https://doi.org/10.1016/j.trc.2023.104269

Efficient Heuristic Methods for Berth Allocation at Multi-line, Multi-berth Curbside Bus Stops

Minyu Shen[a,b], Weihua Gu[b*], Sangen Hu[c], and Feng Xiao[a]

[a]School of Management Science and Engineering, Southwestern University of Finance and Economics, China
[b]Department of Electrical Engineering, The Hong Kong Polytechnic University
[c]Department of Civil and Transportation Engineering, Guangdong University of Technology, China

Abstract

Transit management agencies often pre-allocate a multi-berth stop's berths to specific bus lines so that passengers can find the right place to wait for their buses. Developing optimal berth allocation plans that minimize the total bus delay is essential for mitigating bus queues at busy multi-berth stops. However, this problem is challenging due to the huge solution space, the high degree of stochasticity in bus queues, and the resulting extremely high computational cost. In this paper, we first propose a simple heuristic method inspired by queueing theory. It is based on the idea that evenly distributing the total traffic intensity (defined as the total bus arrival rate times the mean dwell time) among all the berths would produce a lower bus delay. Numerical results demonstrate that this simple method generated very good berth allocation plans (with optimality gaps < 6% for no-overtaking and free-overtaking stops) in seconds! It does not rely on time-consuming simulation surrogate models or numerous input data such as the stochastic bus arrival processes and dwell time distributions of each bus line. To further improve the simple heuristic's performance (especially for limited-overtaking stops), we develop a cluster-based nested partition algorithm that can find a near-optimal plan (e.g., with an optimality gap of < 3%) in a much shorter time than a previous algorithm. The algorithm employs the simple heuristic plan as the initial solution. Our methods can be applied to stops with various berth numbers, different proximities to nearby traffic signals, and under diverse bus queueing rules.

Keywords: Bus stop queues; Bus delays; Berth allocation; Traffic intensity; Simulation-based optimization; Nested partition

*Corresponding author. Email: weihua.gu@polyu.edu.hk

1 Introduction

1.1 Background and literature review

Busy, multi-berth curbside bus stops are major bottlenecks in a bus system (Fernandez and Planzer, 2002). Bus lines visiting these stops can share the berths to serve the passengers (Gu et al., 2011; Zhao et al., 2019), meaning that an arriving bus can enter any available berth of a stop. This "berth-sharing" strategy creates two problems. First and the most important, passengers waiting at the platform do not know which berth their bus will enter. Thus, they may need to scurry along the platform to catch an arriving bus, creating chaos in the platform. Second, when two or more buses on the same line arrive at a stop simultaneously (a common phenomenon known as "bus bunching"), they would occupy the limited berths and prevent buses on other lines from entering the stop to serve their passengers. To avoid the above problems, many transit agencies opt to allocate each berth of a stop to an exclusive group of lines. This way, passengers can find the right place to wait for their buses and bunched buses on the same line will not occupy multiple berths. On the downside, this "berth-allocating" strategy will diminish the berths' utilization rate and the overall bus discharge flow from the stop, therefore creating bus queues. Inferior allocation plans would further aggravate the queueing problem. Thus, optimally allocating a stop's limited berths to the bus lines is crucial for mitigating bus queues and passenger delays at these stops (Lu et al., 2010; Wu et al., 2011; Tan et al., 2014).

To our best knowledge, Lu et al. (2010) is the first work that studied this problem. However, this work only investigated 2-berth stops where bus overtaking maneuvers are prohibited. (This bus queueing rule is termed the "*no-overtaking*" rule in the literature; see Gu and Cassidy, 2013.) Moreover, they only proposed two empirical guidelines for berth allocation instead of presenting an optimization approach. The two guidelines are: (i) assigning more buses to the downstream berth; and (ii) assigning the bus lines with longer dwell times to the downstream berth. Effects of the two guidelines were assessed in a follow-up study, Wu et al. (2011), by a simulation tool calibrated using real bus arrival and dwell time data.

Built upon guideline (i) of the above studies, Tan et al. (2014) developed a two-stage heuristic algorithm to find the optimal berth allocation plan of 2-berth stops, referred to as Tan's algorithm in the rest of this paper. The stops studied in Tan et al. (2014) follow the "*limited-overtaking*" rule (Gu and Cassidy, 2013), where a bus dwelling in the upstream berth is allowed to overtake a downstream dwelling bus to exit the stop, while no bus can bypass a bus occupying the upstream berth to enter the downstream vacant berth. Stage one of Tan's algorithm partitions the entire solution space into several subsets by *the ratio of bus flows assigned to the two berths*. The most promising subset is then selected by comparing a small number of allocation plans randomly sampled from each subset. In stage two, a heuristic allocation plan is found by searching the most promising subset only. Simulation was used for evaluating the performance (i.e., the mean bus delay) of allocation plans in both stages.

Tan's algorithm has several shortcomings:

(i) The computational cost of the two-stage algorithm largely depends on the number of partitioned subsets. If the number of subsets is large, stage one would be computationally expensive; and if that number is small, each subset would be large and stage two would be time-consuming.

(ii) The partitioning method overlooked the impacts of bus dwell times. Note that the bus dwell time is a major factor affecting bus queueing delays at a stop (Gu et al., 2011; Kittelson & Associates, Inc., 2013). Consider two allocation plans (labeled A and B) for a 2-berth stop, which have similar ratios between the bus flows assigned to the two berths. However, the mean bus dwell time of lines assigned to berth 1 is much greater than that of berth 2 in Plan A, while in Plan B the mean dwell times at the two berths are similar. In this case, the two plans will be grouped into the same subset according to the similar bus flow ratio, but their performance (e.g., the mean bus delays) could be quite different. On the other hand, two plans with distinct bus flow ratios would be sorted into different subsets, but they may have comparable performance due to the joint effect of bus flows and dwell times. Therefore, the near-optimal allocation plans might scatter among several subsets instead of being clustered into one, rendering the "most promising" subset found in stage one questionable.

(iii) The algorithm was only tested for 2-berth, limited-overtaking, mid-block stops that are isolated from the influence of neighboring traffic signals. Even for such a small-scale stop, its runtime would be several hours (see Section 4). Thus, the algorithm might be inapplicable or too time-consuming for stops with more than 2 berths.

Berth allocation problems often require employing a surrogate model to evaluate the performance (e.g., the mean bus delay) of candidate plans. Two types of surrogate models have been used in previous studies: a simulation tool of bus queues (Tan et al., 2014) and an empirical function of failure rate (Alonso et al., 2011). The use of failure rate for calculating bus queueing delays has been shown to be inaccurate (Gu et al., 2015; Bunker, 2018).

Ideally, the surrogate model should employ the analytical solutions to the bus stop queueing models. However, analytical solutions for multi-line, multi-berth stops under the berth allocating strategy are unavailable in the literature. Existing analytical solutions were developed for much simpler, isolated stops under idealized assumptions, e.g., the exponentially distributed bus dwell times (Gu and Cassidy, 2013; Bian et al., 2019). And all these works assumed that the berths are shared by all bus lines. Likewise, analytical methods for other tandem queueing systems (e.g., Gu et al., 2012; He and Chao, 2014) cannot be directly applied to our berth allocation problem. Moreover, the analytical solutions presented in previous studies are already very complicated (see especially Gu et al., 2015), while considering berths allocated to specific bus lines would further increase the complexity.

On the other hand, it is generally easier to develop simulation tools to incorporate realistic operating conditions for various queueing systems (Stamatopoulos et al., 2004; Toledo et al., 2010; Feng et al., 2020; Yang et al., 2020). The optimization approach involving a

simulation tool as the surrogate model is called the simulation-based optimization. This approach has been applied in a growing number of studies in the transportation field (e.g., Wu et al., 2019; Cheng et al., 2019; Li et al., 2022; Zheng et al., 2022). For bus-stop simulation, some works used commercial tool packages, e.g., PARAMICS and MISTRANSIT (Abdulhai et al., 2002; Cortés et al., 2005). However, they are not applicable for our problem because they rely on many calibrated parameters and are computationally expensive. Other studies used simulation tools developed in house (Gibson et al., 1989; Fernandez and Planzer, 2002; Fernández, 2010; Bian et al., 2020).

On a related note, a bus stop's performance (e.g., bus queueing delays) is jointly affected by the berth allocation plan, line-specific bus arrival processes, dwell time distributions, and bus overtaking policies. Regularizing bus headways can reduce dwell time variations and alleviate queueing delays (Newell and Potts, 1964; Hickman, 2001; Daganzo, 2009; Delgado et al., 2012; Estrada et al., 2016; He et al., 2019). Interested readers can find comprehensive reviews of bus bunching studies in Ibarra-Rojas et al. (2015) and Rezazada et al. (2022). Some researchers have explored interactions between bus queueing and bus control strategies (e.g., Bian et al., 2023; Shen et al., 2023).

Concerning bus overtaking policies, the no-overtaking policy has been widely studied (e.g., Gu et al., 2011, 2015; Shen et al., 2019). Gibson et al. (1989) examined the limited-overtaking policy, allowing "overtaking-out" maneuvers (dashed arrow in Fig. 1) but prohibiting "overtaking-in" maneuvers (dotted arrow in Fig. 1). Gu and Cassidy (2013) developed analytical queueing models for limited-overtaking stops. Bian et al. (2019) and Hu et al. (2023) compared stop capacities and bus queueing delays under various overtaking policies using analytical and simulation methods, respectively. Schmöcker et al. (2016) examined a corridor featuring two bus lines that have a shared section and analyzed the impacts of common lines on bus bunching under different bus overtaking policies at shared stops. Wu et al. (2017) investigated the impact of overtaking maneuvers on bus bunching when a late-arrived bus may depart earlier due to fewer boarding passengers. In this paper, we will explore optimal berth allocation under a number of overtaking policies.

1.2 Types of bus stops

Considering the limitations of the previous studies, this paper will develop efficient algorithms for identifying better berth allocation plans for a variety of curbside bus stops with 2 or more berths. The layout of a typical 3-berth curbside stop is illustrated in Fig. 1. Three common types of bus overtaking rules will be examined:

(i) No-overtaking (NO), where no bus overtaking maneuver is allowed in and out of the berths (Gu et al., 2011, 2015). This rule is often enforced on congested roads as overtaking buses that disrupt car traffic in the adjacent lanes are prohibited (Kittelson & Associates, Inc., 2013).

(ii) Limited-overtaking (LO), where a bus can freely exit any berth after serving the pas-

sengers (see the dashed arrow in Fig. 1), but no bus is allowed to enter a vacant berth by overtaking other buses (see the dotted arrow). This rule is also commonly observed in reality since it is generally easier for a bus driver to perform overtaking maneuvers when exiting a stop (Gu and Cassidy, 2013). (For the same reason, the overtaking rule under which buses can freely enter berths by overtaking but cannot exit by overtaking has not been found in the real world. Hence, that rule is ignored in this paper.)

(iii) Free-overtaking (FO), where the overtaking-in and overtaking-out maneuvers are both allowed (Bian et al., 2019). Real-world examples include the sawtooth bus stops described in the Transit Capacity and Quality of Service Manual (Kittelson & Associates, Inc., 2013); see Fig. 2 for illustration.

In addition, our analysis spans from mid-block stops to near- and far-side stops that are located close to a downstream or upstream traffic signal (Shen et al., 2019); see again Fig. 1 for the illustration of a near-side stop.

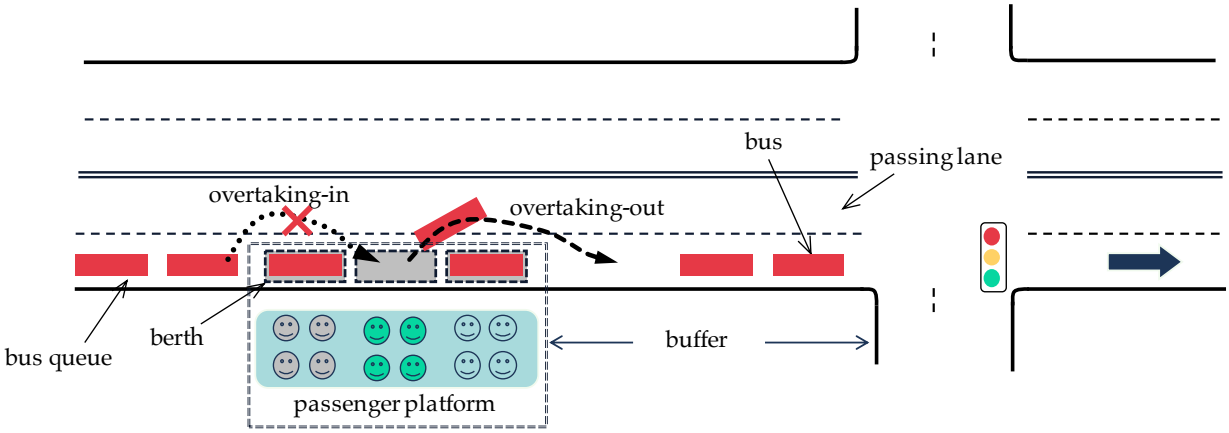


Figure 1: Layout of a near-side bus stop under the LO rule.

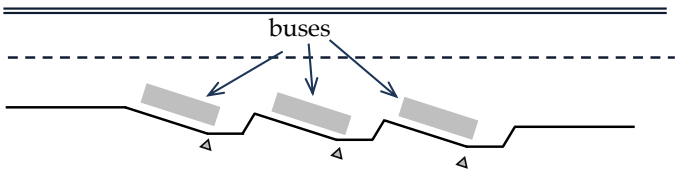


Figure 2: Layout of a sawtooth bus stop.

1.3 The key contribution and overview of the paper

The key contribution of this paper is the development of more computationally efficient methods for berth allocation optimization. This is important due to the following two reasons:

(i) The problem's solution space is huge. Note that the total number of allocation plans for a c-berth stop serving L bus lines is c^L, which is very large when $c \geq 3$ and $L \geq 8$.

(ii) The stochasticity of bus queues calls for employing the simulation-based optimization approach (see Section 1.1). The resulting computational cost would be prohibitively high since each allocation plan must be simulated by thousands of runs to obtain a steady-state performance metric (e.g., the mean bus delay).

The huge solution space, the stochastic bus queues, and the simulation-based optimization approach jointly render it very difficult to find an efficient solution method for the berth allocation problem. In this paper, we will present two novel and computationally efficient heuristic methods.

The first method, termed the simple heuristic, is inspired by queueing theory. It seeks a plan that evenly distributes the total traffic intensity (defined as the total bus arrival rate times the mean dwell time across all bus lines) among all the berths. We find that this simple heuristic solution can often produce near-optimal allocation plans. Notably, this method does not rely on the surrogate model and can generate a good plan in seconds even for stops with four or more berths serving over ten lines! Moreover, the method is parsimonious. It does not rely on the numerous detailed parameters of a bus stop, e.g., the parameters describing bus arrival processes and dwell time distributions for each line. This is nice, since its performance is not affected by the estimation accuracy of those parameters.

The second method is a cluster-based nested partition algorithm designed to further improve the solution quality (especially when the simple heuristic is not satisfactory enough). This search method takes the simple heuristic as the initial solution and employs a bus-stop simulation model developed in-house as the surrogate model. The mean bus delay is selected as the performance metric because it directly reflects the bus service quality and passengers' satisfaction. Compared to other methods (e.g., Tan's algorithm), our method has several advantages. For example, the nested partition structure enables a more efficient search process that can identify high-quality plans by evaluating a relatively small number of candidate allocation plans. Moreover, by clustering the allocation plans using traffic intensity as the indicator, the partitioning of solution space captures the effects of both bus flows and bus dwell times.

Our development of high-efficiency algorithms is necessary due to the following operational aspects:

(1) Large cities like Hong Kong, Santiago, Chicago, and Beijing have thousands of bus stops (Shanmukhappa et al., 2018; Chicago Transit Authority, n.d.; Wikipedia, n.d.). As we shall demonstrate in Section 4, finding a heuristic allocation plan for a simple 2-berth, mid-block stop using previous methods can easily take several hours. Optimizing allocation plans for all stops in a city would require an unacceptably long time.

(2) In practice, bus systems experience frequent changes over time, rendering previously optimized allocation plans suboptimal. For example, when a bus line's timetable or route is modified, the allocation plans for all served stops may need redesigning. Changes in traffic signal timing can affect bus queues at nearby stops and bus arrival

processes at downstream stops. The opening of a new shopping mall or the introduction of a new rail or bus line can alter passenger demands for existing bus lines, and thus their dwell times. Each of these changes may eventually affect the optimal berth allocation plans for several stops. Therefore, having a computationally efficient algorithm for berth allocation optimization is highly beneficial in these situations.

(3) A simulation surrogate model for berth allocation optimization requires many input parameters, some of which (e.g., bus arrival processes and dwell time distributions) are difficult to estimate accurately and may vary over time. To account for this, a stop's allocation plan might need to be optimized multiple times under different parameter settings to obtain a robust plan.

The berth allocation problem formulation and the simulation-based surrogate model are presented in Section 2. The basic idea of the simple heuristic and the cluster-based nested partition algorithm are furnished in Section 3. Section 4 examines the performance of the proposed methods via extensive numerical experiments. Conclusions and future research directions are discussed in Section 5. Notations used in this paper are summarized in Appendix A.

2 Problem Setup

The bus-stop setup and the berth allocation problem formulation are presented in Section 2.1. The simulation-based surrogate model is briefly described in Section 2.2.

2.1 Bus-stop setup and problem formulation

Consider a c-berth curbside stop serving L bus lines. A line-to-berth allocation plan is represented by a binary vector $\boldsymbol{\phi} \equiv (\phi_j^l, l \in \{1, 2, .., L\}, j \in \{1, 2, ..., c\})$, where ϕ_j^l equals 1 if line l is assigned to berth j, and 0 otherwise. Our goal is to find the optimal allocation plan that minimizes the mean bus delay:

$$\boldsymbol{\phi}^* \in \underset{\boldsymbol{\phi} \in \Phi}{\operatorname{argmin}}\, f(\boldsymbol{\phi}), \tag{1}$$

where Φ is the solution space for all possible allocation plans; i.e., $\Phi = \{\boldsymbol{\phi} \mid \phi_j^l \in \{0, 1\}, \forall l \in \{1, 2, \ldots, L\}, j \in \{1, 2, \ldots, c\}; \sum_{j=1}^{c} \phi_j^l = 1, \forall l \in \{1, 2, \ldots, L\}\}$. The size of the solution space is $|\Phi| = c^L$. Function $f : \Phi \mapsto R^+$ returns the steady-state mean bus delay under a specific allocation plan.

The arrival rate of buses on line $l \in \{1, 2, \ldots, L\}$ is denoted by λ_l. The bus headways are assumed to be independent and identically distributed (i.i.d.) random variables following a gamma distribution with mean $\frac{1}{\lambda_l}$ and coefficient of variation C_a^l. A bus's dwell time is the sum of the time for loading and unloading passengers and the time lost to door opening and closing (*not including the delay that occurs when the bus has finished serving passengers, but its*

departure from the stop is blocked by downstream buses or a red signal). The dwell times of buses on line l are also assumed to be i.i.d. gamma-distributed variables with mean $\frac{1}{\mu_l}$ (μ_l is often termed the service rate in the queueing literature) and coefficient of variation C_s^l. Gamma distributions were shown to fit the real-world bus headways and dwell times well (Ge, 2006; Wu et al., 2016), and were often used to model headways and dwell times (Gu et al., 2011; Gu and Cassidy, 2013; Shen et al., 2019). Nevertheless, our numerical experiment results show that the main findings still hold if other distributions of bus headways and dwell times are assumed. Note that bus queueing is typically more pronounced during the morning and evening peak periods, when there is a high demand for passengers and increased bus flows. Conversely, during off-peak times, buses seldom form queues. Therefore, when developing berth allocation plans to minimize bus delays, it is essential to prioritize peak periods and consider arrival processes and dwell time distributions specifically during those peak periods.

For a mid-block stop, we assume that there is always enough space for storing the bus queue formed upstream of the stop. If the stop is a near- or far-side one, more operating parameters need to be specified. These include: (i) the buffer size denoted by d, i.e., the distance between the intersection and the bus stop, normalized as an integer multiple of the bus jam spacing[1] (see Fig. 1); (ii) the signal cycle length, denoted by C; and (iii) the effective green period, denoted by G.

2.2 A simulation-based surrogate model

We develop a discrete-time simulation model using Python to find $f(\phi)$. The simulation model consists of four modules: (1) the entry queue module, (2) the berth module, (3) the passing lane module (for LO and FO stops only), and (4) the signal and buffer module (for near- and far-side stops only). All modules are executed at every time step t.

Using a near-side stop as an example, the basic simulation logic is described as follows:

(1) In the entry queue module, arriving buses queue up at the entry area upstream of the stop if the allocated berth is currently unavailable. The leading bus in the queue checks whether it can proceed to the allocated berth, either by using the passing lane via overtaking-in or directly going through the berth(s).

(2) In the berth module, dwelling buses monitor their remaining service times. Once the dwelling process is completed, a bus checks if it can exit the stop, either by using the passing lane via overtaking-out or directly proceeding through the berth(s).

(3) In the passing lane module, the passing lane used for overtaking is divided into c "cells", each having the same size as the berth and aligned parallel to it. Berths and

[1] If the buffer size is not an integer multiple of the bus jam spacing, it will be rounded down to the nearest integer since only an integer number of buses can be stored in the buffer.

cells are numbered from upstream to downstream as $1, 2, \ldots, c$. Buses performing overtaking-in to enter the allocated berth j advance through the passing lane until reaching cell $j-1$, where they wait to enter berth j. Buses attempting overtaking-out proceed until they exit the passing lane.

(4) In the signal and buffer module, when the signal is red ($t \bmod C > G$), buses departing from the stop form a queue in the buffer. If the number of queued buses reaches d, buses ready to leave the stop must wait. When the signal is green ($t \bmod C \leq G$), buses in the buffer can discharge into the intersection.

For far-side stops, the buffer area is located upstream of the stop, affecting the bus arrivals at the stop. Buses that have traversed the intersection during the green period G fill the buffer if they cannot enter the stop immediately. Buses unable to access a fully occupied buffer must wait upstream of the intersection. The departure process, on the other hand, is unaffected by the signal. All other operations are analogous to those at near-side stops.

The flowchart in Appendix B provides a more comprehensive visualization of the program's logic. Interested readers can also refer to the source code in (https://github.com/Minyu-Shen/bus_berth_allocation) for more details.

To find the steady-state mean bus delay, each simulation run will emulate bus operations for at least 1000 hours. If the average delay per bus does not converge after the 1000-hour period, the simulation will continue to run until convergence. We stipulate that convergence is attained when the mean bus delay's standard deviation is less than 0.1 second. All the simulation runs were performed on a Dell workstation with Intel Xeon Gold 6126 CPU (2.60 GHz×24) and 64 GB DDR4 memory. A simulation run takes 5 minutes on average to complete.

3 Solution approaches

The optimal solution to (1) can be found by exhaustive search if the solution space is small. For example, when $c = 2$ and $L = 6$, we only need to evaluate $c^L = 64$ allocation plans, which takes about 5 hours to complete on our computer. However, for large instances, e.g., when $c = 4$ and $L = 10$ (bus stops of this size are not rare in large cities like Beijing, Chengdu, and Hong Kong), the solution space size soars to $1,048,576$, making the exhaustive search impossible. Thus, efficient solution algorithms are necessary for identifying good allocation plans in short runtimes.

Section 3.1 presents the basic idea upon which our simple heuristic is built. Section 3.2 describes the simple heuristic. Section 3.3 presents the cluster-based nested partition algorithm. For comparison, Tan's algorithm is reproduced and relegated to Appendix C. The code for all the algorithms presented in this paper can be found in (https://github.com/Minyu-Shen/bus_berth_allocation).

3.1 The Basic Idea

The purpose of this section is to show why we choose to use the traffic intensity as a key indicator for allocating bus lines to a stop's berths, and why evenly distributing the traffic intensity among the berths would yield good (although not necessarily optimal) allocation plans. Our belief stems from some facts in queueing theory. Particularly, we show that evenly distributing the traffic intensity among the servers of a queueing system does lead to the minimum mean delay under two special cases.

First of all, note that for bus queues at a given stop, the two most important factors affecting the mean bus delay are the bus arrival rate and the mean bus dwell time[2] (Gu et al., 2011). The joint effect of these two factors can be characterized by the product of them (in other words, the ratio between the bus arrival rate and the service rate). This variable is termed the "*traffic intensity*" or "utilization factor" in the queueing literature (Almeida and Cruz, 2018). It plays a key role in the mean delay models of queueing systems. For example, the well-known Pollaczek-Khintchine mean delay formula (Pollaczek, 1930; Khintchine, 1932) for the M/G/1 queue (a single-server queueing system with Poisson customer arrivals and service times following a general distribution) is as follows:

$$\bar{W} = \frac{\rho}{1-\rho} \cdot \frac{1+C_s^2}{2\mu}, \tag{2}$$

where $\bar{W}$ denotes the mean delay; ρ the traffic intensity; C_s the coefficient of variation of service time; and μ the service rate. As another example, a commonly used mean delay approximation for the G/G/1 queue (a single-server queue with interarrival times and service times both following general distributions), the Kingman's formula (Kingman, 1961), is:

$$\bar{W} \approx \frac{\rho}{1-\rho} \cdot \frac{C_a^2+C_s^2}{2\mu}, \tag{3}$$

where C_a denotes the coefficient of variation of interarrival time.

The critical effect that traffic intensity plays on the mean delay is intuitive: this variable conveniently describes how busy a queueing system is. Specifically, for a single-server system, the traffic intensity represents the long-run proportion of time that the server is occupied by a customer; for a c-server system with parallel servers (that do not have mutual blockage), the traffic intensity divided by c is the average long-run proportion of time that each server is occupied.

With the importance of traffic intensity explained, we introduce the following hypothesis, upon which our simple heuristic berth allocation plan is built:

Hypothesis 1. *The optimal berth allocation plan is one that distributes the total bus traffic intensity*

[2]Other factors, like the coefficients of variation in bus headways and dwell times, have second-order effects on bus delays (Gu et al., 2011; Gu, 2012).

among the berths as evenly as possible.

This hypothesis is also consistent with intuition, since evenly distributing the traffic intensity means making all the berths equally busy. This is intuitively the way to minimize the mean bus delay. Note that as the traffic intensity assigned to a berth increases from 0 to 1, the mean delay of buses served by that berth generally grows at an increasing speed from 0 to infinity. This implies that the mean delay is convex in traffic intensity. Thus, assigning equal traffic intensity to every berth tends to render the lowest mean bus delay.

In what follows, we show that Hypothesis 1 holds for two special cases. Both cases assume that the stop is operating under the FO rule (i.e., a queueing system with parallel servers). For the convenience of analysis, we also assume that the bus flow can be *continuously* assigned to the berths regardless of bus lines. In other words, we show for two special cases that the minimum expected delay is attained when the traffic intensity is evenly distributed among c parallel servers of a queueing system. In the first case, all the servers are assumed to have identical mean service times regardless of the allocation plan. We show for this case that the delay-minimizing allocation plan features an even distribution of customer (bus) arrival rate among the servers, such that the traffic intensities of all servers are equal. In the second case, we assume that the customer arrival rate is always evenly distributed across the servers, but under different allocation plans the mean service times of distinct servers can be different. We show for this case that the optimal allocation plan is achieved when all the servers have an equal mean service time (and also an equal traffic intensity).

To start, we present the following two lemmas extracted from the literature.

Lemma 1. *In a G/G/1 queue, $\lambda\bar{W}(\lambda,\mu)$ is a convex and nondecreasing function of λ, where λ and μ are the arrival and service rates, respectively, and $\bar{W}$ the mean delay.*

For the proof of Lemma 1, please see Proposition 1 of Fridgeirsdottir and Chiu (2005).

Lemma 2. *In a G/G/1 queue, the mean delay $\bar{W}(\lambda,\mu) \equiv \bar{W}\left(\lambda,\frac{1}{\beta}\right)$ is a convex function of the mean service time $\beta = \frac{1}{\mu}$.*

Lemma 2 follows directly from Theorem 1 of Harel (1990).

Built upon the above lemmas, we have the following propositions regarding two special cases of the optimal allocation plan for a queueing system with c parallel servers.

Proposition 1. *For a queueing system with c parallel servers where the customer flow is divided and allocated to each specific server, denote $\lambda^{(j)}$ the customer arrival rate allocated to server j and $\mu^{(j)}$ the service rate of server $j(j \in \{1,2,\ldots,c\})$. If the service rates are equal, i.e., $\mu^{(1)} = \mu^{(2)} = \cdots = \mu^{(c)} \equiv \mu$, then the minimum mean delay is attained when the customer arrival rate is evenly distributed among the servers, i.e., $\lambda^{(1)} = \lambda^{(2)} = \cdots = \lambda^{(c)}$.*

Proof. Such a queueing system with c parallel servers acts like c identical single-server systems (G/G/1 queues) with their queues combined. Let $\bar{W}(\lambda^{(j)},\mu)$ be the mean delay of customers at server j. The steady-state total delay per unit of time is $\sum_{j=1}^{c}\lambda^{(j)}\cdot\bar{W}(\lambda^{(j)},\mu)$.

From Lemma 1, we know that $\lambda\bar{W}(\lambda,\mu)$ is a convex function of λ. Thus, according to the Jensen's inequality,

$$\frac{\sum_{j=1}^{c}\lambda^{(j)}\cdot\bar{W}(\lambda^{(j)},\mu)}{c}\geq\frac{\sum_{j=1}^{c}\lambda^{(j)}}{c}\cdot\bar{W}\left(\frac{\sum_{j=1}^{c}\lambda^{(j)}}{c},\mu\right).$$

Then, we have:

$$\begin{aligned}\sum_{j=1}^{c}\lambda^{(j)}\cdot\bar{W}\left(\lambda^{(j)},\mu\right)&=c\cdot\frac{\sum_{j=1}^{c}\lambda^{(j)}\cdot\bar{W}(\lambda^{(j)},\mu)}{c}\geq c\cdot\left(\frac{\sum_{j=1}^{c}\lambda^{(j)}}{c}\cdot\bar{W}\left(\frac{\sum_{j=1}^{c}\lambda^{(j)}}{c},\mu\right)\right)\\&=\sum_{j=1}^{c}\lambda^{(j)}\cdot\bar{W}\left(\frac{\sum_{j=1}^{c}\lambda^{(j)}}{c},\mu\right).\end{aligned}$$

The above inequality indicates that by evenly distributing the customer flow among the servers (such that each server is allocated a customer flow of $\frac{\sum_{j=1}^{c}\lambda^{(j)}}{c}$), the mean delay would decrease or remain unchanged. □

Proposition 2. *For a queueing system with c parallel servers where the customer flow is divided and allocated to each specific server, if the customer flows allocated to all the servers are equal, i.e., $\lambda^{(1)}=\lambda^{(2)}=\cdots=\lambda^{(c)}\equiv\lambda$, then the minimum mean delay is attained when the allocation ensures that $\mu^{(1)}=\mu^{(2)}=\cdots=\mu^{(c)}$.*

Proof. Let $\bar{W}(\lambda,\mu^{(j)})\equiv\bar{W}\left(\lambda,\frac{1}{\beta^{(j)}}\right)$ be the mean delay of customers at server j, where $\beta^{(j)}=\frac{1}{\mu^{(j)}}$ is the mean service time at server j. The steady-state total delay per unit of time, $\sum_{j=1}^{c}\lambda\cdot\bar{W}\left(\lambda,\frac{1}{\beta^{(j)}}\right)$, satisfies:

$$\sum_{j=1}^{c}\lambda\cdot\bar{W}\left(\lambda,\frac{1}{\beta^{(j)}}\right)=c\lambda\cdot\frac{1}{c}\sum_{j=1}^{c}\bar{W}\left(\lambda,\frac{1}{\beta^{(j)}}\right)\geq c\lambda\cdot\bar{W}\left(\lambda,\frac{1}{\sum_{j=1}^{c}\beta^{(j)}/c}\right)=c\lambda\cdot\bar{W}\left(\lambda,\frac{c}{\sum_{j=1}^{c}\beta^{(j)}}\right).$$

The inequality follows from the Jensen's inequality, which holds true due to the convexity of $\bar{W}$ in mean service time (see Lemma 2). Note that the last term indicates the total delay per unit of time when the customer traffic is distributed in a way that each server has the same mean service rate, $\mu^{(1)}=\mu^{(2)}=\ldots=\mu^{(c)}=\frac{c}{\sum_{j=1}^{c}\beta^{(j)}}$, and the same traffic intensity, $\frac{\sum_{j=1}^{c}\lambda\cdot\beta^{(j)}}{c}$. Hence, by ensuring that all the servers have equal mean service times, the mean delay would decrease or remain unchanged. □

Propositions 1 and 2 reveal that, when the optimal berth allocation plan is achieved, the traffic intensity should be evenly distributed among the berths, or as much so as possible. Note that *this is true irrespective of the exact distributions of bus headways and dwell times.* The limitation here is that the conclusion only applies to the two special cases, i.e., where $\mu^{(j)}$ is a constant across the berths and where $\lambda^{(j)}$ is a constant across the berths. Proving it for the

general case (i.e., where both $\mu^{(j)}$ and $\lambda^{(j)}$ take distinct values across the berths) is difficult, and we do not intend to pursue it in this paper. Nevertheless, it is worth testing whether evenly distributing the traffic intensity can help us quickly find good (if not optimal) berth allocation plans.

Although the above discussion is mainly about FO stops where berths can be treated as parallel servers, we will apply Hypothesis 1 to NO and LO stops as well. Specifically, we will test the idea that a good allocation plan would yield roughly equal traffic loads assigned to each berth, even if the mutual bus blockages under the NO and LO rules are considered. We next present a simple method to find a heuristic plan that distributes the traffic intensity as evenly as possible.

3.2 A simple heuristic

We denote the traffic intensity of line l by $\rho_l = \frac{\lambda_l}{\mu_l}$, and the total traffic intensity by $\rho = \sum_{l=1}^{L} \rho_l$. Define a continuous vector $\boldsymbol{P} = (P_1, ..., P_j, ..., P_c)$ wherein element P_j denotes the traffic intensity assigned to berth $j \in \{1, 2, \ldots, c\}$. We have $0 \leq P_j \leq \rho$ and $\sum_{j=1}^{c} P_j = \rho$. Thus, $\boldsymbol{P}$ lies on a simplex, denoted by Ω. Define $g(\boldsymbol{P})$ the berth allocation plan that matches with $\boldsymbol{P}$ most closely. The $g(\boldsymbol{P})$ can be found by solving the following minimization problem:

$$g(\boldsymbol{P}) = \underset{\boldsymbol{\phi} \in \Phi}{\operatorname{argmin}} \sum_{j=1}^{c} \left(\sum_{l=1}^{L} \phi_j^l \rho_l - P_j \right)^2, \boldsymbol{P} \in \Omega. \tag{4}$$

The above integer quadratic program can be solved efficiently by standard optimization solvers like Gurobi. For example, when $c = 4$ and $L = 15$, solving (4) by Gurobi takes less than 10 seconds. When there are multiple optima, we will choose the one with the minimum mean bus delay (note that this requires calling the surrogate model a few times).

The simple heuristic allocation plan, denoted by $\hat{\boldsymbol{\phi}}$, can be obtained by solving (4) for $\boldsymbol{P} = \left(\frac{\rho}{c}, \ldots, \frac{\rho}{c}\right)$, i.e., $\hat{\boldsymbol{\phi}} = g\left(\frac{\rho}{c}, \ldots, \frac{\rho}{c}\right)$. We will see in Section 4 that this is a good heuristic under a wide range of operating conditions. Moreover, it uses only each bus line's traffic intensity derived from the line-specific arrival rate and mean dwell time, while the detailed distributions of bus headways and dwell times are not needed. Also, it does not need a time-consuming surrogate model for evaluating the candidate plans (unless multiple optima exist). These merits render this heuristic considerably faster than Tan's algorithm or any similar ones that rely on detailed data of bus operations and simulation tools. The heuristic can be readily applied to a variety of stops with different bus overtaking rules and nearby signal settings.

When the simple heuristic is not satisfactory enough, the following search algorithm can be used to find better solutions, which is built upon the simple heuristic.

3.3 A cluster-based nested partition (CNP) algorithm

The algorithm is a modified version of the nested partitions (NP) method proposed by Shi and Ólafsson (2000). The NP method assumes that certain parts of the solution space are more likely to contain the global optima and thus deserve more search efforts. It partitions the solution space into subspaces in a nested manner and concentrates search efforts on promising subspaces, which are identified by considering both the global and local search perspectives. The NP method was shown to converge to a global optimum with probability one in finite time.

The NP method is very suitable for the simulation-based optimization of berth allocation problem, because simulation-based optimization often lacks a structure that can be utilized to identify the optimal solution. Moreover, due to the key role played by the traffic intensity in queueing systems' performance, we believe that some values of the traffic intensity vector $\boldsymbol{P}$ have higher chances to be associated with optimal or near-optimal allocation plans. Thus, instead of partitioning the solution space Φ, we choose to partition the continuous space of $\boldsymbol{P}$, i.e., Ω.[3] Partitioning the continuous space Ω is simpler than partitioning the discrete set Φ, and it allows us to start the search from the simple heuristic described in Section 3.2, which is associated with $\boldsymbol{P} = \left(\frac{\rho}{c}, \ldots, \frac{\rho}{c}\right)$. Using the simple heuristic as the initial solution can further reduce the search effort greatly (as we shall see momentarily). Section 3.3.1 introduces the partitioning of Ω in a nested manner. Section 3.3.2 presents the detailed algorithm.

3.3.1 Partitioning of the solution space

Region Ω is a simplex with c vertices (e.g., $(0, \ldots, 0, \rho, 0, \ldots, 0)$ is a vertex). We first specify Ω as the parent region, indexed by 0. The parent region is divided into c child subregions. Each subregion is a simplex constructed by replacing one of the original c vertices with the centroid of the parent region[4] (i.e., $(\frac{\rho}{c}, \ldots, \frac{\rho}{c})$). Fig. 3 illustrates a case with $c = 3$; i.e., $\boldsymbol{P}$ is a three-dimensional vector and the parent region 0 is a triangle. Using the centroid and any two of the three vertices of the triangle, three triangular subregions are created that partition the original triangle. These three child subregions are numbered 1, 2, and 3, and the partition is termed the depth-1 partition.

Each subregion in the depth-1 partition then becomes a parent region and is further partitioned into c child subregions using the same method. This partition is termed the depth-2 partition, which eventually divides the original simplex into c^2 subregions; see Fig. 3. The partitioning process continues until it reaches a maximum depth of θ. This nested structure

[3] To put it differently, we cluster the berth allocation plans by the traffic intensity vector $\boldsymbol{P}$. This is why our algorithm is termed the cluster-based NP algorithm.

[4] We chose this partitioning method due to the simplicity of implementation. There are multiple partitioning methods and the number of child regions generated from a parent region is not necessarily equal to c. For instance, when $c = 3$, a parent region can be divided into four child regions by connecting the midpoints of all its edges. We tested this alternative partitioning method numerically and found that the results were comparable to those obtained using the method presented in the paper.

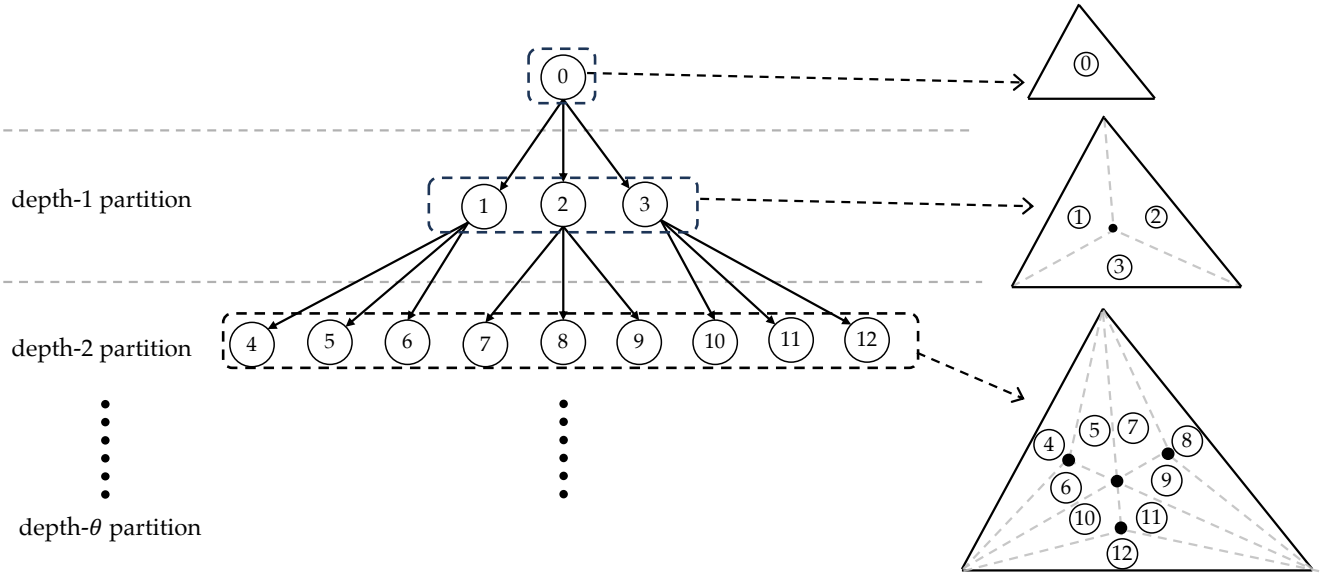


Figure 3: Region partition with $c = 3$.

can be implemented in a tree data structure with θ being the maximum tree depth.

3.3.2 The CNP algorithm

The basic idea of this algorithm is that more searching efforts should be directed to the subregions holding greater chances to contain $\boldsymbol{P}$'s that are associated with optimal or near-optimal allocation plans. To avoid being trapped in a local minimum, we determine whether a subregion is promising or not by comparing the allocation plans randomly sampled from this subregion and its sibling regions on the same depth level.

Following the idea described in Section 3.1, the algorithm starts by specifying that the most promising subregion is the one located at the maximum (θ-th) depth level that contains $\boldsymbol{P} = (\frac{\rho}{c}, \ldots, \frac{\rho}{c})$.[5] At each iteration, it performs the following manipulations:

1. If the most promising subregion, $i \in \{0, 1, 2, \ldots\}$, is at the maximum depth level, uniformly sample N points from this subregion. Find N berth allocation plans by solving (4) for the N sample points. Evaluate the mean bus delay for each plan and calculate the score of subregion i as the mean of the N mean bus delays. Next, merge all the other subregions at the maximum depth into a larger region, uniformly sample N points from this region, and calculate the score of this region. This score is used as a "global benchmark". If the score of subregion i is lower than the benchmark score, keep subregion i as the most promising one. Otherwise, set the parent region of subregion i as the most promising subregion and go to the next iteration.

2. If the most promising subregion i is not located at the maximum depth level, calculate the scores of its c child regions. Also, merge the sibling subregions of subregion i into

[5]Since this point is a vertex shared by multiple subregions, randomly select one of those subregions to start.

a larger region and calculate its score as the global benchmark. If the lowest score of the c child regions is lower than the global benchmark, set the lowest-score child region as the most promising subregion. Otherwise, set the most promising subregion to the parent region of subregion i if the current depth level is greater than 1, and to a randomly selected sibling of subregion i if the current depth level is 1. Then go to the next iteration.

During the iteration process, the algorithm records the mean bus delays of all the allocation plans assessed. The plan with the minimum mean bus delay is returned when the algorithm ends after a maximum number of allocation plans have been assessed. Pseudocode of the algorithm is relegated to Appendix D.

Compared to Tan's algorithm, our CNP algorithm has four advantages.

(i) By using the berth-specific traffic intensities instead of the bus flow ratio as the indicator for partitioning, the CNP algorithm can more effectively cluster the promising allocation plans into one subregion. Recall the limitation of using the bus flow ratio as the indicator for partitioning (see Section 1.1).

(ii) By starting the search from the smallest subregion containing $\boldsymbol{P} = \left(\frac{\rho}{c}, \ldots, \frac{\rho}{c}\right)$ and using the simple heuristic as the initial solution, the CNP algorithm avoids sampling the whole Φ. This significantly reduces the number of plans assessed to find a good solution.

(iii) In the CNP algorithm, the most promising subregion that deserves more searching efforts is dynamically selected by comparing the local search results and a global benchmark. Hence, it is less likely to be trapped in a local minimum. Note that in Tan's algorithm, once the most promising region is identified by evaluating a limited number of sampled plans, all the search efforts will be directed to this region, ignoring the global point of view.

(iv) The CNP algorithm can be easily applied to stops with more than two berths while the current form of Tan's algorithm is applicable to 2-berth stops only.

4 Numerical Analysis

Section 4.1 compares the performance of our simple heuristic and the CNP algorithm with Tan's algorithm for 2- and 3-berth stops. Section 4.2 presents a case study of a real-world 4-berth stop. Sensitivity analyses results to key parameters are presented in Section 4.3. Finally, Section 4.4 shows the added bus delays under the optimal berth-allocating strategy as compared against the berth-sharing strategy.

4.1 Comparisons against Tan's algorithm

4.1.1 $c = 2$

First, consider a 2-berth stop with $L = 12$ bus lines. For this case, the global optimal allocation plan can be developed by enumeration.[6] Thus, we can obtain and compare the optimality gaps of solutions found by the simple heuristic, CNP, and Tan's algorithm. We set the total bus arrival rate to 135 buses/hour and the mean dwell time of the 12 lines combined to 25 seconds. We then use the Dirichlet distribution to randomly generate bus arrival rates and mean dwell times for each line. The coefficients of variation, C_a^l and C_s^l ($\forall l \in \{1, 2, \ldots, L\}$), are set to 0.6 (St. Jacques and Levinson, 1997). Two example sets of line-specific bus arrival rates and mean dwell times are shown in Table 1. For near- and far-side stops, we set $d = 3$, $C = 120$ s, and $G = 60$ s.

For our CNP algorithm, we use $N = 5$ and $\theta = 4$.[7] And for Tan's algorithm, we use the same parameter values as in their paper: $M = 24$ and $r = 0.05$, where M is the number of subsets the solution space is partitioned into, and r each subset's radius of the bus flow ratio (see Appendix C for more details on how these parameters are used in the algorithm). Five randomly selected plans from each subset are assessed and compared to find the most promising subset.

Table 1: Line-specific bus arrival rates and mean dwell times for $c = 2$ and $L = 12$.

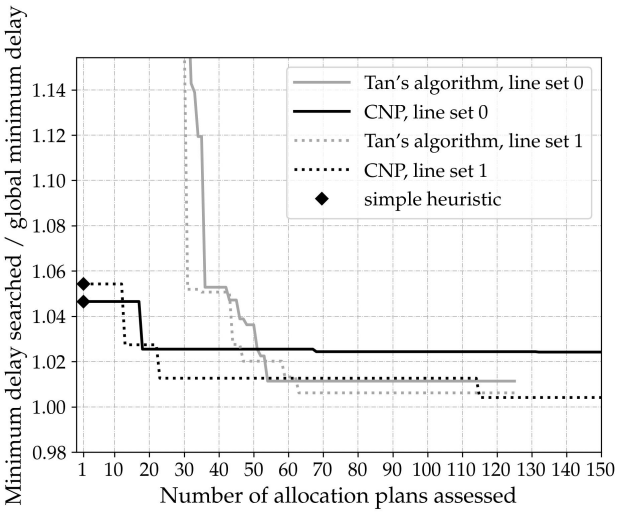
		Line No.											
Set	Parameter	1	2	3	4	5	6	7	8	9	10	11	12
0,1	λ_l (buses/hr)	20.4	16.3	7.1	10.5	18.1	5.8	22.5	6.1	9.0	7.8	4.3	7.1
0	$\frac{1}{\mu_l}$ (seconds)	22.4	25.1	25.2	26.0	27.1	24.0	23.8	25.8	25.2	24.3	26.1	25.0
1	$\frac{1}{\mu_l}$ (seconds)	14.8	25.2	25.5	29.4	34.7	20.6	19.6	28.3	25.6	21.9	29.8	24.6

The results are similar when other sets of bus line data are used. For brevity, we only show the results of the two parameter sets displayed in Table 1. Readers are referred to the online repository (`https://github.com/Minyu-Shen/bus_berth_allocation`) for more results derived from other parameter sets. Specifically, Figs. 4a-i plot the performance of the simple heuristic, the CNP algorithm, and Tan's algorithm at mid-block, near-, and far-side stops under NO, LO, and FO rules, respectively. In each figure, the ratio between the minimum mean bus delay obtained by each method and the global minimum of mean bus delay is plotted against the number of berth allocation plans assessed. This ratio reflects the optimality gap. Bold and light curves are for the proposed CNP and Tan's algorithms, respectively. The solid and dotted curves are drawn under the two sets of line-specific parameter values, respectively. The performance of the simple heuristic is marked by black diamonds.

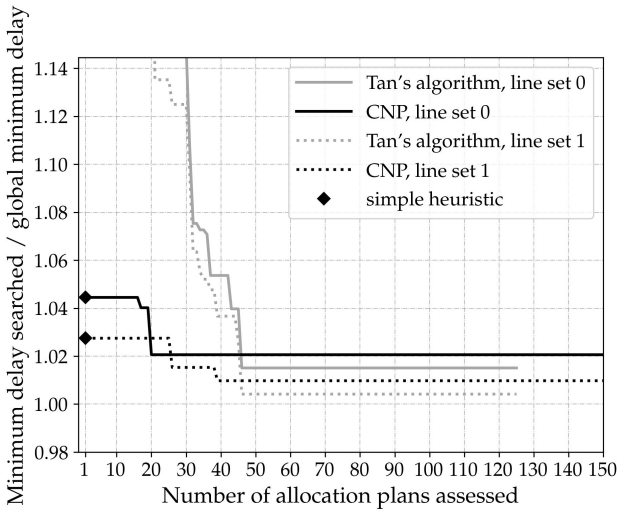
[6] It is nevertheless computationally demanding. We completed this task via parallel computing.

[7] Our parametric tests showed that the general performance of the CNP algorithm was fairly insensitive to these parameter values.

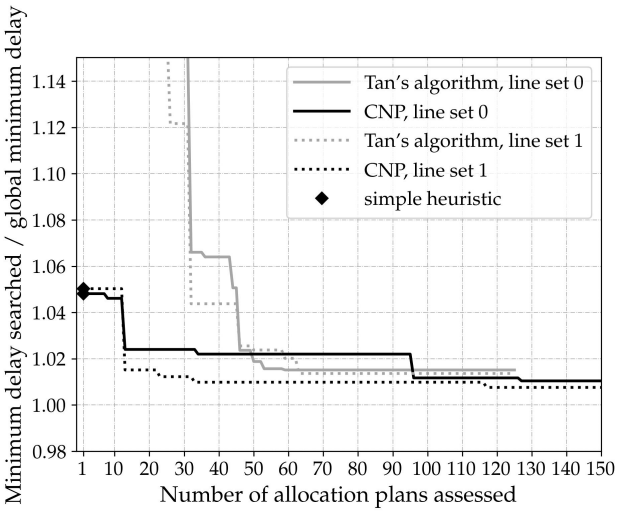
Figs. 4a-c display the results of the three types of stops (mid-block, near-, and far-side) under the NO rule. The figures show that the simple heuristic plan (which can be obtained within 10 seconds) produced a mean bus delay that is only less than 6% higher than the



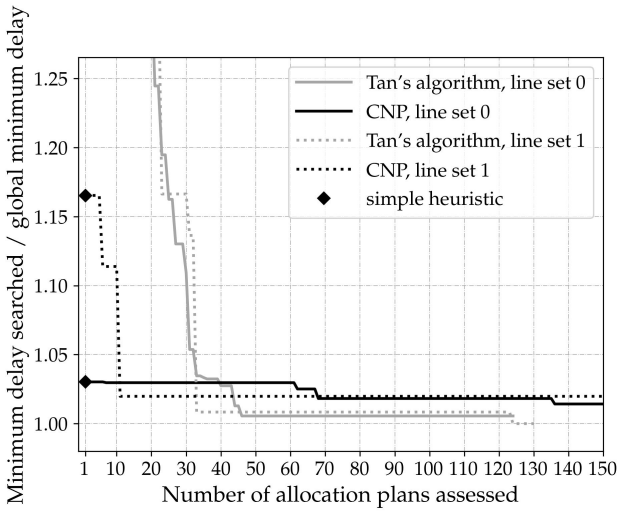
(a) NO rule, mid-block stop



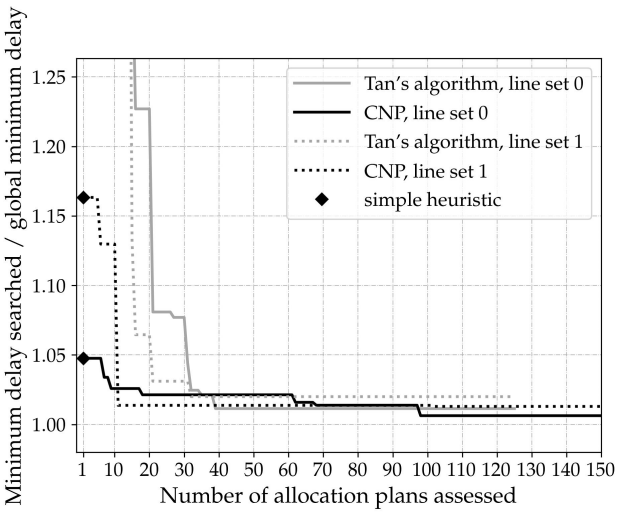
(b) NO rule, near-side stop



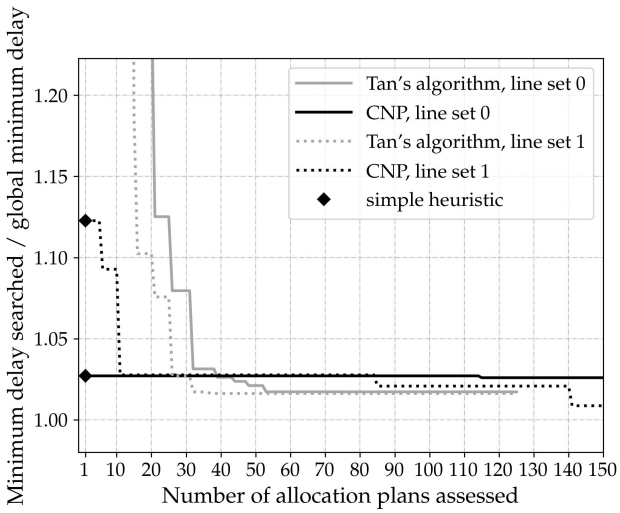
(c) NO rule, far-side stop



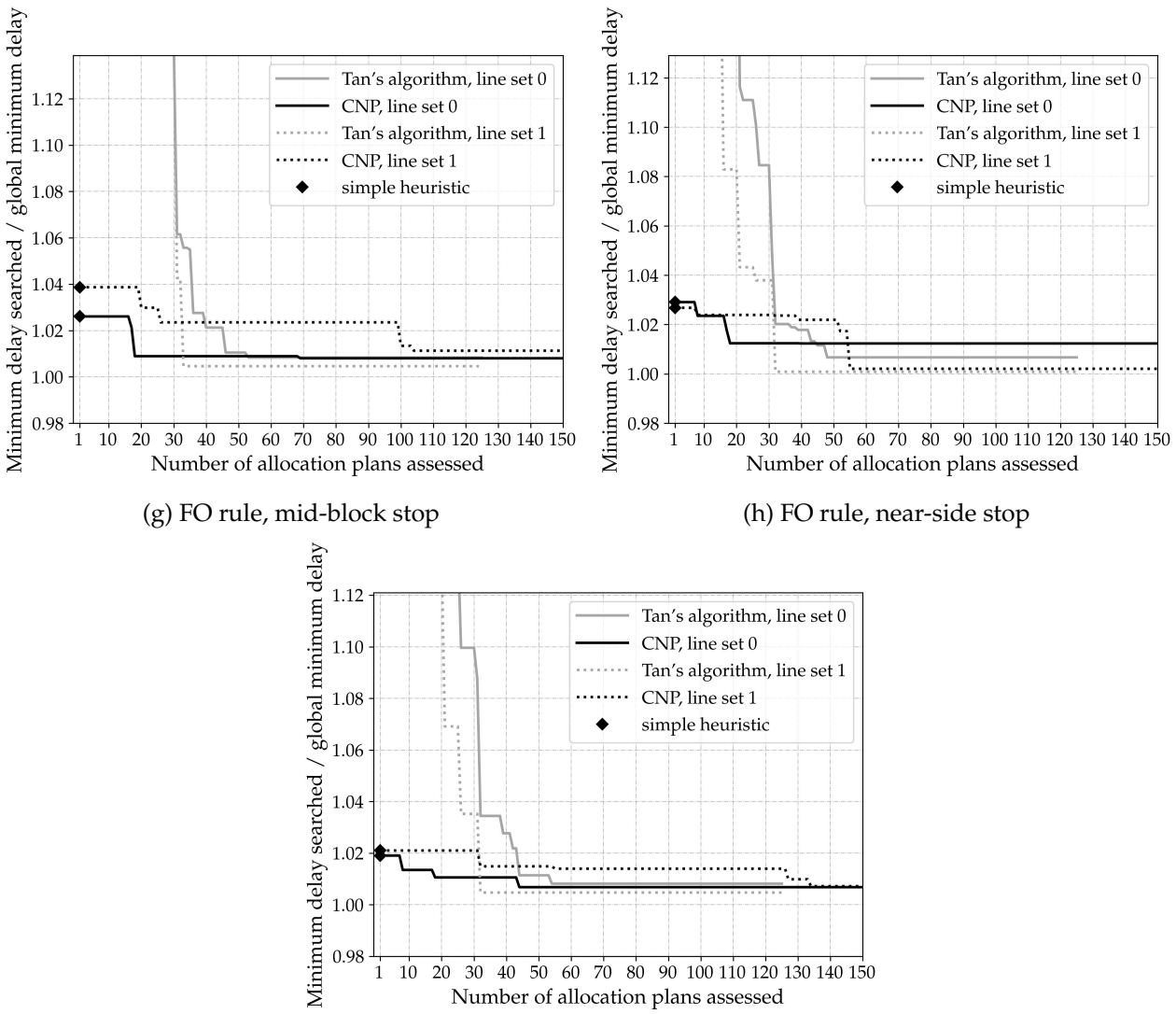
(d) LO rule, mid-block stop



(e) LO rule, near-side stop



(f) LO rule, far-side stop



(g) FO rule, mid-block stop

(h) FO rule, near-side stop

(i) FO rule, far-side stop

Figure 4: Performance assessments of the simple heuristic, the CNP algorithm and Tan's algorithm with $c = 2$ and $L = 12$.

global minimum. Given this, it is unsurprising that further improvement by the CNP algorithm is moderate. In contrast, Tan's algorithm needed to assess 30-40 plans (taking about 3 hours) to attain a similar performance.[8] This finding verifies that *evenly distributing the traffic intensity among berths yields near-optimal allocation plans for NO stops*. In addition, the proximity to signal has a small impact on the performance of these allocation methods.

Figs. 4d-f tell a different story for stops under the LO rule. First, the optimality gap of the simple heuristic became much larger (17% for mid-block and near-side stops) for bus-line set 1. Fortunately, *the CNP algorithm could bring this gap down to below 5% after assessing 10*

[8]Normally, Tan's algorithm needs to sample and evaluate all $M = 24$ subsets before identifying the most promising one for local search. This means at least $24 \times 5 = 120$ plans need to be assessed (taking about 10 hours) before a solution can be produced by the algorithm. Here we assume that the M subsets were evaluated in the ascending order of the bus flow ratio. The best plans recorded during the course of search were extracted to be compared with our methods. Thus, Fig. 4 underestimates the number of plans assessed by Tan's algorithm.

plans, costing less than one hour. On the other hand, Tan's algorithm still took 30-40 plans' assessment to attain a similar optimality gap. The stop's proximity to nearby signals again has no significant effect on these methods' performance.

Finally, it is not surprising to see that *the simple heuristic had the best performance for stops under the FO rule* , as compared to the other two queueing rules. The optimality gaps of the simple heuristic under the FO rule remain below 4% (as illustrated in Figs. 4g-i), while they generally exceed 4% under the NO rule (Figs. 4a-c) and reach up to 16% under the LO rule (Figs. 4d-f.) This is because there is no mutual blockage between buses dwelling in the two berths under the FO rule. Tan's algorithm again needed to assess around 30 plans to find an allocation plan of similar quality.

4.1.2 $c = 3$

We use the bus line parameter sets defined in Section 4.1.1 and increase the total bus arrival rate to 155 buses/hour to keep a 3-berth stop busy. For each line parameter set in Table 1, all the λ_l's are increased proportionally.

The original partitioning method of Tan et al. (2014) based on the ratio between the two berths' assigned bus flows cannot be applied to stops with more than two berths. For comparison, we modified Tan's algorithm by employing a more general partitioning method, which is built upon the idea described in Section 3.3.1. The measure used for partitioning still consists of bus flows assigned to different berths. The modified Tan's algorithm is also presented in Appendix C. The same number of depth levels, θ, is used in both the CNP algorithm and the modified Tan's algorithm.

Fig. 5 plots the performance of these methods against the number of plans assessed. Note that the performance metric (i.e., the vertical axis variable) of Fig. 5 is different from that of Fig. 4. This is because for 3-berth stops, the global optimum cannot be obtained via enumeration due to the very large solution space. Hence, we plot the gap between the minimum mean delays attained by the modified Tan's and CNP algorithms during the search processes. A positive gap means the CNP algorithm produced a lower mean delay. Four mid-block stop scenarios were analyzed with two queueing rules, NO and LO, and the two bus line sets. (Performance under the FO rule is omitted here since it is similar to that under the NO rule.) Note that the left end of a curve in Fig. 5 indicates the advantage of the simple heuristic over the modified Tan's algorithm, since the CNP algorithm starts with the simple heuristic.

The figure shows that the simple heuristic and the CNP algorithm outperformed the modified Tan's algorithm by a large margin (i.e., with a delay saving up to 20 min) when no more than 60 plans were assessed for NO stops. The gap is much smaller (but still significant) for LO stops. For both stop types, the modified Tan's algorithm achieved a similar solution quality as our CNP algorithm only after assessing 70 plans (taking about 6 hours) or more. Comparing this and the results in Section 4.1.1 implies that the computational advantage of our methods becomes greater as the solution space grows.

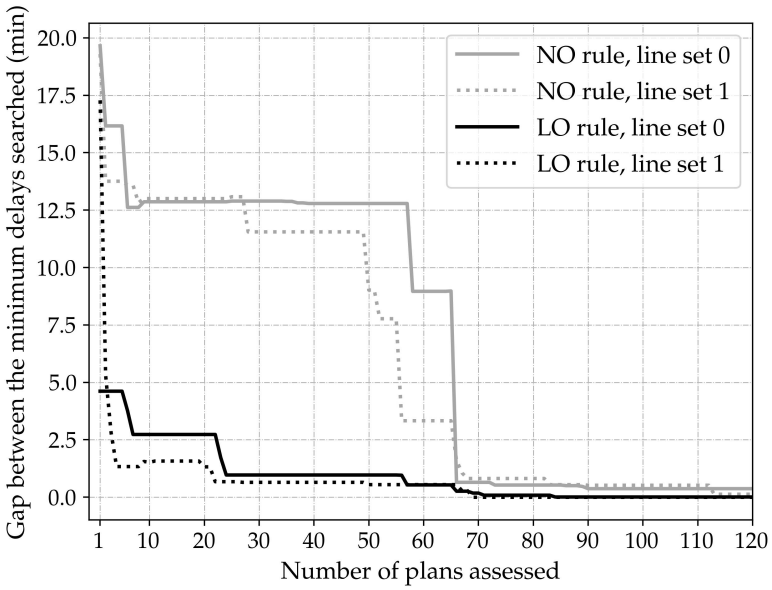


Figure 5: Gap between the minimum mean delays attained by the modified Tan's algorithm and the CNP algorithm for 3-berth mid-block stops.

4.2 A real-world case study

The Cross-Harbour Tunnel (CHT) northbound stop in Hong Kong, is examined in this section. This stop connects to one of the most congested roads in Hong Kong and the world, the Cross-Harbour Tunnel, and is thus among the busiest bus stops in Hong Kong. The stop consists of two substops; see Fig. 6 for the illustration. The downstream one contains two berths and the upstream one contains four. In this case study, we will focus on the upstream substop because the impacts of the downstream substop on the upstream one is marginal.[9] The bus arrival rates and mean dwell times of the 12 bus lines served by the upstream substop are presented in Table 2. They were extracted from the videos taken in the evening peak period on October 25th, 2017. Since our data set size is too small to estimate C_a^l and C_s^l accurately for all the bus lines, we generated those coefficients of variation randomly from a uniform distribution spanning $[0.4, 0.8]$ (St. Jacques and Levinson, 1997). Bus operations in the substop follow the LO rule. A traffic signal is located 60 m (i.e., $d = 5$) downstream of the substop with cycle length $C = 130$ s and green period $G = 60$ s.

Table 2: Line-specific bus arrival rates and mean dwell times for the CHT substop.

Line No.	101	103	106	107	108	109	111	113	115	116	170	182
λ_l (buses/hr)	16.0	2.7	9.3	9.3	4.0	4.0	9.3	2.7	5.3	12.0	4.0	4.0
$\frac{1}{\mu_l}$ (seconds)	38.7	52.0	38.7	67.0	53.7	59.7	25.1	46.0	31.2	53.1	23.3	26.3

Fig. 7 plots the minimum mean bus delay attained by the simple heuristic as the diamond marker, and that by the CNP algorithm against the number of plans assessed as the

[9]This is due to three reasons. First, the two substops are separated by a sufficient distance. Second, there is an adjacent passing lane; see Fig. 6. And last, the bus flow bounded for the downstream substop is small.

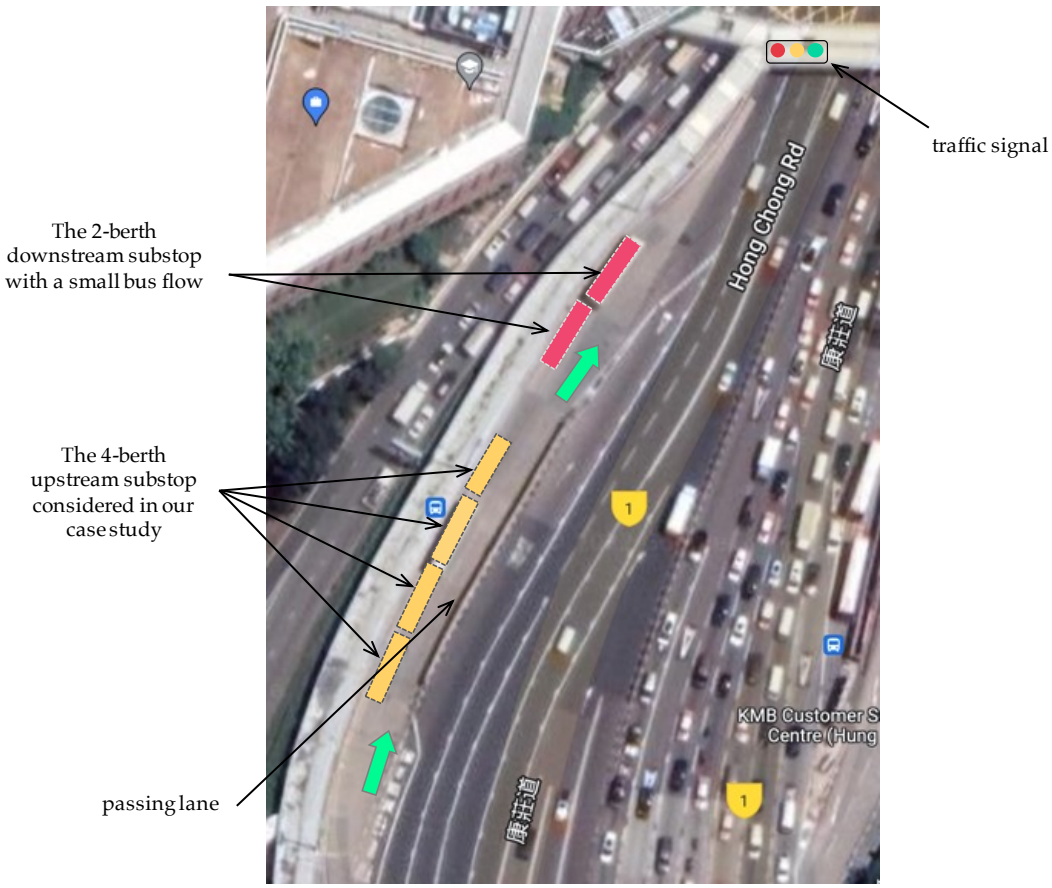


Figure 6: The CHT northbound stop in Hong Kong (background extracted from Google Map).

solid curve. To demonstrate the quality of our heuristic solutions, we randomly sampled 1000 allocation plans from the solution space. (Comparison against Tan's and modified Tan's algorithms is omitted here for simplicity, since the advantage of our methods over those algorithms have been exemplified in Section 4.1.) Their mean bus delays are plotted in ascending order as the dotted curve in Fig. 7 (i.e., the left end of this curve represents the lowest mean bus delay of the 1000 plans, followed by the second lowest, the third lowest, and so on). A little surprisingly, all 1000 allocation plans are worse than the simple heuristic. This is partly due to the extremely large solution space ($4^{12} = 16,777,216$). Still, the comparison manifests how good our simple heuristic is. Owing to the excellent performance of the simple heuristic, the CNP algorithm could only find a slightly better solution after assessing 160 plans during the searching process.

4.3 Sensitivity analyses

We verified that the performance of our simple heuristic and CNP algorithm is robust when the bus dwell time distribution, the total traffic intensity, and the headway distribution change. In the interest of brevity, this section only presents the sensitivity analyses results for varying dwell time distributions.

Figs. 8a and b show the optimality gap of our methods against the number of plans assessed for 2-berth, mid-block stops under the NO and LO rules, respectively. (FO stops

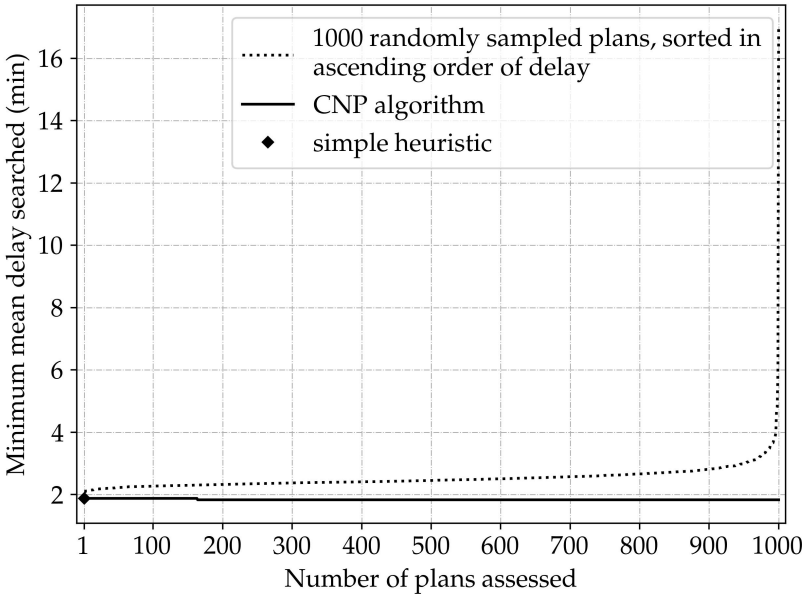
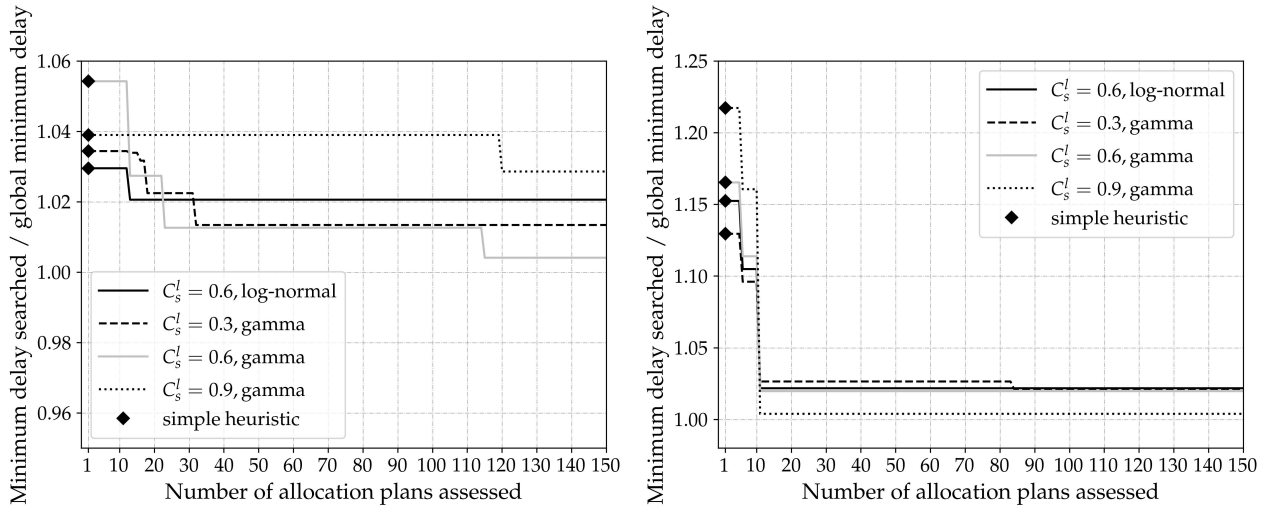


Figure 7: Performance of the simple heuristic and the CNP algorithm at the CHT stop.

are omitted for simplicity since the results are similar to those of NO stops.) Log-normal and gamma distributions of dwell times with various coefficients of variation were tested. (More parameter values were tested than those shown in the figures, and those tests yielded similar results.) The bus line set 1 in Table 1 was used. Other parameter values are the same as in Section 4.1.1. Results show that the simple heuristic performs consistently well for NO stops. For LO stops, although the simple heuristic's performance varies moderately, the CNP algorithm can always bring the optimality gap down to 3% or less within 11 plans searched. Generally speaking, the performance of our methods is robust for various dwell time distributions.



(a) NO rule

(b) LO rule

Figure 8: Sensitivity of the simple heuristic and the CNP algorithm to the dwell time distribution for 2-berth, mid-block stops.

4.4 Comparisons against the berth-sharing strategy

One may also be interested in to what degree the berth-allocating strategy would increase the bus delay compared to the berth-sharing strategy. Fig. 9 plots the mean bus delays under the berth-sharing strategy and the optimal berth allocation plan for 2-berth stops with three overtaking rules and three types of proximities to nearby signals. The data of line set 1 in Table 1 were used. Other parameter values are the same as in Section 4.1.1. Percentages of delay increase when converting stops from berth-sharing to berth-allocating are marked for a clear comparison.

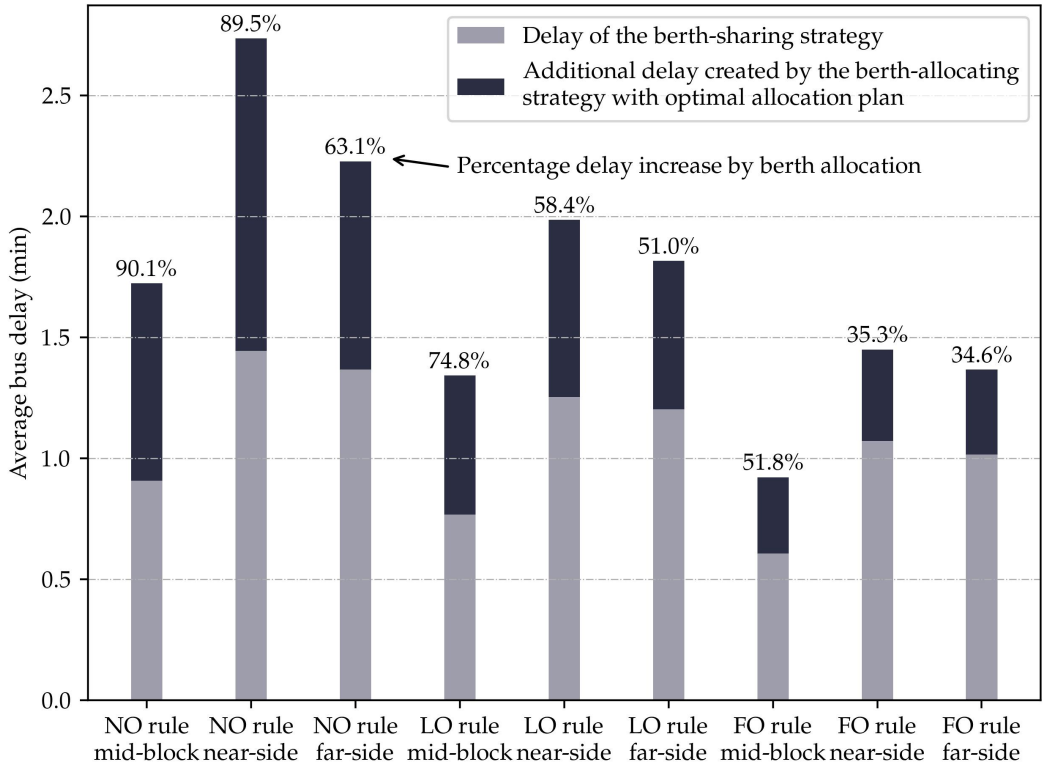


Figure 9: Comparison between the optimal berth allocation plan and the berth-sharing strategy for 2-berth stops.

The figure shows that NO stops exhibit the greatest *percentages* of delay increase by allocating berths to bus lines. This is because under the NO rule a dwelling bus will block all downstream berths. (Consider an extreme example where a bus entering an empty stop is assigned to the upstream-most berth, and no following bus can enter any vacant berth downstream before that bus completes its service.) Hence, the berth-allocating strategy incurs a great waste of available berths at NO stops. On the other hand, FO stops have the lowest percentages of delay increase. This is because a bus can freely enter and leave the assigned berth without being blocked by other dwelling buses. These findings imply that berth allocation is less damaging for stops allowing overtaking maneuvers.

We further find that berth allocation is less damaging for near- and far-side stops. This can also be explained. A near- or far-side stop has a lower capacity than its mid-block counterpart because the stop's capacity is wasted during red periods after the buffer is filled (for

near-side stops) or emptied (for far-side stops); see Shen et al. (2019) for a detailed explanation of this issue. Under the berth-allocating strategy, a stop's bus discharge flow is lower, which means the buffer would take longer to be filled (for near-side stops) or emptied (for far-side stops) in a red period. In other words, the red periods are better utilized under the berth-allocating strategy. Thus, the percentage delay increase is smaller compared to the mid-block case.

5 Conclusions

The major contributions of this paper is the development of two heuristic methods of the berth allocation optimization problem: the simple heuristic and the CNP algorithm. The former finds a near-optimal allocation plan by distributing the total traffic intensity among a stop's berths as evenly as possible. The latter builds upon the simple heuristic to further improve the solution quality via searching in the solution space partitioned in a nested manner. Both methods can be applied to stops under various queueing rules, with two or more berths, and different proximities to traffic signals.

For NO and FO stops, the simple heuristic method can generate a berth allocation plan that is less than 6% worse than a global optimum (in terms of the mean bus delay) within a few seconds. In contrast, a previous method took several hours to obtain a plan of similar quality. The simple heuristic is easy to implement in reality due to its parsimonious nature and negligible computational cost. Recall that it only requires the line-specific bus arrival rates, which can be acquired from the local transit agency, and mean dwell times, which can be computed using passenger demand data. Moreover, it does not rely on a computationally expensive surrogate model.

When the simple heuristic is not good enough (especially for LO stops), the CNP algorithm can bring down the optimality gap to within 3% in less than one hour (see again the bold curves in Figs. 4d-f). The good computational efficiency is mainly due to three reasons. First, the algorithm uses berth-specific traffic intensities to cluster the allocation plans, effectively grouping the near-optimal plans into one or a few subspaces of the solution space. Second, the algorithm balances the tradeoff between exploration and exploitation in the searching process by dynamically selecting the most promising subregion. And lastly, the simple heuristic serves as a very good initial solution. When the real-world input data (e.g., C_a^l and C_s^l) has estimation errors, one can execute the algorithm repeatedly with varying parameter values to obtain a robust allocation plan.

Admittedly, our work has overlooked or simplified some features of real-world bus stop operations. They are briefly discussed as follows:

(i) We did not model the effect of passenger queues on bus dwell times. A notable consequence of this effect is the positive correlation between bus headways and dwell times (Newell and Potts, 1964; Daganzo, 2009). Typically, a bus's dwell time on a given line is modeled as its headway multiplied by a line-specific constant, which is the prod-

uct of passenger arrival rate and unit boarding time per passenger (Daganzo, 2009; Xuan et al., 2011). Consequently, the bus line's traffic intensity (the product of bus flow and mean dwell time) is essentially equal to this line-specific constant. Therefore, our simple heuristic of evenly distributing traffic intensity among berths is equivalent to assigning equal passenger loads to each berth, which intuitively minimizes queues and delays. We modified the surrogate model to account for this linear correlation and found similar results, indicating the effectiveness of our heuristic methods. Detailed results are omitted for brevity but are available in the online repository (`https://github.com/Minyu-Shen/bus_berth_allocation`) for interested readers.

(ii) Furthermore, passenger queues may be influenced by random passenger arrivals, bus capacity, and the presence of passengers who can choose buses on multiple lines, referred to as "common-line passengers" (Schmöcker et al., 2016). Our model does not account for these operational factors. In particular, considering common-line passengers would result in intercorrelation between bus dwell times on different lines sharing these passengers. These factors can be incorporated by modifying the surrogate model. Alternatively, bus lines with a significant number of common-line passengers can be allocated to the same berth, a common practice to facilitate their boarding. A new component can be added to the objective function in Equation (4) to represent how well those common lines are grouped.

(iii) Our study does not account for interactions between buses and general traffic. In practice, buses discharging from near-side stops may compete with right-turning car traffic for buffer space. To address this, we can calibrate the arrival process of right-turning vehicles using field data and incorporate it into our surrogate model. This resembles having a smaller buffer space when excluding right-turning traffic, implying our methods may still be effective. Another real-world scenario involves bus bay stops where exiting buses must wait for a sufficient gap in general traffic to merge back into travel lanes. This issue can be addressed by integrating a gap-acceptance model into the surrogate model.

Real-world bus stops exhibit various operating features, such as segregation from general traffic, curbside or central-island locations, bus bays, and mixed fleets of regular and articulated buses (Kittelson & Associates, Inc., 2013; Hu et al., 2023). Most of these features can be accommodated by adjusting the surrogate model and calibrating it with real data. Nevertheless, comprehensive testing for all bus stop types is beyond this paper's scope and will be explored in future work.

A more interesting research direction is to examine the dynamic berth allocation problem where an arriving bus can be assigned to a specific berth several minutes ahead of its arrival. The passengers waiting at the stop will then be notified and have enough time to walk to the appropriate waiting place. Dynamic berth allocation requires more accurate estimates of bus arrival and dwell times, which can be obtained using existing probabilistic and machine

learning methods (e.g., Yu et al., 2011; Bian et al., 2015; Achar et al., 2022). We expect that this dynamic allocation strategy can significantly improve a stop's bus-carrying capacity and reduce bus delays. The work in this regard is being pursued.

Acknowledgements

The research was supported by a General Research Fund (Project No. 15224317) provided by the Research Grants Council of Hong Kong, the National Natural Science Foundation of China (Project No.'s 72201214 and 72025104), the Sichuan Science and Technology Program (Project No. 2023NSFSC1035) and the Fundamental Research Funds for the Central Universities under Grant JBK2103009. The authors thank Mr. Chi Kit Yeung, a graduate of the Department of Electrical Engineering, the Hong Kong Polytechnic University, for his help in conducting data collections at the CHT northbound bus stop.

Appendix A Table of Notations

Table A.1: List of notations

Notation	Description
c	Number of berths
C	Cycle length
C_a^l	Coefficient of variation in bus headways of line l
C_s^l	Coefficient of variation in bus dwell times of line l
d	Number of buffer spaces
f	$\Phi \mapsto R^+$, a funtion that returns the steady-state mean bus delay given an allocation plan
$g(\boldsymbol{P})$	The berth allocation plan that matches with $\boldsymbol{P}$ most closely.
G	Effective green period
L	Number of bus lines
λ_l	Bus arrival rate of line l
M	Number of subsets in Tan's algorithm
$\frac{1}{\mu_l}$	Mean of bus dwell time of line l
N	Number of sampling points for evaluating a subregion in the CNP algorithm
Ω	A simplex $\{\boldsymbol{P} \mid 0 \leq P_j \leq \rho, \sum_{j=1}^{c} P_j = \rho\}$
$\boldsymbol{P}$	A continuous decision vector of $(P_1, ..., P_j, ..., P_c)$
P_j	Traffic intensity assigned to berth $j \in \{1, 2, \ldots, c\}$
ϕ_j^l	A binary decision variable indicating whether line l is allocated to berth j
$\boldsymbol{\phi}$	A binary vector whose entries are ϕ_j^l's ($l \in \{1, 2, \ldots, L\}$, $j \in \{1, 2, \ldots, c\}$), representing one line-to-berth allocation plan in Φ
$\hat{\boldsymbol{\phi}}$	The simple heuristic allocation plan
$\boldsymbol{\phi}^*$	Optimal allocation plan that minimizes the mean bus delay
Φ	The solution space that includes all the line-to-berth allocation plans
r	Radius of a subset in Tan's algorithm
ρ_l	Traffic intensity of line l
ρ	Total traffic intensity, $\rho = \sum_{l=1}^{L} \rho_l$
θ	Maximum depth of a partition of Ω

Appendix B Simulation flowchart

Fig. B.1 presents the flowchart of the logic performed at each time step, which checks if buses can move towards the next position (either in the passing lane, the berths, or the buffer). Whenever a bus starts moving, it will advance until reaching the next stopping position.

Bus motions are consistent with the simplified kinematic wave theory (Newell, 1993) given the buses' move-up speed and backward wave speed.

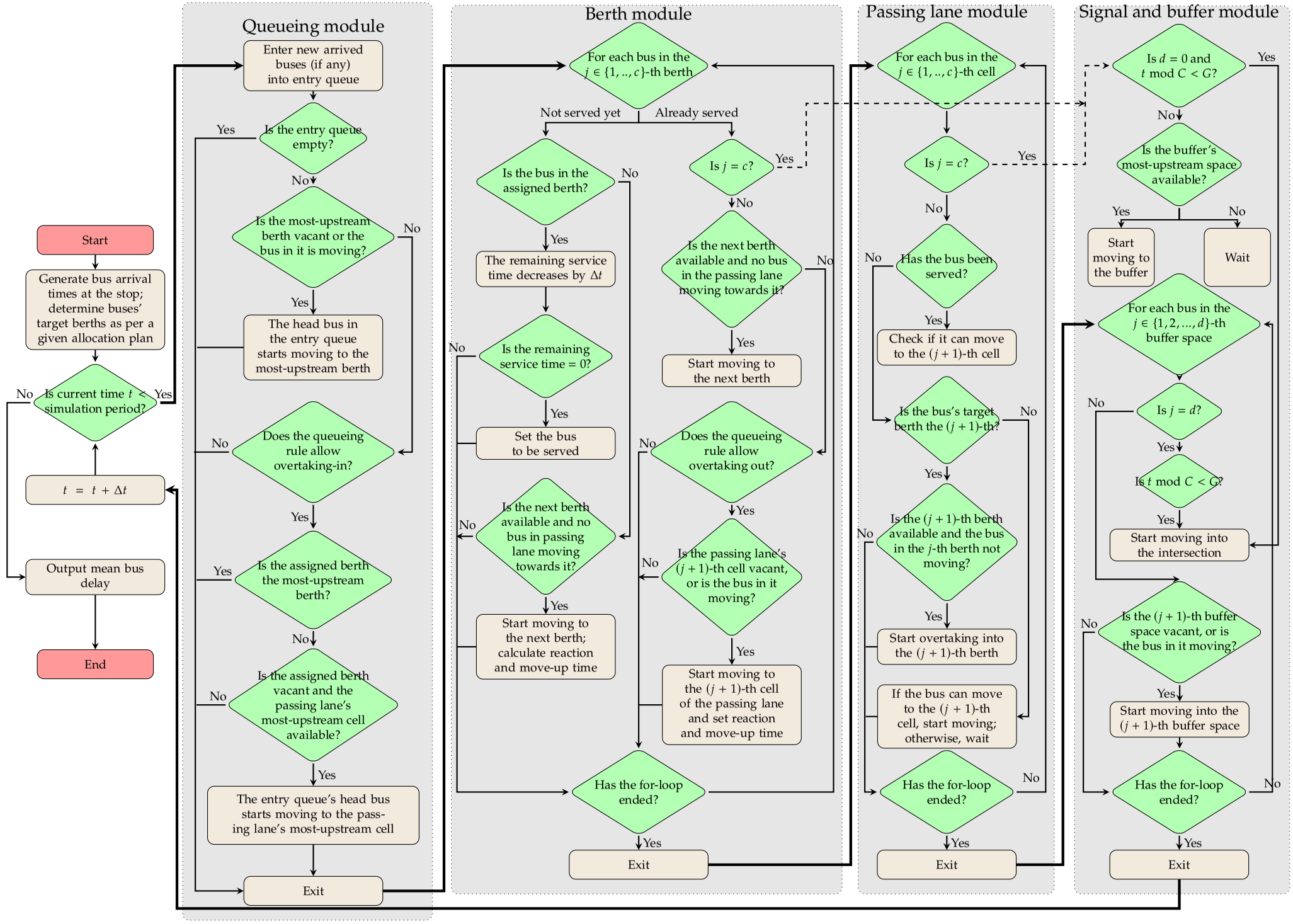


Figure B.1: Flowchart of the simulation program for near-side stops.

Appendix C Pseudocodes of Tan's algorithm and modified Tan's algorithm

The original algorithm proposed in Tan et al. (2014) is only applicable to 2-berth stops. It first partitions the whole solution set into M subsets with a radius of r based on the ratio between bus flows assigned to the two berths. For each subset, it samples five allocation plans and evaluates their performance via simulation. The average performance (e.g., the mean bus delay) is taken across the five plans to represent the subset. The subset with the best average performance is selected as the most promising subset. All the searching efforts thereafter are spent in the most promising subset. Pseudocode of the algorithm is presented as follows.

Algorithm 1: Tan's algorithm for two-berth stops

Divide the total solution space Φ into M subsets by the following loop:
  for *each possible plan* $\phi \in \Phi$ **do**
    Calculate the berth arrival ratio $R(\phi) = \frac{\sum_{l=1}^{a} \lambda_l}{\sum_{l=a+1}^{L} \lambda_l}$, where $1, .., a$ are the bus lines allocated to the downstream berth, and $a+1, ..., L$ are the lines allocated to the upstream berth.
    Assign ϕ to the subset $\Phi(\beta) = \{\phi : |R(\phi) - \beta| < r\}$, where β is the center and r is the radius of the subset. $\beta \in \{r, 3r, 5r, ..., (2M-1)r\}$. The few plans whose $R(\phi)$ are above $2Mr$ are merged into the last subset.
for *all M subsets* **do**
  Randomly sample five plans and evaluate their performance by simulation. Calculate the average performance of the five plans to represent the subset.
Select the subset with the best performance as the most promising subset. Denote ϕ^* as the best plan searched in Steps 5-6.
while *true* **do**
  Sample five plans from the most promising subset and choose the best plan, denoted by ϕ'.
  if ϕ' *is better than* ϕ^* **then**
    $\phi^* = \phi'$.
  else
    break
return ϕ^*

We also modified Tan's algorithm to make it suitable for stops with more than 2 berths. Pseudocode of the modified Tan's algorithm is presented in Algorithm 2.

Algorithm 2: Modified Tan's algorithm for stops with $c > 2$ berths

Define a continuous vector $\boldsymbol{Q} = \{Q_1, Q_2, ..., Q_c\}$ where $Q_j (j = 1, 2, \ldots, c)$ is the total bus flow allocated to berth j. The space of all $\boldsymbol{Q}$ is a simplex with c vertices (e.g., $(0, \ldots, 0, \sum_{l \in L} \lambda_l, 0, \ldots, 0)$ is a vertex). Partition the simplex in a nested manner using the method in Section 3.3.1. In the obtained nested structure, denote Ψ as the set containing all the subregions located at the maximum (θ-th) depth level.

for *a subregion* $i \in \Psi$ **do**

  Randomly sample five points from the subregion i and find the closest berth allocation plans by solving (4). Evaluate the five allocation plans by simulation and calculate the average bus delay to represent the subregion.

Select the subregion with the best average bus delay as the most promising subregion. Denote $\boldsymbol{\phi}^*$ as the best plan searched in Steps 2-3.

while *true* **do**

  Randomly sample five points from the most promising subregion and find the closest berth allocation plans by solving (4). Evaluate the five allocation plans and choose the best plan, denoted by $\boldsymbol{\phi}'$.

  if $\boldsymbol{\phi}'$ *is better than* $\boldsymbol{\phi}^*$ **then**

    $\boldsymbol{\phi}^* = \boldsymbol{\phi}'$.

  else

    break

return $\boldsymbol{\phi}^*$

Appendix D Pseudocode of the CNP algorithm

The following notations are used in the algorithm:

$d(\pi)$ – Depth of a subregion π;

$E(k)$ – Set of subregions that need to be investigated at iteration k;

$f(\boldsymbol{\phi}_k^*)$ – Minimum average bus delay found at iteration k for allocation plan and $\boldsymbol{\phi}_k^*$;

$Q(\pi)$ – Score of a subregion π;

$\pi_k^\star$ – The most promising subregion at iteration k;

Π^d – Set of all the subregions given a depth-d partition, $\{\pi_1^d, \pi_2^d, \ldots, \pi_{|\Pi^d|}^d\}$, where the cardinality $|\Pi^d| = c^d$;

$s(\pi)$ – Parent region of subregion π.

Algorithm 3: Cluster-based nested partition algorithm

Set $\pi^{\star}_{k=1}$ as the subregion located at maximum depth θ that contains $\hat{\boldsymbol{\phi}}$.
$f(\boldsymbol{\phi}^{*}_{k}) \leftarrow f(\hat{\boldsymbol{\phi}})$.
Set the maximum number of allocation plans to be assessed, B.
for *iteration* k **do**
  if $d(\pi^{\star}_{k}) \neq \theta$ **then**
    Obtain $\pi^{\star}_{k}$'s c child subregions: $\pi^{\star}_{k}(1), \pi^{\star}_{k}(2), ..., \pi^{\star}_{k}(r), ...\pi^{\star}_{k}(c)$ and add them into $E(k)$.
  else
    Add $\pi^{\star}_{k}$ into $E(k)$.
  Merge all the subregions in the same depth, $\Pi^{d(\pi^{\star}_{k})} \backslash \pi^{\star}_{k}$, into a "big" subregion and add it into $E(k)$.
  for *each subregion* $e \in E(k)$ **do**
    Uniformly sample N points, $P^{e,1}, P^{e,2}, \ldots, P^{e,n}, \ldots, P^{e,N}$.
    for *each sampled point* $P^{e,n}$ **do**
      Find the closest berth allocation plan $\boldsymbol{\phi}^{g,n}$ via solving (4), and get the average bus delay via simulation $f(\boldsymbol{\phi}^{g,n})$;
      $f(\boldsymbol{\phi}^{*}_{k}) \leftarrow \min(f(\boldsymbol{\phi}^{g,n}), f(\boldsymbol{\phi}^{*}_{k}))$.
      $B \leftarrow B - 1$.
      if $B == 0$ **then**
        stop; return the optimal plan $\boldsymbol{\phi}^{*}_{k}$.
    Estimate the score of each subregion e via:
      $Q(e) = \sum_{n=1}^{N} f(\boldsymbol{\phi}^{e,n})/N$.
  $f(\boldsymbol{\phi}^{*}_{k+1}) \leftarrow f(\boldsymbol{\phi}^{*}_{k})$
  if $\operatorname{argmin}_{e \in E(k)} Q(e)$ *is the merged subregion* **then**
    $\pi^{\star}_{k+1} = s(\pi^{\star}_{k})$.
  else
    $\pi^{\star}_{k+1} = \operatorname{argmin}_{e \in E(k)} Q(e)$.

References

Abdulhai, B., Shalaby, A., Lee, J., Georgi, A., 2002. Microsimulation modelling and impact assessment of street-car transit priority options: The toronto experience. In: Transportation Research Board 81st Annual Meeting, Washington, D.C., USA.

Achar, A., Natarajan, A., Regikumar, R., Kumar, B. A., 2022. Predicting public transit arrival: A nonlinear approach. Transportation Research Part C: Emerging Technologies 144, 103875.

Almeida, M. A., Cruz, F. R., 2018. A note on bayesian estimation of traffic intensity in single-server markovian queues. Communications in Statistics-Simulation and Computation 47 (9), 2577–2586.

Alonso, B., Moura, J. L., Ibeas, A., Ruisánchez, F. J., 2011. Public transport line assignment model to dual-berth bus stops. Journal of Transportation Engineering 137 (12), 953–961.

Bian, B., Pinedo, M., Zhu, N., Ma, S., 2019. Performance analysis of overtaking maneuvers at bus stops with tandem berths. Transportation Science 53 (2), 597–618.

Bian, B., Zhu, N., Ling, S., Ma, S., 2015. Bus service time estimation model for a curbside bus stop. Transportation Research Part C: Emerging Technologies 57, 103–121.

Bian, B., Zhu, N., Meng, Q., 2023. Real-time cruising speed design approach for multiline bus systems. Transportation Research Part B: Methodological 170, 1–24.

Bian, B., Zhu, N., Pinedo, M., Ma, S., Yu, Q., 2020. An optimization-based speed-control method for high frequency buses serving curbside stops. Transportation Research Part C: Emerging Technologies 121, 102860.

Bunker, J. M., 2018. High volume bus stop upstream average waiting time for working capacity and quality of service. Public Transport 10 (2), 311–333.

Cheng, Q., Wang, S., Liu, Z., Yuan, Y., 2019. Surrogate-based simulation optimization approach for day-to-day dynamics model calibration with real data. Transportation Research Part C: Emerging Technologies 105, 422–438.

Chicago Transit Authority, n.d. Cta facts at a glance. [Online]. Available at: `https://www.transitchicago.com/facts`. (Accessed on June 8, 2023).

Cortés, C. E., Pagès, L., Jayakrishnan, R., 2005. Microsimulation of flexible transit system designs in realistic urban networks. Transportation Research Record 1923 (1), 153–163.

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

Delgado, F., Munoz, J. C., Giesen, R., 2012. How much can holding and/or limiting boarding improve transit performance? Transportation Research Part B: Methodological 46 (9), 1202–1217.

Estrada, M., Mensión, J., Aymamí, J. M., Torres, L., 2016. Bus control strategies in corridors with signalized intersections. Transportation Research Part C: Emerging Technologies 71, 500–520.

Feng, X., Hu, S., Gu, W., Jin, X., Lu, Y., 2020. A simulation-based approach for assessing seaside infrastructure improvement measures for large marine crude oil terminals. Transportation Research Part E: Logistics and Transportation Review 142, 102051.

Fernández, R., 2010. Modelling public transport stops by microscopic simulation. Transportation Research Part C: Emerging Technologies 18 (6), 856–868.

Fernandez, R., Planzer, R., 2002. On the capacity of bus transit systems. Transport Reviews 22 (3), 267–293.

Fridgeirsdottir, K., Chiu, S., 2005. A note on convexity of the expected delay cost in single-server queues. Operations research 53 (3), 568–570.

Ge, H., 2006. Traffic impacts of bus stops in urban area and related optimization techniques (in chinese). Ph.D. thesis, Southeast University, China.

Gibson, J., Baeza, I., Willumsen, L., 1989. Bus-stops, congestion and congested bus-stops. Traffic Engineering and Control 30 (6), 291–302.

Gu, W., 2012. Models of bus queueing at isolated bus stops. Ph.D. thesis, University Of California, Berkeley.

Gu, W., Cassidy, M. J., 2013. Maximizing bus discharge flows from multi-berth stops by regulating exit maneuvers. Transportation Research Part B: Methodological 56, 254–264.

Gu, W., Cassidy, M. J., Li, Y., 2012. On the capacity of highway checkpoints: Models for unconventional configurations. Transportation Research Part B: Methodological 46 (10), 1308–1321.

Gu, W., Cassidy, M. J., Li, Y., 2015. Models of bus queueing at curbside stops. Transportation Science 49 (2), 204–212.

Gu, W., Li, Y., Cassidy, M. J., Griswold, J. B., 2011. On the capacity of isolated, curbside bus stops. Transportation Research Part B: Methodological 45 (4), 714–723.

Harel, A., 1990. Convexity results for single-server queues and for multiserver queues with constant service times. Journal of applied probability 27 (2), 465–468.

He, Q., Chao, X., 2014. A tollbooth tandem queue with heterogeneous servers. European Journal of Operational Research 236 (1), 177–189.

He, S., Dong, J., Liang, S., Yuan, P., 2019. An approach to improve the operational stability of a bus line by adjusting bus speeds on the dedicated bus lanes. Transportation Research Part C: Emerging Technologies 107, 54–69.

Hickman, M. D., 2001. An analytic stochastic model for the transit vehicle holding problem. Transportation Science 35 (3), 215–237.

Hu, S., Shen, M., Gu, W., 2023. Impacts of bus overtaking policies on the capacity of bus stops. Transportation Research Part A: Policy and Practice 173, 103702.

Ibarra-Rojas, O. J., Delgado, F., Giesen, R., Muñoz, J. C., 2015. Planning, operation, and control of bus transport systems: A literature review. Transportation Research Part B: Methodological 77, 38–75.

Khintchine, A. Y., 1932. Mathematical theory of stationary queues. Matematicheskii Sbornik 39 (4), 73–84.

Kingman, J. F. C., 1961. The single server queue in heavy traffic. In: Mathematical Proceedings of the Cambridge Philosophical Society. Vol. 57. Cambridge University Press, pp. 902–904.

Kittelson & Associates, Inc., 2013. Transit Capacity and Quality of Service Manual, 3rd Edition. Transit Cooperative Research Program Report 165, Transportation Research Board, Washington, D.C., USA.

Li, Z., Tian, Y., Sun, J., Lu, X., Kan, Y., 2022. Simulation-based optimization of large-scale dedicated bus lanes allocation: Using efficient machine learning models as surrogates. Transportation Research Part C: Emerging Technologies 143, 103827.

Lu, L., Su, Y., Yao, D., Li, L., Li, Z., 2010. Optimal design of bus stops that are shared by multiple lines of buses. In: Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on. IEEE, pp. 125–130.

Newell, G. F., 1993. A simplified theory of kinematic waves in highway traffic, part II: Queueing at freeway bottlenecks. Transportation Research Part B: Methodological 27 (4), 289–303.

Newell, G. F., Potts, R. B., 1964. Maintaining a bus schedule. In: 2nd Australian Road Research Board (ARRB) Conference, Melbourne.

Pollaczek, F., 1930. Über eine aufgabe der wahrscheinlichkeitstheorie. Mathematische Zeitschrift 32 (1), 64–100.

Rezazada, M., Nassir, N., Tanin, E., 2022. Public transport bunching: A critical review with focus on methods and findings for implications for policy and future research. In: Australasian Transport Research Forum Proceedings, 28-30 September, Adelaide, Australia.

Schmöcker, J.-D., Sun, W., Fonzone, A., Liu, R., 2016. Bus bunching along a corridor served by two lines. Transportation Research Part B: Methodological 93, 300–317.

Shanmukhappa, T., Ho, I. W. H., Tse, C. K., 2018. Spatial analysis of bus transport networks using network theory. Physica A: Statistical Mechanics and its Applications 502, 295–314.

Shen, M., Gu, W., Cassidy, M. J., Lin, Y., Ni, W., 2023. Abating a vicious cycle in bus operations along busy corridors by holding (working paper).

Shen, M., Gu, W., Hu, S., Cheng, H., 2019. Capacity approximations for near-and far-side bus stops in dedicated bus lanes. Transportation Research Part B: Methodological 125, 94–120.

Shi, L., Ólafsson, S., 2000. Nested partitions method for global optimization. Operations Research 48 (3), 390–407.

St. Jacques, K., Levinson, H. S., 1997. Operational analysis of bus lanes on arterials. Transit Cooperative Research Program Report 26, Transportation Research Board, Washington, D.C., USA.

Stamatopoulos, M. A., Zografos, K. G., Odoni, A. R., 2004. A decision support system for airport strategic planning. Transportation Research Part C: Emerging Technologies 12 (2), 91–117.

Tan, J., Li, Z., Li, L., Zhang, Y., Lu, L., 2014. Berth assignment planning for multi-line bus stops. Journal of Advanced Transportation 48 (7), 750–765.

Toledo, T., Cats, O., Burghout, W., Koutsopoulos, H. N., 2010. Mesoscopic simulation for transit operations. Transportation Research Part C: Emerging Technologies 18 (6), 896–908.

Wikipedia, n.d. Red metropolitana de movilidad. [Online]. Available at: `https://en.wikipedia.org/wiki/Red_Metropolitana_de_Movilidad`. (Accessed June 8, 2023).

Wu, W., Liu, R., Jin, W., 2016. Designing robust schedule coordination scheme for transit networks with safety control margins. Transportation Research Part B: Methodological 93, 495–519.

Wu, W., Liu, R., Jin, W., 2017. Modelling bus bunching and holding control with vehicle overtaking and distributed passenger boarding behaviour. Transportation Research Part B: Methodological 104, 175–197.

Wu, W., Liu, R., Jin, W., Ma, C., 2019. Simulation-based robust optimization of limited-stop bus service with vehicle overtaking and dynamics: A response surface methodology. Transportation Research Part E: Logistics and Transportation Review 130, 61–81.

Wu, X., Li, Z., Li, L., Su, Y., Lu, L., Tan, J., 2011. Berth assignment planning for multi-berth bus stops. In: Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on. IEEE, pp. 1519–1524.

Xuan, Y., Argote, J., Daganzo, C. F., 2011. Dynamic bus holding strategies for schedule reliability: Optimal linear control and performance analysis. Transportation Research Part B: Methodological 45 (10), 1831–1845.

Yang, F., Gu, W., Cassidy, M. J., Li, X., Li, T., 2020. Achieving higher taxi outflows from a drop-off lane: A simulation-based study. Transportation Research Part C: Emerging Technologies 115, 102623.

Yu, B., Lam, W. H., Tam, M. L., 2011. Bus arrival time prediction at bus stop with multiple routes. Transportation Research Part C: Emerging Technologies 19 (6), 1157–1170.

Zhao, J., Chen, K., Wang, T., Malenje, J. O., 2019. Modeling loading area effectiveness at off-line bus stops with no clear-cut separation of berths. Transportmetrica A: transport science 15 (2), 396–416.

Zheng, L., Liu, P., Huang, H., Ran, B., He, Z., 2022. Time-of-day pricing for toll roads under traffic demand uncertainties: A distributionally robust simulation-based optimization method. Transportation Research Part C: Emerging Technologies 144, 103894.