Robust Stop-Skipping at the Tactical Planning Stage with Evolutionary Optimization

Konstantinos Gkiotsalitis

Abstract
The planning of stop-skipping strategies based on the expected travel times of bus trips has a positive effect in practice only if the traffic conditions during the daily operations do not deviate significantly from those expected. For this reason, we propose a non-deterministic approach which considers the uncertainty of trip travel times and provides stop-skipping strategies which are robust to travel-time variations. In more detail, we show how historical travel-time observations can be integrated into a Genetic Algorithm (GA) that tries to compute a robust stop-skipping strategy for all daily trips of a bus line. The proposed mathematical program of robust stop-skipping at the tactical planning stage is solved using the minimax principle, whereas the GA implementation ensures that improved solutions can be obtained even for high-dimensional problems by avoiding the exhaustive exploration of the solution space. The proposed approach is validated with the use of five months of data from a circular bus line in Singapore demonstrating an improved performance of more than 10% in worst-case scenarios which encourages further investigation of the robust stop-skipping strategy.

The planning of bus operations can be at the strategic, the tactical, or the operational level. Strategic planning determines the layouts of the bus routes and the locations of the bus stops to establish an optimal trade-off between the operators’ and users’ costs (Ibarra-Rojas et al. (1)). Tactical planning comprises the frequency settings stage (Hadad and Shnaiderman (2), Gkiotsalitis and Cats (3)), the timetable design stage (Sun et al. (4), Gkiotsalitis and Kumar (5)) and the vehicle/crew scheduling stage (Boyer et al. (6)). Finally, operational planning focuses on determining corrective actions in a dynamic environment for improving the performance of the bus operations. For instance, bus holding (Newell, Hernández et al., Wu et al. (7–9)), dispatch time control (Hickman (10), Gkiotsalitis and Stathopoulos (11)), stop-skipping (Sun and Hickman, Yu et al., Chen et al. (12–14)) and short-turning (Zhang et al. (15)) are some of the most commonly used corrective actions employed at the operational planning stage.

Although the holding times of buses at intermediate bus stops are decided during the operational stage, decisions regarding stop-skipping and short-turning can be undertaken at the tactical planning stage (Sidi et al. (16)). Deciding which stops should be skipped by a bus trip is a combinatorial problem. As a combinatorial problem, it is NP-Hard prohibiting the computation of a globally optimal solution in polynomial time. For instance, if one bus line has \( N = \{1, \ldots, n, \ldots, |N|\} \) daily trips that serve a set of bus stops \( S = \{1, \ldots, s, \ldots, |S|\} \), then deciding which stops should be skipped by each bus trip \( n \in N \) requires searching a solution space with \( 2^{|S|} \times |N| \) options. This is documented in the recent work of Liu et al. (17). The intuitive proof regarding the size of the solution space is the following: “The decision variable of serving or skipping a bus stop can take two values (skip or serve). For deciding which bus stops should be skipped by a bus trip \( n \in N \), a total number of \( 2^{|S|} \) combinations should be evaluated where \( |S| \) is the number of bus stops. If this decision should be made for all trips of the day at the tactical planning stage, then the number of possible combinations rises to \( 2^{|S|} \times |N| \).”

Because of the exponential growth rate of the solution space, it is not expected to be able to explore the stop-skipping combination options of more than a few daily
trips with the use of exact optimization methods. Therefore, finding a globally optimal solution regarding the stop-skippings of the daily trips of a high-frequency bus line (which typically range from 80 to 300 (Gschwender et al., Gkiotsalitis and Cats, Gkiotsalitis and Maslekar (18–20)) cannot be guaranteed. In addition, for computing an optimal stop-skipping strategy for the daily trips of a bus line, the travel times of the daily trips should be provided as input. Therefore, if one wants to experiment with different travel-time inputs and generate different stop-skipping strategies for studying the performance of the stop-skipping strategies under different travel-time realizations, then the above-mentioned solution space should be explored repeatedly (i.e., dozens or hundreds of times).

This motivates our work; instead of using exact optimization methods that cannot scale, we use a metaheuristic GA that may provide a sufficiently good solution to the stop-skipping problem by exploring a targeted portion of the solution space. In addition, we couple linear programming with the GA metaheuristic allowing us to discover the worst-possible performances of population members of the GA by performing several experiments with different travel-time disturbances. This enhances the solution-space search of the GA which evolves its population by recombining population members that are more robust to travel-time disturbances (each population member of the GA metaheuristic represents a potential stop-skipping strategy for all daily trips of a bus line as presented in Figure 1).

The main contributions of this paper are:

- the adaptation of the model of Fu et al. (21) which focused on dynamic stop-skipping, to the tactical planning of stop-skipping strategies where the skipped stops by all daily trips of a bus line are determined before the beginning of the daily operations;
- the addition of the robust aspect to the stop-skipping problem for favoring the computation of stop-skipping strategies that are robust to travel-time disturbances;
- the adaptation of a metaheuristic GA algorithm to use observed travel-time disturbances for deriving worst-case performance estimates for each one of its population members; and
- the investigation of the performance of robust stop-skipping solutions in common-case and worst-case scenarios.

Related Works

The decision variables of the stop-skipping problem can take two values (1 if a stop is served and 0 if a stop is skipped) resulting in a binary, 0-1 optimization problem. Previous works have addressed the stop-skipping problem as a dynamic control problem where the skipped stops of a bus trip are decided at the time when the trip starts its service (Fu et al., Li et al., Lin et al., Eberlein (21–24)). By deciding independently the stop-skippings of each daily trip, the potential stop-skipping combinations are significantly reduced from $2^{|S|} \times |M|$ to $2^{|S|}$. This permits the computation of the globally optimal stop-skipping solution with exhaustive search methods in the case of dynamic stop-skipping since the computational cost of such a small-scale problem can be in the range of minutes.

![Figure 1. The metaheuristic GA has several φ population members. Each gene of a population member can take the value 1 if the corresponding trip is planned to serve the corresponding stop or 0 otherwise. Each population member has $|N| \times |S|$ genes and represents a daily stop-skipping strategy.](image)
Indeed, several works have solved the dynamic stop-skipping problem using exhaustive search methods (see Sun and Hickman, Fu et al. (12, 21)). Nevertheless, such approaches are reactionary and treat each bus trip independently. Thus, computing the optimal stop-skipping policy for one trip that is about to start its service might affect negatively the performance of the future operations because all bus trips are interconnected. In addition, another reported disadvantage of the dynamic stop-skipping is that passengers are not informed in advance about the stops that will be skipped and might have to wait for at least another time headway to board a bus.

Specifically, Sun and Hickman (12) formulated a real-time stop-skipping strategy for reacting to disruptions that can occur in the middle of a route as a nonlinear integer programming problem. Their formulation included assumptions of random distributions of passenger boardings and alightings and their solution method was an exhaustive search. Fu et al. (21) relaxed the dynamic stop-skipping problem by arguing that if a bus trip skips one stop, then the next bus trip has to serve the entire route in order to guarantee a waiting time of at most two headways for passengers waiting at skipped stations. Similarly to the work of Sun and Hickman (12), Fu et al. (21) performed a simulation-based sensitivity analysis for analyzing the effect of dynamic stop-skipping to the passenger-related and operational-related costs.

In the PhD thesis of Eberlein (25), a simplified transit operation environment was developed for deriving stop-skipping solutions analytically. This thesis and the later published work of Eberlein et al. (26) also studied: (a) the combination of dynamic stop-skipping and bus holding using data from the Massachusetts Bay Transportation Authority (MBTA) Green Line; and (b) the degradation of the optimal solution in the presence of significant stochastic disturbances in the travel times and the headways patterns.

Dynamic stop-skipping has also been combined with short-turning control. Li et al. (27) considered a real-time scheduling problem where buses may skip many bus stops or turn before the terminus in order to respect the schedule. Given the increased complexity, Li et al. (27) developed heuristic procedures for providing quick solutions to the real-time stop-skipping/short-turning problem and evaluated the quality of the results using data from Shanghai.

Past works have also combined the dynamic stop-skipping with the bus holding problem (Lin et al., Eberlein, Cortés et al., Sáez et al. (23, 25, 28, 29)). However, including decisions related to the holding of buses at stops increases the complexity of the problem and expands disproportionally the size of the solution space. Cortés et al. (28) developed a state-space model including the bus position and the expected load and arrival time at stops. The control decisions were applied at discrete events (i.e., when a bus arrives at a bus stop) and given the complexity of the joint dynamic stop-skipping and bus holding problem, a GA-based multi-objective optimization solution method was employed. Sáez et al. (29) integrated also the two aforementioned strategies (dynamic stop-skipping and holding) and solved the real-time public transport control problem with uncertain passenger demand by formulating it as a hybrid predictive control (HPC) problem.

Given that the real-time control strategies affect some passengers in a negative way (i.e., holding passengers inside a bus, preventing them from boarding), the stop-skipping (also known as expressing) has also been studied at the tactical planning level by determining pre-planned stop-skipping strategies for bus lines (i.e., Jordan and Turnquist, Furth (30, 31)). Pre-planned deadheading and stop-skipping have been studied by a limited number of authors (see Liu et al. (17)). Furth and Day (32) and Furth (31) 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. More recently, Delle Site and Filippi (33) proposed an intermediate-level planning of bus operations where short-turnings are pre-planned given the demand patterns over different operational periods, though without analyzing the stop-skipping problem. A similar approach was also proposed by Gkiotsalitis et al. (34) who introduced the concept of virtual lines for pre-planned short-turnings and interlinings.

One observation and one research gap are identified from the previous studies. The observation is that the stop-skipping problem has been mainly addressed at the operational level and limited attention has been given to the pre-planned stop-skipping case. Elaborating on the research gap, most of the stop-skipping optimization models in the above literature review assumed deterministic travel times and constant headways. An exception is the work of Liu et al. (17) which considered travel times as random variables that follow a normal distribution with a constant mean and variance that can be calibrated with the use of automatic vehicle location data. In contrast to the work of Liu et al. (17), this work does not sample travel-time values following a probability distribution and the proposed robust stop-skipping method considers any travel-time value within a pre-defined set. The advantage of this approach is that we can produce a resilient stop-skipping strategy even if the travel times do not follow a specific probability distribution or many links follow distinctly different probability distributions. Compared with the existing works, we show how linear...
programming can be integrated into a metaheuristic search for generating robust pre-planned stop-skipping policies for individual bus lines by using the travel-time variability as a direct input to our optimization problem. Further, we show that incorporating past observations into metaheuristic optimization methods can benefit the robustness to the travel-time variations and provide a balanced performance in both common-case and worst-case scenarios.

**Model Formulation**

**Assumptions and Nomenclature**

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

1. Buses that serve the same line do not overtake each other. This is a common assumption in related works (see Xuan et al., Chen et al., Gkiotsalitis and Masek (35–37)).
2. Passenger arrivals at stops are random because passengers cannot coordinate their arrivals with the arrival times of buses at high-frequency services (Welding, Randall et al. (38, 39)).
3. Passenger demand at skipped stops can be accommodated by the next bus trip of the same line.
4. Passengers use different door channels for boardings and alightings.

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

| N | set of bus trips, $N = \{1, \ldots, n, \ldots, |N|\}$ |
| S | set of bus stops, $S = \{1, \ldots, s, \ldots, |S|\}$ |
| T | matrix of running times where $t_{n,s} \in T$ is the running time of the $n$th trip between stop $s-1$ and $s$ where $s \in S \setminus \{1\}$ |
| $\tau$ | vector of free-flow running times $\tau = (\tau_2, \ldots, \tau_{|S|})$ where $\tau_s$ is the free-flow running time between stop $s-1$ and $s$ where $s \in S \setminus \{1\}$ |
| D | matrix of departure times where $d_{n,s} \in D$ is the departure time of trip $n$ from stop $s$ where $n \in N$ and $s \in S$ |
| A | matrix of arrival times where $a_{n,s} \in A$ is the arrival time of trip $n$ at stop $s$ where $n \in N$ and $s \in S$ |
| K | matrix of dwell times where $k_{n,s} \in K$ is the dwell time of trip $n$ at stop $s$ where $n \in N$ and $s \in S$ |
| H | matrix of bus headways times where $h_{n,s} \in H$ is the headway between trips $n-1$ and $n$ at stop $s$ where $n \in N \setminus \{1\}$ and $s \in S$ |
| W | matrix where each $w_{n,yr} \in W$ denotes the number of passengers waiting for bus $n$ and traveling from stop $s$ to $y$ (note: $w_{n,yr} = 0$, $\forall y \leq s$) |
| L | matrix where each $l_{n,yr} \in L$ denotes the number of passengers traveling from stop $s$ to $y$ skipped by bus $n$ (note: $l_{n,yr} = 0$, $\forall y \leq s$) |
| M | matrix where each $m_{n,s,k} \in M$ denotes the number of passengers at stop $s$ skipped by bus $n$ where $n \in N, s \in S$ (note: $m_{n,s} = \sum_{i=s+1}^{\infty} l_{n,i}$) |
| U | matrix where each $u_{n,s} \in U$ denotes the number of passengers boarding bus $n$ at stop $s$ where $n \in N, s \in S$ (note: $u_{n,s} = 0$, $\forall n \in N$) |
| B | matrix where each $b_{n,yr} \in B$ denotes the number of passengers boarding bus $n$ at stop $s$ whose destination is stop $y$ (note: $b_{n,yr} = 0$, $\forall y \leq s$) |
| V | matrix where each $v_{n,s} \in V$ denotes the number of passengers alighting bus $n$ at stop $s$ where $n \in N, s \in S$ (note: $v_{n,s} = 0$, $\forall n \in N$) |
| r1 | average boarding time per passenger, a constant |
| $\delta$ | average alighting time per passenger, a constant |
| $\lambda_{yr}$ | average bus acceleration plus deceleration time for serving a bus stop, a constant |
| $c_1$ | unit time value associated with the passenger waiting times ($$/hour$$) |
| $c_2$ | unit time value associated with the passenger in-vehicle travel time ($$/hour$$) |
| $c_3$ | unit time value associated with vehicle operation time ($$/hour$$) |
| $x$ | $|N|\times|S|$-dimensional matrix of the decision variables where each $x_{n,s} \in x$ can take a binary value $\{0,1\}$ where $x_{n,s} = 1$ denotes that the $n$th bus will serve stop $s$ and $x_{n,s} = 0$ denotes that the $n$th trip will skip this stop |
Vehicle Movement Model

For formulating the state-space vehicle movement model we extend the formulation of Fu et al. (21) which was focused on the dynamic stop-skipping problem. In more detail, our extended formulation differs from Fu et al. (21) in the following aspects: (i) it considers all daily trips of a bus line because it focuses on tactical planning; and (ii) it considers uncertain trip travel times (robust stop-skipping).

Compared with the dynamic stop-skipping, pre-planned stop-skipping can offer additional advantages. First, the passengers can be informed about what they should expect before using the service. Furthermore, stop-skipping information can be displayed at a bus stop if an electronic device is installed at the stop or it can be announced by bus drivers so that passengers whose destinations will be skipped do not board the wrong bus. The pre-planned stop-skipping might therefore cause much less confusion than the dynamic stop-skipping strategies.

In the vehicle movement model, the arrival time of bus trips $n$ 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 time lost in acceleration and deceleration:

$$a_{n,s} = d_{n,s-1} + t_{n,s} + \frac{\delta}{2} v_{n,s-1} + \frac{\delta}{2} v_{n,s},$$  
\forall n \in N \setminus \{1\}, s \in S \setminus \{1\} \tag{1}$$

In addition, the departure time of bus trip $n$ from stop $s$ is equal to its arrival time plus the dwell time $k_{n,s}$:

$$d_{n,s} = a_{n,s} + k_{n,s}, \forall n \in N \setminus \{1\}, s \in S \setminus \{1\} \tag{2}$$

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

$$h_{n,s} = d_{n,s} - d_{n-1,s}, \forall n \in N \setminus \{1\}, s \in S \tag{3}$$

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:

$$k_{n,s} = r_1 u_{n,s} + r_2 v_{n,s}, \forall n \in N \setminus \{1\}, s \in S \setminus \{1\} \tag{4}$$

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

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 $y$:

$$u_{n,s} = x_{n,s} \sum_{y=s+1}^{|S|} w_{n,s,y} x_{n,y}, \forall n \in N \setminus \{1\}, s \in S \setminus \{|S|\} \tag{5}$$

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

$$b_{n,s,y} = x_{n,s} w_{n,s,y} x_{n,y}, \forall n \in N \setminus \{1\}, 1 \leq s < y \leq |S| \tag{6}$$

The expected number of alighting passengers for bus trip $n$ at stop $s$ (assuming bus $n$ stops at stop $s$) depends on the number of passengers traveling between stops $y$ and $s(y<s)$ and whether the bus will make stop $y$:

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

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 $s$, $l_{n-1,s,y}$, and the average number of passengers who arrive at stop $s$ after bus $n-1$ leaves stop $s$:

$$w_{n,s,y} = l_{n-1,s,y} + \lambda_{s,y} h_{n,s}, \forall n \in N \setminus \{1\}, s \in S \tag{8}$$

The number of passengers destined for stop $y$ who are stranded by bus $n-1$ at stop $s$, $l_{n-1,s,y}$, will be $0$ if bus $n-1$ stops at stops $s$ and $y$ but, otherwise, will equal the number of passengers waiting for bus $n-1$ at stop $s$ who have stop $y$ as their destination:

$$l_{n,s,y} = w_{n,s,y} - w_{n,s,y} x_{n,y}, \forall n \in N \setminus \{1\}, s \in S \tag{9}$$

(Here we assume that passengers waiting to board at stop $s$ always wait for the next trip, $n+1$, if trip $n$ skips stop $s$ or the stop of their destination.)

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

$$m_{n,s} = \sum_{y=s+1}^{|S|} l_{n,s,y}, \forall n \in N \setminus \{1\}, s \in S \tag{10}$$

Objective Function

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:
where the first term of the objective function includes two components. The first component, 

\((u_{n,s} - m_{n-1,s})(\frac{b_{n,s}}{2})\), computes the total waiting time of the passengers who arrive after the departure (or passing) of bus \(n - 1\) at stop \(s\), assuming random arrival with an average passenger waiting time equal to half the headway. The second component represents the total waiting time equal to half the headway of bus \(n - 1\) at stop \(s\) and have to wait for an average amount of time equal to \(m_{n-1,s} + h_{n,s}\). 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:

\[ f(x) := c_1 \sum_{n=1}^{N} \sum_{s=N-1}^{[S]} (u_{n,s} - m_{n-1,s})(\frac{b_{n,s}}{2}) + m_{n-1,s}(\frac{h_{n,s}}{2}) + h_{n,s,0} ) \]

\[ + c_2 \sum_{n=2}^{N} \sum_{s=1}^{[S]} \sum_{y=1}^{[S]} \sum_{z=N-1}^{[S]} [t_z + (k_{n,z} + \delta)x_{n,z}] \]

\[ + c_3 \sum_{n=2}^{N} \sum_{s=1}^{[S]} [t_s + (k_{n,s} + \delta)x_{n,s}] \]

(11)

For computing stop-skipping plans which are robust to the variability of the travel times, one should define the upper and lower limits of the uncertainty sets related to the travel times. Therefore, the following two matrices need to be defined:

\[ E^{\text{min}} \in \mathbb{R} + \times [N] \times [S] - 1 \]

which is a matrix where each element \(e_{n,s}^{\text{min}}\) denotes the lower bound of the travel time of trip \(n\) from stop \(s - 1\) to stop \(s\) where \(n \in N\) and \(s \in S \setminus \{1\}\).

\[ E^{\text{max}} \in \mathbb{R} + \times [N] \times [S] - 1 \]

which is a matrix where each element \(e_{n,s}^{\text{max}}\) denotes the upper bound of the travel time from of trip \(n\) from stop \(s - 1\) to stop \(s\) where \(n \in N\) and \(s \in S \setminus \{1\}\).

Each bus trip \(n \in N\) can take any travel-time value from the uncertainty set \([e_{n,s}^{\text{min}}, e_{n,s}^{\text{max}}] \) when traveling from stop \(s - 1\) to stop \(s\), \(\forall s \in S \setminus \{1\}, n \in N\). This uncertainty set \([e_{n,s}^{\text{min}}, e_{n,s}^{\text{max}}] \) can be defined for each daily trip \(n \in N\) using historical travel-time data from past operations. Although we have complete historical travel-time data for each link, the correlation between links is not considered. Even if it is very likely that the travel times of adjacent links are highly correlated, we allow the link travel times to attain any value within the pre-determined lower and upper bounds of the corresponding sets for investigating the performance of a robust service in worst-case, unexpected scenarios.

The values of the boundaries, \([e_{n,s}^{\text{min}}, e_{n,s}^{\text{max}}] \), control the level of robustness of the stop-skipping solution and, when their range increases, the solution is expected to be more robust to extreme travel-time values. But, at the same time, it is expected to exhibit a worse performance in common-case scenarios. Note that the lower bound of the travel time \(e_{n,s}\) for each trip \(n \in N\) cannot be lower than the free-flow travel time between stops \(s - 1, s\) denoted as \(\tau_{s}\) hence, \([e_{n,s}^{\text{min}}, e_{n,s}^{\text{max}}] \rightarrow \max\{\tau_{s}, e_{n,s}^{\text{min}}, e_{n,s}^{\text{max}}\}\).

Modeling the robust stop-skipping as a minimax problem where the objective is to find the stop-skipping plan for each trip \(n \in N\) which is robust to travel-time variations requires the modification of the mathematical program \((Q)\) that should be stated according to the following minimax model:

Formulating the Robust Stop-Skipping Problem

Given the uncertainty of the trip running times, one can apply robust optimization for calculating stop-skipping solutions which are robust to running time variations. Unlike deterministic stop-skipping approaches which cannot maintain their optimality when there are further disturbances along the route, robust stop-skipping can generate stop-skipping strategies which are resilient to real-time changes and still perform adequately in the presence of disruptions. The level of robustness can be typically defined in practice (i.e., in consultation with the bus operator) since solutions which are robust to extreme-case scenarios tend to perform poorly in the average case and vice versa.

In robust optimization, the uncertain parameters (in our case the trip travel times) can take any value within an uncertainty set (see Bertsimas et al. (40)) where the broader the range of this set, the higher the robustness to extreme disturbances. A robust optimization problem is typically addressed in practice by using a minimax decision rule where the objective is to find the solution with the minimum performance loss at worst-case scenarios (Wald (41)).
\[
\begin{align*}
(\hat{Q}) : \min_{x} \max_{t} f(x, t) := c_1 \sum_{n - 2 s = 1}^{[N]} \sum_{s = 1}^{[S]} \left[ (u_{n,s} - m_{n - 1,s}) \left( \frac{h_{n,s}}{2} \right) + m_{n - 1,s} \left( \frac{h_{n,s} - 1}{2} + h_{n,s} \right) \right] \\
+ c_2 \sum_{n - 2 s = 1}^{[N]} \sum_{y = 1}^{[S]} \sum_{z = s + 1}^{[Y]} \left[ b_{n, s} y \sum_{z = s + 1}^{[Y]} (t_{n,z} + (k_{n,z} + \delta) x_{n,s}) \right] \\
+ c_3 \sum_{n - 2 s = 2}^{[N]} \sum_{s = 1}^{[S]} (t_{n,s} + (k_{n,s} + \delta) x_{n,s}) \\
s.t. \text{Equations } 1-10, 14-16 \\
t_{n,s} \in [\max \{ r_s, e_{n,s}^{\min} \}, e_{n,s}^{\max}], \forall n, s \in S \setminus \{1\}
\end{align*}
\] (17)

In the mathematical program \((\hat{Q})\), each travel time \(t_{n,s}, n \in N, s \in S \setminus \{1\}\) becomes a decision variable that can take any value from the uncertainty set \([\max \{ r_s, e_{n,s}^{\min} \}, e_{n,s}^{\max}]\) with the objective to maximize the function \(f(x^0, t)\) for a specific solution \(x^0\).

### Solution Method

The minimax problem of \((\hat{Q})\) is a two-player game where player 1 chooses his/her strategy (i.e., finds the travel times \(t^0\) from the uncertainty sets that maximize the objective function \(f(x^0, t)\)) for a specific solution \(x^0\) and player 2 updates the stop-skipping solution by minimizing the objective function \(f(x, t^0)\) for a specific travel-time variability \(t^0\). In theory, this two-player game can continue iteratively but the resulting solutions will oscillate infinitely because every optimal stop-skipping solution corresponds to a specific, worst-case travel-time disturbance which is immediately updated (and therefore not valid) by the time the stop-skipping solution is computed.

### A Genetic Algorithm-Based Metaheuristic for Converging to a Robust Stop-Skipping Solution

Using a GA we can prevent the oscillation problem since the stop-skipping solutions that perform well under unknown travel-time variabilities will have higher priority to reproduce and evolve into next generations.

A typical GA contains several strings which form the problem population. Each string is a population member and represents a solution to the optimization problem (in our case, each population member is a stop-skipping strategy \(x^0, x^1, \ldots\)). Each string is also considered as a chromosome and the GA uses a population rather than a single chromosome during its search; something that allows a GA to explore several areas of the search space and avoid local optima (Bakirtzis et al. (42)).

A GA requires only the existence of a fitness function which can be evaluated. In the program \((\hat{Q})\), the fitness function of a population member \(x^0\) is the solution of the mathematical program:

\[
(\hat{Q}) : \max_{t} f(x^0, t) \\
s.t. t_{n,s} \in [\max \{ r_s, e_{n,s}^{\min} \}, e_{n,s}^{\max}], \forall n, s \in S \setminus \{1\}
\] (18)

which is an easy-to-solve, continuous linear program because: (i) the objective function is a sum of linear functions; and (ii) the decision variables (worst-case travel times for the stop-skipping strategy \(x^0\)) can take any real value from the uncertainty set \([\max \{ r_s, e_{n,s}^{\min} \}, e_{n,s}^{\max}]\), \(\forall n, s \in S \setminus \{1\}\).

To execute the GA, we need first to define the population size, which is a GA hyperparameter and can be part of the calibration stage. The chromosomes of the initial population \((x^0, x^1, \ldots)\) can be randomly generated and evaluated. The GA evolution comprises the steps of: (1) parent selection; (2) crossover; and (3) mutation that are executed iteratively. In the parent selection stage, the fittest individuals are selected and they are allowed to pass their genes to the next generation. At each parent selection, two chromosomes from the population are selected where chromosomes with better fitness values have a high probability of being selected for producing an offspring. This is achieved in our case by using the well-known roulette-wheel selection method (see Goldberg and Deb (43)).

In the crossover stage, each pair of parents is combined to produce new chromosomes that inherit parts of the genes which belonged to the parent chromosomes. Typically, the genes of the parent chromosomes are exchanged at a single point (the crossover point) for generating offspring via recombination. In the mutation stage, the GA explores new information that does not belong to the pair of parents that were used at the crossover stage. In our case, we allow a very small probability to revert to the stop-skipping option (0-1) which is encoded at each gene of the generated offspring. Note that each population member should satisfy the constraints of Equation 15. To achieve this, the crossover and mutation steps have restrictions which do not allow two consecutive trips \(n - 1, n\) to skip the same stop \(s \in S \setminus \{1, |S|\}\). For instance, if for a population member \(x^0\) we have \(x^0_{n, s} = 0\) and \(x^0_{n, s} = 1\), then a mutation that will revert the \(s^0_{n, s} = 1\) option into \(s^0_{n, s} = 0\) is not allowed.
After completing the three stages, a new generation is produced and the GA terminates when the maximum number of generations is reached or if no further improvement is observed. After the termination, the best chromosome represents the currently best solution as presented in Figure 2 and in the following compact algorithm:

Step 0: Select a population size $\phi$ and generate population members, $(x^0, x^1, ..., x^{\phi-1})$, where each population member represents a stop-skipping strategy for all trips of a bus line;

Step 1: For each population member $x^k \in (x^0, x^1, ..., x^{\phi-1})$ solve the linear program $(Q)$ and obtain the value of $f(x^k, t^k)$ where $t^k$ is the solution of $(Q)$;

Step 2: Select the fittest population members from the set $(x^0, x^1, ..., x^{\phi-1})$ with the lowest $f(x^k, t^k)$ values for reproduction using the roulette-wheel method;

Step 3: For each couple of population members that are selected for reproduction, perform the crossover and mutation steps;

Step 4: If the maximum number of population generations is reached or there are no further improvements, STOP;

Step 5: Else: Calculate the incumbent best solution $x^*$ for which $\max_t f(x^*, t) \leq \max_t f(x^k, t), \forall k \in (0, 1, ..., \phi - 1)$ and pass the population to the new generation. Then, go to Step 1.

Figure 2. Genetic algorithm for defining a robust stop-skipping strategy based on the minimax decision rule.
Numerical Experiments

Case Study Description

The case study is a high-frequency circular bus service in Singapore with $|N| = 132$ trips from 07:00 to 19:00 and $|S| = 22$ bus stops. The circular service covers 7.5 km with an average trip travel time of 37 min. The circular bus service is a feeder service covering residential blocks, schools, public amenities and connecting them to a Mass Rapid Transit station as presented in Figure 3. Detailed Automated Vehicle Location (AVL) and Automated Passenger Counting (APC) datasets are available for this line for a five-month period. The datasets contain a total number of 2,254 trips with complete information regarding arrival times at stops, link travel times, boardings, and alightings.

For illustration purposes, Figure 4 visualizes the upper and lower limits for the link travel times of the first five links as a function of the time of the day. These limits are used to form each uncertainty set, $[c_{e_{n,s}}^{\text{min}}, c_{e_{n,s}}^{\text{max}}]$, $\forall n \in N$, $\forall s \in S \setminus \{1\}$, of the link travel times where the lower and upper bounds of each set are defined using Gaussian processes (see Rasmussen and Williams (44)) on the five-month AVL data. Gaussian processes assume that the underlying process, in this case the travel time at each link, is Gaussian and has a correlation function which is given by a kernel function. In our experiment we use the GPy python library (GPy (45)) and RBF ((Gaussian) radial basis function) or square exponential kernel. Note that the range of the lower and upper bounds of the link travel times affects the level of robustness of the stop-skipping solution to the extreme cases. Given that a strategy which is robust to extreme scenarios is expected to have a deteriorated performance in the common-case scenarios with mild disturbances, the defined bounds in Figure 4 do not include travel-time disturbances that appear as outliers. This is also mentioned in other works such as Liu et al. (46) who suggested that using a broad lower/upper bound interval might be too conservative because the probability of the situation that the travel times at all links reach simultaneously their extreme values is very low.

Computation of the Robust Stop-Skipping Strategy

In this study we use the same weight factor values for the passenger waiting times, on-board passenger delays and bus trips travel times as in the work of Fu et al. (21)—namely, $c_1 = $20/hour, $c_2 = $10/hour and $c_3 = $50/hour.

The objective of the robust stop-skipping problem is to find a solution $x^*$ which performs best in the worst-case scenario of link travel times. In other words, the
optimal solution \( \mathbf{x}^* \) should be the one for which the resulting value of the maximization problem \( \max f(\mathbf{x}, t) \) is the lowest. In Figure 5, the result of the maximization problem \( \max f(\mathbf{x}, t) \) for the incumbent elite solution \( z \) of each population generation \( z \in \{1, 2, \ldots \} \) of the GA is presented in the y-axis. Note that during the initialization, we select as the elite solution of the first generation \( (z = 1) \) a solution that does not perform any stop-skippings. The algorithm is coded in Python 2.7 and implemented on a conventional computer with Intel Core i7-7700HQ CPU @ 2.80GHz and 16 GB RAM resulting in a total running time of 4 min and 27 s.

From Figure 5 one can note that at the first iterations the elite solutions of the GA oscillate significantly. However, as the population generations increase, the population of the new generations becomes more homogeneous and all population members have a good performance in scenarios of worst-case travel-time disturbance (thus, avoiding oscillations). This is evident after the 50th population generation where the incumbent best solutions do not improve any further (at least significantly) and their performances stabilize. It should be noted here that the GA does not guarantee global optimality, but an improvement with respect to the initial solution guess.

In Table 1 we summarize the performance of the robust solution \( \mathbf{x}^* \) at the worst-case scenario of travel-time disturbances. The performance of this solution in terms of average waiting times of passengers at stops, average in-vehicle times of on-board passengers and average bus trip times is compared against the performance of the solution of the 1st population generation \( (z = 1) \) in which no stop-skipping was applied.

Table 1 demonstrates a significant improvement potential in both the average waiting time and bus trip travel times at the worst-case scenario of travel-time disturbances. This is an expected outcome because our robust solution is appropriate for scenarios with disturbances. A more meaningful evaluation of the robust stop-skipping strategy can be derived when applying the robust strategy in realistic daily operations where only a few link travel times might vary significantly from their expected values. This evaluation is performed in the next section and the results are summarized in the discussion.

**Comparative Analysis of the Robust and the Average-Case Stop-Skipping Strategy in Realistic Scenarios**

As previously discussed, a robust stop-skipping strategy to an extreme travel-time disturbance level might affect negatively the performance of the service in a typical day with mild disturbances. For this reason, we compute the optimal stop-skipping strategy for the average case by:

- using the average link travel time values from the five-month AVL data which are denoted with the red line in Figure 4; and
- solving the mathematical program \( (Q) \) by assuming the link travel-time values as problem parameters (their values are set equal to the average link travel-time values in Figure 4).

This yields another stop-skipping solution, \( \mathbf{x}' \), that performs best when all link travel-time values on a day of operations are equal to their expected (average) values. If we consider two deterministic travel-time scenarios (one where the link travel times of the daily trips are always equal to the expected values and another where they take the values of the worst-possible disturbance within the uncertainty sets presented in Figure 4), then the average-case stop-skipping solution, \( \mathbf{x}' \), will perform better at the former scenario and the robust solution, \( \mathbf{x}^* \), to the uncertainty sets of Figure 4 will perform better at the latter.

Because of that, we validate the performance of the robust solution \( \mathbf{x}^* \) in different scenarios using real travel-time data from the five-month AVL dataset. For each day of the five months, we apply the average-case stop-skipping strategy \( \mathbf{x}' \) and the robust stop-skipping strategy \( \mathbf{x}^* \) and calculate the performance of the average
Table 2. Validation Results of Performance Improvement When Comparing the Robust Stop-Skipping Solution Case Against the Optimal One for the Average-case Scenario

<table>
<thead>
<tr>
<th></th>
<th>Average waiting time</th>
<th>Average in-vehicle time</th>
<th>Average bus trip time</th>
<th>Weighted total</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximum</td>
<td>32.10%</td>
<td>6.20%</td>
<td>7.10%</td>
<td>13.24%</td>
</tr>
<tr>
<td>Q3</td>
<td>14.50%</td>
<td>1.70%</td>
<td>4.60%</td>
<td>6.71%</td>
</tr>
<tr>
<td>Q2 (median)</td>
<td>4.70%</td>
<td>–0.20%</td>
<td>2.80%</td>
<td>2.90%</td>
</tr>
<tr>
<td>Q1</td>
<td>4.30%</td>
<td>0.60%</td>
<td>2.70%</td>
<td>2.84%</td>
</tr>
<tr>
<td>minimum</td>
<td>9.12%</td>
<td>1.70%</td>
<td>2.80%</td>
<td>4.24%</td>
</tr>
</tbody>
</table>

Discussion

Summarizing the results from Table 2, applying a robust stop-skipping solution can yield an overall performance improvement of 13.24% in actual operations which are not close to the average case. In contrast, for daily operations which are close to the average case (i.e., average expected day) the robust strategy does not offer a significant improvement (only 2.9%). A logical explanation behind this is that the robust strategy does not try to improve common-case scenarios and does not have a competitive advantage over deterministic stop-skipping strategies that do not consider the travel-time variability. For this reason, bus operators should be aware that there might even be a case where their performance will deteriorate slightly in common-case scenarios when they strive to improve operations in the presence of strong disruptions.

Concluding Remarks

This study investigated the problem of determining stop-skipping strategies at the tactical planning stage which are robust to travel-time variations. Expanding the mathematical program of Fu et al. (21) which used deterministic travel times for computing only the stop-skipping options of the bus that is about to be dispatched (dynamic stop-skipping), we focused on the determination of the optimal stop-skipping options of all daily trips that yields an NP-Hard problem which cannot be solved to global optimality. In addition, we modified the mathematical program of Fu et al. (21) so that it can incorporate uncertain travel-time parameters which can take values from uncertainty sets.

The resulting mathematical program proposed in this work is solved with a problem-specific GA that solves numerous linear programming problems internally for evaluating the fitness of each population member at the presence of worst-case travel-time disturbances. The proposed solution approach uses the minimax decision rule to find stop-skipping strategies with better performance in the presence of worst-case travel-time disturbances.

In contrast to the dynamic stop-skipping, stop-skipping at the tactical planning stage cannot adjust to the travel-time disturbances during the actual operations. For this reason, determining robust strategies that perform well in both common-case and worst-case scenarios becomes more important. This aspect was investigated using a five-month dataset and comparing the performance of a robust solution with a set of observed travel-time disturbances against the performance of the average-case stop-skipping solution. The analysis was based on the reasonable assumption that the link travel times of trips are not significantly affected by the stop-skipping strategies because most disturbances are due to road traffic. This analysis showed that robust stop-skipping strategies perform very close to the optimal solutions that consider the average values of the link travel times in most of the cases. This notwithstanding, applying a robust stop-skipping solution can yield a performance improvement of more than 10% in actual operations which are not close to the average case (for such operations, one can expect an improvement in the range of 13.24% according to the results of Table 2).

In future research, one could examine the performance of robust stop-skipping strategies, when the boundaries of the uncertainty sets vary, in order to find a set of solutions that perform well in both worst-case and common-case scenarios. In addition, the effect of stop-skipping to the load levels of the buses can be examined by considering the bus capacity and the optimal bus occupancy. Finally, the impact of pre-planned stop-skipping strategies to the passengers' trips can be considered as an additional, passenger-based key performance indicator in a.

passenger waiting times, average in-vehicle times and average bus trip times assuming that the observed link travel times of trips will remain the same when applying stop-skipping (a reasonable assumption because, in contrast to the dwell times that are affected by stop-skipping, the link travel times are affected only by the road traffic and other exogenous factors such as the driving behavior of the bus driver).

For each one of the daily travel-time realizations that are based on real data observations, we evaluate the performance of the average case and the robust stop-skipping strategy. After performing this procedure for all days, the results are presented in Table 2 using the Tukey boxplot convention where: (i) $Q_1$ is the middle number between the smallest number and the median of the data set (1st quartile); (ii) $Q_2$ is the median of the data set; (iii) $Q_3$ is the third quartile; (iv) the “minimum” is the lowest datum still within 1.5 of the interquartile range (IQR) of the first quartile; and (v) the “maximum” is the highest datum still within 1.5 IQR of the third quartile.
broader objective function to avoid reducing the quality of service. This passenger-based key performance indicator can be modeled as a utility function that considers not just the extra waiting times of passengers due to a skipped stop, but also the potential loss of their trip connections and other passenger-related factors.

References


The Standing Committee on Bus Transit Systems (AP050) peer-reviewed this paper (19-05489).