Periodic Stop Skipping: NP-hardness and computational limitations

Dr Konstantinos Gkiotsalitis
Assistant Professor
University of Twente

Center for Transport Studies (CTS)
Department of Civil Engineering
P.O. Box 217
7500 AE Enschede
The Netherlands
Email: k.gkiotsalitis@utwente.nl

99th Annual Meeting of the Transportation Research Board, Washington D.C.
Paper no. 20-00069

January 2020

ABSTRACT
Stop-skipping (also known as expressing) is a typical control strategy in public transit operations with a dual objective: (i) reduce the trip delays, and (ii) improve the travel times of onboard passengers. Stop-skipping approaches decide about the stop-skipping strategy of each bus trip in isolation, neglecting the effect of the skipped stops to future trips. To rectify this, we introduce a rolling-horizon stop-skipping model that determines the skipped stops of several trips within a rolling horizon. Then, we model the rolling-horizon stop-skipping problem as an integer nonlinear program, and we prove that it is an NP-Hard problem which can be solved to global optimality for small-scale scenarios. Extensive numerical experiments in a high frequency bus line in Singapore investigate the number of trips that can be optimized in a rolling horizon subject to the number of candidate stops that can be skipped. This investigation reveals the computational limitations of this problem and provides useful lessons to public transport practitioners that want to introduce periodic stop-skipping control to their daily operations.

Keywords: periodic stop skipping; integer nonlinear programming; rolling horizon optimization; combinatorial optimization; public transit.

INTRODUCTION

From the tactical planning stage, bus lines are expected to have a fixed service interval (headway) determined by the selected line frequency Hadas and Shnaiderman, Gkiotsalitis and Cats (*1, 2*). Nevertheless, travel time and passenger demand variations during the actual operations result in unreliable and inconsistent services Chen et al., Daganzo, Gkiotsalitis et al., Gkiotsalitis and Alesiani (*3, 4, 5, 6*). To rectify this, several flexible scheduling approaches have emerged over the past 40 years with a shifted focus towards operational control. Operational control includes a variety of options, such as bus holding Bartholdi III and Eisenstein, Delgado et al., Gkiotsalitis and Cats (*7, 8, 9*), stop-skipping Liu et al., Chen et al., Gkiotsalitis (*10, 11, 12*), short-turning Furth, Cortés et al. (*13, 14*), interlining Delle Site and Filippi, Gkiotsalitis et al. (*15, 16*), re-scheduling Gkiotsalitis, Gkiotsalitis and Van Berkum (*17, 18*), and speed control Daganzo and Pilachowski, Muñoz et al. (*19, 20*).

In this study, we specifically focus on the problem of stop-skipping at the operational planning stage. Stop-skipping can correct service inconsistencies due to the inherent travel time and passenger demand variations but might result in increased waiting times at the locations of the skipped stops Chen et al. (*11*). Thus, we address the problem in a holistic manner considering the waiting times of passengers, their in-vehicle times, and the total bus trip travel times. The two former objectives concern the passenger-related costs, whereas the last objective concerns the cost of the operator.

Addressing the stop-skipping problem at the operational level requires to compute a stop-skipping solution within a short period of time. Given the computational complexity of the stop-skipping problem, several works consider the stop-skipping strategy of only one trip at a time to reduce the size of the solution space Liu et al., Fu et al. (*10, 21*). Such treatment enables the computation of a stop-skipping solution, but results in a myopic control option because it addresses every bus trip separately without acknowledging that it belongs to a chain of trips Bartholdi III and Eisenstein (*7*). Other approaches calculate a stop-skipping plan for the entirety of daily trips Gkiotsalitis (*12*), but they cannot be applied in operational control because of the significant computational costs associated with the computation of a daily schedule of skipped stops.

In this study, we investigate the potential of a hybrid strategy where the skipped stops of a pre-selected number of trips are determined in rolling horizons that can contain more than one trip. Consistent with the theory of rolling horizon optimization, a rolling horizon is understood as a time period that includes a pre-determined number of trips for which a stop-skipping decision needs to be made Bostel et al. (*22*). When the rolling horizon is over, a new rolling horizon starts, and this continues until the end of the daily operations.

The remainder of this paper is structured as follows: in section 2 we provide the literature review in the area of stop-skipping and report the incremental contribution of our work. In section 3, we model the stop-skipping problem in rolling horizons extending the model of Fu et al. (*21*) and proving its NP-hardness. In section 4, we present exact solution methods, such as simple enumeration (brute-force), and we investigate their theoretical computational costs. In section 5, the numerical experiments are performed starting from a small demonstration of our model in a toy network. Then, the computational costs of the periodic stop-skipping problem are investigated in the test-case of bus line 302 in Singapore, considering different numbers of (a) trips in the rolling horizon, and (b) candidate stops to be skipped. The main findings and the limitations of our study are discussed in section 6 which concludes our work.

LITERATURE REVIEW

Stop-skipping strategies can be devised at the tactical planning level or at the operational level (dynamic stop skipping). Depending on the level of control, the objectives of a stop-skipping strategy might differ. At the tactical planning stage, the focus is on developing reliable, resilient or robust strategies that will maintain a good performance in case of disruptions during the actual operations. On the contrary, dynamic stop-skipping strategies at the operational level are reactionary and less sophisticated because they need to be simple and computationally efficient. A review of past works at the different planning levels is provided below.

Stop-skipping at the Tactical Planning Level

A line of research addresses the stop-skipping problem at the tactical planning stage Jordan and Turnquist, Furth (*23, 24*). At the tactical planning stage, a stop-skipping strategy is devised prior to the start of the daily operations and is not updated ever since. The main benefit is that the stop-skipping strategy serves as a fixed plan which can be communicated to both the bus drivers and the passengers well in advance. On the other side, it cannot be adjusted during the operational stage and cannot react to changes during the actual operations.

Furth and Day (*25*) and Furth (*24*) analyzed the effect of four pre-planned strategies (short-turning, restricted zonal service, semi-restricted zonal service, and stop-skipping) to bus lines with unbalanced demand between directions. The explored objectives were the minimization of the fleet size and the improvement of the passenger-related cost. Gkiotsalitis (*12*) proposed a combination of genetic algorithm and linear programming to develop a stop-skipping strategy for the entire day of operations which performs well at worst-case scenarios (robust stop-skipping plan). The approach was tested in a circular bus line in Singapore demonstrating a potential performance improvement of more than 10% at worst-case scenarios.

Jamili and Aghaee (*26*) focused on finding optimum stop-skipping patterns in railway systems. As in Gkiotsalitis (*12*), they developed robust stop-skipping plans using metaheuristics (namely, a decomposition-based algorithm and a simulated annealing-based algorithm). After testing their solution in an Iranian metro line, the results demonstrated that the simulated annealing metaheuristic offers better results in large-scale problems.

Dynamic Stop-Skipping

In dynamic control, several approaches determine the skipped stops of a bus trip when it is about to be dispatched Fu et al., Li et al., Lin et al., Eberlein (*21, 27, 28, 29*). Determining the skipped stops for each trip in isolation reduces the problem complexity and limits the solution space. In more detail, skipping a stop is modeled as a 0-1 decision problem, where 0 denotes a skipped stop. If only one trip is considered, the solution space comprises of $2^{|S|}$ different options where $|S|$ is the total number of stops that can be optionally skipped. Note that the solution space increases exponentially with the number of stops and cannot be explored for large values of $|S|$. Nevertheless, several works resort to exhaustive search methods (brute-force) to solve the dynamic stop-skipping problem taking advantage of the relatively small scale of the problem in bus lines with less than 20 stops Fu et al., Sun and Hickman (*21, 30*).

Sun and Hickman (*30*) modeled the stop-skipping problem as a nonlinear integer program including assumptions of random distributions of passenger boardings and alightings. Then, the problem was solved with exhaustive search. Similarly, Fu et al. (*21*) used an exhaustive search to determine the skipped stops of one trip at a time. Fu et al. (*21*) considered the total waiting times of passengers, the in-vehicle time, and the total trip travel time as problem objectives. The potential benefit was tested with a simulation of route 7D in Waterloo, Canada.

While in Fu et al. (*21*) two consecutive bus trips were not allowed to skip the same stop, Liu et al. (*10*) used a more strict rule. In Liu et al. (*10*), if a bus trip skips one (or more) stops its preceding and following trip should not skip any stops. The formulation of Liu et al. (*10*) resulted in a mixed integer nonlinear program with a non-convex objective function. Hence, Liu et al. (*10*) used a genetic algorithm incorporating Monte Carlo simulations for the solution of the problem. Contrary to the more sophisticated models, Eberlein (*31*) developed a simplified transit operation environment to derive the stop-skipping solutions analytically. In this simplification, the stop-skipping problem was modeled as an integer nonlinear program with quadratic objective function and constraints.

Other approaches have considered the stop-skipping problem in combination with short-turning. Li et al. (*27*) considered both the stop-skipping and short-turning problems formulating them as a single 0-1 stochastic programming model accounting for both the deviations from the schedule and the unsatisfied passenger demand. Finally, given the problem complexity, Li et al. (*27*) used heuristic approaches and tested the solution performance with sample data from the Shanghai Transit Company.

Stop-skipping has also been combined with bus holding Lin et al., Eberlein, Cortés et al., Sáez et al. (*28, 31, 32, 33*). In Cortés et al. (*32*), the control decisions were applied when buses arrived at stops. Given the disproportionate increase of the problem complexity when accounting for both stop-skipping and bus holding, the problem was solved with a genetic algorithm-based multi-objective optimization solution method. Lin et al. (*28*) and Sáez et al. (*33*) integrated also the two aforementioned strategies. Lin et al. (*28*) measured the system performance in terms of passenger in-vehicle time and waiting time, and Sáez et al. (*33*) considered uncertain passenger demand by formulating it as a hybrid predictive control problem. Finally, Eberlein et al. (*34*) considered the disturbance of travel times and headway patterns when combining the dynamic stop-skipping and bus holding problems in an application at the Green Line of the Massachusetts Bay Transportation Authority.

Contribution

From the current literature, we identify a main research gap. Whereas there is an extensive body of works on dynamic stop-skipping, these works concentrate predominantly on determining the skipped stops of one trip at a time. Hence, they do not account for the (potential) negative effect of such decision to future trips that operate in the same line. This myopic decision mechanism motivates our work which focuses on periodic optimization that determines the skipped stops of several trips in a holistic manner.

In this pursuit, we incorporate the rolling horizon optimization theory in the dynamic stop-skipping problem. In more detail, the incremental contributions of this study to the state-of-the-art are:

- the modeling, for the first time, of the dynamic stop-skipping problem as a rolling horizon optimization problem by expanding the classical formulation of Fu et al. (*21*);
- the mathematical analysis of the resulting integer nonlinear program and the proof of its NP-hardness;

- the in-depth investigation of the scalability of the periodic stop-skipping problem with respect to the number of trips in the rolling horizon and the stop-skipping candidate stops.

MODEL FORMULATION

In rolling horizon optimization, the duration of the daily operations is split into discrete time windows. Trips are allocated to a time window if they are expected to be dispatched within its time period. At the start of each time window (also known as rolling horizon or epoch), the skipped stops of all trips belonging to this time window are determined simultaneously by solving a combinatorial problem. This is in strike contrast to most approaches that decide the stop-skipping strategy of one trip at a time Liu et al., Fu et al. (*10, 21*). Note that in extreme cases a rolling horizon can contain only one trip (and then the problem is reduced to the problem of Liu et al., Fu et al. (*10, 21*)) or the entirety of the daily trips of the bus line (and then the problem is transformed into a tactical planning problem Gkiotsalitis (*12*)).

In this work, a rolling horizon can contain any number of trips within the two extreme cases. One should note that the decision about the skipped stops of all trips within a rolling horizon is made at the start of the rolling horizon. If a rolling horizon contains a large number of trips, it is advised to re-evaluate the decisions every time a new trip is about to be dispatched to incorporate potential updates on the estimated travel times and passenger demand Eberlein et al. (*35*). In the remainder of this section, we introduce the mathematical model of the stop-skipping problem in rolling horizons starting from the main assumptions and the nomenclature.

Assumptions and nomenclature

The modeling part of this work relies on the following assumptions:

- Buses that serve the same line do not overtake each other. This is a common assumption in bus operations (see Xuan et al., Chen et al., Gkiotsalitis and Maslekar (*36, 37, 38*));
- The passenger arrivals at stops are random because passengers cannot coordinate their arrivals with the arrival times of buses at regularity-based services Welding, Randall et al. (*39, 40*);
- The passenger demand at skipped stops is accommodated by the next bus trip of the same line Liu et al., Fu et al., Sun and Hickman (*10, 21, 30*);
- Passengers use different door channels for boardings and alightings.

Before proceeding to the modeling, we introduce the following nomenclature:

NOMENCLATURE

Sets

N	ordered set of bus trips in the rolling horizon, $N = \langle 1, ...n, ..., \|N\| \rangle$. Note that trip 1 is the first trip to be dispatched in this rolling horizon;
S	set of ordered bus stops, $S = \langle 1, ..., s, ..., \|S\| \rangle$;

Parameters

T	is a $\|N\| \times (\|S\|-1)$ matrix of running times. Each element $t_{n,s} \in \mathbb{R}_{\geq 0}$ of matrix **T** is the expected running time of the n-th trip between stop $s-1$ and s, where $s \in S \setminus \{1\}$;
r_1	average boarding time per passenger, a constant;
r_2	average alighting time per passenger, a constant;
δ	average bus acceleration plus deceleration time for serving a bus stop, a constant;
Λ	$\|S\| \times \|S\|$ matrix where each element $\lambda_{sy} \in \mathbb{R}_{\geq 0}$ of matrix Λ denotes the average passenger arrival rate at stop s whose destination is stop y (note: $\lambda_{sy} = 0,\ \forall 1 \leq y \leq s$);
c_1	unit time value associated with the passenger waiting times ($/h);
c_2	unit time value associated with the passenger in-vehicle travel time ($/h);
c_3	unit time value associated with the vehicle operation time ($/h);
$\tilde{d}_{n,1}$	planned departure time of every trip $n \in N$ from the first stop;
$\tilde{w}_{1,sy}$	number of passengers waiting for trip 1, which is the first trip of the rolling horizon, and traveling from stop $s \in S$ to stop $y \in S$;

Decision Variables

x	$\|N\| \times \|S\|$-dimensional matrix of the decision variables where each $x_{n,s} \in \mathbf{x}$ can take a binary value $\{0,1\}$ with $x_{n,s} = 1$ denoting that the n-th bus trip will serve stop s.

Variables

D	$\|N\| \times \|S\|$ matrix of departure times where $d_{n,s} \in \mathbb{R}_{\geq 0}$ is the departure time of trip n from stop s, where $n \in N$ and $s \in S$;
A	$\|N\| \times \|S\|$ matrix of arrival times where $a_{n,s} \in \mathbb{R}_{\geq 0}$ is the arrival time of trip n at stop s, where $n \in N$ and $s \in S$;
K	$\|N\| \times \|S\|$ matrix of dwell times where $k_{n,s} \in \mathbb{R}_{\geq 0}$ is the dwell time of trip n at stop s, where $n \in N$ and $s \in S$;
H	$(\|N\|-1) \times \|S\|$ matrix of bus headways where $h_{n,s} \in \mathbb{R}_{\geq 0}$ is the headway between trips $n-1$ and n at stop s, where $n \in N \setminus \{1\}$ and $s \in S$;

W	$\lvert N\rvert \times \lvert S\rvert \times \lvert S\rvert$ matrix where each $w_{n,sy} \in \mathbf{W}$ denotes the number of passengers waiting for bus n and traveling from stop s to y (note: $w_{n,sy} = 0,\ \forall y \leq s$);
L	$\lvert N\rvert \times \lvert S\rvert \times \lvert S\rvert$ matrix where each $l_{n,sy} \in \mathbb{R}_{\geq 0}$ denotes the number of passengers traveling from stop s to stop y skipped by bus n (note: $l_{n,sy} = 0,\ \forall y \leq s$);
M	$\lvert N\rvert \times \lvert S\rvert$ matrix where each $m_{n,s} \in \mathbb{R}_{\geq 0}$ denotes the number of passengers at stop s skipped by bus n, where $n \in N, s \in S$ (note: $m_{n,s} = \sum_{i=s+1}^{\lvert S\rvert} l_{n,si}$);
U	$\lvert N\rvert \times \lvert S\rvert$ matrix where each $u_{n,s} \in \mathbb{R}_{\geq 0}$ denotes the number of passengers boarding bus n at stop s, where $n \in N, s \in S$ (note: $u_{n,\lvert S\rvert} = 0,\ \forall n \in N$);
B	$\lvert N\rvert \times \lvert S\rvert \times \lvert S\rvert$ matrix where each $b_{n,sy} \in \mathbb{R}_{\geq 0}$ denotes the number of passengers boarding bus n at stop s whose destination is stop y (note: $b_{n,sy} = 0,\ \forall y \leq s$);
V	$\lvert N\rvert \times \lvert S\rvert$ matrix where each $v_{n,s} \in \mathbb{R}_{\geq 0}$ denotes the number of passengers alighting bus n at stop s, where $n \in N, s \in S$ (note: $v_{n,1} = 0,\ \forall n \in N$);
μ	$\lvert S\rvert$-valued vector, where each $\mu_s \in \mathbb{R}_{\geq 0}$ denotes the average passenger arrival rate at stop s (note: $\mu_s = \sum_{i=s+1}^{\lvert S\rvert} \lambda_{si}$).

Variable values

Our formulation differs from the common formulations of the dynamic stop-skipping problem because it considers all trips, $N = \langle 1, ..., n, ..., |N| \rangle$, within a rolling horizon when determining the skipped stops.

The number of passengers destined to bus stop y who are stranded by bus trip n at stop $s, l_{n,sy}$, will be 0 if bus trip n serves stops s and y. Otherwise, it will equal the number of passengers waiting for bus trip n at stop s and have bus stop $y > s$ as their destination. Therefore, the value of variable $l_{n,sy}, \forall n \in N, \forall s \in S, \forall y \in S$ can be calculated as:

$$l_{n,sy} \triangleq \begin{cases} 0, \text{ if } y \leq s \\ w_{n,sy} - w_{n,sy} x_{n,s} x_{n,y}, \text{ if } y > s \end{cases} \tag{1}$$

Additionally, the number of passengers at stop s skipped by bus trip n is:

$$m_{n,s} \triangleq \sum_{y=s+1}^{|S|} l_{n,sy},\ \forall n \in N, s \in S \tag{2}$$

The number of passengers waiting for bus n at stop s whose destination is stop y depends on the number of passengers skipped by bus $n-1$ at stop s, $l_{n-1,sy}$, and the average number of passengers who arrive at stop s after bus $n-1$ leaves stop s:

$$w_{n,sy} \triangleq \begin{cases} l_{n-1,sy} + \lambda_{sy} h_{n,s},\ \forall n \in N \setminus \{1\}, \forall s \in S \\ \tilde{w}_{1,sy},\ \text{for } n = 1, \forall s \in S \end{cases} \tag{3}$$

Note that $w_{n,sy} = \tilde{w}_{1,sy},$ for $n = 1, \forall s \in S$ reflects the boundary condition which is imposed at the first trip of the rolling horizon. The value of $\tilde{w}_{1,sy}$ does not depend on the decisions in this rolling horizon. Hence, in the current rolling horizon $\tilde{w}_{1,sy}$ is a parameter.

The expected number of passengers who will board bus trip n at stop s (assuming bus n stops at stop s) depends on the number of passengers traveling between stops s and y $(y > s)$ and whether the bus will stop at stop y:

$$u_{n,s} \triangleq x_{n,s} \sum_{y=s+1}^{|S|} w_{n,sy} x_{n,y},\ \forall n \in N, s \in S \setminus \{|S|\} \tag{4}$$

Note that at the last stop we have no boardings. Thus, we introduce the boundary condition:

$$u_{n,|S|} = 0,\ \forall n \in N \tag{5}$$

From the total amount of passengers boarding bus trip n at stop s ($u_{n,s}$), the number of passengers boarding bus trip n at stop s whose destination is stop y is:

$$b_{n,sy} \triangleq \begin{cases} x_{n,s} w_{n,sy} x_{n,y}, \text{ if } y > s \\ 0, \text{ if } y \leq s \end{cases} \tag{6}$$

The expected number of alighting passengers for bus trip n at stop s depends on the number of passengers traveling between stops y and s $(y < s)$ and whether the bus will make stop y. Thus, the value of $v_{n,s}$, $\forall n \in N, s \in S \setminus \{1\}$ can be derived by

$$v_{n,s} \triangleq x_{n,s} \sum_{y=1}^{s-1} w_{n,ys} x_{n,y},\ \forall n \in N, s \in S \setminus \{1\} \tag{7}$$

A special case is the first stop of a bus trip where we do not have passenger alightings. This introduces the boundary condition:

$$v_{n,1} = 0,\ \forall n \in N \tag{8}$$

The dwell time of each bus trip n at each stop s depends on the number of passengers who will board and alight at the stop, denoted by $u_{n,s}$ and $v_{n,s}$, respectively:

$$k_{n,s} \triangleq r_1 u_{n,s} + r_2 v_{n,s},\ \forall n \in N, s \in S \setminus \{1\} \tag{9}$$

Note that if passengers use different door channels for boardings/alightings; then, the dwell time can be expressed as $k_{n,s} \triangleq \max\left(r_1 u_{n,s}; r_2 v_{n,s}\right)$.

The arrival time of bus trip n at stop s is equal to its departure time at stop $s-1$ ($d_{n,s-1}$), plus the travel time between the two stops, plus the time lost in acceleration and deceleration:

$$a_{n,s} \triangleq d_{n,s-1} + t_{n,s} + \frac{\delta}{2}(x_{n,s-1} + x_{n,s}),\ \forall n \in N, s \in S \setminus \{1,2\} \tag{10}$$

Eq.(10) requires a boundary condition for the arrival time at the second stop, $a_{n,2}$, $\forall n \in N$. This is provided by the originally planned dispatching time $\tilde{d}_{n,1}$ of every trip $n \in N$:

$$a_{n,2} \triangleq \tilde{d}_{n,1} + t_{n,2} + \frac{\delta}{2}(x_{n,1} + x_{n,2}),\ \forall n \in N \tag{11}$$

In addition, the departure time of bus trip n from stop $s \in S \setminus \{1\}$ is equal to its arrival time at that stop plus the dwell time $k_{n,s}$:

$$d_{n,s} \triangleq a_{n,s} + k_{n,s},\ \forall n \in N, s \in S \setminus \{1\} \tag{12}$$

Assuming that overtaking between buses of the same line is not allowed, the departure headway between bus trip n and its preceding one reads:

$$h_{n,s} \triangleq d_{n,s} - d_{n-1,s},\ \forall n \in N \setminus \{1\}, s \in S \setminus \{1\} \tag{13}$$

Finally, note that the departure headway at the first stop is calculated based on the planned departure times of the respective trips:

$$h_{n,1} \triangleq \tilde{d}_{n,1} - \tilde{d}_{n-1,1},\ \forall n \in N \setminus \{1\} \tag{14}$$

Objective function and mathematical program

Stop-skipping strategies can have several (occasionally conflicting) objectives such as the minimization of passenger waiting times, on-board passenger delays and trip travel times. This yields a binary, multi-objective optimization problem that can be formulated with the use of weight factors c_1, c_2, c_3 (that convert all values to common units of cost in dollars) in order to minimize the equivalent weighted cost of passenger waiting time and passenger in-vehicle time as well as vehicle travel time:

$$\begin{aligned} f(\mathbf{x}) \triangleq\ & c_1 \sum_{n=2}^{|N|} \sum_{s=1}^{|S|} \Big[(u_{n,s} - m_{n-1,s}) \frac{h_{n,s}}{2} \\ & \qquad\qquad + m_{n-1,s} \Big(\frac{h_{n-1,s}}{2} + h_{n,s} \Big) \Big] \\ & + c_2 \sum_{n=2}^{|N|} \sum_{s=1}^{|S|-1} \sum_{y=s+1}^{|S|} \Big[b_{n,sy} \sum_{z=s+1}^{y} (t_{n,z} + (k_{n,z} + \delta) x_{n,z}) \Big] \\ & + c_3 \sum_{n=2}^{|N|} \sum_{s=2}^{|S|} (t_{n,s} + (k_{n,s} + \delta) x_{n,s}) \end{aligned} \tag{15}$$

where the generalized cost of the objective function includes three terms. The first term includes two components. The first component, $(u_{n,s} - m_{n-1,s}) \Big(\frac{h_{n,s}}{2} \Big)$, computes the total waiting time of the passengers who arrive after the departure (or passing) of bus $n-1$ at stop s, assuming random arrivals with an average passenger waiting time equal to half the headway. The second component represents the total waiting time of those passengers who have been stranded by bus $n-1$ ($m_{n-1,s}$) and have to wait for an average amount of time equal to $m_{n-1,s} \Big(\frac{h_{n-1,s}}{2} + h_{n,s} \Big)$. The second term of the objective function calculates the total in-vehicle time of passengers summed over all O-D pairs and the final term computes the total bus trip time.

Incorporating the previously formulated vehicle movement equations, yields the following mathematical program:

$$
\begin{aligned}
(Q): \quad & \min_{\mathbf{x}} \; f(\mathbf{x}) && (16)\\
& \text{s.t. } \mathbf{x} \in \mathscr{F}(x) = \left\{ \; \mathbf{x} \mid \mathbf{x} \text{ satisfies Eq. (1)-(14)} \; \right\} && (17)\\
& \quad x_{n,1} = x_{n,|S|} = 1 \; , \forall n \in N && (18)\\
& \quad x_{n,s} + x_{n-1,s} \geq 1, \; \forall n \in N \setminus \{1\}, \; \forall s \in S && (19)\\
& \quad x_{n,s} \in \{0,1\} \; , \forall n \in N, \; \forall s \in S && (20)
\end{aligned}
$$

Note that the equality constraint of eq.(18) ensures that the first and last stops of a bus trip cannot be skipped and the inequality constraints of eq.(19) that if a bus stop is skipped by one trip, it will be served by the next one.

Program (Q) is an integer nonlinear programming problem (INLP). Due to its combinatorial nature, the problem can be solved to global optimality with exhaustive search of the solution space. In terms of computational complexity, it is an NP-Hard problem as it is formally proved in theorem 3.1.

Theorem 3.1. *The rolling horizon stop-skipping problem, (Q), is an NP-Hard decision problem with an exponential computational complexity that requires to explore $2^{|N|\times|S|}$ potential solutions to obtain a globally optimal one.*

Proof. Given that the constraints limit $x_{n,s}$ to either 0 or 1, any feasible solution to the integer program (Q) is a subset of vertices. The first constraint implies that at least one end point of every edge is included in this subset. Therefore, the solution describes a *vertex cover*. Additionally, given some vertex cover C, $x_{n,s}$ can be set to 1 for any $(n,s) \in C$ and to 0 for any $(n,s) \notin C$, thus giving us a feasible solution to the integer program. Hence, our problem can be reduced to the *minimum vertex cover* which is one of Karp's 21 NP-Complete decision problems Karp (*41*) that demonstrates the NP-Hardness of problem (Q). That is, there is no polynomial algorithm that can solve all instances of (Q), unless P≡NP.

Now, if we want to find the globally optimal stop-skipping strategy of one trip, we need to explore a set of $2^{|S|}$ potential solutions because at each stop $s \in \{1,2,...,|S|\}$ we have two options: serve or skip. In a rolling horizon with $|N|$ trips, we should make a total number of $|N| \times |S|$ simultaneous stop-skipping decisions. That is, the potential solutions that need to be evaluated are $2^{|N|\times|S|}$. □

Evaluating the feasibility and the performance of $2^{|N|\times|S|}$ potential stop-skipping solutions using the equations of program (Q) is not a trivial task. First, an increase in the number of trips and/or stops increases exponentially the solution space. Second, and equally important, the equations of program (Q) include several recursive relationships that increase the computational cost when the number of trips and stops is increased.

EXPLORATION OF THE SOLUTION SPACE

Solution Method

An obvious solution method that returns a globally optimal solution of this discrete optimization problem is the exhaustive search (brute force method). The brute force solution method evaluates exhaustively the values of the variables and the objective function covering the entire solution space. To cover the entire solution space, the number of solution evaluations is $2^{|N|\times|S|}$. For a typical bus line with 20 stops, the total number of solution evaluations varies with respect to the number of trips, $|N|$. This is depicted in Fig.1 which is plotted in logarithmic scale. Fig.1 indicates the increase of the solution space subject to the increase of the number of trips in the rolling horizon for a typical bus line with 20 stops.

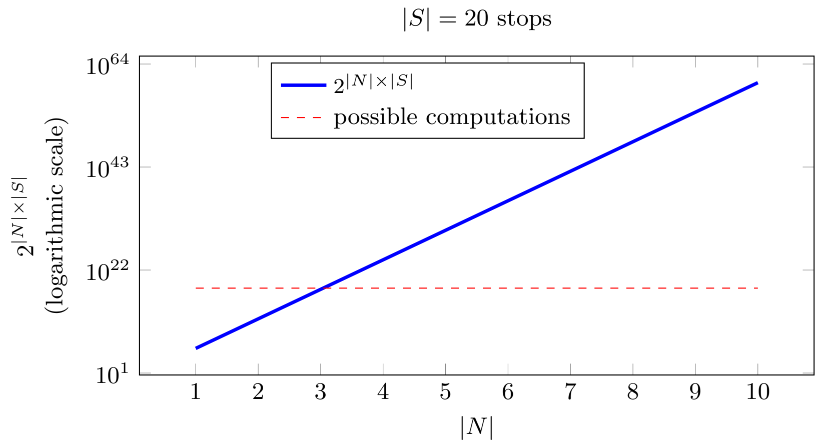

FIGURE 1 : Solution space size for a bus line with $|S| = 20$ bus stops when the number of trips in the rolling horizon, $|N|$, varies.

Other exact optimization approaches for combinatorial optimization include branch and bound (B&B); however, in our case, B&B is reduced to an exhaustive search because our 0-1 problem does not have a continuous relaxation. We finally note that this study is concerned with exact solutions and will not resort into the application of heuristics that do not guarantee global optimality. Notwithstanding this, in our numerical experiments we investigate the scalability of our problem even in the case of using heuristics.

Solution Space Pruning

As previously proved, comparing the performance of all potential solutions to obtain a globally optimal one requires to evaluate the objective function f of program (Q) for any potential solution $\mathbf{x}^i$ in the solution space $\mathscr{P} = \langle 1, 2, ..., 2^{|N|\times|S|} \rangle$. However, evaluating the objective function score, $f(\mathbf{x}^i)$, for a given solution $\mathbf{x}^i \in \mathscr{P}$ requires to perform several recursive computations expressed in eq.(1)-(14) that result in significant computational costs. Consequently, it is important to reduce the required number of the time-consuming objective function evaluations as much as possible.

In this pursuit, one can exploit the infeasibility of several solution candidates. First, note that the constraints of eq.18 do not allow any trip to skip the first or the last stop of the bus service. Hence, our feasible solution space $\tilde{\mathscr{P}}$ is reduced from $2^{|N|\times|S|}$ to $2^{|N|\times(|S|-2)}$ solution candidates. Second, the constraints of eq.19 in (Q) do not allow two consecutive bus trips to skip the same stop because this will result in unacceptable waiting times. Exploiting this constraint, several solution candidates in $\tilde{\mathscr{P}}$ are infeasible and can be discarded without performing time-consuming objective function evaluations.

NUMERICAL EXPERIMENTS

Demonstration in a Toy Network

The computational tests are performed in a general-purpose computer with Intel Core i7-455 7700HQ CPU @ 2.80GHz and 16 GB RAM. To facilitate the reproduction of our results, we provide below the input data of a toy network and report its exact solution by solving program (Q) with brute force.

In the toy network, we consider a circular bus line with 5 bus stops, as presented in Fig.2. We also consider 4 trips in the rolling horizon, and fixed travel times $t_{n,s} = 60$ sec, $\forall n \in N$, $\forall s \in S \setminus \{1\}$. Additionally, the average number of passenger arrival rate at stop $s \in S$ whose destination is stop $y \in S$ is set to $\lambda_{n,sy} = 0.1$ passengers/sec (that is, 6 passengers per minute) if $y > s$ and 0 if $y \leq s$. The number of passengers waiting for trip 1, which is the first in this rolling horizon, and traveling from stop $s \in S$ to stop $y \in S$ is set as $\tilde{w}_{1,sy} = 12$ if $y > s$ and 0 otherwise.

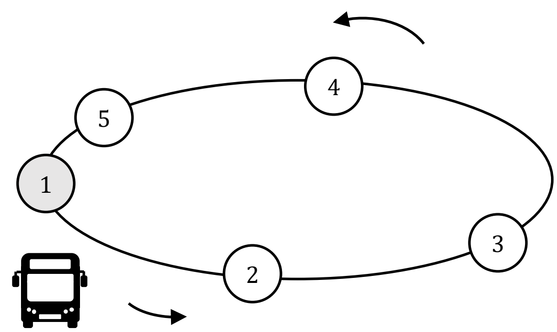

FIGURE 2 : Topology of the toy network operating one bus line.

The values of the other parameters of the idealized scenario(s) in this toy network are presented in Table 1.

TABLE 1 : Parameter values of the idealized scenario

Parameter	Value	Parameter	Value
r_1	4 sec	c_1	10 \$/h
r_2	2 sec	c_2	5 \$/h
δ	20 sec	c_3	7 \$/h

Finally, the planned dispatching times of the 4 trips are $\tilde{d}_{n,1} = 600(n-1), \forall n \in \{1,2,...,4\}$. That is, trip $n = 1$ is dispatched at $\tilde{d}_{n,1} = 0$ sec, trip $n = 2$ at $\tilde{d}_{n,1} = 600$ sec and so forth. This indicates a 10-minute dispatching headway among trips.

To demonstrate the application of our model, we evaluate the cost of the objective function of program (Q) for any potential stop-skipping combination. This requires ca. $2^{|N|\times(|S|-2)} = 2^{4\times 3}$ evaluations of the objective function with the brute-force method. The mathematical program (Q) is programmed and solved in Python 3.6 using an exhaustive search. Its globally optimal solution is:

$$\mathbf{x}^* = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

with a generalized objective function cost,

$$f(\mathbf{x}^*) \simeq 2,602,701 \ \$$$

The computational cost of evaluating all possible stop-skipping options for the case of 5 stops and 4 trips was 82 min and 21 sec.

Computational Cost Analysis: remarks from the edge of impossibility

Evaluating the performance of ca. $2^{|N|\times(|S|-2)}$ potential stop-skipping solutions to obtain the globally optimal solution of (Q) is a computational intensive task because:

- an increase in the number of trips and/or stops increases exponentially the solution space.
- the equations of program (Q) include several recursive relationships that increase the computational cost of evaluating the performance of a stop-skipping solution when the number of trips and stops is increased.

In this sub-section, we perform computational tests to investigate how many trips can we include in a rolling horizon depending on the number of stop-skipping candidate stops. Since our work is focusing on high-frequency services where the passenger arrivals at stops are random Welding, Randall et al. (*39, 40*), we consider that services operate with high frequencies of at least 6 trips per hour. In this context, every pair of trips must be dispatched with (at most) a 10-minute headway. Consequently, in periodic optimization of high frequency services, there is a time limit of up to 10 minutes to compute a new stop-skipping strategy by the time a trip is dispatched. This time limit is used in our numerical experiments to investigate how many trips can we include in our rolling horizon to derive an optimal stop-skipping strategy depending on the number of stops of the bus line.

To perform computational cost experiments in a realistic context, we consider the high frequency, circular bus line 302 in Singapore. Bus line 302 has 22 bus stops departing from Choa Chu Kang Loop - Choa Chu Kang Int (44009) and ending at the same stop. It is operated by SMRT and its regularity is monitored by the Land Transport Authority (LTA). Normally, starts operating at 05:30 and ends at 00:55. Its route length is 8.1 km and its total travel time typically ranges from 35 to 40 minutes. Bus line 302 is monitored in terms of service regularity and is placed under the Bus Service Reliability Framework (BSRF) from the LTA Leong et al. (*42*).

Bus line 302 is a feeder service that serves residential blocks, schools, and public amenities, connecting them to Choa Chu Kang Town Centre and Yew Tee Mass Rapid Transit (MRT) station. The topology of bus line 302 and its 22 bus stops are presented in Fig.3. Note that since it is a circular line, the terminal counts as both the first and the last stop of the bus line.

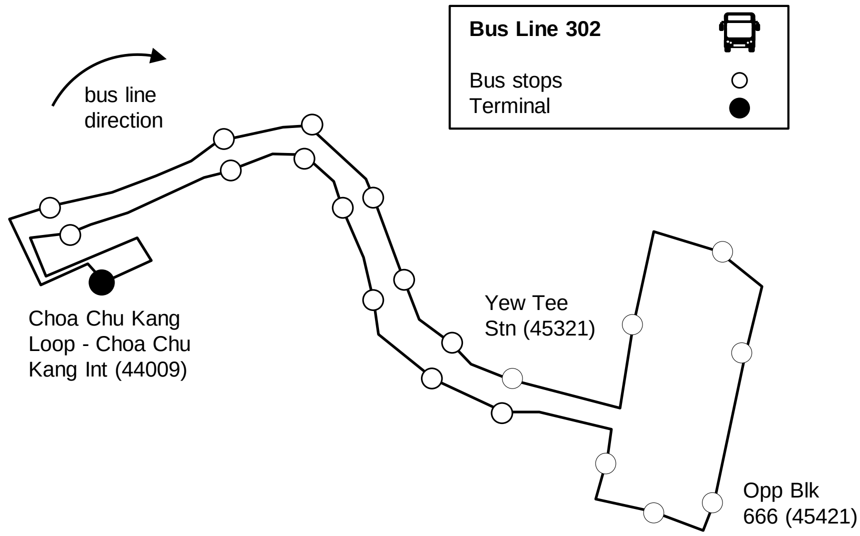

FIGURE 3 : Topology and bus stops of bus line 302 in Singapore

In our numerical experiments, we consider a rolling horizon that can include from 1 to 8 trips and we calculate the respective computational costs of finding a globally optimal stop-skipping solution. As it will become apparent after analyzing the results in Table 2, the reason why we consider 1 to 8 trips is that we cannot compute a globally optimal solution within a reasonable time for more than 8 trips.

From Table 2 one can note that as the number of trips in the rolling horizon increases, the computational costs exhibit a disproportionate increase. That is, after some point we cannot obtain a solution within a reasonable time if every bus stop of bus line 302 is a stop-skipping candidate. Thus, we perform several computational experiments in scenarios with varying numbers of stop-skipping candidates (i.e., from 3 up to 22 stops) and report the results in Table 2.

TABLE 2 : Computational costs in CPU minutes for different numbers of trips in the rolling horizon subject to the number of stop-skipping candidate stops. Computational times that violate the computational time limit of 10 minutes are not reported.

Stops	**Trips in the rolling horizon**							
	1	2	3	4	5	6	7	8
3	0.00	0.00	2E-04	3E-03	0.02	0.26	2.05	**25.61**
4	0.00	0.00	0.01	0.37	**18.23**	**121.71**	**>10.00**	**>10.00**
5	0.00	0.01	0.42	**82.36**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
6	0.00	0.13	**315.66**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
7	0.00	1.74	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
8	0.00	**23.72**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
9	0.00	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
10	0.00	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
11	2E-04	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
12	7E-04	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
13	2E-03	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
14	3E-03	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
15	7E-03	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
16	0.01	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
17	0.03	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
18	0.05	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
19	0.11	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
20	0.24	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
21	0.51	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**
22	1.06	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**	**>10.00**

From Table 2, one can note that:

- similarly to Fu et al. (*21*), rolling horizons with a single trip can be solved to global optimality regardless of the number of candidate stops to be skipped within a limited time of less than 2 CPU minutes.
- considering two trips in a rolling horizon increases significantly the computational cost of the combinatorial stop-skipping problem. In this case, one must consider at most 7 stop-skipping candidate stops to return a solution within 10 minutes.
- considering three trips in a rolling horizon can return a solution within 10 minutes only if the stop-skipping candidate stops are less than 6.
- considering five to seven trips in a rolling horizon can return a solution within 10 minutes only if we consider three (or less) stop-skipping candidate stops.
- eight trips in a rolling horizon cannot return an optimal solution within a reasonable time.

The aforementioned observations are supported by Fig.4 which plots the maximum number of trips that can be in a rolling horizon for computing a globally optimal solution in less than 10 minutes subject to the number of stop-skipping candidate stops.

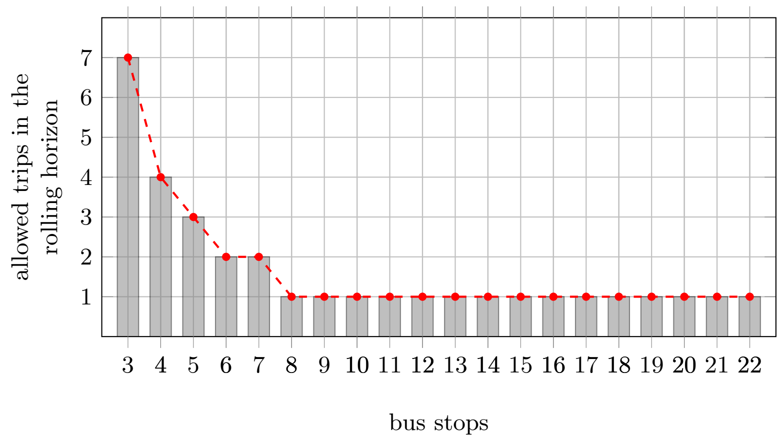

FIGURE 4 : Maximum number of trips that can be in a rolling horizon for computing an exact stop-skipping solution in less than 10 min subject to the number of bus stops that are stop-skipping candidates.

From Table 2 and Fig.4 one can note that the periodic stop-skipping problem cannot be solved to global optimality within a reasonable time when the number of trips in the rolling horizon increases (i.e., for 2 trips we can consider up to 7 stop-skipping candidates, for 3 trips up to 5, for 4 trips up to 4, and so forth).

This disproportionate increase in computational costs provides a motivation towards resorting into heuristics. Heuristics can return a solution which might exhibit a large optimality gap. In return, they require to evaluate only a fraction of the the pruned solution space, $2^{|N|\times(|S|-2)}$. Notwithstanding, even if one explores a small fraction of the solution space, the computational cost can still be prohibitive. The reason is that the computational cost does not depend only on the number of objective function evaluations, but also on the cost of a single objective function evaluation which increases when the number of stops and trips increase given the recursive relations in program (Q). That is, heuristics might not be able to return an approximate solution within the time limit of 10 minutes in scenarios with many stops and trips.

To investigate the size of the problems that can be treated with heuristics, in Table 3 we report the computational cost of a single evaluation of the objective function $f(\mathbf{x})$ when the stop-skipping candidates and the trips in the rolling horizon increase. As expected, the recursive relationships in (Q) result in increased computational costs for evaluating the performance of one solution when the number of trips and stops increases.

TABLE 3 : Computational costs in CPU seconds for a single evaluation of the objective function for different numbers of trips in the rolling horizon subject to the number of stops of the bus line

Stops	**Trips in the rolling horizon**							
	1	2	3	4	5	6	7	8
3	0.00	0.00	0.01	0.01	0.02	0.31	1.11	5.89
4	0.00	0.00	0.03	0.15	1.14	7.37	53.39	**360.59**
5	0.00	0.01	0.13	1.26	13.56	**133.62**	**1366.06**	**>90.00**
6	0.00	0.03	0.58	9.94	**130.42**	**>90.00**	**>90.00**	**>90.00**
7	0.00	0.09	3.01	58.12	**>90.00**	**>90.00**	**>90.00**	**>90.00**
8	0.00	0.34	12.78	**321.96**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
9	0.00	1.06	50.17	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
10	0.00	3.45	**190.20**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
11	0.00	10.46	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
12	0.00	29.78	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
13	0.00	**91.39**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
14	0.00	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
15	0.00	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
16	0.00	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
17	0.00	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
18	0.00	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
19	0.00	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
20	0.00	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
21	0.00	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**
22	0.00	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**	**>90.00**

In more detail, from Table 3 one can note that the computational cost of a single evaluation of the objective function is more than 1½ minutes in cases with:

- 2 trips in the rolling horizon and 13 stop-skipping candidates
- 3 trips in the rolling horizon and 10 stop-skipping candidates
- 4 trips in the rolling horizon and 8 stop-skipping candidates
- 5 trips in the rolling horizon and 6 stop-skipping candidates
- 6 or 7 trips in the rolling horizon and 5 stop-skipping candidates

Note that a computational cost of more than 1½ minutes for evaluating the objective function is almost prohibitive for computing an acceptable solution. For instance, for a time limit of up to 10 minutes, a heuristic algorithm will start from a random initial solution guess and will be allowed to evaluate the objective function score up to 6 times until reaching the 10-minute time limit. That is, it is highly unlikely to return a solution close to the globally optimal one if our heuristic can evaluate only 6 points out of the vast number of points in our solution space (ca. $|N| \times (|S|-2)$). Nevertheless, even if we assume that we are satisfied with a very large optimality gap, such a solution cannot be always computed because several scenarios require more than 1½ minutes for the evaluation of the objective function score of a potential solution.

For instance, let us consider the case of 4 trips in the rolling horizon and the case of 4 candidate stops to be skipped. As presented in Fig.5, in both cases the computational cost of evaluating the objective function becomes prohibitive when considering 4 trips and 8 stops, or 4 stops and 8 trips. Hence, the recursive relations in program (Q) do not permit the solution of the periodic stop-skipping problem in cases with several trips and stops even if one does not seek a globally optimal solution.

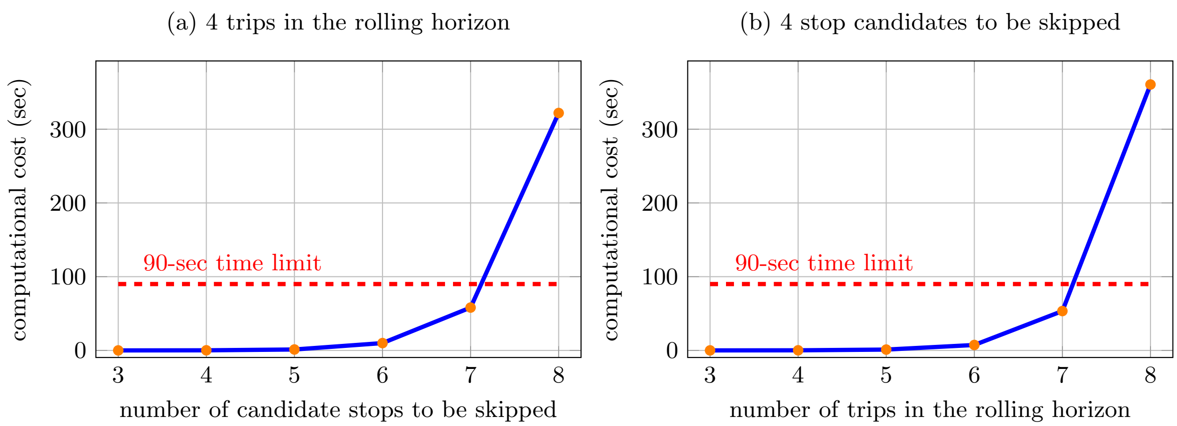

FIGURE 5 : Computational cost when evaluating the performance of a single solution in the case of fixed trips (a) and fixed stops (b).

DISCUSSION AND CONCLUDING REMARKS

Discussion of the main findings

This study introduced a model formulation for the periodic stop-skipping problem. The proposed model is an integer nonlinear programming problem that is proved to be NP-Hard. Consequently, a globally optimal stop-skipping solution can be found after an exhaustive search of the solution space.

Implementing stop-skipping strategies in rolling horizons is a challenging problem because - unlike the bus holding, timetabling and other scheduling problems - stop-skipping is an NP-Hard, 0-1 problem which is computationally intractable in cases with many trips and stops, even if we explore a small fraction of the solution space. The main reason is the recursive relations in program (Q) that do not allow us to evaluate the performance of a candidate solution in scenarios with several stop-skipping candidates or several trips in the rolling horizon.

The only possibility to obtain a globally optimal solution within a reasonable time (i.e., less than 2 minutes) is to consider only one trip in the rolling horizon. This was also reported in the work of Fu et al. (*21*) which considered only a single trip in the dynamic stop-skipping problem. For two trips or more, there should be a compromise with respect to the number of stop-skipping candidates, since we cannot consider every stop as a potential stop to be skipped. Interestingly, even if we relax the requirement of obtaining a globally optimal solution by resorting to heuristics, the scalability issues persist and the periodic stop-skipping problem remains intractable for instances with several trips and stop-skipping candidates. In smaller instances where the stop-skipping problem is still tractable, employing a heuristic can reduce the computational costs (i.e., if we have 2 trips in the rolling horizon a heuristic can compute an approximate solution in cases with up to 12 stop-skipping candidates, whereas an exhaustive search can compute a solution when the number of candidate stops is reduced to 7).

Limitations

To facilitate the reproducibility of our work, we explicitly state the main limitations that are associated with our model and the solution method(s):

- our model can be applied only in services where the passengers do not coordinate their arrivals at stops with the arrival times of the buses (that is, high frequency services);
- our work cannot be applied in bus line services with a significant number of overtakes among buses that operate in the same line;
- our model, and the periodic stop-skipping problem in general, cannot be solved to global optimality if the number of candidate stops for stop-skipping or the trips in the rolling horizon are high. Hence, a meticulous pre-selection of stop-skipping candidates is required and the bus operators should not plan the skipped stops of several trips in each periodic optimization.

With respect to the pre-selection of stop-skipping candidates, we note that restricting the number of stop-skipping candidates to reduce the computational costs is not necessarily a negative aspect. For instance, a pre-selection of stop-skipping candidates can have positive secondary effects because skipping too many stops can increase the inconvenience of passengers and affect the coordination of the bus operations at the network level.

Managerial Implications and Future Research Directions

Notwithstanding the potential benefits, this research suggests some important managerial implications. One obvious managerial implication is the proper communication of the skipped stops to passengers, bus drivers, and other stakeholders. Another managerial implications is the preselection of major stops as stop-skipping candidates. For this purpose, specific attention should be given to the network structure and potential transfer points among different bus lines to avoid missed connections because of skipped stops.

In future research, the proposed stop-skipping approach can be applied to other problems that consider the synchronization of services and the passenger transfers. Moreover, the secondary effects of a stop-skipping strategy to the implementation of the timetable and the passenger satisfaction can be a potential topic for future research.

REFERENCES

[1] Hadas, Y. and M. Shnaiderman, Public-transit frequency setting using minimum-cost approach with stochastic demand and travel time. *Transportation Research Part B: Methodological*, Vol. 46, No. 8, 2012, pp. 1068–1084.

[2] Gkiotsalitis, K. and O. Cats, Reliable frequency determination: Incorporating information on service uncertainty when setting dispatching headways. *Transportation Research Part C: Emerging Technologies*, Vol. 88, 2018, pp. 187–207.

[3] Chen, X., L. Yu, Y. Zhang, and J. Guo, Analyzing urban bus service reliability at the stop, route, and network levels. *Transportation research part A: policy and practice*, Vol. 43, No. 8, 2009, pp. 722–734.

[4] Daganzo, C. F., A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons. *Transportation Research Part B: Methodological*, Vol. 43, No. 10, 2009, pp. 913–921.

[5] Gkiotsalitis, K., O. A. Eikenbroek, and O. Cats, Robust network-wide bus scheduling with transfer synchronizations. *IEEE transactions on intelligent transportation systems*, 2019.

[6] Gkiotsalitis, K. and F. Alesiani, Robust timetable optimization for bus lines subject to resource and regulatory constraints. *Transportation Research Part E: Logistics and Transportation Review*, Vol. 128, 2019, pp. 30–51.

[7] Bartholdi III, J. J. and D. D. Eisenstein, A self-coördinating bus route to resist bus bunching. *Transportation Research Part B: Methodological*, Vol. 46, No. 4, 2012, pp. 481–491.

[8] Delgado, F., J. C. Munoz, and R. Giesen, How much can holding and/or limiting boarding improve transit performance? *Transportation Research Part B: Methodological*, Vol. 46, No. 9, 2012, pp. 1202–1217.

[9] Gkiotsalitis, K. and O. Cats, Multi-constrained bus holding control in time windows with branch and bound and alternating minimization. *Transportmetrica B: Transport Dynamics*, Vol. 7, No. 1, 2019, pp. 1258–1285.

[10] Liu, Z., Y. Yan, X. Qu, and Y. Zhang, Bus stop-skipping scheme with random travel time. *Transportation Research Part C: Emerging Technologies*, Vol. 35, 2013, pp. 46–56.

[11] Chen, X., B. Hellinga, C. Chang, and L. Fu, Optimization of headways with stop-skipping control: a case study of bus rapid transit system. *Journal of advanced transportation*, Vol. 49, No. 3, 2015, pp. 385–401.

[12] Gkiotsalitis, K., Robust Stop-Skipping at the Tactical Planning Stage with Evolutionary Optimization. *Transportation research record*, Vol. 2673, No. 3, 2019, pp. 611–623.

[13] Furth, P. G., Short turning on transit routes. *Transportation Research Record*, Vol. 1108, 1987, pp. 42–52.

[14] Cortés, C. E., S. Jara-Díaz, and A. Tirachini, Integrating short turning and deadheading in the optimization of transit services. *Transportation Research Part A: Policy and Practice*, Vol. 45, No. 5, 2011, pp. 419–434.

[15] Delle Site, P. and F. Filippi, Service optimization for bus corridors with short-turn strategies and variable vehicle size. *Transportation Research Part A: Policy and Practice*, Vol. 32, No. 1, 1998, pp. 19–38.

[16] Gkiotsalitis, K., Z. Wu, and O. Cats, A cost-minimization model for bus fleet allocation featuring the tactical generation of short-turning and interlining options. *Transportation Research Part C: Emerging Technologies*, Vol. 98, 2019, pp. 14–36.

[17] Gkiotsalitis, K., Bus rescheduling in rolling horizons for regularity-based services. *Journal of intelligent transportation systems*, 2019, pp. 1–20.

[18] Gkiotsalitis, K. and E. Van Berkum, An exact method for the bus dispatching problem in rolling horizons. *Transportation Research Part C: Emerging Technologies*, Vol. 110, 2020, pp. 143–165.

[19] Daganzo, C. F. and J. Pilachowski, Reducing bunching with bus-to-bus cooperation. *Transportation Research Part B: Methodological*, Vol. 45, No. 1, 2011, pp. 267–277.

[20] Muñoz, J. C., C. E. Cortés, R. Giesen, D. Sáez, F. Delgado, F. Valencia, and A. Cipriano, Comparison of dynamic control strategies for transit operations. *Transportation Research Part C: Emerging Technologies*, Vol. 28, 2013, pp. 101–113.

[21] Fu, L., Q. Liu, and P. Calamai, Real-time optimization model for dynamic scheduling of transit operations. *Transportation Research Record: Journal of the Transportation Research Board*, , No. 1857, 2003, pp. 48–55.

[22] Bostel, N., P. Dejax, P. Guez, and F. Tricoire, Multiperiod planning and routing on a rolling horizon for field force optimization logistics. In *The vehicle routing problem: latest advances and new challenges*, Springer, 2008, pp. 503–525.

[23] Jordan, W. C. and M. A. Turnquist, Zone scheduling of bus routes to improve service reliability. *Transportation science*, Vol. 13, No. 3, 1979, pp. 242–268.

[24] Furth, P. G., Zonal route design for transit corridors. *Transportation Science*, Vol. 20, No. 1, 1986, pp. 1–12.

[25] Furth, P. G. and F. B. Day, Transit routing and scheduling strategies for heavy-demand corridors (Abridgment). *Transportation Research Record*, , No. 1011, 1985.

[26] Jamili, A. and M. P. Aghaee, Robust stop-skipping patterns in urban railway operations under traffic alteration situation. *Transportation Research Part C: Emerging Technologies*, Vol. 61, 2015, pp. 63–74.

[27] Li, Y., M. Gendreau, et al., Real time scheduling on a transit bus route: a 0-1 stochastic programming model, 1991.

[28] Lin, G., P. Liang, P. Schonfeld, and R. Larson, *Adaptive control of transit operations, Final Report*. MD-26-7002. US Department of Transportation, 1995.

[29] Eberlein, X. J., Real-time control strategies in transit operations: Models and analysis. *Transportation Research Part A*, Vol. 1, No. 31, 1997, pp. 69–70.

[30] Sun, A. and M. Hickman, The real–time stop–skipping problem. *Journal of Intelligent Transportation Systems*, Vol. 9, No. 2, 2005, pp. 91–109.

[31] Eberlein, X. J., *Real-time control strategies in transit operations: Models and analysis*. Ph.D. thesis, Massachusetts Institute of Technology, Department of Civil and Environmental Engineering, 1995.

[32] Cortés, C. E., D. Sáez, F. Milla, A. Núñez, and M. Riquelme, Hybrid predictive control for real-time optimization of public transport systemsâĂŹ operations based on evolutionary multi-objective optimization. *Transportation Research Part C: Emerging Technologies*, Vol. 18, No. 5, 2010, pp. 757–769.

[33] Sáez, D., C. E. Cortés, F. Milla, A. Núñez, A. Tirachini, and M. Riquelme, Hybrid predictive control strategy for a public transport system with uncertain demand. *Transportmetrica*, Vol. 8, No. 1, 2012, pp. 61–86.

[34] Eberlein, X. J., N. H. Wilson, C. Barnhart, and D. Bernstein, The real-time deadheading problem in transit operations control. *Transportation Research Part B: Methodological*, Vol. 32, No. 2, 1998, pp. 77–100.

[35] Eberlein, X. J., N. H. Wilson, and D. Bernstein, The holding problem with real–time information available. *Transportation science*, Vol. 35, No. 1, 2001, pp. 1–18.

[36] Xuan, Y., J. Argote, and C. F. Daganzo, Dynamic bus holding strategies for schedule reliability: Optimal linear control and performance analysis. *Transportation Research Part B: Methodological*, Vol. 45, No. 10, 2011, pp. 1831–1845.

[37] Chen, Q., E. Adida, and J. Lin, Implementation of an iterative headway-based bus holding strategy with real-time information. *Public Transport*, Vol. 4, No. 3, 2013, pp. 165–186.

[38] Gkiotsalitis, K. and N. Maslekar, Towards transfer synchronization of regularity-based bus operations with sequential hill-climbing. *Public transport*, Vol. 10, No. 2, 2018, pp. 335–361.

[39] Welding, P., The instability of a close-interval service. *Journal of the operational research society*, Vol. 8, No. 3, 1957, pp. 133–142.

[40] Randall, E. R., B. J. Condry, M. Trompet, and S. Campus, International bus system benchmarking: Performance measurement development, challenges, and lessons learned. In *Transportation Research Board 86th Annual Meeting, 21st-25th january*, 2007.

[41] Karp, R. M., Reducibility among combinatorial problems. In *Complexity of computer computations*, Springer, 1972, pp. 85–103.

[42] Leong, W., K. Goh, S. Hess, and P. Murphy, Improving bus service reliability: The Singapore experience. *Research in Transportation Economics*, Vol. 59, 2016, pp. 40–49.