}^{\text{pass,i,1}}(t) = \mu \cdot \left[ \frac{|\delta_{n}^{i}(t)|}{V} \cdot \frac{O_{n}^{i}(t) \cdot v_{n}^{i}(t)}{l/2} \right] \left( 2 - \frac{v_{n}^{i}(t)}{l/2} \right) + \frac{1}{2} \lambda \left[ \frac{v_{n}^{i}(t)}{v_{n}^{i}(t)} \right]^2, \forall i \in \{L, T\}
\]  
(63)
\[
C_{n}^{\text{pass,i,2}}(t) = \mu \cdot \left[ \frac{|\delta_{n}^{i}(t)|}{V} \cdot \frac{O_{n}^{i}(t) \cdot v_{n}^{i}(t)}{l/2 + \Delta_{n}^{i}(t)} \right] \left( 2 - \frac{v_{n}^{i}(t)}{l/2 + \Delta_{n}^{i}(t)} \right), \forall i \in \{L, T\}
\]  
(64)
Here, \(\mu\) is the value of time. Eqs. (63) and (64) calculate the passengers’ cost when the platoon is in full-service mode and drop-off-only mode, respectively. The first term in Eq. (63) is the schedule deviation cost for all passengers alighting the considered platoon at time \(t\). It is calculated as the product of the deviation converted to a time unit, \(\frac{|\delta_{n}^{i}(t)|}{V}\), and the number of passengers alighting, \(\frac{O_{n}^{i}(t) \cdot v_{n}^{i}(t)}{l/2} \cdot \left( 2 - \frac{v_{n}^{i}(t)}{l/2} \right) + \frac{1}{2} \lambda \left[ \frac{v_{n}^{i}(t)}{v_{n}^{i}(t)} \right]^2\). The second term in Eq. (63) is the passenger waiting time where the average waiting time for each passenger is approximated as half of the headway (Petit et al., 2018). The minimization of waiting time is justified by the well-known fact that the expected waiting time is minimized when the headways between buses are even. Eq. (64) is similar to Eq. (63), except that it has no term for waiting time cost since the platoon does not pick up passengers when it is in drop-off-only mode. Incorporating with the indicator variables gives Eqs. (61) and (62). It is worth noting that the minimization of schedule deviation cost and waiting time cost also plays an effective role in minimizing the total travel time. Since the trip distance for each passenger and the bus cruising speed is fixed, the travel time thus depends on the dwell time experienced by all passengers. Note that the total number of boarding passengers is fixed, and so is the total dwell time for the bus fleet. When the headways are not even, buses facing a larger/smaller headway tend to have more/fewer passengers onboard experiencing longer/shorter dwell time. This will result in longer experienced dwell time per passenger (and thus longer travel time) for the whole system. Therefore, reducing travel time implies maintaining even headways and desired schedules.

The second component is the operational cost for bus running and splitting.
\[
C_{n}^{\text{op}}(t) = c^{\text{d}} \cdot z_{n}(t) + c^{\text{c}} \cdot [1 - u_{n}(t)] \cdot v_{n}^{L}(t) + u_{n}(t) \cdot \left[ c^{\text{d}}(K_{n}^{L}(t)) \cdot v_{n}^{L}(t) + c^{\text{d}}(K_{n}^{T}(t)) \cdot v_{n}^{T}(t) \right]
\]  
(65)
where \(c^{\text{d}}\) denotes the bus splitting cost which involves the manipulation cost for a pair of splitting and merging operations and the depreciation due to mechanical wear and tear. \(c^{\text{c}}\) denotes the per-distance cost for buses running in the coupled state. \(c^{\text{d}}(\cdot)\) is an increasing function in platoon capacity \(K_{n}^{i}(t)\), which denotes the per distance cost for bus platoons running in the decoupled state. It is assumed that \(c^{\text{c}} \leq c^{\text{d}}(K_{n}^{L}(t)) + c^{\text{d}}(mK - K_{n}^{L}(t))\) since the cost for operating a single platoon is less than that when the same number of units operates separately as two platoons.

The third cost component considered in this study is the in-vehicle crowding cost.
\[
C_{n}^{\text{crowd}}(t) = u_{n}(t) \cdot [\Phi_{n}^{L}(t) + \Phi_{n}^{T}(t)] + [1 - u_{n}(t)] \cdot \Phi_{n}^{L}(t) = \Phi_{n}^{L}(t) + u_{n}(t) \cdot \Phi_{n}^{T}(t)
\]  
(66)
where $\Phi^i_n(t)$ is the in-vehicle congestion cost for the leading (if $i = L$) or trailing (if $i = T$) platoon of bus $n$ at time $t$, which is calculated via an in-vehicle congestion cost function $\Phi(O^i_n(t), K^i_n(t))$. Here, we adopt the discomfort function proposed in Hamdouch et al. (2014).

$$\Phi^i_n(t) = \Phi(O^i_n(t), K^i_n(t)) = \psi \left[ \frac{O^i_n(t)}{K^i_n(t)} \right]^{\beta}, \forall i \in \{L, T\}$$

where $\beta > 1$ is a parameter that makes the in-vehicle congestion increase rapidly in terms of the relative occupancy and $\psi$ is a coefficient converting the in-vehicle congestion into monetary cost. This discomfort function ensures that the crowding cost remains low when the occupancy is lower than the capacity but increases quickly when overloading happens.

In summary, the infinite-horizon stochastic optimization model for a modular bus system is established as follows, where the objective is to minimize the expected system cost over all buses and the whole horizon.

$$\min_{z, k} \sum_{n=1}^{N} \sum_{t=0}^{N} \mathbb{E} \left[ C_{n}^{\text{pax}}(t) + C_{n}^{\text{opr}}(t) + C_{n}^{\text{crowd}}(t) \right] \quad \text{s.t.} \ (23) - (59)$$

where constraints (23) - (40) specify the dynamics of bus trajectory, speed and occupancy, while constraints (41) - (59) describe the relationship between the decisions and system states. Note that although we define five decision variables, namely $z = \{ z^i_n(t) \}_{n \in \mathcal{N}, t \in \{1,2,...\}, i \in \{e,l\}}$, $k = \{ k_n(t) \}_{n \in \mathcal{N}, t \in \{1,2,...\}}$, $r = \{ r^i_n(t) \}_{n \in \mathcal{N}, t \in \{1,2,...\}, i \in \{e,l\}}$, $h = \{ h^i_n(t) \}_{n \in \mathcal{N}, t \in \{1,2,...\}, i \in \{L,T\}}$ and $g = \{ g^i_n(t) \}_{n \in \mathcal{N}, t \in \{1,2,...\}, i \in \{e,l\}}$, the latter three can be determined automatically by the system states through constraints (43)-(56). The essential decisions we need to make in this system are $z$ and $k$ when a bus is in the coupled state. After we decide to decouple a bus, the decoupling epoch can be regarded as a random event over which we have rare control.

### 2.2.2 Model II: hard capacity constraint

Without considering the hard vehicle capacity constraint, Model I with a soft capacity constraint lends us some simplicity in modeling while preventing the occupancy from being too much higher than the capacity and providing an indicator for us to identify potential overcrowding issues. It, however, lacks the capability of reflecting the reality that passengers may not be able or willing to board an overcrowded platoon and thus have to wait for the next platoon. In Model II, we introduce the hard vehicle capacity constraint. When there are more boarding passengers than the remaining capacity, the platoon can only pick up passengers up to its full occupancy while the excess passengers have to wait for the next platoon.

Compared with Model I, the introduction of hard capacity constraint impacts the model development in three major aspects.

- First, a catch-up or holding split may not be launched immediately at the instant of decision if the occupancy is too high to spare enough empty units.
- Second, it is necessary to keep track of the number of passengers who fail to board the first platoon and thus have to wait for the following platoons.
- Third, the moving speed estimation need to be adjusted as the number of passengers actually boarding a platoon may be less than the number of intended passengers, which affects the dwell time calculation.

In what follows, we will develop Model II by responding to each of these aspects based on Model I.
To address the first aspect, we introduce an additional *pre-split* phase on top of the eight phases described in Model I. A bus enters a pre-split phase whenever the bus operator decides to launch a split but the split is not feasible due to high occupancy. In the pre-split phase, the bus continues to operate in coupled state but only drops off passengers. The pre-split phase ends when the occupancy becomes low enough to enable the split. At this moment, if the bus is still behind (or ahead of) schedule, the catch-up (or holding) split will then be launched; otherwise, the bus will resume to full-service mode without decoupling. The pre-split phase can be considered as a preparation phase for decoupling. When the bus is crowded, the passengers may have to spend more time moving to the designated platoon, and the pre-split phase can provide more time for the passengers to move.

Possible scenarios with pre-split phase $\mathcal{g}$ is depicted in Figs. 6 and 7. These scenarios will appear only when the split is not feasible at the moment of decision; otherwise, we will still observe the scenarios depicted in Figs. 4 and 5. To model these new scenarios, we additionally define the following variables.

$$\hat{z}_n(t) = \begin{cases} 1, & \text{if bus } n \text{ decides to launch a catch-up split at time } t \\ 0, & \text{otherwise} \end{cases} \quad (69)$$

$^4$Though scenario 7 for holding splits is a rare scenario, we include it here for completeness.
Figure 7: Four additional scenarios of a holding split in Model II
\[
\hat{z}_n(t) = \begin{cases} 
1, & \text{if bus } n \text{ decides to launch a holding split at time } t \\
0, & \text{otherwise}
\end{cases}
\]

(70)

\[
\hat{k}_n(t) : \text{ number of units decided to be in the leading platoon when bus } n \text{ decides to split}
\]

(71)

\[
\hat{\tilde{r}}_n(t) = \begin{cases} 
1, & \text{if bus } n \text{ resumes to full-service mode from a pre-split phase at time } t \\
0, & \text{otherwise}
\end{cases}
\]

(72)

\[
\hat{\tilde{u}}_n(t) = \begin{cases} 
1, & \text{if bus } n \text{ is in the pre-split mode at time } t \\
0, & \text{otherwise}
\end{cases}
\]

(73)

With these variables, we differentiate the decision (i.e., \( \hat{z}_n(t) \), \( \hat{\tilde{r}}_n(t) \) and \( \hat{k}_n(t) \)) and the execution (i.e., \( \hat{z}'_n(t) \), \( \hat{\tilde{r}}'_n(t) \) and \( \hat{k}_n(t) \)) of a split, which in Model I are regarded as the same. In Model II, the essential decision variables are \( \hat{z}'_n(t) \), \( \hat{\tilde{r}}'_n(t) \) and \( \hat{k}_n(t) \) while \( \hat{z}_n(t) \), \( \hat{\tilde{r}}_n(t) \) and \( k_n(t) \) will be automatically determined by the decisions \( \hat{z}'_n(t) \), \( \hat{\tilde{r}}'_n(t) \) and \( k_n(t) \) as well as the system states. Decisions can only be made when a bus is not in decoupled state nor in a pre-split phase. Hereinafter, we term a state in which a bus is allowed to make a split decision as an **actionable state**.

All variables defined in Model I will still be used in Model II with the definitions unchanged. However, to correctly incorporate the pre-split phase in the model while respecting the scenarios established in Model I (Figs. 4 and 5), we need to make some necessary modifications to the constraints and state transition equations.

First of all, the boundary constraints (41) and (57) imposed on variables \( z_n(t) \) and \( k_n(t) \) are no longer needed since they are now not the decision variables. Instead, boundary constraints should be imposed on variables \( \hat{z}'_n(t) \), \( \hat{\tilde{r}}'_n(t) \) and \( \hat{k}_n(t) \).

\[
\hat{z}_n(t) \leq 1 - [u_n(t-1) + \hat{\tilde{u}}_n(t-1)]
\]

(74)

\[
\hat{z}_n(t) \leq \hat{k}_n(t) \leq (m-1) \cdot \hat{z}_n(t)
\]

(75)

where \( \hat{z}_n(t) \) is defined as \( \hat{z}'_n(t) + \hat{\tilde{r}}'_n(t) \) for conciseness. Constraint (74) ensures that split decisions can only be made at time \( t \) when the bus is in an actionable state at time \( t-1 \) (\( u_n(t-1) = 1 \) or \( \hat{\tilde{u}}_n(t-1) = 1 \)). Constraint (75) is the bound for decision \( \hat{k}_n(t) \). Given decisions \( \hat{z}_n(t) \), \( \hat{\tilde{r}}_n(t) \) and \( \hat{k}_n(t) \), variables \( z_n(t) \), \( \tilde{r}_n(t) \) and \( k_n(t) \) are now determined by the following set of constraints.

\[
k_n(t) = k_n(t-1) - k_n(t-1)[r_n(t) + \hat{\tilde{r}}_n(t)] + \hat{k}_n(t)
\]

(76)

\[
M[p_n^5(t) - 1] \leq [m - k_n(t-1)]K - O_n^e(t) \leq M p_n^5(t)
\]

(77)

\[
M[p_n^6(t) - 1] \leq k_n(t-1)K - O_n^e(t) \leq M p_n^6(t)
\]

(78)

\[
\hat{z}'_n(t) = p_n^1(t) \cdot p_n^5(t) \cdot [\hat{z}_n(t) + \hat{\tilde{u}}_n(t-1)]
\]

(79)

\[
\hat{\tilde{r}}'_n(t) = [1 - p_n^1(t)] \cdot p_n^6(t) \cdot [\hat{\tilde{r}}_n(t) + \hat{\tilde{u}}_n(t-1)]
\]

(80)

By Eq. (76), \( k_n(t) \) records any decision \( \hat{k}_n(t) > 0 \) until the bus recouples \( (r_n(t) = 1) \) or resumes to full-service mode \( (\hat{\tilde{r}}_n(t) = 1) \). Constraints (77) and (78) play a similar role as (45) - (48). The binary variable \( p_n^5(t) \) (or \( p_n^6(t) \)) will equal 1 if the catch-up (or holding) split is feasible. Eq. (79) and (80) determine the value of \( \hat{z}'_n(t) \) and \( \hat{\tilde{r}}'_n(t) \), respectively. For example, a catch-up split should be launched at time \( t \) (\( \hat{z}'_n(t) = 1 \)) if the following conditions are met: a) catch-up split decision is made at time \( t \) (\( \hat{z}_n(t) = 1 \)) or the bus is in pre-split state at time \( t-1 \) (\( \hat{\tilde{u}}_n(t-1) = 1 \)) and b) the split is feasible at time \( t \) (\( p_n^5(t) = 1 \)) and c) the bus is behind schedule at time \( t \) (\( p_n^1(t) = 1 \)).
For variables $\hat{r}_n(t)$ and $\hat{u}_n(t)$ defined exclusively for Model II, we need additional equations to determine their values. According to definition (72), $\hat{r}_n(t)$ should equal 1 if and only if the schedule deviation is recovered during the pre-split phase, which can be expressed by

$$\hat{r}_n(t) = \hat{u}_n(t - 1) \cdot |p^L_n(t) - p^L_n(t - 1)|$$  (81)

This equation ensures that $\hat{r}_n(t) = 1$ if a) the bus is in a pre-split phase ($\hat{u}_n(t - 1) = 1$) at time $t - 1$ and b) $p^L_n(t) \neq p^L_n(t - 1)$ which indicates $\delta^L_n(t)$ and $\delta^L_n(t - 1)$ have different signs. The transition function for $\hat{u}_n(t)$ is given by

$$\hat{u}_n(t) = \hat{u}_n(t - 1) - z_n(t) - \hat{r}_n(t) + \hat{z}_n(t)$$  (82)

which guarantees that the bus enters a pre-split phase at time $t$ ($\hat{u}_n(t) = 1$) if and only if a split decision is made ($\hat{z}_n(t) = 1$) but the split is not launched at time $t$ ($z_n(t) = 0$), and that the pre-split phase ends ($\hat{u}_n(t) = 0$) when the split is launched ($z_n(t) = 1$) or the bus resumes to full-service mode ($\hat{r}_n(t) = 1$).

Due to the additional scenarios of platoon mode transitions, the following transition functions for variables $q^L_n(t)$, $w^T_n(t)$, $\hat{Y}^L_n(t)$ and $\hat{Y}^T_n(t)$ should be adopted in Model II instead of their counterparts developed in Model I.

$$q^L_n(t) = q^L_n(t - 1) - \frac{1}{n} - \hat{r}_n(t) - r^l_n(t) + \hat{z}_n(t)$$  (83)

$$w^T_n(t) = w^T_n(t - 1) - \hat{z}_n(t) - h^T_n(t) + z^l_n(t) + \hat{r}_n(t) + g^T_n(t) + r^l_n(t)[1 - w^T_n(t - 1)]$$  (84)

$$\hat{Y}^L_n(t) = \hat{Y}^L_n(t - 1) + [\hat{z}_n(t) - z_n(t)] \cdot \hat{y}^L_n(t) - [z^l_n(t) + g^L_n(t) + \hat{r}_n(t)] \cdot \hat{y}^L_n(t - 1)$$  (85)

$$\hat{Y}^T_n(t) = \hat{Y}^T_n(t - 1) + [\hat{z}_n(t) - z_n(t)] \cdot \hat{y}^T_n(t) + \hat{y}^L_n(t) \cdot \hat{y}^T_n(t) - [z_n(t) + r^l_n(t) + \hat{r}_n(t)] \cdot \hat{y}^T_n(t - 1)$$  (86)

Eq. (83) establishes that the leading platoon of bus $n$ switches to drop-off-only mode at time $t$ ($q^L_n(t) = 1$) if the operator decides to launch a holding split ($\hat{z}_n(t) = 1$) or the bus enters a pre-split phase due to an infeasible catch-up split ($\hat{z}_n(t) = 1$ and $z^l_n(t) = 0$). And the platoon switches to full-service mode ($q^L_n(t) = 0$) if the bus resumes to full-service mode from a pre-split phase ($\hat{r}_n(t) = 1$) or recouples from a holding split ($r^l_n(t) = 1$). Compared with its counterpart Eq. (56) in Model I, Eq. (84) covers an additional pair of situations where a trailing platoon switches between full-service mode and drop-off-only mode. It switches to drop-off-only mode ($w^T_n(t) = 0$) when the bus enters a pre-split phase due to a catch-up split decision ($\hat{z}_n(t) = 1$ and $z^l_n(t) = 0$) and $w^T_n(t)$ resumes to 1 when the bus resumes full service from a pre-split phase ($\hat{r}_n(t) = 1$). Similarly, Eqs. (85) and (86) are derived by extending Eqs. (30) and (31) to account for the additional situations incurred by the pre-split phase.

Now we proceed to address the second aspect regarding the occupancy dynamics considering the excess demand. To calculate the actual number of boarding passengers, we need to determine the available capacity for accommodating passengers between any time $t$ and $t + 1$, which depends on the capacity of the platoon, the occupancy at time $t$ and the number of passengers alighting between time $t$ and $t + 1$. Considering the possibility of passengers failing to board a platoon, the onboard passengers’ origin distribution may be different from that in Model I which may in turn affect the alighting proportion. In Appendix B, we prove that the alighting proportion derived in Petit et al. (2018) still holds in a probabilistic sense even if the platoon has a limited capacity. Therefore, the available capacity of bus $n$’s platoon $i$ between time $t$ and $t + 1$, denoted by $\overline{K}^i_n(t)$, is calculated as the total capacity $K^i_n(t)$ minus the number of passengers that will remain onboard at time $t + 1$, i.e.,

$$\overline{K}^i_n(t) = K^i_n(t) - O^i_n(t) \left[ 1 - \frac{v^i_n(t)}{t/2} \left( 2 - \frac{v^i_n(t)}{t/2} \right) \right]$$  (87)
Note that we only care about the available capacity for full-service platoons so Eq. (87) is only valid for full-service platoons.

The number of passengers intending to board platoon $i$ of bus $n$ between time $t$ and $t + 1$ is calculated as

$$\hat{\alpha}_n^i(t) = \lambda s_n^i(t) + \int_{\mathcal{D}_n^i(t)} \pi(\omega; t) d\omega$$

(88)

where $\pi(\omega; t)$ is the excess (or stranded) demand from previous platoons waiting at position $\omega$ at time $t$ and $\mathcal{D}_n^i(t)$ is the bus route segment traveled by platoon $i$ of bus $n$ between time $t$ and time $t + 1$. When $x_n^i(t) \leq x_n^i(t+1)$, $\mathcal{D}_n^i(t) = [x_n^i(t), x_n^i(t+1)]$; otherwise, $\mathcal{D}_n^i(t) = [x_n^i(t), l] \cup [0, x_n^i(t+1)]$. Thus, the integral $\int_{\mathcal{D}_n^i(t)} \pi(\omega; t) d\omega$ calculates the excess demand intending to board platoon $i$ of bus $n$ when it travels on segment $\mathcal{D}_n^i(t)$. By summing the newly generated demand $\lambda s_n^i(t)$ and the excess demand, $\hat{\alpha}_n^i(t)$ denotes the total demand intending to board platoon $i$ of bus $n$.

Due to the capacity constraint, the actual number of passengers a platoon can pick up between time $t$ and $t + 1$ is the minimum of the available capacity and the total demand intending to board. Let $\alpha_n^i(t)$ denote the actual number of passengers boarding platoon $i$ of bus $n$ between time $t$ and $t + 1$. Then it is given by

$$\alpha_n^i(t) = \min\{\hat{\alpha}_n^i(t), \bar{K}_n^i(t)\}$$

(89)

When there are more passengers intending to board than the available capacity, the excess demand is assumed to be evenly distributed over the segment $\mathcal{D}_n^i(t)$, which gives the equation for the excess demand

$$\pi(\omega; t + 1) = \frac{\hat{\alpha}_n^i(t) - \alpha_n^i(t)}{y_n^i(t + 1) - y_n^i(t)}, \forall \omega \in \mathcal{D}_n^i(t)$$

(90)

The occupancies for the leading and trailing platoons of bus $n$ before recoupling, similar to Eqs. (38) and (39), are respectively calculated as

$$\bar{O}_n^L(t + 1) \approx w_n^L(t)[1 - q_n^L(t)] \cdot \left\{O_n^L(t) \left[1 - \frac{v_n^L(t)}{l/2} \left(2 - \frac{v_n^L(t)}{l/2}\right)\right] + \alpha_n^L(t + 1)\right\}$$

(91)

$$+ w_n^L(t)q_n^L(t) \cdot O_n^L(t) \left[1 - \frac{v_n^L(t)}{l/2 + \Delta_n^L(t)} \left(2 - \frac{v_n^L(t)}{l/2 + \Delta_n^L(t)}\right)\right]$$

$$\bar{O}_n^T(t + 1) \approx \left[1 - w_n^T(t)\right][1 - q_n^T(t)] \cdot O_n^T(t) \left[1 - \frac{v_n^T(t)}{l/2 + \Delta_n^T(t)} \left(2 - \frac{v_n^T(t)}{l/2 + \Delta_n^T(t)}\right)\right]$$

(92)

$$+ w_n^T(t) \cdot \left\{O_n^T(t) \left[1 - \frac{v_n^T(t)}{l/2} \left(2 - \frac{v_n^T(t)}{l/2}\right)\right] + \alpha_n^T(t + 1)\right\}$$

The last aspect we have to deal with is the calculation of the moving speed $v_n^i(t)$ for platoons in full-service mode. The moving speed estimation derived in Daganzo and Pilachowski (2011) and Petit et al. (2018) and used in Model I is not valid here since the derivation is based on the assumption of unlimited vehicle capacity. In their derivation, the bus travel time includes the cruising time and the dwell time for passenger boarding. With no capacity constraint, all passengers can board the first bus and there will be no stranded demand. Therefore, the dwell time is simply a function of the arrival rate and the headway, which
contributes to the closed-form formulas for the moving speed. In Model II, however, we must consider the capacity constraint and the stranded demand when determining the dwell time. To this end, we propose to determine the moving speed $\hat{v}_n^i(t)$ for a full-service platoon by the following nonlinear equation system.

$$\frac{v_n^{i,1}(t)}{E} + B \left[ \lambda s_n(t) + \int_{\mathcal{D}(v_n^{i,1}(t), v_n^{i,1}(t))} \pi(\omega;t) d\omega \right] = 1$$

$$\frac{v_n^{i,2}(t)}{E} + B \left\{ K_n^i(t) - O_n^i(t) \left[ 1 - \frac{v_n^{i,2}(t)}{l/2} \right] \right\} = 1$$

$$\hat{v}_n^i(t) = \max \{v_n^{i,1}(t), v_n^{i,2}(t)\} \leq E$$

where $v_n^{i,1}(t)$ and $v_n^{i,2}(t)$ are artificial variables introduced as unknowns for Eqs. (93) and (94), respectively. The derivation, validation and solution for this nonlinear system are detailed in Appendix C.

Finally, we need an adjustment to the objective function. Since we explicitly model the excess demand due to the capacity constraint, the in-vehicle crowding cost in the cost function is not necessary. Instead, we need to measure the expected extra waiting time cost for the excess demand, which is given as the product of the number of excess passengers and the expected headway, i.e.,

$$C_n^{\text{excess}}(t) = \mu H / V \left\{ u_n(t) \cdot [\alpha_n^L(t) - \alpha_n^T(t)] + [1 - u_n(t)] \cdot [\alpha_n^L(t) - \alpha_n^T(t)] \right\}$$

$$= \mu H / V \left\{ u_n(t) \cdot [\alpha_n^T(t) - \alpha_n^L(t)] + [1 - u_n(t)] \cdot [\alpha_n^L(t) - \alpha_n^T(t)] \right\}$$

The cost function now consists of three components, namely the time cost for passengers who succeed in boarding the bus ($C_n^{\text{pax}}$), the bus operational cost ($C_n^{\text{opr}}$) and the extra waiting time cost for the excess demand ($C_n^{\text{excess}}$). The overall infinite-horizon stochastic optimization model for a modular bus system with vehicle capacity constraint is established as

$$\min_{\hat{z}, \hat{k}} Z = \sum_{t=0}^{\infty} \sum_{n=1}^{N} \mathbb{E} \left[ C_n^{\text{pax}}(t) + C_n^{\text{opr}}(t) + C_n^{\text{excess}}(t) \right]$$

s.t. (23) - (26), (28) - (29), (32) - (37), (40), (43) - (53), (55), (58) - (59), (74) - (95)

### 3 Solution Algorithm

The infinite-horizon stochastic optimization models developed in the previous section require us to make decisions $\hat{z}_n^i(t)$, $\hat{z}_n^e(t)$ and $k_n(t)$ (for Model I) or $\hat{z}_n^i(t)$, $\hat{z}_n^e(t)$ and $\hat{k}_n(t)$ (for Model II) for each bus in an actionable state in an MAV-aided transit system, specifying when (to decide) to decouple and the number of units in each platoon when decoupling. It is highly dynamic and stochastic with complicated interactions between numerous system state variables and decisions. In this section, we propose to use the technique of reinforcement learning to provide a satisfactory solution to the two models. Since the time when different buses appear in an actionable state is not synchronized, we hence regard the buses as decision agents who make asynchronized decisions to minimize their own long-term expected costs. The buses will be trained to improve their decision policy based on the feedback of their interactions with the environment. For conciseness, we unify the notations of variables for both models by additionally defining variables $\hat{z}_n^i(t) = z_n^i(t)$, $\hat{z}_n^e(t) = z_n^e(t)$, $\hat{k}_n(t) = k_n(t)$, $\hat{r}_n(t) = 0$ and $\hat{u}_n(t) = 0$ for Model I.
3.1 Recursive decision for each bus

The recursive decision procedure faced by a given bus $n$ during the operation horizon can be formally described as follows. We first denote the state of the system at time $t$ by $S_t^s$; i.e., it is a collection of $w(t) = \{w^i_n(t)\}_{i \in \{L,T\}, n \in \mathcal{N}}$, $q(t) = \{q^i_n(t)\}_{i \in \{L,T\}, n \in \mathcal{N}}$, $u(t) = \{u_n(t)\}_{n \in \mathcal{N}}$, $\hat{u}(t) = \{\hat{u}_n(t)\}_{n \in \mathcal{N}}$, $\delta(t) = \{\delta^i_n(t)\}_{i \in \{L,T\}, n \in \mathcal{N}}$, $\hat{\delta}(t) = \{\hat{\delta}_n(t)\}_{n \in \mathcal{N}}$, $\hat{\gamma}(t) = \{\hat{\gamma}_n^i(t)\}_{i \in \{L,T\}, n \in \mathcal{N}}$, $O(t) = \{O^i_n(t)\}_{i \in \{L,T\}, n \in \mathcal{N}}$ and $K(t) = \{K^i_n(t)\}_{i \in \{L,T\}, n \in \mathcal{N}}$. These are the necessary variables required to fully capture the system at time $t$. Other variables such as speed and spacing can be derived from these variables.

When bus $n$ decides $\hat{z}^l_n(t)$, $\hat{z}^e_n(t)$ and $\hat{k}_n(t)$ at time $t$, we say a decision event happens for bus $n$ at time $t$. Since a bus only makes decisions when it is in an actionable state, it will not experience a decision event at every time step. Moreover, after a bus decides to decouple in a decision event, when it will experience the next decision event is random since the duration of a decoupling epoch is random. Nevertheless, a bus will sequentially experience decision events with the same structure during the operation horizon.

To differentiate decision events and system time steps. We use $\tau$ to index the order of decision events. The state of the system at the beginning of decision event $\tau$ from the perspective of bus $n$ is denoted by $S_{n,\tau}^s$. When bus $n$ observes the system, it considers itself the origin and the first bus, and all bus indices and locations are converted to the relative indices and locations with respect to bus $n$. Hence, state $S_{n,\tau}^s$ is an absolute description of the system while state $S_{n,\tau}$ is a relative description of the system by considering bus $n$ as the reference point. We then encode the decision $\hat{z}^l_n(t)$, $\hat{z}^e_n(t)$ and $\hat{k}_n(t)$ made at decision event $\tau$ at time $t$ as an action $a_{n,\tau} = [\hat{z}^l_n(t) - \hat{z}^e_n(t)] \cdot \hat{k}_n(t)$. Accordingly, $a_{n,\tau} = \{1-m, 2-m, \ldots, 0, 1, \ldots, m-1\}$. Note that given action $a_{n,\tau}$, we can uniquely determine the decisions made at decision event $\tau$. When $a_{n,\tau} = 0$,

\[ \text{Figure 8: State transition} \]
it means bus $n$ decides not to decouple at decision event $\tau$. When $a_{n, \tau} > (\leq) 0$, it means bus $n$ decides to launch a catch-up (holding) split with $|a_{n, \tau}|$ units in the leading platoon at decision event $\tau$.

Before making decisions at decision event $\tau$ at time $t$, the bus observes the state of the system $(S_{n, \tau} \leftarrow S_{n, \tau}')$. It then decides whether and how many units to decouple based on the state while considering its decision’s expected impact on the future. If it decides to take $a_{n, \tau} = 0$ at time $t$, then with the system advancing by one time step, the bus observes the realized cost $\hat{C}_\tau(S_{n, \tau}, a_{n, \tau})$ and enters the next decision event $\tau + 1$ at time $t + 1$. Then it will observe the state $S_{n, \tau+1} \leftarrow S_{n, \tau+1}'$ and take action $a_{n, \tau+1}$ based on state $S_{n, \tau+1}$. If bus $n$ decides to decouple ($a_{n, \tau} \neq 0$) at decision event $\tau$, then it enters a decoupling epoch with random duration. Say at time $t'$ bus $n$ recouples and enters decision event $\tau + 1$. Then the bus observes the realized cost $\hat{C}_\tau(S_{n, \tau}, a_{n, \tau})$ due to the action taken at decision event $\tau$, which equals the accumulated cost during the decoupling epoch from time $t$ to time $t'$, and the state $S_{n, \tau+1} \leftarrow S_{n, \tau+1}'$. Fig. 8 depicts an example of the state transition process in which the bus decides not to decouple at decision event $\tau - 1$ at time $t - 1$, then decides to launch at catch-up split at decision event $\tau$ at time $t$ and enters the next decision epoch $\tau + 1$ at time $t'$.

### 3.2 Deep Q-Network (DQN) algorithm

In order to determine the best decision policy for a bus, we develop a customized Deep Q-Network (DQN) algorithm based on the idea of Deep Q-learning, which has achieved promising performance in off-policy optimization for dynamic problems with discrete action space (Dong et al., 2020). In what follows, we will describe the routine design of a DQN algorithm in the context of our problem. Then we will discuss how we adapt the routine design based on the features of our problem.

DQN uses a neural network $Q(s, a; \theta)$ parameterized by $\theta$ to approximate the state-action value function $J(S_{n, \tau}, a_{n, \tau})$ for each bus, which is defined as

$$J(S_{n, \tau}, a_{n, \tau}) = \mathbb{E}[C(S_{n, \tau}, a_{n, \tau}) + \gamma \min_{a} J(S_{n, \tau+1}, a)|S_{n, \tau}, a_{n, \tau}]$$

(98)

where $C(S_{n, \tau}, a_{n, \tau})$ is the sum of passengers’ cost, bus operational cost and in-vehicle crowding (for Model I) or extra waiting time (for Model II) cost incurred during the period from decision event $\tau$ to decision event $\tau + 1$. It is worth noting that the calculation of passengers’ cost for $C(S_{n, \tau}, a_{n, \tau})$ is slightly different from Eq. (63) in which the passengers’ cost for bus $n$ is the sum of the schedule deviation cost and the waiting time cost for bus $n$. Here, the passengers’ cost for bus $n$ is the sum of the schedule deviation cost for bus $n$ and the waiting time cost for its immediately following bus. We make such an adjustment because bus $n$’s decision has a more significant impact on the headway and speed (which determine the waiting time) of its immediately following bus rather than itself. Therefore, it may better establish the relationship between different decisions and the changes in cost during the training process. Note that although we adjust the calculation, the sum of $C(S_{n, \tau}, a_{n, \tau})$ over all buses still equals the original objective function. Based on our preliminary experiments, this adjustment indeed yields better performance in terms of cost reduction and result stability. $\gamma \in (0, 1)$ is a coefficient discounting the future cost. For bus $n$, the state-action value function represents the expected long-term cost of taking action $a_{n, \tau}$ given state $S_{n, \tau}$ at the beginning of decision event $\tau$.

Note that the notation $J$ does not have a bus index $n$, because all buses share a common state-action value function due to the symmetry of the bus route, homogeneity of the buses, and relativity of the observed state $S_{n, \tau}$. Hereinafter, we will drop the subscript $n$ in $S_{n, \tau}$, $a_{n, \tau}$ and $\hat{C}_{n, \tau}$ whenever it is unnecessary to specify the bus index, implying they apply to any bus.
During the training process, a bus will recursively observe the current state $S_t$, determine the possible action set $A_{n, \tau}$, take an action $a_\tau \in A_{n, \tau}$ using some policy (e.g., $\epsilon$-greedy) based on the current Q network, and then observe the cost $\hat{C}_t$ and the next state $S_{t+1}$. The possible action set $A_{n, \tau}$ includes actions that will be considered in the current decision event. For example, a late/early bus will not launch a holding/catch-up split. Moreover, for Model II, a bus will not consider infeasible splits that lead to a pre-split phase unless there is no feasible split. A tuple of $\{S_t, a_\tau, \hat{C}_t, S_{t+1}\}$ is called a transition. As the buses interact with the system, they will collect many transitions. An important component of DQN is the replay buffer $M$ where a collection of recent transitions are stored (i.e., transitions are stored in first-in-first-out (FIFO) principle) (O’Neill et al., 2010). At every training step, a mini-batch of transitions $\mathcal{B}$ will be randomly drawn from the replay buffer. For each sampled transition $\{S_t, a_\tau, \hat{C}_t, S_{t+1}\} \in \mathcal{B}$, a target network $\hat{Q}(s; a; \hat{\theta})$ is used to compute the target output $\hat{J}$ by

$$\hat{J} = \hat{C}_t + \gamma \min_{a \in A_{n, \tau}} \hat{Q}(S_{t+1}; a; \hat{\theta})$$

(99)

Then the input $(S_t, a_\tau)$ and the corresponding output $\hat{J}$ will serve as one of the data points for training the Q network $Q(s; a; \theta)$. The target network $\hat{Q}(s; a; \hat{\theta})$ has the same structure with the Q network but its parameter $\hat{\theta}$ has a lower update frequency than $\theta$ (i.e., $\hat{\theta}$ is synchronized with $\theta$ for every $t_{syn}$ steps where $t_{syn}$ is a hyperparameter).

Note that for both $S_t$ and $S_{n, \tau}$, the state space is $\Omega = \{0, 1\}^{2|\mathcal{M}|} \times \{0, 1\}^{2|\mathcal{M}|} \times \{0, 1\}^{2|\mathcal{M}|} \times \mathbb{R}^{2|\mathcal{M}|} \times \mathbb{R}^{2|\mathcal{M}|} \times \mathbb{R}^{2|\mathcal{M}|}$. Approximating the state-action value function with such a large scale of state variables may be inefficient. Moreover, in real-world operations, the global information regarding the whole transit system may not be available to each individual bus. Therefore, we assume that when each bus takes an action, it only references the local information most relevant to the cost computation, rather than all information about the system. Specifically, based on the cost formulation for $C(S_{n, \tau}, a_{n, \tau})$, we select the deviation $\delta_n(t)$, occupancy $O_n^t(t)$ and spacing faced by the immediately following leading platoon $s_{n+1}^t(t)$ as the local information referenced by bus $n$ when it makes decisions at time $t$.

Another significant challenge in applying reinforcement learning to solve our problem lies in the imbalanced data set. As shown in Fig. 8, when a bus takes $a_\tau = 0$, it observes the cost immediately at the next time step. Hence, action $a_\tau = 0$ produces a transition every time step. However, when a bus decides to split at time $t$, it cannot determine the cost from the splitting action until the decoupling epoch ends. That is, it takes several time steps for a bus to collect a transition with $a_\tau \neq 0$. As a result, the replay buffer will be dominated by transitions with $a_\tau = 0$, which makes it hard for the bus to learn from the decoupling experiences with $a_\tau \neq 0$. Moreover, when a bus decides not to decouple at decision event $\tau$ at time $t$, it observes the cost $\hat{C}_\tau$ at time $t+1$, where $\hat{C}_\tau$ is the cost incurred within one time step. In contrast, when a bus decides to decouple at decision event $\tau$, the cost it observes at the end of the decoupling epoch is the cost accumulated during multiple time steps. Therefore, the observed cost for action $a_\tau \neq 0$ is expected to be multiple times greater than that for action $a_\tau = 0$. In other words, a decoupling action may be frequently related to a greater observed cost. This correlation will be learned by the Q network during the training process. As a result, the bus may tend to stay in the coupled state even when the deviation is large.

To address these issues, we propose two remedies. First, instead of maintaining one FIFO replay buffer for all transitions, we maintain multiple FIFO replay buffers, one for each action value. There are hence $2m - 1$ buffers in our DQN algorithm. $M(a), a \in \{1 - m, 2 - m, ..., 0, 1, ... m - 1\}$ represents the replay buffer for action $a_\tau = a$ in which all transitions has action $a_\tau = a$. At each training step, the same number of transitions is sampled from each of the buffers. By doing this, the Q network sees the expected results of different actions when updating $\theta$, and thus may gain a better performance in predicting the state-action
Algorithm 1: DQN reinforcement learning algorithm

1 **Initialization:** Q network \( Q(S, a; \theta) \), target network \( \hat{Q}(S, a; \hat{\theta}) \leftarrow Q(S, a; \theta) \), FIFO replay buffers \( \mathcal{M}(a) = \emptyset \) with capacity \( N \) each, synchronization frequency \( t_{syc} \), maximum training time step \( T_{max} \)

2 **Warm start:** initialize the system, then implement random actions to collect transitions \( \{S_t, a_t, \hat{C}_t, S_{t+1}\} \) to fill up the replay buffers \( \mathcal{M}(a) \) until all buffers have been filled.

3 Initialize the system, set \( t = 0 \) and initial state \( S_0^s \)

4 while \( t < T_{max} \) do

5     for bus \( n \in \mathcal{N} \) do

6         if bus \( n \) is in an actionable state then

7             Determine the possible action set \( \mathcal{A}_{n,t} \)

8             if \( |\delta^L_n(t)| > H / 2 \) then

9                 Decouple the bus, use \( \epsilon \)-greedy policy to select \( a_{n,t} \in \mathcal{A}_{n,t} \setminus \{0\} \)

10                else

11                    Choose an action \( a_{n,t} \in \mathcal{A}_{n,t} \) using \( \epsilon \)-greedy policy

12             else

13                 Record the cost \( \hat{C}_{n,t} \leftarrow \hat{C}_{n,t} + C_n(t - 1) \)

14         Implement the actions \( a_{n,t} \) for all buses in coupled state and advance the system

15         Observe the costs \( C_n(t) = C_{n}^{\text{max}}(t) + C_{n}^{\text{opt}}(t) + C_{n}^{\text{crowd}}(t) \), and the next state \( S_{t+1}^s \)

16     for bus \( n \in \mathcal{N} \) do

17         if bus \( n \) is in an actionable state then

18             Record the cost \( \hat{C}_{n,t} \leftarrow \hat{C}_{n,t} + C_n(t) \)

19             Observe the system from bus \( n \)'s perspective \( S_{n,t+1}^s \leftarrow S_{t+1}^s \)

20             Put transition \( \{S_{n,t}, \hat{C}_{n,t}, a_{n,t}, S_{n,t+1}\} \) in buffer \( \mathcal{M}(a_{n,t}) \) and removing the oldest transition

21             Set \( S_{n,t} \leftarrow S_{n,t+1} \) and reset \( \hat{C}_{n,t}, a_{n,t}, S_{n,t+1} \) to null

22         Draw a mini-batch of transitions from the buffers

23         Calculate for each sampled transition the target variable \( \hat{J} \)

24         Perform one step of stochastic gradient descent updating \( \theta \) based on the sampled transitions and the corresponding target output

25         Synchronize the target network \( \hat{Q}(S, a; \hat{\theta}) \leftarrow Q(S, a; \theta) \) every \( t_{syc} \) steps

26     \( t \leftarrow t + 1 \)

27 return \( Q(S, a; \theta) \).
value for $a_\tau \neq 0$ even though these actions take a long time to produce a new transition. Second, we introduce a ruin state with a penalty in the training process so that the bus will be aware that $a_\tau = 0$ is an unfavorable action when the deviation is large. Specifically, when a bus takes action $a_\tau = 0$ which brings it to the next state with $|\delta^L_n(t)|$ greater than some threshold, say $H/2$, we say the bus triggers a ruin state and it is forced to decouple at decision event $\tau + 1$ to recover from the deviation. Then, when calculating the target output $\hat{J}$ of the transition $\{S_\tau, a_\tau, \hat{C}_\tau, S_{\tau+1}\}$ triggering the ruin state, we apply a slightly different calculation from Eq. (99).

$$\hat{J} = \begin{cases} \hat{C}_\tau + \gamma \min_{a \in \mathcal{A}_n} \hat{Q}(S_{\tau+1}; a; \hat{\theta}), & \text{if } \hat{C}_\tau + \gamma \min_{a \in \mathcal{A}_n} \hat{Q}(S_{\tau+1}; a; \hat{\theta}) > Q^*(S_\tau; \theta) \\ Q^*(S_\tau; \theta) + e, & \text{otherwise} \end{cases} \quad (100)$$

where $Q^*(S_\tau; \theta) = \min_{a \in \mathcal{A}_n \setminus \{0\}} Q(S_\tau; a; \theta)$ is the predicted optimal cost if the bus decouples given state $S_\tau$ based on the current Q network, and the penalty term $e = Q^*(S_\tau; \theta) - [\hat{C}_\tau + \gamma \min_{a \in \mathcal{A}_n} \hat{Q}(S_{\tau+1}; a; \hat{\theta})]$ denotes the difference between $Q^*(S_\tau; \theta)$ and the target output calculated using Eq. (99). The motivation of Eq. (100) is to produce a target value higher than $Q^*(S_\tau; \theta)$, thereby reinforcing the fact that given state $S_\tau$ taking action $a_\tau = 0$ yields a higher cost than taking a decoupling action based on the Q network. If the target value calculated using Eq. (99) is already greater than $Q^*(S_\tau; \theta)$, we use the target value directly. Otherwise, the target output is $Q^*(S_\tau; \theta)$ plus a positive penalty $e$ which is adaptive based on the difference. The design of the ruin state with a penalty resembles episode-based reinforcement learning. When the system reaches a terminating state, the current episode ends and the system is reset to start a new episode. And the terminating transition will have zero future rewards when calculating the target output. In our case, the ruin state acts as a terminating state, and the penalty imposed serves as a counterpart of the zero future rewards in a return maximization setting. The ruin state condition is set as $|\delta^L_n(t)| > H/2$ because this is the exact threshold to prevent bus bunching. If each bus is not allowed to deviate from the scheduled trajectory by $H/2$, then it is guaranteed that no bus bunching will happen under any situation since their trajectories will not cross each other. Bus bunching may happen whenever the deviation is allowed to be greater than $H/2$. Any condition stricter than $|\delta^L_n(t)| > H/2$ will also provide a bunch-proof guarantee, but these conditions will be triggered more frequently, which may severely distort the approximation of the state-action value function due to the penalty imposed.

The DQN reinforcement learning algorithm is summarized in Algorithm 1.

### 4 Numerical Example

In this section, we perform numerical experiments to validate and showcase the effectiveness of our models and solution algorithm. For both models, we begin by presenting an illustrative example involving two vehicles, each consisting of two units, to demonstrate the optimal bus splitting and merging operations obtained from the DQN algorithm. Subsequently, we conduct a comprehensive strategy comparison and sensitivity analysis across various operational scenarios.

#### 4.1 Model I

##### 4.1.1 Illustrative example

The system configurations for the illustrative example are set primarily based on the settings in Petit et al. (2018). Specifically, the bus system consists of $N = 2$ buses with $m = 2$ units each serving a bus route of
length $l = 5$ km at cruising speed $E = 25$ km/h. The system time step is set as 1 minute and the operation horizon lasts for 4 hours. The standard deviation of the trajectory noise for a coupled bus is $\xi = 0.045$ km/min (Daganzo and Pilachowski, 2011). For platoons in drop-off-only and deadheading modes, the standard deviations of noise ($\xi$ and $\hat{\xi}$) are set as half and a quarter of $\xi$, respectively. The nominal capacity of each unit is $K = 10$ passengers. The passenger demand rate is set as $\lambda = 60$ pax/h. At this demand rate, the equilibrium occupancy of each bus will be approximately 11 passengers which is about half of the bus capacity$^6$. The boarding and alighting times per passenger are set as $B = 8$ s/pax and $A = 4$ s/pax, respectively. The distance threshold enabling two units to recouple is set as $\rho = 1$ m. During a holding split, when the leading platoon’s occupancy is less than $\eta = 0.5$ passenger, we assume the platoon is empty. The value of time is set as $\mu = $10/h. The running cost for the bus in coupled state $c^c$ is set as $37/veh-h$. The splitting cost for a pair of decoupling and recoupling operations is set as $c^s = $5. The crowding cost function parameters are set as $\beta = 2$ and $\psi = $2. To capture the impact of platooning on the bus running cost, we adopt the cost function for truck platooning proposed in Abdolmaleki et al. (2021). Specifically, the per distance bus running for a platoon with $x$ units is $c^u(x) = c^c \alpha x/m$, where $\alpha = 1.06$ according to the empirical result in Zabat et al. (1995).

For the DQN algorithm, since the Q network only has 3 inputs, we hence use an architecture with a modest structure for the Q network and target network. Both the Q network and target network have 2 hidden layers with 24 neurons each. The activation function is ReLU function which generally has a more stable performance than Sigmoid function. The discount factor is set as $\gamma = 0.99$. We maintain a buffer of 200 capacity for each action. 10 transitions will be sampled from each buffer at each training step. The target network synchronization frequency is set as 30 training steps. The algorithm runs for $T_{max} = 200,000$ steps. An $\epsilon$-greedy policy is used for action selection; i.e., a bus will choose a random action with probability $\epsilon$ and choose the best action based on the current Q network with probability $1 - \epsilon$. $\epsilon$ is set as 1 at the beginning and is annealed towards 0 following a harmonic sequence. This will ensure both enough exploration at the first iterations and adequate exploitation after some useful experience has been learned.

Fig. 9 shows how the average costs evolve during the DQN training process, which illustrates the convergence property of the DQN algorithm. In the first few iterations, the average costs are relatively low because we use an ideal state (i.e., each bus is evenly spaced with zero deviation and equilibrium occupancy) as the initial state. However, such an initial state is not stable, so the costs increase after a few iterations as the system deviates from the equilibrium state. When the Q network is not sufficiently trained at the early stage of the training process, the decision policy is inferior with the average costs maintained at a relatively high level. As the training process proceeds, the Q network starts to predict more accurately and produce a high-quality decision policy, which consistently reduces the average costs in terms of the passenger cost, bus operational cost, in-vehicle crowding cost, and total cost. At the final stage, the cost reduction trend tends to diminish, indicating convergence of the algorithm.

After obtaining the trained Q network from the DQN algorithm, we implement the strategy by selecting the action with best state-action value predicted by the Q network. Fig. 10 demonstrates the strategy’s capability of preventing bus bunching. In Fig. 10(a), we plot the schedule deviation of the two buses when no bunching alleviation strategy is applied. It shows the typical process of how bus bunching happens. Recall that bus 1 and bus 2 are the only buses on the loop-shaped bus route. As bus 2 runs ahead of its schedule at the first few time steps, it picks up fewer passengers while bus 1 has to pick up more passengers, which makes bus 1 gradually fall behind schedule. As a result, the spacings of bus 1 and bus 2 become more

---

$^6$The equilibrium occupancy $O_{eq} = \frac{\lambda H}{2V(L/E - 2V/E^2)}$ according to Petit et al. (2018)
Figure 9: Convergence of the DQN algorithm
Figure 10: Schedule deviation with (a) no control and (b,c) MAV-aided bunching alleviation strategy
and more uneven as the system advances, which finally causes the buses to bunch at time step 29. When they bunch, bus 1 falls behind schedule by approximately 1.5 km and bus 2 runs ahead by around 1 km. After the bunching happens, they are not able to recover with the absence of intervention, and overtake each other alternately along the route. Fig. 10(b) and (c) are the schedule deviations for both platoons of bus 1 and bus 2, respectively, when we implement our strategy in the same scenario. We use solid and dotted lines to represent the leading and trailing platoons’ deviation, respectively. When a bus is in coupled state, the two lines overlap and only the solid line is visible. It is obvious that multiple holding splits (the ones with negative deviations) are launched during the operation horizon. After each holding split, the deviation of the early bus is reduced to different degrees. Note that though the leading platoon may experience a large deviation after launching a holding split, it imposes little impact on the bus system. This is because the leading platoon switches to drop-off-only mode at the moment when a holding split is launched. It then functions similarly as a “retired bus” in Petit et al. (2018) that should not be considered a working bus in the system. If we focus only on the solid lines, we find that the deviation of both buses remains within the interval of [-0.7km, 0.5km]. This interval is a fairly safe deviation range implying bus bunching does not happen. Catch-up splits are not as apparent as holding splits since they tend to be launched when the bus falls only slightly behind schedule to realize a quick recovery from schedule deviation. Moreover, the strategy tends to keep the bus a little ahead of schedule. This phenomenon might be attributed to two reasons. First, though the schedule deviation cost (i.e., the first term of Eq. (63)) is symmetric for both positive and negative deviations, the waiting time cost (i.e., the second term of Eq. (63)) is not. An early bus is expected to have a smaller spacing from the preceding bus and a greater moving speed. Therefore, an early bus usually means less waiting time costs imposed on passengers, which makes early buses preferable to late buses. Second, holding splits tend to be more controllable than catch-up splits. In a typical holding split, both the leading and trailing platoons will switch to dwelling mode with no trajectory uncertainty for some while, and the leading platoon will dwell to wait for the trailing platoon to catch up. In contrast, trajectory noises exist for both leading and trailing platoons throughout a catch-up split. A catch-up split launched by a late bus with a large deviation may suffer from great uncertainty which extends the decoupling epoch, thereby leading to increased running and crowding costs during the split. Given that the system is susceptible to disturbances, it is advantageous to maintain the system in a more favorable and manageable state.

4.1.2 Strategy comparison

In this subsection, we compare the strategy obtained from the DQN algorithm (MAV-DQN) with alternative strategies for both MAV-aided and conventional transit systems across various operation scenarios. For alternative strategies applicable to MAV transit systems, we use a myopic strategy as a benchmark. The myopic strategy is inspired by the threshold strategy proposed in Khan et al. (2023) and Khan and Menendez (2023) in which a decision of splitting or holding is made when the realized headway meets a threshold condition. However, their threshold policy is proposed for a discrete model and applies only to the case with two modular units for each bus. Nevertheless, we adopt a similar decision threshold based on the deviation to decide when to split. A split is launched whenever the deviation is greater than half of the scheduled spacing, similar to the threshold of 1.5 times the scheduled headway tested in Khan et al. (2023). To determine the number of units to split, we adopt a myopic rule. Since a split will result in capacity reduction and increase the crowding cost, we always select the number of units that leads to a minimum crowding cost increase based on the current occupancy. For the alternative strategy designed for conventional transit systems, we use the two-way-looking strategy proposed by Daganzo and Pilachowski (2011) as a benchmark. The two-way-looking strategy is a speed adjustment strategy based on a CA model which can be readily implemented
within our modeling framework for comparison.

Table 2: Demand rates (pax/h) for different combinations of \( N \) and \( m \)

| \( N = 2 \) | \( m = 2 \) | \( m = 3 \) | \( m = 4 \) | \( N = 3 \) | \( m = 2 \) | \( m = 3 \) | \( m = 4 \) | \( N = 4 \) | \( m = 2 \) | \( m = 3 \) | \( m = 4 \) | \( N = 5 \) | \( m = 2 \) | \( m = 3 \) | \( m = 4 \) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 60 | 70 | 80 | 45 | 55 | 60 | 35 | 45 | 50 | 30 | 38 | 45 |

To compare the performance of the three strategies in different operation scenarios, we implement them in transit systems with \( N \in \{2,3,4,5\} \) and \( m \in \{2,3,4\} \). The route lengths are set such that the equilibrium spacing is \( H = 2.5 \) km. The demand rates for these scenarios are shown in Table 2, which ensures the equilibrium occupancy is approximately half of the bus capacity. Other parameters are set the same as specified in the illustrative example. For each scenario, we test the three strategies with an identical set of 20 realizations of the random noises, and then assess the average performance per bus over the 20 tests.

Table 3: Average performance per bus for different strategies (4-hour operation horizon)

<table>
<thead>
<tr>
<th>( N )</th>
<th>( m )</th>
<th>Two-way-looking</th>
<th>MAV Myopic</th>
<th>MAV DQN</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_{\text{pax}} )</td>
<td>( C_{\text{opr}} )</td>
<td>( C_{\text{crowd}} )</td>
<td>( C_{\text{total}} )</td>
<td>( C_{\text{pax}} )</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1004</td>
<td>75</td>
<td>503</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1511</td>
<td>81</td>
<td>387</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2335</td>
<td>80</td>
<td>422</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>564</td>
<td>68</td>
<td>389</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>851</td>
<td>70</td>
<td>345</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>1019</td>
<td>70</td>
<td>353</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>413</td>
<td>70</td>
<td>364</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>613</td>
<td>72</td>
<td>310</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>726</td>
<td>72</td>
<td>285</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>336</td>
<td>67</td>
<td>361</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>469</td>
<td>69</td>
<td>295</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>607</td>
<td>70</td>
<td>327</td>
</tr>
<tr>
<td>Average</td>
<td>2274</td>
<td>37</td>
<td>183</td>
<td>2494</td>
</tr>
</tbody>
</table>

All three strategies can provide bunch-proof operations during the operation horizon, and Table 3 presents their passenger cost, bus operational cost, in-vehicle crowding cost, and total cost. We can see that the MAV-DQN strategy outperforms the other two strategies in terms of passenger cost and total cost across all fleet configurations. The two-way-looking strategy for a conventional transit system has the lowest bus operational cost since it does not involve splitting and merging operations. Also, it has the lowest crowding cost on average compared to other strategies because its capacity does not vary and it produces highly regular headways such that the occupancy does not fluctuate dramatically. However, it yields significantly higher passenger costs in all scenarios. This is because it achieves regular headways at the expense of reduced commercial speed which makes all buses fall behind schedule at the same pace. While the myopic strategy has a lower bus operational cost than MAV-DQN strategy, it causes higher passenger and in-vehicle crowding costs. This indicates that the myopic strategy may not have taken full advantage of the splitting
and merging operations. The MAV-DQN strategy, in contrast, makes more intelligent use of the MAVs to significantly reduce passenger and crowding costs with a moderate increase in the bus operational cost.

Besides comparison with different fleet settings, we also conduct sensitivity analyses to evaluate the strategy performance with different system parameters. First, we test the effect of different demand rates with other parameters fixed as the values we set in the illustrative example. For each demand rate, we test the performance for each strategy with 20 different random seeds and assess the average performance. Fig. 11 shows the mean and range of the cumulative costs against time under different strategies with different demand rates. It is obvious that the MAV-DQN strategy has a significantly lower variance in its performance, indicating its enhanced stability. Bus bunching happens for the uncontrolled systems under different demand rates. The slope change indicates that after bunching happens, the cumulative cost will steadily increase at a larger rate. Moreover, bus bunching tends to happen earlier as the demand rate increases. This is because when there are more passengers arriving, a late bus will suffer from more delay, which accelerates the bus bunching. The total cost for both the myopic strategy and the MAV-DQN strategy over the 4 hours of operation increases with the increase in demand rate because of the mounting passenger cost and crowding cost, but they are both bunch-proof. It is worth noting that the benefit of implementing the MAV-DQN strategy compared with the myopic strategy becomes more prominent under scenarios with higher demand.

![Figure 11: Strategy performance under different levels of demand](image)

We then compare the system performance under different levels of trajectory noises by varying the value of $\sigma$. A larger $\sigma$ indicates greater disturbance on the bus trajectories. For each value of $\sigma$, we run 20 independent tests with different random seeds for each strategy and assess the average performance. The result is presented in Fig. 12. As expected, as the bus trajectories suffer from more randomness, an uncontrolled system will be more fragile and bus bunching happens earlier. Moreover, when the uncertainty is high, the myopic strategy results in higher cost and performs less stably (Fig. 12(c)) though it is able to avoid the two buses bunching together, whereas the MAV-DQN strategy exhibits relatively stable performance across scenarios with varying levels of noise.
Figure 12: Strategy performance under different levels of trajectory noises

(a) $\sigma = 0.025$
(b) $\sigma = 0.045$
(c) $\sigma = 0.065$

Figure 13: Convergence of the DQN algorithm

(a) Passenger cost
(b) Bus operational cost
(c) Excess cost
(d) Total cost
Figure 14: Schedule deviation with (a) no control and (b,c) MAV-aided bunching alleviation strategy.
4.2 Model II

4.2.1 Illustrative example

For Model II, the illustrative example adopts the same parameter setting for the operating scenario and DQN algorithm as specified for Model I. Fig. 13 shows the convergence of the DQN algorithm applied to Model II. Due to the similar structure between Model I and Model II, the DQN algorithm exhibits similar convergence stages when applied to both models.

Fig. 14 presents the trajectory deviation for the system with and without the control of MAV-DQN strategy. When there is no control, the system is not stable and the two buses bunch together at time $t = 36$. Similar to Model I, the MAV-DQN strategy for Model II successfully prevents the bus bunching from happening. However, the controlled deviation in Figs. 14(b) and (c) tends to be evenly distributed around zeros while in Model I the MAV-DQN strategy tends to maintain both buses slightly ahead of schedule (see Figs. 10(b) and (c)). This may relate to the presence of pre-split phases. Note that pre-split phases are more likely to occur to catch-up splits as late buses tend to carry more passengers. Catch-up splits hence may not take effect as quickly as they do in Model I due to the potential delayed execution. On the other hand, in pre-split phases, the whole bus switches to drop-off-only mode, which leads to some passengers being skipped. Such degradation in the level of service will be captured by the increased waiting time cost for the immediately following bus. As a result, pre-split phases are usually associated with higher costs. Therefore, the cost incurred during the pre-split phases prevents the bus from launching catch-up splits as frequently as it does in Model I where there is no pre-split phase and the passenger service is uninterrupted by the splits.

Note that whether to enter a pre-split phase depends on the demand relative to the capacity. With more spare capacity, a bus may be less likely to enter a pre-split phase, and the system may behave as what we see for Model I. Moreover, entering a pre-split phase may be an optional operation in real-world implementation depending on how the operator assesses the current occupancy relative to the capacity. For example, an operator may decide to directly launch the split without entering a pre-split phase if they think the overloading issue is acceptable. Therefore, in practice, a preferable pattern may be somewhere between the two cases (Figs. 10 and 14).

4.2.2 Strategy comparison

In Table 4, we compare the performance of the two-way-looking, MAV-Myopic and MAV-DQN strategies across different fleet configurations. Although the two-way-looking strategy is developed based on its performance under the unlimited capacity assumption, it is also implementable under the setting with hard capacity constraint as it is simply a speed-changing strategy that does not rely on the assumption. Specifically, the bus speed will be adjusted according to the two-way-looking rule while the simulation of system dynamics and the cost calculation will respect the hard capacity constraint as established in Section 2.2.2. From Table 4, we can see that under the current parameter setting, the two-way-looking strategy will not cause an overloading issue (and thus has no excess cost). Therefore, the two-way-looking strategy will perform as expected (as if there is no capacity constraint) and thus has the same passengers’ costs and operational costs as in Table 3. As discussed in Section 4.1.2, though the two-way-looking strategy has zero excess cost and lower operational cost, it has much higher passengers’ cost as it causes constant schedule deviation. Across all fleet configurations, the MAV-DQN strategy, compared with the MAV-Myopic strategy, can significantly reduce the passengers’ cost and excess cost at the expense of bus operational cost, which contributes to a substantial reduction in the total cost. This means the MAV-DQN strategy can leverage
Table 4: Strategy performance under different fleet configurations

<table>
<thead>
<tr>
<th>N</th>
<th>m</th>
<th>Two-way-looking</th>
<th>MAV Myopic</th>
<th>MAV DQN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>C_pax</td>
<td>C_bus</td>
<td>C_excess</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2828</td>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>3862</td>
<td>29</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>5177</td>
<td>25</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1657</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2386</td>
<td>35</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>2818</td>
<td>33</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>1094</td>
<td>44</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1675</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2022</td>
<td>37</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>849</td>
<td>46</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>1250</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1671</td>
<td>39</td>
<td>0</td>
</tr>
<tr>
<td>Average</td>
<td>2274</td>
<td>37</td>
<td>0</td>
<td>2311</td>
</tr>
</tbody>
</table>

more sophisticated operations to improve the level of service.

In Model II, the capacity has a profound impact on the system performance as it not only directly affects the excess cost but also mediates the moving speed of a platoon. Based on the illustrative example, we conduct a comparison among strategies across different MAV unit capacities. For each strategy and each capacity ($K$), we run 20 independent tests with different random seeds. In Table 5, we list the average per bus system performance with no control strategy. As expected, the excess cost decreases as the capacity increases. The passengers’ cost, however, increases with the increase in capacity. When unit capacity $K = 8$, the passengers’ cost is significantly lower than other scenarios. This is because the two buses tend to bunch later when $K = 8$. For $K = 8$, the buses bunch together at $t = 45$ on average while in other scenarios the two buses bunch much earlier. Later bunching implies a saving in the schedule deviation cost and thus in the passengers’ cost. The later bunching is a result of the self-regulating effect which appears as the bus becomes crowded. When a bus falls behind schedule, it tends to have more boarding passengers and thus a higher occupancy. Due to the hard capacity constraint, it may not be able to pick up all waiting passengers, which limits the mounting of the dwell time and in return accelerates the bus. The hard capacity constraint entails spontaneous limited boarding which prevents a crowded late bus from further delay. In other words, for a crowded late bus, the dwell time and moving speed are determined by its available capacity rather than the enlarged spacing to its preceding bus. This means consecutive buses are less susceptible to each other, implying higher system stability. It is worth noting that the self-regulating effect is more prominent when the buses tend to be crowded and comes with higher excess cost (the excess cost when $K = 8$ is significantly higher). The bus operational cost is almost the same across all scenarios since there are no split and merge operations. The scenario with $K = 8$ has a slightly higher operational cost due to the increased average speed from the self-regulating effect. The buses therefore travel a longer distance within the same planning horizon, which results in a higher distance-based bus running cost.

In Table 6, we present the average per bus performance of MAV-Myopic and MAV-DQN strategies across different MAV unit capacities. Compared with the uncontrolled system, the MAV-Myopic strategy can effectively reduce the passengers’ cost and the total cost at the expense of higher operational costs and excess costs. The higher excess cost for the MAV-Myopic strategy is due to the reduced capacity during
Table 5: Uncontrolled system performance under different MAV unit capacity

<table>
<thead>
<tr>
<th>K</th>
<th>C_pax</th>
<th>C_opr</th>
<th>C_excess</th>
<th>C_total</th>
<th>Ave. bunching time</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>1826.8</td>
<td>41.2</td>
<td>28.7</td>
<td>1896.7</td>
<td>45.0</td>
</tr>
<tr>
<td>10</td>
<td>1882.2</td>
<td>41.1</td>
<td>6.7</td>
<td>1930.0</td>
<td>35.3</td>
</tr>
<tr>
<td>12</td>
<td>1893.9</td>
<td>41.1</td>
<td>1.7</td>
<td>1936.8</td>
<td>32.5</td>
</tr>
<tr>
<td>14</td>
<td>1895.7</td>
<td>41.1</td>
<td>0.1</td>
<td>1936.9</td>
<td>32.2</td>
</tr>
</tbody>
</table>

A split whereas in the uncontrolled system the buses will not split. Similar to the uncontrolled system, the excess cost decreases while the passengers’ cost becomes higher as the capacity increases. The higher passengers’ cost is attributed to the weakened self-regulating effect, which results in buses tending to deviate from their schedule more quickly. This also explains why the MAV-Myopic strategy suggests launching more splits and hence endures higher operational costs as the capacity increases. Since the passengers’ cost and the operational cost are increasing while the excess cost is decreasing in $K$, the total cost is therefore not necessarily monotonic in $K$. As expected, the decoupling duration (i.e., the total time spent in decoupled state) increases with the number of splits. The pre-split duration (i.e., the total time spent in pre-split phase), however, is not monotonically increasing as we decide to launch more splits. Deciding to launch a split is only the precondition for a bus to enter a pre-split phase while the duration of a pre-split phase depends on the occupancy relative to the capacity at the instant of split decision. With more split decisions, a bus is thus more likely to enter a pre-split phase. On the other hand, larger capacity also implies shorter pre-split phases, which may explain the drop in pre-split duration for $K = 14$.

Table 6: Strategy comparison under different MAV unit capacity

<table>
<thead>
<tr>
<th>K</th>
<th>C_pax</th>
<th>C_opr</th>
<th>C_excess</th>
<th>C_total</th>
<th>Decoupling duration</th>
<th>Pre-split duration</th>
<th># of split executions</th>
<th># of split decisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>802.4</td>
<td>55.7</td>
<td>153.9</td>
<td>1011.9</td>
<td>43.3</td>
<td>5.6</td>
<td>2.9</td>
<td>2.9</td>
</tr>
<tr>
<td>MAV-Myopic</td>
<td>10</td>
<td>812.0</td>
<td>63.0</td>
<td>80.2</td>
<td>955.2</td>
<td>70.2</td>
<td>6.6</td>
<td>4.3</td>
</tr>
<tr>
<td>12</td>
<td>858.2</td>
<td>65.7</td>
<td>59.8</td>
<td>983.6</td>
<td>76.8</td>
<td>7.0</td>
<td>4.9</td>
<td>4.9</td>
</tr>
<tr>
<td>MAV-DQN</td>
<td>14</td>
<td>870.6</td>
<td>66.4</td>
<td>38.0</td>
<td>975.0</td>
<td>78.8</td>
<td>5.9</td>
<td>5.0</td>
</tr>
<tr>
<td>8</td>
<td>684.6</td>
<td>55.0</td>
<td>11.2</td>
<td>750.7</td>
<td>9.9</td>
<td>30.2</td>
<td>2.8</td>
<td>25.7</td>
</tr>
<tr>
<td>MAV-DQN</td>
<td>10</td>
<td>552.5</td>
<td>84.9</td>
<td>9.9</td>
<td>647.4</td>
<td>51.0</td>
<td>10.7</td>
<td>8.7</td>
</tr>
<tr>
<td>12</td>
<td>535.3</td>
<td>95.8</td>
<td>5.2</td>
<td>636.2</td>
<td>98.0</td>
<td>3.3</td>
<td>10.9</td>
<td>10.9</td>
</tr>
<tr>
<td>MAV-DQN</td>
<td>14</td>
<td>537.6</td>
<td>101.4</td>
<td>0.8</td>
<td>639.8</td>
<td>108.3</td>
<td>3.3</td>
<td>12.0</td>
</tr>
</tbody>
</table>

For the MAV-DQN strategy, we observe a similar overall trend for the number of split executions (increasing), operational cost (increasing) and excess cost (decreasing) as the capacity increases. However, the operations suggested by the MAV-DQN strategy under different scenarios are actually more nuanced than those by the MAV-Myopic strategy.

- For scenario with $K = 8$: the MAV-DQN strategy reduces the costs by means of a different mechanism from what we have seen for the MAV-Myopic strategy. During the operation horizon, the MAV-DQN strategy makes a great number of split decisions while few of them are executed, which also leads to a long pre-split duration. In other words, for most of the split decisions, the bus just enters a pre-split phase (due to the low capacity) and resumes full-service mode without splitting. In this case, the catch-up split decision is more like a stop-skipping decision as the bus just stops picking up passengers for a while and then resume full service. This happens when the bus prefers to decide on a catch-up
split when it is just slightly to moderately behind schedule. Moreover, such timely responses to schedule deviation result in shorter decoupling epochs and thus contribute to the significantly shorter decoupling duration. Compared with the MAV-Myopic strategy, the MAV-DQN strategy yields a much lower excess cost due to the shorter decoupling duration in which a bus has to provide service with reduced capacity. The operational cost is also comparable with that of the MAV-Myopic strategy since they execute approximately the same number of splits. Through the holding splits and frequent "stop skipping", the MAV-DQN strategy succeeds in reducing the schedule deviation cost without increasing the operational cost and the excess cost. It, however, causes additional waiting time due to the extended pre-split duration. This is why this scenario still has a significantly higher passengers’ cost and total cost compared with the rest scenarios for the MAV-DQN strategy.

• For the scenario with $K = 10$: We still observe the use of "stop skipping" in the scenario since the number of split executions is less than that of split decisions. But the MAV-DQN strategy makes much fewer split decisions than it does for the scenario with $K = 8$ and most split decisions here end up being executed. This is because with the demand rate fixed, it becomes easier for a bus with larger MAV units to spare an empty unit when it decides to split. That is, it becomes harder to enter and stay in a pre-split phase when the capacity is higher. In this case, making too many split decisions will lead to unnecessarily longer decoupling duration which may increase the operational cost substantially. The passengers' cost is lower than that in the scenario with $K = 8$ because of the shorter pre-split duration and thus lower extra waiting time.

• For scenarios with $K = 12$ and 14: We hardly see "stop skipping" in these scenarios, and the overall pattern becomes similar to what we have seen for the MAV-Myopic strategy except that here the split decisions are made more proactively. Hence, they have a higher operational cost but a better level of service compared with the MAV-Myopic strategy. Compared with the scenario with $K = 10$, the passengers’ costs are further reduced due to the even shorter pre-split duration.

The above discussion reveals that the MAV-DQN strategy can improve the system performance by implementing more sophisticated MAV operations based on the property of different scenarios. In practice, the transit operator may need to carefully design the capacity when implementing an MAV-aided bus system since the optimizing mechanisms may be different for systems with different MAV unit capacities.

In Figs. 15 and 16, we present the results for strategy comparison under scenarios with different levels of demand and random noise. The findings are similar to those for Model I. We, therefore, abstain from revisiting the discussion here.

5 Conclusion

In this paper, we investigate a novel bus bunching alleviation strategy in MAV-based transit systems. Taking advantage of the splittable nature of MAVs, two types of splits, namely catch-up splits and holding splits are proposed to help late buses and early buses to get back to their scheduled trajectories, respectively. To quantify the performance of such a strategy, we establish two CA models, one with a soft vehicle capacity constraint (Model I) and the other with a hard vehicle constraint (Model II), to capture the system dynamics considering the MAV splitting and merging operations. Based on the CA models, we then derive the corresponding infinite-horizon stochastic optimization models to optimize the strategy in terms of passengers’ cost, bus operational cost, and in-vehicle crowding cost (for Model I) or excess cost (for Model II). Since each bus makes asynchronous decisions, we thus consider each bus a decision agent and develop a DQN
Figure 15: Strategy performance under different levels of demand

(a) $\lambda = 20$

(b) $\lambda = 40$

(c) $\lambda = 60$

Figure 16: Strategy performance under different levels of trajectory noises

(a) $\sigma = 0.025$

(b) $\sigma = 0.045$

(c) $\sigma = 0.065$
algorithm to learn the optimal strategy for each bus. Numerical results suggest that the strategy obtained from the DQN algorithm (MAV-DQN) is a bunch-proof strategy. Moreover, based on Model I, we find that the MAV-DQN strategy tends to launch a catch-up split when the bus falls slightly behind while launching a holding split when the bus is ahead of schedule to some extent. By doing this, the MAV-DQN strategy keeps the bus overall a little ahead of schedule, which is more controllable and expected to yield lower costs. Such phenomenon, however, is not apparent in Model II since the cost incurred during the pre-split phase prevents the bus from launching catch-up splits frequently. Through comparison under scenarios with different fleet sizes and unit sizes, we show that the MAV-DQN strategy outperforms the myopic strategy for MAV-aided systems and the two-way-looking strategy for conventional transit systems. Further comparisons are conducted to examine the benefits of the MAV-DQN strategy across scenarios with varying demand rates and levels of trajectory uncertainty. For Model II with a hard capacity constraint, we also conduct the comparison under different MAV unit capacities and reveal the self-regulating effect for crowded buses, which plays a mediating role in delaying bus bunching. Moreover, the MAV-DQN strategy can offer subtle operations based on the characteristics of different scenarios. The findings in this paper provide managerial insights and guidance for leveraging MAVs to alleviate bus bunching, and the model developed in this study can serve as a useful tool for evaluating different transit operation strategies with MAVs.

There are a few possible extensions to this work. In our model, whenever a bus launches a split, passengers are all directed to one of the platoons such that an empty platoon can decouple from the bus to catch up or dwell. This may impose unnecessary transfer costs on the passengers. Future work may incorporate optimal passenger distribution among the units with the splitting and merging operations to further improve the level of service. In addition, though the DQN algorithm proposed here considers each bus a decision agent, they share a common value function and adopt the same policy. Therefore, the problem of policy optimization for multiple agents is essentially treated as a single-agent reinforcement learning problem. However, in a multi-agent system, the agents may cooperate (which is true for buses in a transit system) with each other and may not adopt an identical policy, especially when heterogeneous vehicles are considered. Hence, a multi-agent reinforcement algorithm may be developed to enhance the overall system performance. In addition, the emerging intelligent seat booking systems for transit service may help to alleviate the bus bunching by controlling the passengers’ arrival rates, further exploration for integrating booking systems in MAV-aided bus service is worth investigating.

CRediT authorship contribution statement

Yuhao Liu: Methodology, Formal Analysis, Software, Visualization, Writing – original draft. Zhibin Chen: Conceptualization, Formal Analysis, Validation, Writing – review & editing, Supervision, Funding Acquisition. Xiaolei Wang: Validation, Writing – review & editing.

Acknowledgement

This work was partially supported by the National Natural Science Foundation of China (72101153, 72022013, 72021002, 72061127003), Shanghai Chenguang Program (21CGA72), Shanghai Frontiers Science Center of Artificial Intelligence and Deep Learning, Shanghai Eastern Scholar Program (QD2020057), the NYU Shanghai Doctoral Fellowships, and the NYU Shanghai Boost Fund.
## Appendix A: Notation list

### Table 7: List of important notations

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td></td>
</tr>
<tr>
<td>$N$</td>
<td>Number buses</td>
</tr>
<tr>
<td>$m$</td>
<td>Number units per bus</td>
</tr>
<tr>
<td>$l$</td>
<td>Length of the bus route</td>
</tr>
<tr>
<td>$K$</td>
<td>Nominal capacity of a modular unit</td>
</tr>
<tr>
<td>$H$</td>
<td>Scheduled spacing</td>
</tr>
<tr>
<td>$E$</td>
<td>Bus cruising speed</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Demand rate</td>
</tr>
<tr>
<td>$A$</td>
<td>Alighting time per passenger</td>
</tr>
<tr>
<td>$B$</td>
<td>Boarding time per passenger</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Maximum distance allowed for two platoons to recouple</td>
</tr>
<tr>
<td>$c_s$</td>
<td>Bus splitting cost</td>
</tr>
<tr>
<td>$c_c$</td>
<td>Per distance bus running cost</td>
</tr>
<tr>
<td>$\mu$</td>
<td>Value of time</td>
</tr>
<tr>
<td><strong>Decision Variables</strong></td>
<td></td>
</tr>
<tr>
<td>$z_n(t)$</td>
<td>$=1$ if bus $n$ launches a holding split at time $t$; $=0$, otherwise</td>
</tr>
<tr>
<td>$z'_n(t)$</td>
<td>$=1$ if bus $n$ launches a catch-up split at time $t$; $=0$, otherwise</td>
</tr>
<tr>
<td>$z_n(t)$</td>
<td>$=z_n^e(t) + z_n^l(t)$</td>
</tr>
<tr>
<td>$r_n(t)$</td>
<td>$=1$ if bus $n$ recouples from a holding split at time $t$; $=0$, otherwise</td>
</tr>
<tr>
<td>$r'_n(t)$</td>
<td>$=1$ if bus $n$ recouples from a catch-up split at time $t$; $=0$, otherwise</td>
</tr>
<tr>
<td>$r_n(t)$</td>
<td>$=r_n^e(t) + r_n^l(t)$</td>
</tr>
<tr>
<td>$k_n(t)$</td>
<td>number of units in the leading platoon if bus $n$ decouples at time $t$</td>
</tr>
<tr>
<td>$h_n(t)$</td>
<td>$=1$ if bus $n$’s leading platoon resumes full service in a catch-up split at time $t$</td>
</tr>
<tr>
<td>$h_n(t)$</td>
<td>$=1$ if bus $n$’s trailing platoon switches to drop-off-only mode in a catch-up split at time $t$</td>
</tr>
<tr>
<td>$g_n(t)$</td>
<td>$=1$ if bus $n$’s leading platoon switches to dwelling mode at time $t$</td>
</tr>
<tr>
<td>$g_n(t)$</td>
<td>$=1$ if bus $n$’s trailing platoon switches from dwelling to full-service mode at time $t$</td>
</tr>
<tr>
<td>$z_n(t)$</td>
<td>$=z_n^l(t) + z_n^e(t)$</td>
</tr>
<tr>
<td>$\hat{z}_n(t)$</td>
<td># of units decided to be in the leading platoon when bus $n$ decides to split at time $t$</td>
</tr>
<tr>
<td>$\hat{r}_n(t)$</td>
<td>$=1$, if bus $n$ resumes full service from a pre-split phase at time $t$; $=0$, otherwise</td>
</tr>
<tr>
<td><strong>State Variables</strong></td>
<td></td>
</tr>
<tr>
<td>$u_n(t)$</td>
<td>$=1$ if bus $n$ is in the decoupled state at time $t$</td>
</tr>
<tr>
<td>$w_n^f(t)$</td>
<td>$=1$ if bus $n$’s leading platoon is in full or drop-off-only service at time $t$</td>
</tr>
<tr>
<td>$w_n^d(t)$</td>
<td>$=1$ if bus $n$’s leading platoon is in full-service mode at time $t$</td>
</tr>
<tr>
<td>$q_n^f(t)$</td>
<td>$=1$ if bus $n$’s leading platoon is in drop-off-only or dwelling mode at time $t$</td>
</tr>
<tr>
<td>$q_n^d(t)$</td>
<td>$=1$ if bus $n$’s trailing platoon is in dwelling mode at time $t$</td>
</tr>
<tr>
<td>$y_n(t)$</td>
<td>Trajectory of platoon $i$ of bus $n$</td>
</tr>
<tr>
<td>$x_n(t)$</td>
<td>Location of platoon $i$ of bus $n$</td>
</tr>
<tr>
<td>$s_n(t)$</td>
<td>Spacing faced by platoon $i$ of bus $n$</td>
</tr>
</tbody>
</table>

Continued on next page
Table 7 – Continued from previous page

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_i^n(t) )</td>
<td>Capacity of platoon ( i ) of bus ( n ) at time ( t )</td>
</tr>
<tr>
<td>( O_i^n(t) )</td>
<td>Occupancy of platoon ( i ) of bus ( n ) at time ( t )</td>
</tr>
<tr>
<td>( v_i^n(t) )</td>
<td>Speed of platoon ( i ) of bus ( n ) at time ( t )</td>
</tr>
<tr>
<td>( \delta_i^n(t) )</td>
<td>Schedule deviation of platoon ( i ) of bus ( n ) at time ( t )</td>
</tr>
<tr>
<td>( \xi_i^n(t) )</td>
<td>Trajectory noise for platoon ( i ) of bus ( n ) at time ( t )</td>
</tr>
<tr>
<td>( \xi_n(t) )</td>
<td>Trajectory noise for bus ( n ) in the coupled state at time ( t )</td>
</tr>
<tr>
<td>( \xi_{i,n}(t) )</td>
<td>Trajectory noise for platoon ( i ) of bus ( n ) in drop-off-only mode at time ( t )</td>
</tr>
<tr>
<td>( \delta_{i,n}(t) )</td>
<td>Schedule deviation of platoon ( i ) of bus ( n ) at time ( t )</td>
</tr>
<tr>
<td>( \hat{u}_n(t) )</td>
<td>=1, if bus ( n ) is in the pre-split mode at time ( t ); = 0 otherwise</td>
</tr>
<tr>
<td>( p_i^n(t) )</td>
<td>( i \in {1,2,...,5} ). Binary variables indicating the signs of continuous variables</td>
</tr>
</tbody>
</table>

Appendix B: Derivation of the alighting probability in Model II

In Model II, when a full-service platoon arrives, a passenger will attempt to board the platoon but may fail due to the vehicle capacity constraint. If they fail, they have to wait for the next arriving full-service platoon to make a second attempt. Let \( \xi \in \{1,2,...\} \) be a random variable representing the number of attempts a passenger has made until succeeding in boarding. Let \( p_i = P\{\xi = i\} \) denote the probability of a passenger succeeding in boarding a platoon in the \( i \)th attempt. Since the bus route is homogeneous, the probability \( p_i \) is independent of the passenger’s origin and destination.

Now, we can formally derive the alighting probability for Model II. Suppose at time \( t \), the platoon is at location \( y \in [0,l] \) and the moving speed is \( v(t) \). We want to derive the probability density function (PDF) of the destination of a passenger who succeeds in boarding the platoon in the \( i \)th attempt and remains on board at location \( y \), thereby determining the alighting proportion over the segment \([y, y + v(t)]\). Let \( Z \) be a random variable representing a passenger’s destination. Given that a passenger onboard at any given location \( y \) with \( \xi = i \), we need to derive the conditional PDF of \( Z \), i.e.,

\[
f_{Z|\text{onboard at } y \cap \xi = i}(z) = \frac{dF_{Z|\text{onboard at } y \cap \xi = i}(z)}{dz} \quad (101)
\]

\[
= \frac{dP\{Z \leq z \cap \text{onboard at location } y \cap \xi = i\}}{dz} \quad (102)
\]

\[
= \frac{d}{dz} \frac{P\{Z \leq z \cap \text{onboard at location } y \cap \xi = i\}}{P\{\text{onboard at location } y \cap \xi = i\}} \quad (103)
\]

To complete the derivation, we need the PDF of a passenger’s destination and the PDF of a passenger’s origin conditioned on her destination. The passengers’ destinations are uniformly distributed along the route. Hence, the unconditional PDF of \( Z \) is

\[
f_Z(z) = \frac{1}{l}, \forall z \in [0,l] \quad (104)
\]

Given the destination \( Z = z \), the passenger’s origin, denoted by random variable \( X \), can only be located within interval \([z - l/2, z]\). Hence, the conditional PDF of \( X \) is given by

\[
f_{X|Z=z}(x) = \frac{2}{l}, \forall x \in [z - l/2, z] \quad (105)
\]
With these, the numerator and denominator of Eq. (103) can be respectively calculated as follows.

\[ P\{Z \leq z \cap \text{onboard at location } y \cap \xi = i\} \]
\[ = P\{X \in [Z - l/2, y] \cap Z \in [y, z] \cap \xi = i\} \quad \text{(Equivalence)} \]  
\[ = P\{X \in [Z - l/2, y] \cap Z \in [y, z]\} \cdot P\{\xi = i\} \quad \text{(Independence)} \]
\[ = p_i \cdot \int_{y-l/2}^{y} f_Z(z) \int_{y-l/2}^{y} f_X|Z=y(x) dx du = p_i \cdot \left[ \frac{2y+l}{l} (z-y) - \frac{1}{l} (z^2 - y^2) \right] \quad \text{(109)} \]

\[ P\{\text{onboard at location } y \cap \xi = i\} \]
\[ = P\{\xi = i \cap X \in [Z - l/2, y] \cap Z \in [y, y+l/2]\} \quad \text{(Equivalence)} \]
\[ = P\{\xi = i\} \cdot P\{X \in [Z - l/2, y] \cap Z \in [y, y+l/2]\} \quad \text{(Independence)} \]
\[ = p_i \cdot \int_{y}^{y+l/2} f_Z(z) \int_{y-l/2}^{y} f_X|Z=z(x) dx dz = \frac{p_i}{4} \quad \text{(113)} \]

The conditional PDF \( f_{Z|\text{onboard at } y \cap \xi = i}(z) \) is then derived as

\[ f_{Z|\text{onboard at } y \cap \xi = i}(z) = \frac{d}{dz} \frac{p_i[(2y+l)(z-y) - (z^2 - y^2)]/l}{p_i/4} = \frac{4}{l} - \frac{8(z-y)}{l^2} \quad \text{(114)} \]

Note that the conditional PDF \( f_{Z|\text{onboard at } y \cap \xi = i}(z) \) turns out to be independent of \( i \) and identical to that derived in Petit et al. (2018). Therefore, this alighting probability applies to any passenger onboard irrespective of the number of attempts they have made. It depends only on the difference between location \( y \) and the concerned location \( z \). The alighting proportion between interval \([y, y+v(t)]\) can be derived as

\[ f(t) = \int_{y}^{y+v(t)} f_{Z|\text{onboard at } y \cap \xi = i}(z) dz = \int_{y}^{y+v(t)} \frac{4}{l} - \frac{8(z-y)}{l^2} dz \quad \text{(115)} \]

which is equivalent to calculating the area of a trapezoid as indicated in Petit et al. (2018) and ends up as the proportion used in both Model I and II. Following similar procedures, we can derive the same alighting probability as in Petit et al. (2018) for a drop-off-only platoon.

**Appendix C: Derivation and validation of the moving speed for a full-service platoon in Model II**

For any given full-service platoon at any given time \( t \), suppose we know its position \( x(t) \), forward spacing \( s(t) \), capacity \( K(t) \), occupancy \( O(t) \) and the stranded demand along the route \( \pi(\omega; t) \). We now want to estimate its moving speed for the next time step (i.e., the speed between time \( t \) and \( t+1 \)).

First, we assume that there is no capacity constraint but consider the potential stranded demand and let \( V_1 \) denote the unknown moving speed under this assumption. This gives the following equation in terms of \( V_1 \)

\[ \frac{V_1}{E} + B \left[ \lambda s(t) + \int_{\mathcal{D}(x(t), V_1)} \pi(\omega; t) d\omega \right] = 1 \quad \text{(116)} \]
where \( D(x(t), V_1) \) denotes the route segment traveled by the platoon at location \( x(t) \) with speed \( V_1 \) in one unit of time. In this equation, the first term is the time spent on cruising calculated as the traveled distance \( (V_1) \) divided by the cruising speed \( (E) \). The second term is the dwell time spent on picking up passengers where the number of boarding passengers is calculated as the sum of the number of newly generated passengers \( (\lambda s(t)) \) and the number of stranded passengers from the previous platoons \( (\int \pi(\omega; t) d\omega) \). And the sum of the time spent on cruising and dwelling should equal 1 unit of time. The speed \( V_1^* \) solved from this equation may result in a number of boarding passengers greater than the available capacity of the platoon as the capacity constraint is not considered here. Hence, we need to check whether \( V_1^* \) complies with the capacity constraint. Note that based on Eq. (87), the available capacity is also a function of the moving speed. For conciseness, we define two functions \( f_1(V) \) and \( f_2(V) \) in terms of \( V \)

\[
\begin{align*}
f_1(V) &= \lambda s(t) + \int_{D(x(t), V)} \pi(\omega; t) d\omega \\
f_2(V) &= K(t) - O(t) \left[ 1 - \frac{V}{l/2} \left( 2 - \frac{V}{l/2} \right) \right]
\end{align*}
\]

If a platoon travels at speed \( V \) in the following time step, then \( f_1(V) \) represents the passenger demand intending to board and \( f_2(V) \) denotes capacity available during the time step.

Given \( V_1^* \) obtained from solving Eq. (116), we need to check whether the following inequality holds

\[
f_1(V_1^*) \leq f_2(V_1^*)
\]

If this inequality holds, then \( V_1^* \) is a valid estimate of the moving speed; otherwise, it underestimates the actual speed because it assumes more boarding passengers than the available capacity, and hence is invalid.

To make up for the inadequacy of Eq. (116), we propose the following equation in terms of the unknown moving speed \( V_2 \)

\[
\frac{V_2}{E} + B f_2(V_2) = 1
\]

which is established based on the same logic as Eq. (116) except that the number of boarding passengers is now equal to the available capacity. This equation applies to the situation where the capacity constraint is binding. Therefore, given the solution \( V_2^* \), we have to check whether the following condition holds

\[
f_2(V_2^*) \leq f_1(V_2^*)
\]

If this condition holds, then \( V_2^* \) is a valid estimate of the moving speed since the vehicle capacity constraint is binding; otherwise, it overestimates the actual moving speed as it assumes fewer boarding passengers which in fact can be more.

Although we now have two equations for speed estimation that apply to the two situations we are expecting for the system, the validity of the proposed equations needs further examination. First of all, we need to examine the existence and uniqueness of the solutions to both equations, respectively.

**Proposition 1.** There exists a unique solution \( V_1^* \in (0, E) \) to Eq. (116) if and only if \( B\lambda s(t) < 1. \)

**Proof.** Note that the left-hand side of Eq. (116) is strictly increasing in variable \( V_1 \). Therefore, if the equation has a solution, it must be unique.
We first prove the necessity. Let $V_1^* \in (0, E)$ denote the unique solution to Eq. (116). Since $V_1^*/E > 0$ and $\int_{\mathcal{D}(x(t), \omega)} \pi(\omega; t) d\omega \geq 0$, from Eq. (116) we have

$$B\lambda s(t) = 1 - \frac{V_1^*}{E} - B \int_{\mathcal{D}(x(t), \omega)} \pi(\omega; t) d\omega < 1$$  \hspace{1cm} (122)

We now prove the sufficiency. Since $B\lambda s(t) < 1$, from Eq. (116) we have

$$\frac{V_1}{E} + B \int_{\mathcal{D}(x(t), \omega)} \pi(\omega; t) d\omega = 1 - B\lambda s(t) > 0$$  \hspace{1cm} (123)

Note that $\pi(\omega; t)$ is rarely a continuous function in $\omega$. However, owing to the continuity of the CA model, at any discontinuous point $\hat{\omega}$ of function $\pi(\omega; t)$, we still have

$$\lim_{V_1 \to \hat{\omega}^-} \int_{\mathcal{D}(x(t), V_1)} \pi(\omega; t) d\omega = \lim_{V_1 \to \hat{\omega}^+} \int_{\mathcal{D}(x(t), V_1)} \pi(\omega; t) d\omega$$  \hspace{1cm} (124)

Therefore, the left-hand side of Eq. (123) is a continuous function in $V_1$. When $V_1 = 0$, $\frac{V_1}{E} + B \int_{\mathcal{D}(x(t), V_1)} \pi(\omega; t) d\omega = 0 < 1 - B\lambda s(t)$; when $V_1 = E$, $\frac{V_1}{E} + B \int_{\mathcal{D}(x(t), V_1)} \pi(\omega; t) d\omega \geq 1 > 1 - B\lambda s(t)$. Therefore, there exists a unique solution $V_1^* \in (0, E)$ for Eq. (123) and hence for Eq. (116). This completes the proof.

The condition $B\lambda s(t) < 1$ requires that for every time step, the newly generated passengers should not be too many to be picked up in one unit of time, which intuitively makes sense and is also the least condition for the speed estimation in Daganzo and Pilachowski (2011) and Petit et al. (2018) to hold. To ensure this condition, the value of $B, \lambda$ and the expected spacing $H$ should be properly configured. Moreover, during the simulation, the instantaneous spacing $s(t)$ may suddenly become too large as some buses switch to drop-off-only mode. In this case, the time headway may not be well approximated via the space headway $s(t)$, which may result in the violation of condition $B\lambda s(t) < 1$. To circumvent this violation, an upper bound (e.g., $2H$ as we expect no passenger to be skipped by two consecutive drop-off-only platoons) can be set on $s(t)$ to smooth the approximation.

**Proposition 2.** If $E < l/2$ and $B(K(t) - O(t)) < 1$, there exists a unique solution $V_2^* \in (0, E)$ to Eq. (120).

**Proof.** Note that Eq. (120) is a quadratic equation. Its determinant

$$\Delta = \frac{8BO(t)}{lE} + \frac{1}{E^2} + \frac{16B^2 O(t)K(t)}{l^2} - \frac{16BO(t)}{l^2} > \frac{1}{E^2} + \frac{16B^2 O(t)K(t)}{l^2} > 0$$  \hspace{1cm} (125)

where the first inequality holds due to the condition $E < l/2$. With this, we claim that Eq. (120) possesses two real zeros if $E < l/2$. Let $f(V_2)$ denote the quadratic function corresponding to Eq. (120). $f(V_2)$ has a negative quadratic term coefficient. The maximizer of $f(V_2)$ is

$$\hat{V}_2 = \frac{4BEIO(t) + l^2}{8BEO(t)} > \frac{4BEIO(t)}{8BEO(t)} > \frac{l}{2} > E$$  \hspace{1cm} (126)

Therefore, to prove that there exists a unique solution $V_2^* \in (0, E)$ to Eq. (120), it suffices to prove that $f(0) < 0$ and $f(E) > 0$. The former is equivalent to condition $B(K(t) - O(t)) < 1$ while the latter holds if $E < l/2$.  \hspace{1cm} $\Box$
Note that condition $E < l/2$ is hardly a strict condition as it requires that the cruising distance within one time step is less than half of the route length, which can be satisfied with any reasonable length of the time step. In Eq. (120), we are assuming we will pick up passengers up to the available capacity. Violating condition $B(K(t) - O(t)) < 1$ means we have to spend the entire time step picking up passengers to fill the platoon even with no onboard passengers alighting, which will absolutely lead to failure in estimating the moving speed for the concerned time step. In this case, the solutions to Eq. (120), one being non-positive and the other greater than $l/2$, are not qualified as valid moving speeds. There is no guarantee that condition $B(K(t) - O(t)) < 1$ will hold throughout the simulation. However, we will soon show that Eq. (116) must yield a valid moving speed when condition $B(K(t) - O(t)) < 1$ for Eq. (120) does not hold.

Now suppose we obtain solution $V_1^* \in (0, E)$ to Eq. (116) and solution $V_2^* < E$ to Eq. (120). $V_2^*$ is not guaranteed to be positive as condition $B(K(t) - O(t)) < 1$ may be violated. Note that the applicability conditions (119) and (121) for the two equations depend on their solutions $V_1^*$ and $V_2^*$, respectively. It is therefore in question that whether both solutions will indicate themselves as valid or invalid. In the former case, we are unsure about which speed estimate to use while in the latter case, we fail to obtain any valid speed estimate. We disprove both possibilities via the following proposition.

**Proposition 3.** At any time step,

a) At least one of the two solutions $V_1^*$ and $V_2^*$ is the valid speed;

b) Both of them are valid if and only if $V_1^* = V_2^*$.

c) When $V_1^* \neq V_2^*$, the valid speed $V^* = \max\{V_1^*, V_2^*\}$.

**Proof.** We first consider the case where $B(K(t) - O(t)) \geq 1$ (i.e., $Bf_2(0) \geq 1$). In this case, $V_2^* \leq 0$ is not a valid estimate for the moving speed. We need to prove that $V_1^*$ must be a valid estimate for the moving speed (i.e., $f_1(V_1^*) \leq f_2(V_2^*)$).

Since $V_1^*$ is the solution to Eq. (116), by rearranging the terms we have

$$f_1(V_1^*) = \frac{1}{B} - \frac{V_1^*}{BE}$$

(127)

Notice that function $f_2(V)$ is increasing in $V$ for $V < E < l/2$. Hence,

$$f_2(V_1^*) > f_2(0) \geq \frac{1}{B} > \frac{1}{B} - \frac{V_1^*}{BE} = f_1(V_1^*)$$

(128)

Therefore, when $B(K(t) - O(t)) \geq 1$, $V_1^*(> 0 \geq V_2^*)$ must be a valid estimate for the moving speed.

Now we proceed to the case where $B(K(t) - O(t)) < 1$. In this case, $V_1^* \in (0, E)$ and $V_2^* \in (0, E)$. To prove a), we need to prove at least one of the two conditions (119) and (121) holds. We prove this by contradiction and the fact that both $V/E + f_1(V)$ and $V/E + f_2(V)$ are increasing in $V$. Suppose none of $V_1^*$ and $V_2^*$ is valid, i.e., $f_1(V_1^*) > f_2(V_1^*)$ and $f_2(V_2^*) > f_1(V_2^*)$, we have

$$\frac{V_1^*}{E} + Bf_2(V_1^*) < \frac{V_1^*}{E} + Bf_1(V_1^*) = 1 = \frac{V_2^*}{E} + Bf_2(V_2^*) \Rightarrow V_1^* < V_2^*$$

(129)

$$\frac{V_2^*}{E} + Bf_1(V_2^*) < \frac{V_2^*}{E} + Bf_2(V_2^*) = 1 = \frac{V_1^*}{E} + Bf_1(V_1^*) \Rightarrow V_1^* > V_2^*$$

(130)

which elicits contradiction. Therefore, at least one of the two conditions (119) and (121) holds.
For b), we first prove the necessity. Since both $V_1^*$ and $V_2^*$ are valid, i.e., $f_1(V_1^*) \leq f_2(V_1^*)$ and $f_2(V_2^*) \leq f_1(V_2^*)$, we have

$$\frac{V_1^*}{E} + B f_2(V_1^*) \geq \frac{V_1^*}{E} + B f_1(V_1^*) = 1 = \frac{V_2^*}{E} + B f_2(V_2^*) \Rightarrow V_1^* \geq V_2^*$$  (131)

$$\frac{V_2^*}{E} + B f_1(V_2^*) \geq \frac{V_2^*}{E} + B f_2(V_2^*) = 1 = \frac{V_1^*}{E} + B f_1(V_1^*) \Rightarrow V_1^* \leq V_2^*$$  (132)

which implies $V_1^* = V_2^*$. We now prove the sufficiency. Since $V_1^* = V_2^*$, by Eqs. (116) and (120) we have

$$f_1(V_2^*) = f_1(V_1^*) = f_2(V_2^*) = f_2(V_1^*)$$  (133)

which implies that both conditions (Eqs. (119) and (121)) are satisfied.

Finally, when $V_1^* \neq V_2^*$, by properties a) and b) we claim that exactly one of the two conditions (119) and (121) is satisfied. Now property c) is the direct conclusion from Eq. (131) and (132).

The nonlinear system (Eqs. (93)-(95)) for speed estimation is proposed based on the aforementioned discussion. The proposed approach requires no stricter condition compared with the speed estimation in Daganzo and Pilachowski (2011) and Petit et al. (2018) but effectively extends its applicability to CA models with a hard capacity constraint.
References


