ORIGINAL RESEARCH



Holding times to maintain quasi-regular headways and reduce real-time bus bunching

Citlali M. Olvera-Toscano¹ · Yasmín Á. Ríos-Solís² · Roger Z. Ríos-Mercado¹ · Romeo Sánchez Nigenda¹

Accepted: 16 April 2023 / Published online: 17 May 2023 © The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2023

Abstract

Real-time control strategies deal with the day's dynamics in bus rapid transit systems. This work focuses on minimizing the number of buses of the same line cruising head-to-tail or arriving at a stop simultaneously by implementing bus holding times at the stops as a control strategy. We propose a new mathematical model to determine the bus holding times. It has quadratic constraints but a linear objective function that minimizes the bus bunching penalties. We also propose a beam-search heuristic to reduce computational solution time to solve large instances. Experimental results on a bus rapid transit system simulation in Monterrey, Mexico, show a bus bunching reduction of 45% compared to the case without optimization. Moreover, passenger waiting times are reduced by 30% in some scenarios. For real-world instances with 60 buses, the beam-search approach provides solutions with an optimality gap of less than 5% in less than 3 s.

Keywords Bus bunching \cdot Real-time \cdot Holding time \cdot Transit operations \cdot Beam search

Yasmín Á. Ríos-Solís yasmin.riossolis@tec.mx

> Citlali M. Olvera-Toscano citlali.olverats@uanl.edu.mx

Roger Z. Ríos-Mercado roger.rios@uanl.edu.mx

Romeo Sánchez Nigenda romeo.sanchezng@uanl.edu.mx

- ¹ Graduate Program in Systems Engineering, Universidad Autonoma de Nuevo Leon, Av. Universidad s/n, 66455 San Nicolas de los Garza, NL, Mexico
- ² Engineering and Sciences Department, Tecnologico de Monterrey, Av. Eugenio Garza Sada 2501, 64700 Monterrey, NL, Mexico

1 Introduction

Every city encourages their public transport companies to mobilize its population more sustainably. Nevertheless, public transport systems often lack high-quality service, making them unattractive to users with private vehicles (National Academies of Sciences, Engineering, and Medicine 2013; He et al. 2020). The transport dynamics are challenging to understand and to incorporate into the operation. Public transportation companies aim to reduce variability in the user waiting times at the stops to increase customer service satisfaction and attract more customers. Indeed, quality is positively correlated to reliable systems (Ceder 2016; Prassas and Roess 2020; Verbich et al. 2016). For this, technology and infrastructure in transportation services enable better communication with the bus drivers and extensive data collection such as user flows, bus occupancy, the geospatial location of buses, or road reports. This information provides real-time feedback to the bus control system that aims to improve the users' satisfaction (National Academies of Sciences, Engineering, and Medicine 2013; Ceder 2016; Deng et al. 2020).

Once a public transportation company establishes the buses' departure times for each stop with a specific frequency, the company uses real-time control strategies to palliate the day's dynamics such as traffic, passenger flow, weather, or accidents. As Wang and Sun (2020) and Ceder (2016) mention, a slight frequency deviation can cause many buses to be delayed on their schedule.

In this work, we aim to improve the dynamic aspects of transport systems in real time. We focus on the time difference between trips of consecutive buses at a particular stop, known as *headway* (Ceder 2016). In particular, we deal with *bus bunching* that occurs when the value of the headway is equal to zero (or equivalently, buses arriving at a bus stop simultaneously). Bus bunching implies frequency variability that affects the user waiting times at the stops and traveling times. Also, bus bunching involves unbalanced occupancy rates since many waiting users choose to board the first bus that arrives at the stop, especially when the information on the arrival of the next bus is unknown, as in most cities in developing countries.

Figure 1 shows an example of bus bunching for a single bus line. The time is plotted on the horizontal axis, while bus stops are on the vertical axis. There are three bus trips (red, blue, and yellow) with an established departure time of 7:00, 7:10, and 7:20, respectively. Each line shows the time-space trajectories of the three trips. An ideal behavior of the buses without bus bunching is when the headway is equal to 10 min at any time and space. Nevertheless, in this figure, the blue and yellow trips (solid trajectories) deviate from the planned ones (dotted trajectories). Indeed, the second bus (blue) finds more congestion than usual between stops 1 and 2. Thus, more persons are at stop 2, implying a larger dwelling time. At stop 2, some passengers that should have taken the yellow trip boarded the blue one; eventually, the blue and the yellow trips bunch.

Our objective is to reduce the number of bus bunching events by maintaining *quasi-regular* headways between each pair of buses to reduce user waiting times



Fig. 1 Bus bunching event at stop *s* between the buses that departed at 7:10 (blue) and 7:20 (yellow), caused by congestion between stations 1 and 2 (color figure online)

at the stops and their overall traveling times. We use a control strategy consisting of holding the buses for some minutes at some stations after the boarding and alighting processes have been completed. There are other control strategies, such as skipping stops or speed control. However, according to Verbich et al. (2016), the bus holding strategy has an essential impact on the reduction of bus bunching problem (BBP) (Delgado et al. 2012; Hernández-Landa et al. 2015; Sánchez-Martínez et al. 2016; Wang and Sun 2020) that seeks to determine a set of bus holding times at some stops to minimize the number of bus bunching events.



Fig. 2 Framework for the real-time data retrieving and the mathematical model

As shown in Fig. 2, our proposed solution approach to this BBP has two main processes. One of them is an optimization stage that determines the holding times of each bus at each stop by considering the current information along the bus line and by predicting the future events in the line: persons at each bus stop, dwelling and alighting times, and the capacity of each of the buses. This stage is modeled as a quadratically constrained integer linear program that yields the holding times for the buses, which are immediately communicated to the drivers to recover quasi-regular headways between them. This model is one of the main contributions of this work. While most related studies deal with a non-linear objective function that minimizes the total user waiting time at the optimization stage (see the overview by Ibarra-Rojas et al. 2015), our BBP methodology uses a linear objective function that speeds up its computational time. The main advantage of our model is that it includes the forecasting of the departure, boarding, and alighting times for each bus at each future stop, compared to previous works by Delgado et al. (2012) and Hernández-Landa et al. (2015). Only a few approaches consider the bus capacity every time the optimization model computes holding times during the evolution of the simulation process, as we do in this study. Another significant feature of our model is that it does not use origin-destination matrices as most of the other approaches (Delgado et al. 2009; Hernández-Landa et al. 2015; Sánchez-Martínez et al. 2016; Wang and Sun 2020; Xuan et al. 2011), which affects their efficiency. Instead, we use the user's arrival rate and the proportions of users that alight at stops. Our integer programming model can be solved by an exact method, such as branch-and-bound, or a heuristic to speed up the computational effort (beam-search heuristic in Sect. 3.4).

The other process of our methodology (see Fig. 2) is collecting data in real time: the position of the buses, the actual number of users waiting at stops, and the actual number of users on each bus. This information is the input of the optimization stage. These two processes are periodically applied one after the other along the working day to maintain quasi-regular headways between each pair of buses. We developed a discrete-event simulation program representing a bus rapid transit (BRT) system to simulate the real-data collection and stress the transit system under different parameters. Thus, we can retrieve simulated real-time data from the system to validate the efficiency of our mathematical model. This simulation takes as input the holding times of the optimization stage. Then, it evolves over time, and when asked, it returns its actual state, which is the input of the optimization stage.

Maintaining quasi-regular headways in a transit line differs from sticking to the buses' planned departure times at the stops. Quasi-regular headways imply a reliable service for users since the variance in the frequency of the buses is small at each bus stop and has a significant effect on travel time reliability (Soza-Parra et al. 2021). This tactic is beneficial in cities where the users do not know the exact scheduling of the buses, since they only have an expected frequency estimation, as is the case with systems in developing countries (Hernández-Landa et al. 2015).

The main contributions of this study are the following.

 A new mathematical model for the BBP with quadratic constraints, whose linear objective function minimizes the penalties caused by non-quasi-regular head-

- A BRT discrete-event microsimulation that effectively represents rush-hours to analyze the impact of the holding times obtained by the optimization model under different scenarios. This simulation is stochastic with random components on the travel times and the arrival of the users.
- A beam-search heuristic based on our mathematical model to reduce computational solution time to solve larger instances in real time.

This paper is organized as follows. In Sect. 2, we present a related literature review. Section 3 describes the system characteristics and the description of the mathematical model, the discrete-event simulation that represents a bus rapid transit system, and the beam-search approach. We validate the efficiency of our BBP methodology with a case study from Monterrey city in Mexico in Sect. 4. Finally, Sect. 5 presents the conclusions of this work.

2 Literature review

Significant efforts have been made to address the BBP with real-time monitoring strategies, categorized in *inter-station* and *station control* (Ibarra-Rojas et al. 2015). Inter-station control strategies are those where decisions are made at some bus line stops, such as holding times (Wang and Sun 2020; Daganzo 2009; Xuan et al. 2011), skipping stops (Sun and Hickman 2005; Fu et al. 2003; Cortés et al. 2010), boarding time limits (Delgado et al. 2012; Barnett 1974; Delgado et al. 2009), station control strategies such as bus speed regulation (He 2015; Daganzo and Pilachowski 2011; He et al. 2019), or traffic signal priority (Estrada et al. 2016; Liu et al. 2003; Ling and Shalaby 2004).

Among all the BBP strategies, holding time is the most used and the one that users resent the least. Table 1 shows the studies related to the holding time strategy (we do not include hybrid strategies). The first column presents the reference. The second column indicates whether the vehicle capacity is considered or not. The third column indicates the different types of control points that the system considers, such as single preset control points (SPCP), multiple preset control points (MPCP), or multiple control points (MCP) defined by the solution strategy. Column "Alighting" is how the users leave the transport system in the solution approach: a proportion of the onboard passengers or determined by an origin–destination (OD) matrix. The fifth and sixth columns indicate the considered objective function and model type, respectively (QP is for quadratic MILP, MIQCP is for quadratically constrained integer linear programming). The solution approach is in the "Method" column (B&B is for branch-and-bound), while the "Sim" one indicates whether the approach is analyzed through any simulation. Finally, the last column reports if a case study was tested.

Table 1 Literature review o	of the h	olding time strates	gy					
References	Cap	Control points	Alighting	Objective Func	Model type	Method	Sim	Case study
Eberlein et al. (2001)	No	SPCP	Proportional onboard pas- sengers	Minimize waiting time	QP	Local search	Yes	Yes
Hickman (2001)	No	SPCP (one bus)	OD matrix	Minimize headway vari- ance	Stoch. QP	Gradient search	No	No
Sun and Hickman (2008)	No	MPCP	Proportional onboard pas- sengers	Minimize waiting time	QP	Heuristic	No	No
Xuan et al. (2011)	No	MPCP	OD matrix	Schedule adherence and headway regularity	QP	Local gradient search	No	No
Delgado et al. (2012)	Yes	MPCP	OD matrix	Minimize waiting time	QP	Reduced gradient	Yes	No
Hernández-Landa et al. (2015)	Yes	MCP	OD matrix	Headway regularity	MILP	B&B	Yes	Yes
Sánchez-Martínez et al. (2016)	Yes	MCP	OD matrix	Minimize waiting time	QP	Iterative quadratic approximation	Yes	No
Wang and Sun (2020)	No	MCP	OD matrix	Reduce the headway mean and variance	QP	Multi-agent system	Yes	No
Our approach	Yes	MCP	Proportional onboard pas- sengers	Headway regularity	MIQCP	B&B	Yes	Yes

The last line of Table 1 considers the approach we propose. Our BBP approach searches for headway regularity which we show later implies reducing waiting times. This search for headway regularity allows us to have a linear objective function that reduces computational times compared to other approaches that directly optimize waiting times with a quadratic function. We also consider that buses have a capacity. The users must wait for the next bus if this capacity is reached. Our optimization process determines the stops where the buses must hold and the holding times. An enormous gain is obtained by considering that passengers alight a bus via a probabilistic approach since an OD matrix is rarely updated and reliable.

Argote-Cabanero et al. (2015) employ a combination of dynamic holding times and en-route driver guidance to improve schedule adherence. This method is analytically evaluated with simulations to improve reliability. As in our work, their simulation reflects a real BRT. Hall et al. (2001) develop analytical models to obtain the optimal holding times and waiting times at transfer stations. Their instances are based on actual data from the Los Angeles transit network. Daganzo (2009) proposes an adaptive control scheme that dynamically determines holding times at the stops. The proposed scheme aims to provide quasi-regular headways. The method proves to be effective when minor disturbances arise. Chen et al. (2013) present a problem that is a non-convex optimization problem with linear constraints that focuses on minimizing the user waiting time. Their model considers that the user arrival rate is uniform. The buses have the available capacity to let all users board at a stop. They use a suitable heuristic algorithm for real-time situations as a solution method.

More recently, Gkiotsalitis and Cats (2019) consider a periodic holding time control method where holding times of all running trips are computed simultaneously within each optimization period. They model the BBP as a discrete non-linear optimization problem. They do not consider additional user arrivals, when the bus is waiting at a stop, or the capacity of buses, since all users can board the first bus that arrives, even if this implies longer headways.

He et al. (2020) present a dynamic target-headway-based holding strategy. They observe that the average total waiting time increases at the beginning of their experiment but decreases over time, which is the desired behavior. Delgado et al. (2012) incorporate two strategies: holding time and boarding limits. They propose a mathematical programming model that minimizes total delays with a quadratic function evaluated in a simulation environment. Nevertheless, their simulation considers very short headways for computing holding times when a bus reaches a stop. Although they obtain the value of the holding times for all the buses in their following stops, they only use the holding time value for the current stop discarding the other values.

Liang et al. (2019) combine bus holding and stop-skipping with the objective of automatically equalizing bus headways on a single line. Their results show good performance in a simplified scenario where bus capacity is not considered. Sun and Schmöcker (2018) consider passenger choices and bus overtaking with uncapacitated buses; when a delay occurs in any of the buses, the departure times of all buses are calculated using a set of discrete state equations. Similarly, Drabicki et al. (2023) consider that users have arrival information for the next pair of buses. Thus, users may decide whether to board the first bus arriving at the station or wait for the next one. They developed a boarding choice model implemented in a simulation framework. Wang and Sun (2020) incorporate global coordination and long-term operation in holding time with a multi-agent deep reinforcement learning framework. However, they do not consider the bus capacity. Sánchez-Martínez et al. (2016) present a quadratic model to compute passengers' average waiting time costs and separately obtain the holding times. Their simulation only considers ten buses and equal arrival rates for all OD pairs. Li and Li (2022) present a probabilistic prediction model to classify the headways into bunching, stable and large gap, and decide which control strategy is suitable to use (holding, speed-adjusting, or stop-skipping) to minimize the deviation of the headway. However, their experiments consider the same arrival rate at all stops, a fixed traffic light time, and the alighting rate at intermediate stations is a random number between 0 and 1.

Tian et al. (2022) propose a model that uses the short-turning strategy in which they perform a rescheduling of the trips and whose objective is to minimize the deviation from the original schedule. They consider that the buses' scheduled arrival time at the stops must be known and that the capacity of the buses is sufficient to meet passenger demand. Laskaris et al. (2019) coordinate different bus lines using bus holding for minimizing the travel times of the users. Nevertheless, the capacity of their vehicles is not considered, and they use OD matrices. Sajikumar and Bijulal (2022) propose a model for a network with multiple origins. The model uses a schedule-based slack control strategy to reduce bus bunching and establish a redesigning route schedule. Bus capacity is not considered.

A close but simplified work, compared to ours, is presented by Liang et al. (2021). They present an algorithm that determines the optimal holding times to solve the nonlinear optimization problem. However, the obtained solution is local since, at the control point, the model only considers the bus that arrived at the station and its front and rear bus.

Yu et al. (2016) use supervised learning techniques by using support vector machine techniques and transportation smart-cards data to predict bus schedules and bus bunching. Shi et al. (2022) propose a distributed deep reinforcement learning to solve the bus bunching problem by maintaining schedule adherence, and headway regularity. Degeler et al. (2021) prove that it is possible to extract and detect individual cases of bus bunching using fully unsupervised techniques.

In this paper, we integrate simulation and optimization to reduce the number of bus bunching events and indirectly reduce the total user waiting times. On the one hand, we develop a discrete-event simulation of a BRT system. On the other hand, the mathematical model has quadratic constraints, but its objective function is linear, speeding up its computational time. Our BBP solution approach has two main processes (see Fig. 2). This section describes our bus rapid transit (BRT) system and introduces its notation. Then, we present the mathematical model of the optimization stage that determines the holding times of each bus. Finally, we present the simulation stage that represents the real-time data collection of a BRT case study.

3.1 The bus rapid transit description

Consider a one-way bus line corridor with a set *S* of stops plus the depot denoted as 0, as represented in Fig. 3, operated by a high-frequency bus service. A set of *B* buses with capacity C_b for each $b \in B$ and an average speed of *V* constitute the bus service. All buses start their trip at the depot following a departure frequency of *F* minutes (ideal headway). Each bus $b \in B$ sequentially visits all the stops in the corridor. The buses are numbered in ascending order in the corridor; bus b + 1 leaves the deposit after bus *b*, for $b \in B \setminus \{|B| - 1\}$. Notice that bus passing is not allowed. When a bus *b* arrives at the last stop, all bus users must alight, and the bus returns to the depot (no deadheads, since it is a circular corridor). The travel time between a pair of stops (s - 1, s) is denoted as d^s , for $s \in S$. The limit case d^1 is the distance between stop 1 and the depot.

The users arrive at each stop $s \in S$ following a Poisson distribution rate λ^s . The boarding and alighting times per user are \overline{U} , and \underline{U} , respectively. The total time taken by the bus doors for opening and closing is \overline{G} . Similarly to Eberlein et al. (2001), our work assumes that the number of users alighting from bus $b \in B$ at stop $s \in S$ is a proportion γ^s of the number of onboard users, a crucial point since we avoid using OD matrices that make heavier models. Additionally, the holding time assigned to a bus at a stop must not exceed a number H of minutes per station to avoid the irritation of the onboard users.



Fig. 3 Circular bus line corridor with s stops (green circles) and five buses (color figure online)

Time t_0 is when the real data is collected to compute the required decisions to recover a quasi-regular frequency. We collect the following data for each $b \in B$: the distance between bus b and its following stops denoted as m_b^s , the number of onboard users on bus b is q_b , and l_b is the last station visited by bus b. Note that if bus b is at station s at time t_0 , then $l_b = s$. Also, at time t_0 , we collect the number of users waiting at station s denoted by w^s for every station $s \in S$.

Let us properly define the concept of quasi-regular headways and a bus bunching event to incorporate them into our mathematical model (Daganzo 2009; Ibarra-Rojas et al. 2015). Maintaining ideal headways *F* along the BRT is not practical. Thus, let $0 \le \kappa \le 1$ be the tolerated percentage deviation from the ideal headway such that $\kappa F < F/2$.

Definition 1 The headway between two buses is *quasi-regular* if it is in the interval $[(1 - \kappa)F, (1 + \kappa)F]$. Thus, there is a *bus bunching* event between two buses at a stop if their headway is between $[0, (1 - \kappa)F]$ or if it is larger than $(1 + \kappa)F$.

3.2 The BBP optimization stage

The main variables to formulate the quadratically constrained integer linear program of the BBP methodology are h_b^s that indicate the holding time for bus $b \in B$ for all the future stops $s = l_b + 1, ..., |S|$ that the bus must visit. With the position of the buses at time t_0 and the holding time variables, we can derive the rest of the variables of each bus $b \in B$ at stop $s = l_b + 1, ..., |S|$. Let t_b^s be the variables that represent the departure time of bus b from stop s. With these last variables, we can compute the number of users who want to board bus b at stop s, the number of users who indeed boarded the bus, and the number of persons that alight that bus, denoted by z_b^s , x_b^s , and y_b^s , respectively, for $b \in B$ and $s = l_b + 1, ..., |S|$. See Table 2 for a notation summary.

The headway between the bus pair (b, b + 1) is determined by the difference of its departure times at a stop $s: t_b^s - t_{b+1}^s$, for $s \in S$. Let variable p_b^s be the penalty related to the headway between two buses for $b \in B$ and $s \in S$:



Fig. 4 The penalty values between a bus pair at a certain stop

Parameters	
В	Set of buses $(b \in B)$
S	Set of stops ($s \in S$)
С	Bus capacity
F	Ideal headway at every stop
V	Average bus speed
$\overline{U}, \underline{U}$	Boarding and alighting time per user
G	Time taken by the bus doors for opening and closing at each stop
Н	Maximum holding time value
Κ	Percentage deviation from the ideal frequency F
d^s	Distance between stations $s - 1$ and $s, s \in S$
λ^s	User arrival rate at station $s, s \in S$
γ^s	Proportion of users alighting from bus <i>b</i> at the station $s, s \in S$
t_0	Time when information is obtained
l_b	Last station visited by bus <i>b</i> at time $t_0, b \in B$
w ^s	Users waiting at stop s at time $t_0, s \in S, b \in B$
q_b	Number of onboard users on bus <i>b</i> at time $t_0, b \in B$
m_b^s	Distance between bus <i>b</i> and its next stop <i>s</i> at time $t_0, s \in S, b \in B$
Variables (for $s \in S, b \in B$)	
t_b^s	Departure time of bus b from stop s
z_b^s	Number of users who want to board bus b at stop s
x_b^s	Number of users who indeed boarded the bus b at stop s
\mathcal{Y}_{b}^{s}	Number of persons that alight bus b at stop s
h_b^s	Holding time for bus b at stop s
p_b^s	Headway penalty between buses b and $b + 1$ at stop s
n_b^s	1 if $x_b^s \leq C$ for bus <i>b</i> at station <i>s</i> , 0 otherwise
r_b^s	1 if $x_b^s > C$ for bus <i>b</i> at station <i>s</i> , 0 otherwise

 Table 2
 Notation for the BBP mathematical model

1

$$p_b^s = \begin{cases} 0, & \text{if } t_{b+1}^s - t_b^s \in [(1-\kappa)F, (1+\kappa)F] \\ t_b^s - t_{b+1}^s + (1-\kappa)F, & \text{if } t_{b+1}^s - t_b^s \in [0, (1-\kappa)F) \\ t_{b+1}^s - t_b^s - (1+\kappa)F, & \text{if } t_{b+1}^s - t_b^s > (1+\kappa)F \end{cases}$$

Figure 4 shows the headway values between a pair of buses at any fixed stop on its horizontal axis while the vertical axis represents the headway penalty values. Since the buses cannot overtake each other, this difference is positive. To incorporate the penalty value of bus b at stop $l_b + 1$ in the objective function, we use constraints (1) and (2). Note that p_b^s are positive variables:

$$p_b^s \ge t_b^s - t_{b+1}^s + (1 - \kappa)F, \qquad b \in B, s = l_b + 1,$$
 (1)

$$p_b^s \ge t_{b+1}^s - t_b^s - (1+\kappa)F, \qquad b \in B, s = l_b + 1.$$
 (2)

Changing the slopes of the right-hand side of the previous constraints would imply different importance to each one of the cases. Here we take the same slopes for both cases. In this way, the objective function (3) that minimizes the bus bunching penalties for each pair of buses to maintain quasi-regular headways between the buses is:

$$\min \sum_{b \in B} \sum_{s=l_b+1}^{|S|-1} p_b^s.$$
(3)

By minimizing the bus bunching occurrences and the number of separated buses, we aim to approach quasi-regular headways and, indirectly, to reduce user waiting times, as it is empirically shown in Sect. 5.

There are two cases for establishing the buses' departure times at each future station that they will visit after time t_0 . For handling the first case, constraints (4) set the departing time of bus *b* at stop $l_b + 1$ as the time t_0 plus the remaining time to arrive at stop $l_b + 1$, the boarding and alighting times, its holding time, and the opening and closing of the bus doors. For the second case, constraints (5) define the departure time of bus *b* from stop $s = l_b + 2, ..., |S|$. Constraints (6) do not allow a bus to leave a stop earlier than its predecessor, thus ensuring that passing between buses is impossible. Constraints (7) ensure that the holding times do not exceed the maximum allowed value. These last constraints were also included in the model by Hernández-Landa et al. (2015):

$$t_b^s = t_0 + \frac{m_b^s}{V} + \overline{U}x_b^s + \underline{U}y_b^s + h_b^s + G, b \in B, s = l_b + 1,$$

$$\tag{4}$$

$$t_{b}^{s} = t_{b}^{s-1} + \frac{d^{s}}{V} + \overline{U}x_{b}^{s} + \underline{U}y_{b}^{s} + h_{b}^{s} + G, b \in B, s = l_{b} + 2, \dots, |S|,$$
(5)

$$t_b^s \ge t_{b-1}^s, b \in B, s = l_b + 1,$$
 (6)

$$h_b^s \le H, b \in B, s = l_b + 1. \tag{7}$$

The number of users at stop *s* who are waiting to board bus *b* is established by constraints (8). They consider the persons w^s who are already at the stop at time t_0 plus the persons who arrive with a Poisson rate during the time that takes the bus to arrive at this stop *s*:

$$z_b^s = w^s + \lambda^s \left(\frac{m_b^s}{V}\right), \qquad b \in B, s = l_b + 1, \dots, |S|.$$
(8)

Note that not all these users z_b^s will be able to board bus *b* at stop *s*, since we are considering that buses have a finite capacity.

Constraints (9) determine the number of alighting users from bus *b* at stop *s*. They consider the onboard passengers and the users already alighted in previous stops s' < s, multiplied by the estimated proportion γ^s of passengers alighting at this stop, for $s \in S$:

$$y_b^s = \gamma^s \left(q_b + \sum_{s'=l_b+1}^{s-1} x_b^{s'} - y_b^{s'} \right), \qquad b \in B, s = l_b + 1, \dots, |S|.$$
(9)

The boarding users at stop *s* are determined with the following constraint (10), for $b \in B$, $s = l_b + 1, ..., |S|$:

$$x_b^s = \min\left\{C_b - q_b - \sum_{s'=l_b+1}^s x_b^{s'} + \sum_{s'=l_b+1}^s y_b^{s'}, z_b^s - \sum_{b'=0}^{b-1} x_{b'}^s\right\}.$$
 (10)

The right side of this minimum considers the case where the number of users waiting at *s* is less than the bus capacity. Thus, all users board except the ones that have already boarded a previous bus. The left side of the minimum limits the boarding users to the bus's current capacity. It subtracts the users who boarded previous buses. Constraints (10) are not linear, but a classical linearization (Williams 2013) for the minimum cannot be applied in this case because the objective function (3) includes the difference of $t_b^s - t_{b+1}^s$ that could be negative. Thus, we introduce two binary indicator variables for $b \in B$ and $s = l_b + 1, \ldots, |S|$. The first one, n_b^s , is equal to 1 if the number of users waiting for bus *b* at stop *s* is less than the actual bus capacity, and 0 otherwise. The second one, r_b^s , equals 1 if the available bus capacity is less than the number of users waiting for bus *b* at stop *s*, and is 0 otherwise. With these two binary sets of variables, we replace equation (10) with linear constraints (11) and (12), for $b \in B$ and $s = l_b + 1, \ldots, |S|$:

$$1 = n_b^s + r_b^s, (11)$$

$$x_{b}^{s} = n_{b}^{s} \left(z_{b}^{s} - \sum_{b'=0}^{b-1} x_{b'}^{s} \right) + r_{b}^{s} \left(C - q_{b} - \sum_{s'=l_{b}+1}^{s} x_{b'}^{s'} + \sum_{s'=l_{b}+1}^{s} y_{b'}^{s'} \right).$$
(12)

Constraints (11) indicate that only one case is possible. Either all the users waiting at stop *s* are going to board bus *b*, implying $n_b^s = 1$, or some of them will wait for the next bus b + 1; thus $r_b^s = 1$. By adding constraints (12), we explicitly choose between these two options and assign the correct value to the boarding variable x_b^s .

To reinforce the mathematical model, we add two families of valid inequalities. Constraints (13) and (14) bound the boarding passengers variable x_b^s by either the passengers at the stop or its remaining capacity, respectively, for $b \in B$ and $s = l_b + 1, ..., |S|$. These constraints are not necessary for the model to be correct, but they strengthen the convex hull of the discrete solution space, which reduces the resolution time of the mathematical model (Wolsey 2020).

$$x_b^s \le z_b^s - \sum_{b'=0}^{b-1} x_{b'}^s, b \in B, s = l_b + 1, \dots, |S|,$$
(13)

$$x_b^s \le C_b - q_b - \sum_{s'=l_b+1}^s x_b^{s'} + \sum_{s'=l_b+1}^s y_b^{s'}, b \in B, s = l_b + 1, \dots, |S|.$$
(14)

To summarize, the mathematical model that determines the best holding times for the buses at their future stops at time t_0 is named H-BBP(t_0):

$$\begin{array}{l} \min(3),\\ \text{s.t.} (1), (2),\\ (4) - (9)\\ (11), (12),\\ (13), (14),\\ h_b^s, x_b^s, t_b^s, z_b^s, y_b^s, p_b^s \in \mathbb{R}^+, s \in S, b \in B\\ n_b^s, r_b^s \in \{0, 1\}, s \in S, b \in B. \end{array}$$
(valid inequalities)

In the H-BBP(t_0) model, the forecast of the departure times and the boarding and alighting times play an essential role for each bus at each future stop. These approximations are one of the critical points of our study since they are much more precise than the ones made by other related approaches (Hernández-Landa et al. 2015; Delgado et al. 2012).

In Sect. 4, we solve model H-BBP(t_0) by a branch-and-bound algorithm to obtain high-quality solutions. Also, we propose a beam-search heuristic (see Sect. 3.4) to find a trade-off between quality and CPU time.

Table 2 summarizes parameters and variables for the BBP model.

3.3 The simulation of the real-time collection stage of the BBP

Our BBP methodology requires real-time data collection. We can obtain this data by using global positioning or automatic vehicle location systems for each bus in the BRT system. However, this kind of data is not always available, so we developed a discrete-event microsimulation to mimic a real-time situation described in Sect. 3.1. Besides, with our simulation, we can analyze the impact of the optimization model under different scenarios.

Our discrete-event simulation model is stochastic (with random components on the travel times and the arrival of the users) but is not dynamic, since time is not variable (Leemis and Park 2006; Sokolowski and Banks 2011). Indeed, we simulate a single rush-hour period.

The system state of our microsimulation is composed of four components that contain enough information to describe the evolution of the transportation system over time: the activation of a bus, the user's generation, the movement of the buses that consider the boarding and alighting users, and the holding times obtained by the optimization stage.

The ActiveBus function ensures that every F minutes, a new bus leaves the depot, recording all its information at every step of the simulation: the departure times, the number of persons on board or alighting, travel times, and holding times.

At each step of the simulation and for each stop s, the UserGeneration(s) function randomly generates users following a Poisson distribution with mean λ^s , for each $s \in S$. Each generated user at s is recorded together with its waiting time to catch the first bus and its traveling time on board. In this manner, at the end of the simulation, we can compute the total user waiting time.

Algorithm 1 MoveBus (s, b)
1: $s' \leftarrow l_b + 1$
2: while bus is not at stop s do
3: advance b up to s' during m_b^s/V of time
4: at stop s', hold bus b of $\overline{U}\bar{x}_{b}^{s'} + \underline{U}\bar{y}_{b}^{s'} + h_{b}^{s'} + G$ time
5: $s' \leftarrow s' + 1$
6: end while

The MoveBus(*s*, *b*) function, described by Algorithm 1, describes how the buses move in the BRT system. A bus *b* may advance or stop at a stop during each simulation step. A bus dwells at a stop s' < s until the users board and alight $\bar{x}_b^{s'} + \bar{y}_b^{s'}$, plus the holding time h_b^s established by the H-BBP(t_0) model, plus the time needed by the doors to open and close. This addition reflects the case where boarding and alighting occur through the same door. Note that $\bar{x}_b^{s'}$ and $\bar{y}_b^{s'}$ are not variables; they are values computed with the actual users at the stops and the actual capacity of the bus. Alighting users $\bar{y}_b^{s'}$ are a proportion $\gamma^{s'}$ of the onboard users. The boarding process follows a First-in-First-out (FIFO) rule, so the first user at the stop is the first to board the bus, provided the bus has available capacity. Otherwise, the user waits until the next bus arrives with available capacity. Therefore, $\bar{x}_b^{s'}$ is the minimum value between the available capacity of the bus *b* and the users waiting at a stop. The holding times are obtained from the H-BBP(t_0) model presented in Sect. 3.2.

At every step of the simulation, we keep track of the bus capacities and their location, the onboard users, and the ones at the stops. Therefore, different measures of effectiveness (MOE) may be calculated. The MOEs reflecting the system performance statistics are the total user waiting time to board a bus, the total travel time of the users, and the number of bus bunching events.

At the initial state of the simulation, there are only $B = \lfloor \frac{s}{2} \rfloor$ active buses in the corridor, uniformly distributed through the stops. The total execution time of the simulation represents a high demand period of T_{max} minutes. Ω is the frequency at which the holding times are computed by the H-BBP(t_0) model where t_0 is distributed evenly in the interval $[10\% T_{\text{max}}, 90\% T_{\text{max}}]$ (the planning period starts at time 0). The first 10% of the simulation time is the *warm-up time* to reach a steady point,

while the last 10% is mainly for the users to alight the buses, so there is not much interest in computing holding times anymore at the end of the period.

3.4 Beam-search heuristic

While exact methods can solve relatively small instances, larger ones take considerably longer. Therefore, heuristic methods for delivering quick, high-quality solutions become very important. In this work, we consider beam search, a classical heuristic search method that has been very effective in several combinatorial optimization problems (Akeb et al. 2009; Bennell et al. 2018). The beam-search idea is applied within a branch-and-bound framework, where the most promising nodes, according to certain criteria, are considered for further branching at each level while the rest of the nodes are permanently pruned. Thus, this can be seen as a truncated branch-and-bound method where optimality can no longer be guaranteed, but the computational effort is considerably smaller.

Tamannaei and Irandoost (2019) solve a carpooling problem, and Yavuz (2017) addresses a vehicle routing problem using heuristic beam-search algorithms. They show that their solution methods are efficient in terms of computational time, and they find near-optimal solutions for large-scale instances. More related to public transportation, Islam et al. (2019) developed a stochastic beam-search algorithm for a multicriteria route design problem for a transit network.

Our particular beam-search heuristic works as follows. Given a relative optimality gap parameter ϵ , we discard nodes (corresponding to subproblems) whose best objective function value falls beyond ϵ percent from the global optimum. That is, if *LB* is the linear programming relaxation value of a given node, we discard it if $LB + \frac{\epsilon}{100} > UB$, where *UB* is the best-known upper bound (for a minimization problem). By proceeding this way, the algorithm eliminates many nodes from the enumeration tree. Ultimately, it finds a solution with a guaranteed optimality gap of at most ϵ . Naturally, computational effort and solution quality trade-off depend on ϵ . The smaller the ϵ value, the better the solution will be at



Fig. 5 Beam-search approach example on a branch-and-bound enumeration tree

a higher computational effort. This trade-off is empirically assessed in Sect. 4.8. Another critical issue in beam-search heuristics is the search strategy. The nodes can be explored by breadth- or depth-first search strategies or using some heuristic to improve the current solution. This issue is also investigated in Sect. 4.8.

Figure 5 shows some nodes within the branch-and-bound enumeration tree when a beam-search approach is applied. Only teal nodes are considered in the beam search. In contrast, yellow nodes are discarded from the beam-search tree because their best solutions have an optimality gap larger than the ε percentage.

4 Experimental results

In this section, we test a set of experiments on a case study with various scenarios to verify the effectiveness of our BBP strategy to mitigate the bus bunching effect and reduce the user waiting times.

The system's current state at time t_0 is the input that the mathematical model H-BBP(t_0) receives. This model is coded in C++ and solved using the commercial solver Gurobi 8.11. Since we have a linear objective function, valid inequalities, and we avoid origin–destination matrices, our formulation can be solved in less than 15 s for most instances. In this manner, the rapidly obtained holding times are input for the discrete-event simulation coded in R 3.6.0 every Ω minutes along the planning period.

The experiments were carried out on a computer with macOS Catalina 10.15.5 with an Octa-core of 3 GHz Intel Xeon E5 processor and 16GB 1866MHz DDR3 of RAM. All the instances and the code of the mathematical model are available at https://doi.org/10.6084/m9.figshare.16688770.v1.

4.1 Case study: Ecovía in Monterrey, Mexico

The case study analyzed in this work contemplates the morning rush hours (6:00–8:00) of the Ecovía, a BRT corridor that provides service in Monterrey city and its metropolitan area in Mexico. The one-way corridor has 40 stations, including two terminal stops (Lincoln and Valle Soleado) located at opposite ends of the line, as schemed by the red line in Fig. 6. We consider the BRT Ecovía from Lincoln to Valle Soleado, which is around 30.2 km long. Notice that it intersects two city underground rapid transit rail system lines: Line 1 at Mitras station and Line 2 at Regina station. Mitras is the most critical stop since this bus/subway station has higher arrival and alighting rates. Although the Ecovía BRT has three types of buses with different capacities, 80% of the buses are of the same class, so our experiments are performed with only the type of buses with a maximum capacity of 80 persons.

The Ecovía simulation period is $T_{\text{max}} = 120$ min with a frequency F of 2 min during the rush hour. Thus, there could be more than 60 buses simultaneously in the corridor. The step size of the simulation is 10 s. This parameter was determined empirically.



Fig. 6 Map of Monterrey, Mexico. Bus rapid transit Ecovía (red) intersects Line 1 (yellow) and Line 2 (green) of the underground rapid transit rail system. Adapted with permission of the author from [Ecovía Monterrey], from Google, n.d., https://www.google.com/maps/@25.6983801,-100.3365817,12.41z. All rights reserved 2021 by Google (color figure online)

We have statistically determined parameter vectors $(\lambda^s)_{s \in S}$ and $(\gamma^s)_{s \in S}$ used in the simulation and the H-BBP(t_0) model through a field survey. At each stop, the users' arrival process λ^s , $s \in S$, is a Poisson distribution with around eight passengers per minute rates at Mitras or Lincoln stations, while as few as 0.12 passengers per minute for the final stations close to Valle Soleado. We use the expected value of these distributions for the H-BBP(t_0) model. We have also determined the proportion of users alighting at each station. The first stops have a small proportion of $\gamma^s = 1\%$, while in Mitras it is $\gamma^{\text{Mitras}} = 75\%$. Also, from our field survey, we fixed the boarding and alighting time per person in $\overline{U} = \underline{U} = 2$ s, similar to what was reported in Fernández et al. (2008) and Tirachini (2014). These times already consider a congested rush-hour system with passenger friction since users board and alight by the same door. When a bus arrives at a station, opening and closing its doors takes an average of G = 5 s (Dueker et al. 2004).

For the simulation, the travel times along the corridor are a Lognormal distribution with a mean of 0.77 and variance of 0.4 min as in Delgado et al. (2012) and Hernández-Landa et al. (2015). Still, one could use the collected real date to fine-tune these parameters as in Ricard et al. (2022). To mimic a real bus line, the simulation should not stop its evolution while the optimization stage is computing the bus holding times. Nevertheless, we pause the simulation in our experimental setting until the model is executed.

For the H-BBP(t_0) model, we consider an average speed of 60 km/h for all buses and $\kappa = 0.20$. Preliminary experimentation regarding the values of κ showed a

612

positive relationship with the number of bus bunching events. However, our results were not conclusive when related to the total user waiting time. The company validated this value of κ by arguing that diminishing the service quality was a detriment to their user demand share.

To validate our simulation (without the optimization phase), we compared the number of bus bunching events, the average number of persons in the main bus stops, and the average bus load with observations of the real BRT system.

As mentioned in Sect. 2, a few works dealing with bus bunching adhere to the features of our BBP problem and our case study. The work of Delgado et al. (2012) is the closest to our research. However, we cannot make a comparison between their methodologies and ours. Indeed, they limit the boarding passengers at stops since they do not consider a dynamic bus capacity at each stop as we do in this work. The user's boarding limits strategy cannot be applied in our case study, since it would create user displeasure, which would worsen the service reliability. Moreover, their instances are unsuitable for our case study, since there is almost no flexibility for the holding times. Nevertheless, we use our instances and the model proposed in this paper to compare our policy with the optimization policy presented by Delgado et al. (2012). The comparison results can be found in Sect. 4.7.

Since we are using a simulation of the Ecovía, we can stress the system to determine high-quality holding policies. Thus, different scenarios address the following questions (bold parameters are the real case). We wish to investigate the following critical issues in our experimental work.

- We wish to investigate if the boarding and alighting times are the leading cause of bus bunching. If this is the case, the companies may evaluate investing in other vehicles with two doors instead of one or implement different queuing processes. Thus, we consider $U = \overline{U} \in \{1, 2, 3\}$ s per person, as in Tirachini (2013).
- We want to know how the bus frequency F impacts the BRT behavior and whether higher frequencies reduce bus bunching. We consider $F \in \{1, 2, 3, 4\}$ min per bus. F = 4 corresponds to the COVID-19 situation where traveling in public transportation was not recommended.
- We wish to establish how many times the holding times must be computed during the planning period. Thus, we test with Ω ∈ {2, 5, 7, 15, 20, Ø}, where Ø means that no holding times are applied.
- We want to evaluate whether holding times should be an integer number so drivers adopt them more easily. In addition, we want to know the effect of rounding a holding time, for example, from 35.8 s to 30 s or 1 min. Thus, we consider both cases; the holding times in the H-BBP(t_0) may be integer or continuous values.

Therefore, $|\underline{U}| \times |F| \times |\Omega| \times 2 = 144$ configurations were tested to evaluate and compare the H-BBP(t_0) model under different operating conditions. Ten simulation runs were carried out for each scenario to guarantee statistically sound results.



Fig. 7 Bus-bunching pairs with respect to optimization process frequency Ω when the bus boarding and alight times are **a** 1, **b** 2, and **c** 3 s

4.2 Number of bus-bunching pairs

The main objective of the H-BBP(t_0) model is to avoid paired bus bunching. Thus, we apply our BBP methodology to the scenarios with boarding and alighting $\underline{U} = \overline{U} \in \{1, 2, 3\}$ s per person corresponding to the three box-plot graphs of Fig. 7. The vertical axis indicates the total number of bus-bunching pairs throughout the 2 h. Note that if there are four buses at a station, then there are three bus bunching events. The horizontal axis represents the optimization process's frequency Ω . The four different types of boxes correspond to the different frequencies *F*.

Our BBP methodology significantly reduced the number of bus-bunching pairs in the evaluation scenarios, so users will perceive a more reliable service. Note that generating the holding times more often ($\Omega < 2$) does not necessarily imply a higher reduction of bus-bunching pairs. A period of $\Omega = 5$ or $\Omega = 7$ min obtains the best results without overwhelming the drivers with frequent holding requests. On the one hand, when the boarding time of users $\overline{U} = \underline{U}$ is 2 or 3 s (Fig. 7b and c), the smallest amount of bus-bunching pairs occurs when the optimization process Ω is performed every 7 min. On the other hand, when the boarding time of the users $\overline{U} = \underline{U}$ is 1 s (Fig. 7a), it is recommended that the optimization phase runs every 5 min. For systems with a low bus frequency F = 4, the bus bunching reduction is not as drastic as for the high-frequency ones of F = 1 min. The case study is observed in Fig. 7b with a bus frequency of F = 2 without a bus bunching strategy, and there are around 470 bunching pairs. With an intervention every $\Omega = 5$ min, we could reduce busbunching reductions we get with the BBP methodology. For Fig. 7c, corresponding to a COVID-19 situation where line frequency is augmented, we observe that the causes of bus bunching are the boarding and alighting times.

In model H-BBP(t_0), constraints (7) bound the holding times by *H* minutes. We used H = 5 for the previous experimentation of Fig. 7, but we carried out other experiments without this bound obtaining similar results about the bus bunching reduction. However, longer holding times could be resented by the users.

In our BRT case study, the distance between stations is short (approximately 700 ms), and the frequency of bus departures during rush hour is very high. It is a heavily loaded system with little freedom of movement. A system with these characteristics naturally tends to become bunched. Although there are alternatives that can be addressed that would possibly allow a significant decrease in the number of bus bunching events, such as a redesign of the network or the bus stops, through the results of our experiments, we found that our methods will help alleviate the bus bunching events and the user waiting times (Fig. 7).

4.3 Time-space bus trajectories

Bus trajectories in time-space graphs allow us to observe the bus bunching cases at a glance whenever two or more lines collide. The four graphs in Fig. 8 are similar to the ones in Sect. 1. Each line is a bus trip, where the time in minutes is on the horizontal axis, while the position of the buses is on the vertical one. Figure 8 corresponds to the case study parameters: boarding and alighting times equal to 2 s and a bus frequency of 2 min. Figure 8d presents the simulation when no holding times are applied ($\Omega = \emptyset$). We observe many bus-bunching pairs and long intervals without buses at the stops, especially after kilometer 10. The optimization stage is done every 20 and 10 min, respectively, in the time-space Fig. 8b and c. We can observe a significant change since there are fewer bus bunching cases than without considering holding times. Indeed, the trips tend to be more equidistant, so there are fewer long intervals without buses. We obtain the best solution when $\Omega = 5$ min corresponds to Fig. 8a. A bus bunching reduction of around 30% is achieved compared to the first scenario.



Fig. 8 Bus trajectories in 2 h with bus boarding and alight times equal to 2 s ($\overline{U} = \underline{U} = 2$), bus frequency of 2 min, and considering different optimization process frequencies **a** every 5 min, **b** every 10 min, **c** every 20 min, and **d** without optimization process

4.4 User waiting times

The objective function in the H-BBP(t_0) model seeks to maintain quasi-regular headways to reduce the amount of bus-bunching pairs. The main question is if this strategy positively correlates with a reduction of user waiting times, which is the essential characteristic of the quality in a public transport system (Ben-Akiva and Lerman 2018; Eberlein et al. 2001).

Figure 9 shows that in our BBP methodology, the minimization of the bus-bunching pairs implies a reduction of the user waiting times. In box-plot graphs of Fig. 9, the vertical axis corresponds to the average user waiting time until they board the first bus. This average was obtained considering all the system stations and based on real data. The horizontal axis is the optimization process frequency Ω . As in the



 $(0) \quad 0 = \underline{0} = 0.$

Fig. 9 Average user waiting times with respect to optimization process frequency Ω when the bus boarding and alighting times are of **a** 1, **b** 2, and **c** 3 s

previous box-plots of Fig. 7, we vary the bus frequency *F* and the user's boarding and alighting times $\overline{U} = \underline{U}$: (a) 1, (b) 2, and (c) 3 s.

Figure 9a and b, show that the more frequent the optimization interventions are, the more the average waiting times are reduced. For the case study (Fig. 9b with F = 2), the average waiting times are reduced by 30% for $\Omega = 5$ min. The box plot of Fig. 9c shows the case with boarding and alighting times of 3 s, which is not usual but observed during the COVID-19 pandemic. In this case, for the bus frequency of 2 and 3 min, the waiting times are only reduced when the optimization frequency Ω is small. The variances of the average waiting times are short for the case of 1 s for boarding and alighting (Fig. 9a), but they increase as $\overline{U} = U$ augment.



Fig. 10 Average user waiting times with respect to optimization process frequency $\Omega = \emptyset$ (inner ring) and $\Omega = 7$ (outer ring) with a bus frequency of F = 2, and boarding and alighting waiting times of $\overline{U} = \underline{U} = 2$

Figure 10 allows us to analyze further the average user waiting times when the BBP methodology is used. In the figure, there are two percentage ring graphs; both have a bus frequency of F = 2, and boarding and alighting waiting times of $\overline{U} = \underline{U} = 2$ correspond to the case study. The inner ring is without holding times $(\Omega = \emptyset)$, while the outer one has a frequency of holding times computation of $\Omega = 7$ min. The user waiting times are classified into five categories: 0–5 min, 5–10 min, 10–15 min, 15–20 min, and more than 20 min.

From Fig. 10, we observe that computing the holding times via our mathematical model every $\Omega = 7$ min positively impacts user waiting time. Indeed, 15% of the users improved their average waiting times. The bus bunching problem cannot be eradicated with our BBP method, but now only 11% of the passengers wait for more than 20 min to board a bus.

4.5 Bus loads

When two buses arrive at a station simultaneously, most of the waiting users decide to board the first bus. This behavior causes the load on the first bus to be significantly higher than on the bus following it, creating unbalanced occupancy rates among the buses serving the system. The users in a crowded bus are uncomfortable and perceive poor service quality.

Figure 11 shows the load of each bus from the initial station (Lincoln) to the final station (Valle Soleado). Each line represents a bus trip in a setting with a departure frequency of F = 2 and boarding and alighting waiting times of $\overline{U} = U = 2$. The



Fig. 11 Bus load at each station when the optimization process is $\Omega = \emptyset$ (blue trips) and $\Omega = 5$ (red trips) with F = 2 and $\overline{U} = \underline{U} = 2$ (color figure online)



(a) Continuous holding time variables.

(b) Integer holding time variables.

Fig. 12 Values of the holding time variables obtained by $\text{H-BBP}(t_0)$ when they are **a** continuous, and **b** discrete

blue trips do not apply the holding times computed by our model ($\Omega = \emptyset$), while the red trips compute the holding times every $\Omega = 5$ min.

Figure 11 reveals that when the optimization process is not applied (blue trips), there is more variability in the bus loads compared to the case where the optimization is performed every 5 min (red trips). Thus, by minimizing the bus bunching we also obtain a more balanced bus load pattern since we observe less variability.

4.6 Holding times analysis

In this section, we analyze if there is a consequence of considering integer holding times which are easier to communicate to the drivers. Thus, Fig. 12 considers both cases, the holding times in the H-BBP(t_0) model are integer (Fig. 12b) or continuous values (Fig. 12a). The boarding and alighting waiting times are $\overline{U} = \underline{U} = 2$, and we consider all the optimization frequencies Ω and all the different headways *F*. In both figures, the pentagon's vertices indicate the different values the holding times can take (accumulated values of all instances and different parameters). In Fig. 12a, vertex i = 1, ..., 4 groups all values in [i - 1, i), while vertex 5 groups the values of the holding times between [4, 5]. Each colored pentagon represents the number of occurrences of a holding value type differentiated by the different bus frequencies *F*.

Figure 12 shows that the integer holding times are less applied than real ones, which is a great advantage for the practical implementation of our methodology. Indeed, Fig. 12a exhibits holding times have values ranging between [0, 60] s, which can be challenging to apply in a real case. Independently of the nature of the holding time variables in the H-BBP(t_0) model, the value of 5 min is the one with the highest recurrence regardless of service frequency.

We did not notice a significant solving time difference when the holding variables take discrete or continuous values in the H-BBP(t_0) model. On average, the running times for both cases were 15 s. Moreover, the number of bus-bunching pairs is not affected by the nature of the holding variables. We also analyzed the running times when solving the mathematical model with and without valid inequalities (13) and (14), considering discrete or continuous holding time variables. A slight improvement in the running time was observed when using valid inequalities, especially with discrete holding variables.

It is also interesting to analyze at which stations the holding times are more recurrent. We could determine the necessity of applying holding times at all the stations or only at some of them with this information. Figure 13 consists of 12 histograms whose horizontal axis corresponds to the stops (stop 14 is Mitras), while the vertical one corresponds to the number of times that holding times were needed. We consider the bus frequency F(1,2,3, and 4 min, corresponding to the columns of the figure) and the boarding and alighting times (1, 2, and 3 s, corresponding to the lines of the figure). The colors represent the different optimization frequencies Ω . This figure shows the case with integer holding time values.

In Fig. 13, the case study is when the bus frequency is F = 2 and the boarding and alighting times are 2 s. Before the Mitras stop, there is a large number of holding times needed. This behavior holds when $\overline{U} = \underline{U} = 2$ and when F = 2 and F = 3. Nevertheless, when the boarding and alighting times are short, $\overline{U} = \underline{U} = 1$,



Fig. 13 Number of holding times needed at each BRT station



Fig. 14 Comparison between optimizing every 5 min and optimizing every time a bus arrives at a stop concerning **a** amount of bus-bunching pairs, and **b** average user waiting times

the holding times are more uniform through the stops. Thus, the whole Ecovía BRT is affected by the users' boarding and alighting times and ideal headways. When the boarding and alighting times are equal to 2 or 3 s, the tendency is to hold the buses only in the first half of the stops.

4.7 Optimization at every stop

This section aims to compare our methodology with other literature approaches that compute the holding times each time a bus arrives at a stop as in Delgado et al. (2009, 2012) and Sánchez-Martínez et al. (2016). In Delgado et al. (2012), they compute the holding times for all buses at all future stops. However, they only keep the holding time value that affects the bus arriving at the station. All other holding time values found by the model are discarded. We compare the Delgado et al. (2012) policy against our case, which performs the optimization stage every 5 min. We consider the real parameters of our case study, $\overline{U} = \underline{U} = 2$ s and F = 2 min. Our comparison is only on the number of times the optimization stage is called. While we implicitly reduce bus bunching to decrease user waiting times, the model by Delgado et al. (2012) directly minimizes the user waiting times, which employs a quadratic objective function that is more time-consuming to solve.

The comparison between the two policies is in Fig. 14. Figure 14a shows the bus bunching events, while Fig. 14b shows the user waiting times over 2 h. We use our beam-search heuristic with a relative optimality gap of 5% to solve our model for both policies.

, and heuristic	
t search.	
adth-firs	
arch, bre	
oth-first se	
egies: dep	
arch strat	
ig the sea	
odes usir	
kplored n	
aber of ex	
and nun	
ng times	
3 Runni	
Table	

Search	ε	Buses									
		20		30		40		50		60	
		Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes
DFS	20	32.7	1	7.8	1	3.6	1	3.2	1	2.1	1
	10	37.3	56	15.0	1	7.1	9	9.6	1	2.1	1
	5	38.0	88	21.9	7	7.3	12	10.1	7	7.2	1
	1	42.5	139	23.4	190	10.1	114	12.7	82	7.2	1
	1×10^{-5}	4.4	152	I	10,591,108	10.0	126	I	9,851,332	I	7,720,914
BFS	20	23.6	1	8.9	1	4.0	1	3.6	1	2.6	1
	10	24.4	1	20.7	1	5.9	1	12.2	1	2.6	1
	5	38.2	29	20.7	1	10.7	1	15.0	68	2.7	1
	1	33.7	185	24.7	1	12.9	213	16.6	1243	7.0	1485
	1×10^{-5}	31.9	1815	29.2	10	13.2	1070	29.2	2169	7.0	1485
HEU	20	29.3	1	7.36	1	6.4	374	3.1	1	2.3	1
	10	27.1	1	14.2	1	6.1	374	8.3	26	2.2	1
	5	33.5	88	14.0	1	6.1	374	13.9	292	4.3	194
	1	32.0	183	29.7	1581	10.1	1761	15.0	1612	5.4	1770
	1×10^{-5}	34.4	229	30.8	2007	10.8	1923	24.1	3304	5.6	2027

From Fig. 14a, we observe that the median for $\Omega = 5$ is just under 375 bus bunching-pairs, while for the other case, the median is close to 435. Thus, making fewer optimization calls is better in terms of bus bunching events than making optimization calls almost every 10 s (optimization every time a bus arrives at a stop). From Fig. 14b, we can observe that the average user waiting time is between 52 and 55 min for $\Omega = 5$, which is already a long time (we have validated this time in the Ecovía). When the optimization process is performed every time a bus arrives at a stop, there are average waiting time values lower than 50 min, but there are also values higher than 60. The variance is much higher than in the previous case, implying a less reliable service.

4.8 Beam-search results

As explained in Sect. 3.4, there is a trade-off between running time and solution quality that depends on the relative optimality gap parameter ϵ and the used search strategy. This section assesses this trade-off by carrying out the following experiment based on the real case corresponding to a frequency of F = 2, boarding and alighting times of $\overline{U} = U = 2$, and 40 stops.

We execute the beam-search heuristic using $\varepsilon \in \{0.2, 0.1, 0.05, 0.01, 1 \times 10^{-5}\}$. The last ε value corresponds to the optimal solution case for comparison purposes. We implement our beam search by directly using the branch-and-bound library of the Gurobi solver by using three different node exploration strategies: (1) depthfirst search chooses the most recently created node and explores as far as possible along each branch before backtracking; (2) breadth-first search explores each node by level; (3) family of internal heuristics to improve the node search.

The running times can be found in Table 3, where the first column displays the beam-search exploration strategy. DFS, BFS, and HEU denote depth-first search, breadth-first search, and heuristic, respectively. The second indicates the ϵ value used. The remaining columns show the running times and the number of explored nodes for instances with 20, 30, 40, 50, and 60 buses. Each value is the average of ten instances. The symbol "–" indicates that the time limit of 10,800 s was reached.

There are several interesting observations. First, as expected, the smaller the value of ε , the larger the computational effort; however, the increment in time is marginal, kept in the same order of magnitude. Going from an ε value of 0.05–0.01, the increment in time is slightly more, also depending on the search strategy. For instance, going from 0.05 to 0.01 produces around 70–100% increment under the breadth-first search and heuristic strategies, but merely 15–20% under the depth-first search strategy. When strategies are compared, for the 20- and 30-bus instances, the heuristic search strategy works slightly better than the other two; however, for the 40-, 50-, and 60-bus instances, the depth-first search strategy yields the best results. Now, comparing the running times obtained by the different versions of the beam search with the exact method (corresponding to ε equal to 1×10^{-5}), we can observe that under the depth-first search strategy, there were three cases where the algorithm was unable to optimally solve the problem reaching the time limit of 10,800 s.

Note that in most cases, when there are more buses, the execution times are lower, probably because there is less flexibility in the values of the decision variables. In instances with 60 buses, we can find excellent quality solutions in less than 5 s. Moreover, the best performance is found with the heuristic strategy.

5 Conclusions

With the BBP methodology, we minimize the amount of bus bunching, that is, the number of buses of the same line cruising head-to-tail or arriving at a stop simultaneously by determining holding times at the bus stops.

Our BBP solution approach presents an optimization stage that determines the holding times of each bus at the stops by considering the actual information along the bus line and predicting the future events in the line. This stage is a novel quadratically constrained integer linear programming model that yields the holding times, which are immediately communicated to the drivers to recover quasi-regular headways between the buses. We also present a discrete-event simulation program representing a bus rapid transit system. In this manner, we can retrieve simulated real-time data from the system to validate the efficiency of our mathematical model.

The experimental results show that our methodology mitigates bus bunching by reducing it up to 45%, especially in scenarios with high bus frequency. Moreover, the user waiting times also decrease considerably in scenarios where the dwelling and alighting times are 1 or 2 s, reduced by up to 30%. Every 5 min is the frequency in which the holding times must be computed to have the best quasi-regular head-ways between the buses and a balanced bus load. In addition, it leads to a more reliable transport service than optimizing every time a bus arrives at a station, as many works in the literature do. Furthermore, we used a beam-search heuristic using different search strategies and relative optimality gap parameters. We obtained solutions for instances up to 60 buses with a gap of less than 5% in less than 3 s.

Even if our methodology is based on the Ecovía, it can be easily adapted to different BRT systems. For example, in our case study, there is only one door for boarding and alighting actions. Simultaneous boarding and alighting will only impact the model and simulation parameters corresponding to a similar behavior with the $\overline{U} = U = 1$ (Figs. 7a, 9a).

The impact of the values, such as the headway tolerance from the ideal headway, the importance between bus bunching events, and separated buses (slopes of the penalty function), will lead to a better understanding of the problem. Future research lines may consider additional strategies, such as bus insertion at stations with a high passenger flow. Besides, we have considered 40 one-way stations. Nevertheless, the corridor has the functionality of a round-trip one. Only available buses leave at Valle Soleado (the end station). Thus, when bus bunching is severe, there could be departure times that cannot be fulfilled. This framework may be enhanced by considering uncertainty in the user's travel and arrival times, bus speeds, and scenarios where accidents or bus breakdowns may occur. Finally, one may implement heuristic methods to determine holding times for larger BRT systems.

Acknowledgements The research was supported by the Mexican Council for Science and Technology (Grant CONACyT FC 2016-2/1948). The first author was financially supported by graduate scholarship 781117, also by CONACyT. Additional support was obtained by SEP (PRODEP).

References

- Akeb H, Hifi M, M'Hallah R (2009) A beam search algorithm for the circular packing problem. Comput Oper Res 36(5):1513–1528
- Argote-Cabanero J, Daganzo CF, Lynn JW (2015) Dynamic control of complex transit systems. Transp Res Part B Methodol 81:146–160
- Barnett A (1974) On controlling randomness in transit operations. Transp Sci 8(2):102-116
- Ben-Akiva M, Lerman SR (2018) Discrete choice analysis: theory and application to travel demand. Transportation studies. MIT Press, Cambridge
- Bennell JA, Cabo M, Martinez-Sykora A (2018) A beam search approach to solve the convex irregular bin packing problem with guillotine cuts. Eur J Oper Res 270(1):89–102
- Ceder A (2016) Public transit planning and operation: modeling, practice and behavior, 2nd edn. CRC Press, Boca Raton
- Chen Q, Adida E, Lin J (2013) Implementation of an iterative headway-based bus holding strategy with real-time information. Public Transp 4(3):165–186
- Cortés CE, Sáez D, Milla F, Núñez A, Riquelme M (2010) Hybrid predictive control for real-time optimization of public transport systems' operations based on evolutionary multi-objective optimization. Transp Res Part C Emerg Technol 18(5):757–769
- Daganzo CF (2009) A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons. Transp Res Part B Methodol 43(10):913–921
- Daganzo CF, Pilachowski J (2011) Reducing bunching with bus-to-bus cooperation. Transp Res Part B Methodol 45(1):267–277
- Degeler V, Heydenrijk-Ottens L, Luo D, Van Oort N, Van Lint H (2021) Unsupervised approach towards analysing the public transport bunching swings formation phenomenon. Public Transp 13(3):533–555
- Delgado F, Muñoz JC, Giesen R, Cipriano A (2009) Real-time control of buses in a transit corridor based on vehicle holding and boarding limits. Transp Res Rec 2090(1):59–67
- Delgado F, Muñoz JC, Giesen R (2012) How much can holding and/or limiting boarding improve transit performance? Transp Res Part B Methodol 46(9):1202–1217
- Deng Y-J, Liu X-H, Hu X, Zhang M (2020) Reduce bus bunching with a real-time speed control algorithm considering heterogeneous roadway conditions and intersection delays. J Transp Eng Part A Syst 146(7):04020048
- Drabicki A, Kucharski R, Cats O (2023) Mitigating bus bunching with real-time crowding information. Transportation 50:1003–1030. https://doi.org/10.1007/s11116-022-10270-3
- Dueker KJ, Kimpel TJ, Strathman JG, Callas S (2004) Determinants of bus dwell time. J Public Transp 7(1):2–40
- Eberlein XJ, Wilson NHM, Bernstein D (2001) The holding problem with real-time information available. Transp Sci 35(1):1–18
- Estrada M, Mensión J, Aymamí JM, Torres L (2016) Bus control strategies in corridors with signalized intersections. Transp Res Part C Emerg Technol 71:500–520
- Fernández R, del Campo MA, Swett C (2008) Data collection and calibration of passenger service time models for the Transantiago system. In: Proceedings of the European transport conference, volume 30, Noordwiijkerhout, Netherlands
- Fu L, Liu Q, Calamai P (2003) Real-time optimization model for dynamic scheduling of transit operations. Transp Res Rec 1857(1):48–55
- Gkiotsalitis K, Cats O (2019) Multi-constrained bus holding control in time windows with branch and bound and alternating minimization. Transportmetrica B Transp Dyn 7(1):1258–1285
- Hall R, Dessouky M, Lu Q (2001) Optimal holding times at transfer stations. Comput Ind Eng 40(4):379–397
- He S-X (2015) An anti-bunching strategy to improve bus schedule and headway reliability by making use of the available accurate information. Comput Ind Eng 85:17–32

- He S-X, Dong J, Liang S-D, Yuan P-C (2019) An approach to improve the operational stability of a bus line by adjusting bus speeds on the dedicated bus lanes. Transp Res Part C Emerg Technol 107:54–69
- He S-X, Liang S-D, Dong J, Zhang D, He J-J, Yuan P-C (2020) A holding strategy to resist bus bunching with dynamic target headway. Comput Ind Eng 140:106237
- Hernández-Landa LG, Morales-Marroquín ML, Sánchez-Nigenda R, Ríos-Solís YÁ (2015) Linear bus holding model for real-time traffic network control. In: Miguel MM, De La Mota IF, Serrano DG (eds) Applied simulation and optimization. Springer, Cham, pp 303–319
- Hickman MD (2001) An analytic stochastic model for the transit vehicle holding problem. Transp Sci 35(3):215–237
- Ibarra-Rojas OJ, Delgado F, Giesen R, Muñoz JC (2015) Planning, operation, and control of bus transport systems: a literature review. Transp Res Part B Methodol 77:38–75
- Islam K, Moosa IM, Mobin J, Nayeem MA, Rahman MS (2019) A heuristic aided stochastic beam search algorithm for solving the transit network design problem. Swarm Evol Comput 46:154–170
- Laskaris G, Cats O, Jenelius E, Rinaldi M, Viti F (2019) Multiline holding based control for lines merging to a shared transit corridor. Transportmetrica B Transp Dyn 7(1):1062–1095
- Leemis LM, Park SK (2006) Discrete-event simulation: a first course. Pearson Prentice Hall, Upper Saddle River
- Li M, Li Y (2022) Bus predictive-control method considering the impact of traffic lights. J Adv Transp 2022:5280109
- Liang S, Ma M, He S, Zhang H, Yuan P (2019) Coordinated control method to self-equalize bus headways: an analytical method. Transportmetrica B Transp Dyn 7(1):1175–1202
- Liang S, He S, Zhang H, Ma M (2021) Optimal holding time calculation algorithm to improve the reliability of high frequency bus route considering the bus capacity constraint. Reliab Eng Syst Saf 212:107632
- Ling K, Shalaby A (2004) Automated transit headway control via adaptive signal priority. J Adv Transp 38(1):45–67
- Liu H, Skabardonis A, Zhang W-B (2003) A dynamic model for adaptive bus signal priority. In: Proceedings of the 82nd transportation research board annual meeting, Washington
- National Academies of Sciences, Engineering, and Medicine (2013) Transit capacity and quality of service manual. TCRP Report 165
- Prassas ES, Roess RP (2020) The highway capacity manual and the committee on highway capacity and quality of service. The highway capacity manual: a conceptual and research history. Volume 12 of Springer tracts on transportation and traffic, vol 2. Springer, Cham, pp 1–16
- Ricard L, Desaulniers G, Lodi A, Rousseau LM (2022) Predicting the probability distribution of bus travel time to measure the reliability of public transport services. Transp Res Part C Emerg Technol 138:103619
- Sajikumar S, Bijulal D (2022) Zero bunching solution for a local public transport system with multipleorigins bus operation. Public Transp 14:655–681
- Sánchez-Martínez GE, Koutsopoulos HN, Wilson NHM (2016) Real-time holding control for high-frequency transit with dynamics. Transp Res Part B Methodol 83:1–19
- Shi H, Nie Q, Fu S, Wang X, Zhou Y, Ran B (2022) A distributed deep reinforcement learning-based integrated dynamic bus control system in a connected environment. Comput Aided Civ Infrastruct Eng 37(15):2016–2032
- Sokolowski JA, Banks CM (2011) Principles of modeling and simulation: a multidisciplinary approach. Wiley, New York
- Soza-Parra J, Muñoz JC, Raveau S (2021) Factors that affect the evolution of headway variability along an urban bus service. Transportmetrica B Trans Dyn 9(1):479–490
- Sun A, Hickman M (2005) The real-time stop-skipping problem. J Intell Transp Syst 9(2):91-109
- Sun A, Hickman M (2008) The holding problem at multiple holding stations. Computer-aided Systems in Public Transport. Volume 600 of Lecture notes in economics and mathematical systems. Springer, Berlin, pp 339–359
- Sun W, Schmöcker J (2018) Considering passenger choices and overtaking in the bus bunching problem. Transportmetrica B Transp Dyn 6(2):151–168
- Tamannaei M, Irandoost I (2019) Carpooling problem: a new mathematical model, branch-and-bound, and heuristic beam search algorithm. J Intell Transp Syst 23(3):203–215
- Tian S, Li X, Liu J, Ma H, Yu H (2022) A short-turning strategy to alleviate bus bunching. J Ambient Intell Humaniz Comput 13(1):117–128

- Tirachini A (2013) Bus dwell time: effect of different fare collection systems, bus floor level and age of passengers. Transportmetrica A Transp Sci 9(1):28–49
- Tirachini A (2014) The economics and engineering of bus stops: Spacing, design and congestion. Transp Res Part A Policy Pract 59:37–57
- Verbich D, Diab E, El-Geneidy A (2016) Have they bunched yet? An exploratory study of the impacts of bus bunching on dwell and running times. Public Transp 8(2):225–242
- Wang J, Sun L (2020) Dynamic holding control to avoid bus bunching: a multi-agent deep reinforcement learning framework. Transp Res Part C Emerg Technol 116:102661
- Williams HP (2013) Model building in mathematical programming, 5th edn. Wiley, New York

Wolsey LA (2020) Integer programming, 2nd edn. Wiley, New York

- Xuan Y, Argote J, Daganzo CF (2011) Dynamic bus holding strategies for schedule reliability: Optimal linear control and performance analysis. Transp Res Part B Methodol 45(10):1831–1845
- Yavuz M (2017) An iterated beam search algorithm for the green vehicle routing problem. Networks 69(3):317–328
- Yu H, Chen D, Wu Z, Ma X, Wang Y (2016) Headway-based bus bunching prediction using transit smart card data. Transp Res Part C Emerg Technol 72:45–59

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.