

# DEPART: Dynamic Route Planning in Stochastic Time-Dependent Public Transit Networks

Peng Ni<sup>1,2</sup>, Hoang Tam Vo, Daniel Dahlmeier  
<sup>1</sup>SAP Research & Innovation

Wentong Cai  
<sup>2</sup>Nanyang Technological University

Jordan Ivanchev, Heiko Ayt  
TUM CREATE

**Abstract**—While providing intelligent urban transportation services for commuters is one of the key enablers for developing smart cities, existing route planners mainly rely on static schedules and hence fall short in dealing with uncertain and time-dependent traffic situations. In this paper, by leveraging a large set of historical travel smart card data, we propose a method to build a stochastic time-dependent model for public transit networks. In addition, we develop DEPART<sup>1</sup>—a dynamic route planner that takes the stochastic models of both bus travel time and waiting time into account and optimizes both the speediness and reliability of routes. Experiments on real bus data set for the entire city confirm the quality and accuracy of the routes returned by DEPART in comparison to state-of-the-art route planners.

## I. INTRODUCTION

Intelligent urban transportation systems are becoming increasingly important for commuters in smart cities. Such systems enable city-dwellers to quickly identify an optimal route in a transit network between specified origin and destination when planning daily trips. Route planning is fundamental to intelligent transportation systems, and in fact, the problem of finding shortest paths in road and public transit networks has been extensively studied. Specifically, several techniques based on the classical Dijkstra algorithm have been proposed in recent decades to speed up system response time [1].

The underlying assumption of traditional route planners is that means of public transportation such as buses follow a fixed schedule. However, such an assumption does not really hold in a realistic transit network where buses' arrival times are not as accurate as scheduled since they are largely dependent on real traffic situations. In fact, the travel time in an urban traffic environment is highly stochastic and time-dependent. Hence, the results returned by static route planners are often inadequate in real world and cause user dissatisfaction.

**Motivation.** We take Singapore's bus network as an example. The bus network is the backbone of public transportation in Singapore and accounts for sixty percent of the total public transportation trips [17]. Existing route planners for Singapore, such as Google Maps and Gothere.sg, have two major drawbacks. Firstly, although users can specify their desired departure time, the query result is the same no matter whether the departure time falls in peak or off-peak periods. As will be shown in our experiments using real smart card data collected from Singapore's bus network in a period of three months, the same journey takes 20% longer on average during weekdays' peak hours than off-peak hours and weekends due to traffic congestion. Secondly, the travel time estimated by

these route planners is not accurate. For example, Google Maps returns a travel time of 30 minutes for a journey consisting of 30 bus stops. In practice, the journey takes at least 50 minutes based on the personal experience of some of the authors due to the time for passengers boarding and alighting at each stop.

State-of-the-art stochastic time-dependent route planning solutions are generally proposed for road networks where travel time within a time interval is modeled as a discrete variable with a probability mass function [18], [25], [9]. Other researches in public transportation focus on stochastic bus waiting time and assume that once commuters get on the bus, the travel time is deterministic [5], [2]. Besides, various bus running models are proposed to provide estimations for bus waiting time and travel time [12]. Nevertheless, none of them aims to design and develop a complete route planning system in which both expected bus waiting time and travel time are considered to recommend optimal routes for users.

**Challenges.** To provide a practical route planning system that can effectively deal with uncertain and time-dependent characteristics of urban traffic, several challenges need to be addressed. Firstly, the stochastic model of bus travel time has to be carefully devised so that transit networks ensure first-in-first-out (FIFO) property, i.e., buses do not overtake each other. Unlike road networks where drivers on private cars can speed at their own discretion and departing early does not guarantee an early arrival, in public transit networks a commuter who gets on an earlier bus usually arrives early too. It is noteworthy that a network without FIFO property may not have optimal substructures, i.e., the concatenation of the shortest paths from A to B and from B to C is not necessarily the shortest path from A to C. In fact, it is shown that there is no optimal substructure in a general stochastic time-dependent network (STD network) [11].

Secondly, it is challenging to properly incorporate the reliability of a route with its speediness, i.e., expected travel time. Most of research works done in this area use a mean-risk model [15], [21], [19] which combines the mean and variance of the travel time as a single linear objective function that needs to be optimized. Nevertheless, choosing appropriate coefficients for such a linear objective function is quite tricky and often heuristically done via experiments.

Thirdly, the routing algorithm needs to take into account that the travel time between two bus stops is not equal to the sum of travel times of each pair of consecutive stops between them. More specifically, given a bus line passing stops S1, S2 and S3 consecutively, the sum of the travel times from S1 to S2 and from S2 to S3 is smaller than the travel time of S1 to S3 due to the time taken for passengers boarding

<sup>1</sup>DEPART: Dynamic routE PIANning tRansit neTworks

and alighting at the middle stop S2. When the bus travels a long journey, the aggregate error in travel time estimation grows even bigger. In other words, the bus travel time lacks of component aggregation property.

**Contributions.** In order to address the aforementioned challenges, we build a stochastic time-dependent model with FIFO property for a bus network based on the collected travel smart card data, and propose an algorithm that can deal with the aggregated error of travel time and incorporate the reliability of a route as part of the output. The main contributions of this paper are as follows:

- To the best of our knowledge, this is the first work to provide a practical solution to route planning problem in a stochastic time-dependent public transit network, and take both travel time and waiting time into account.
- We propose a method to build a stochastic time-dependent transit network that enforces FIFO property where the travel time is modeled as a time-dependent continuous distribution function.
- We introduce a new algorithm that solves the lack of component aggregation property of bus travel time and optimize both the speediness and reliability of the returned routes.
- We develop *DEPART* – a dynamic route planner and evaluate the system with the real bus network of the entire city. The results confirm the quality and accuracy of its returned routes in comparison to other state-of-the-art route planners.

The remainder of the paper is organized as follows. In Section II, we present an overview of our proposed dynamic route planner. We introduce a model for FIFO stochastic time-dependent public transit networks in Section III. In Section IV, we describe our algorithm which is specially designed to work with stochastic time-dependent networks. We evaluate the system in Section V and provide a literature review in Section VI. Finally, we conclude the paper in Section VII.

## II. SYSTEM OVERVIEW

In this section, we present an overview of *DEPART* – our proposed solution for dynamic route planning in time-dependent transit networks which leverages historical travel smart card data to recommend routes that are more adapted to traffic situations. Figure 1 depicts the overall architecture of the system.

The system incorporates both dynamic and static data sources. The former includes historical travel smart card data from which information is extracted to build stochastic models for the bus travel time and waiting time in the form of probability distributions (see Section III). The latter includes information of bus stops and static schedule of bus lines. These public transportation information are used to create a time-dependent graph representing the entire bus network (see Section IV-A). The stochastic models of bus travel time and waiting time are associated as costs of the edges in the time-dependent transportation graph.

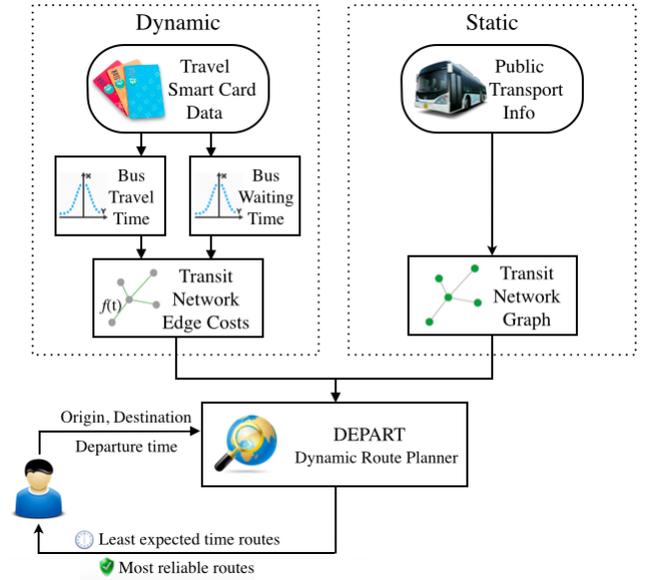


Fig. 1. Architecture of *DEPART* – a dynamic route planning system.

The created stochastic time-dependent graph constitutes the core data structure in our routing engine. Given a query submitted by a user which basically consists of origin, destination, and departure time, the system runs an optimization algorithm to find routes with the least expected travel time and the highest reliability (see Section IV-B).

## III. MODELING A FIFO STOCHASTIC TIME-DEPENDENT PUBLIC TRANSIT NETWORK

In this section, we first describe the collected travel smart card data. Then, we present our method to extract information from those data and build basic stochastic time-dependent models for bus travel times and waiting times. Finally, we propose modifications to the basic models in order to enforce FIFO property in a transit network.

### A. Travel Smart Card Data

The public transportation system in Singapore is fully integrated with travel smart card. The collected data comprise of the recorded usage of all bus lines in Singapore for three months in 2011. Every row in the data set consists of a recorded trip of a commuter. The trip starts when the smart card is used to tap in the vehicle and ends when the passenger taps out. The format of a trip record is the following: [ Bus Line, Start Station, End Station, Boarding Time, Alighting Time, Trip Distance, Trip Date ].

### B. Information Extraction From Travel Smart Card Data

The key information that we need to extract from the data is the estimated travel time and its variance between any two stops of a bus line during each time interval of a day. Another piece of information that we would need is the estimated waiting time for the passengers at the bus stop until the desired bus arrives. The information from the smart card records only points out when the passenger boarded the bus at a bus stop

and not when the passenger actually was at that bus stop, it is therefore impossible to extract the waiting time information directly from the data. Instead, we estimate these two sets of information as follows.

In order to get the first set of information (i.e., travel time between any two stops including boarding and alighting of passengers in the middle), we treat every record of the data as a sample point. It reveals what are the boarding/alighting stops and what is the boarding time and total travel time for that trip. We group all records according to the start stops, alighting stops, start time interval and bus line number. From all entries within a single group, we are then able to extract the mean and standard deviation of the corresponding travel time. Finally, we estimate the waiting time by figuring out the frequency of bus arrivals of the same bus line at a given stop during a given time interval.

TABLE I. NOTATIONS FOR MODELING BUS TRAVEL AND WAITING TIME

$\mu_{i,k}^j$	mean of travel time between stop $i$ and $k$ in time interval $j$
$\sigma_{i,k}^j$	standard deviation of travel time between stop $i$ and $k$ in time interval $j$
$T_{i,k}^j$	time for a bus to travel from the $i$ -st stop to $k$ -th stop in time interval $j$
$M$	number of buses for the whole day
$R$	number of time intervals the day is split into
$R_l$	length of a time interval
$f^j$	frequency of buses starting during time interval $j$
$w_i^j$	expected waiting time at the $i$ -th stop during the $j$ -th time interval
$B_i^m$	the time when bus $m$ is at stop $i$

### C. Basic Models for Bus Travel Time and Waiting Time

Here, we briefly describe our basic models [12] that will be leveraged in the next section. Notations are listed in Table I.

1) *Estimation of transit times from station to station:* For each of the time intervals in a day, the distribution for the transit times is extracted from the data. This aims at modelling rush hour situations slowing down the buses and thus achieving a realistic congestion temporal profile. The travel time of a bus from the  $i$ -th stop to the  $k$ -th stop during time interval  $j$  can be represented as a sample from a log normal distribution as in Equation 1, which has the best *Anderson-Darling* goodness of fit tests on the data [12].

$$T_{i,k}^j \sim \log \mathcal{N}(\mu_{i,k}^j, \sigma_{i,k}^j) \quad (1)$$

The parameters needed for the distribution, namely the mean and the deviation, are then extracted from the data by finding all samples that have the same starting stop, ending stop, bus number and are within the respective time interval. The result of this process is a lookup table containing information about the distribution parameters of all possible stop to stop combinations for all time intervals of the day.

2) *Estimating Bus Waiting Time:* We first extract the times  $B_1^m$ , which is the time when bus  $m$  is at the first stop in its scheduled route. Then, in order to get the estimated arrival time of the bus at other stops we add the mean travel time:

$$B_k^m = B_1^m + \mu_{i,k}^j \quad (2)$$

In order to estimate the frequency of buses arriving at a stop during a specific time interval  $j$  we define the function  $f$  that tells us if the estimated arrival time of a bus is within the chosen time interval:

$$f(B_i^m, j) = 1 \quad \text{if } B_i^m \in j \quad (3)$$

$$f(B_i^m, j) = 0 \quad \text{if } B_i^m \notin j \quad (4)$$

Then, the frequency of bus arrivals at a station can be defined as the number of buses that are expected to arrive during this period divided by the length of the time interval:

$$f_i^j = \frac{\sum_{k=1}^M f(B_i^k, j)}{R_l} \quad (5)$$

The expected waiting time at a bus stop for a specific time interval can be calculated as:

$$w_i^j = \frac{1}{2f_i^j} \quad (6)$$

### D. Building a FIFO Stochastic Time-Dependent Network

As discussed above, time-dependent travel time is a step function. For both weekday and weekend, a number of time intervals are predefined and data are aggregated based on which time interval they fall into. Then, for each time interval, continuous distributions of bus travel time and waiting time are calculated. For example, as shown in Figure 2, the mean travel time between time interval [9:00, 9:30] is 35 minutes, while traveling between [8:30, 9:00], which is in peak hours, takes up to 45 minutes. Hence, if two users U1 and U2 start their journeys at 8:59 and 9:01 respectively, their expected travel times would be 45 and 35 minutes.

In fact, there are two issues with step functions of time-dependent travel time. Firstly, the travel time estimation is not logical as discussed in the above example: with only two-minute difference in departure time, the expected travel time difference is as much as 10 minutes. Secondly, in this scenario, the FIFO property of a transit network is violated, i.e., user U1 starts earlier but ends up arriving later than user U2.

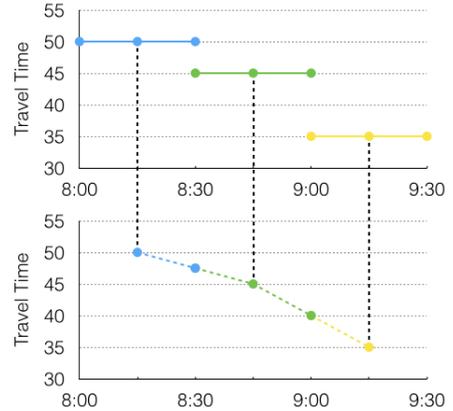


Fig. 2. Linearizing step-functions of time-dependent travel times.

To solve the above issues, we propose a method to linearize the step functions of time-dependent travel time. Specifically, instead of choosing the best matching time interval that the given departure time falls into, we take the best matching and the second best matching intervals, then use their weighted average as the expected travel time. The detailed implementation is shown in Algorithm 1.

With the above technique, a linear function is created between any two neighboring time intervals. In this way, the travel time across time intervals during a day follows a continuous piece-wise linear function instead of step functions. Applying this to the above example, user U1 departing at 8:59

and user U2 departing at 9:01 are now expected to travel 40.3 minutes and 39.7 minutes respectively, which follows the FIFO property.

After linearizing the step functions of travel time, the FIFO property is guaranteed as long as the slope of the linear function is greater than  $-1$ , which intuitively means that by departing  $t$  units of time later, the expected travel time shortened should not be larger than  $t$ . Thus, departing later results in arriving at the destination at a later time too. We did Monte Carlo experiments [3] and the results confirm that when the FIFO property is guaranteed, the optimal substructures of shortest paths also hold for linearized log normal continuous distribution functions.

TABLE II. NOTATIONS FOR LINEARIZATION OF BUS TRAVEL TIME MODEL

$t_0$	the starting time of the first interval of a day
$I_j$	time interval, $I_j = \{t   t \in (t_0 + j * R_l, t_0 + (j + 1) * R_l)\}$
$p_j$	the middle point of time interval $I_j$ , $p_j = t_0 + (j + 0.5) * R_l$
$T^j$	the travel time for a bus in time interval $j$

### Algorithm 1: Compute linearized travel time

```

Input: Departure time  $t$ 
Output: Expected bus travel time  $T$ 
/* check if the departure time is within
bus operating time */
1 if  $t < t_0$  or  $t > t_0 + R * R_l$  then
2   | return  $T = \infty$ 
3 end
/* first get the index of the best
matching interval  $i_s$  */
4  $i_s = (t - t_0) \bmod R_l$ 
/* next find the index of the upper and
lower bounding intervals  $i_u$  and  $i_l$  */
5 if  $t < p_0$  or  $t > p_{R-1}$  then
6   | //  $t$  falls in the first/last interval
7   |  $i_u = i_l = i_s$ 
8 else if  $t < p_{i_s}$  then
9   | // 2nd best matching is on the left
10  |  $i_u = i_s$ 
11  |  $i_l = i_s - 1$ 
12 else
13  | // 2nd best matching is on the right
14  |  $i_u = i_s + 1$ 
15  |  $i_l = i_s$ 
16 end
17 return  $T = T^{i_l} + (t - p_{i_l})/R_l * (T^{i_u} - T^{i_l})$ 

```

## IV. STOCHASTIC ROUTE PLANNING IN TIME-DEPENDENT NETWORKS

In this section, we first describe how to construct a time-dependent transportation graph [24], given the static information including bus stops and schedules of bus lines. Then, we introduce an algorithm that can deal with the lack of component aggregation property of travel time, and optimize both speediness and reliability of routes.

### A. Time-dependent bus network model

Since there can be multiple bus lines serving the same bus stop, we create nodes of two types in a transportation graph, namely transfer nodes and route nodes. Specifically, a physical bus stop  $S$  corresponds to a single transfer node  $t_S$  and multiple route nodes, e.g.,  $r_S^{l_1}, \dots, r_S^{l_k}$  if there are  $k$  different bus lines serving that bus stop. A route returned to users always starts from an origin transfer node and ends at a destination transfer node.

There are three categories of edges in this transportation graph. The edge from a transfer node  $t_i$  to a route node  $r_i^l$  represents a bus boarding process, and its associated cost is the expected waiting time  $w_i^l$  at bus stop  $i$  of bus line  $l$ . The edge from a route node  $r_i^l$  to a transfer node  $t_i$  represents the alighting of a commuter and we assume its associated time cost is 0. The edge from a route node  $r_i^l$  to another route node  $r_j^l$  of the same bus represents the bus traveling process, and its associated cost is the stochastic time-dependent bus travel time  $T_{i,j}^l$  between stops  $i$  and  $j$  of bus line  $l$  as we modeled in section III.

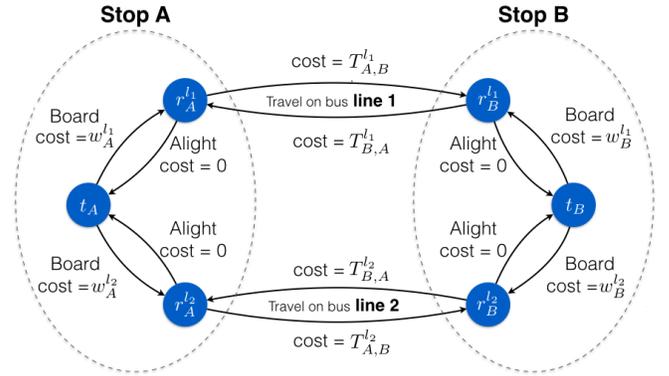


Fig. 3. Time-dependent transportation graph.

Figure 3 illustrates a simple time-dependent transportation graph, where two bus lines  $l_1$  and  $l_2$  run between bus stops  $A$  and  $B$ . In this case, a stop (say,  $A$ ) is modeled as a single transfer node (i.e.,  $t_A$ ) plus two route nodes (i.e.,  $r_A^{l_1}$  and  $r_A^{l_2}$ ). The edge from  $t_A$  to  $r_A^{l_1}$  represents a boarding process, and the cost of the edge is the expected waiting time for bus line  $l_1$ . The reversed edge  $r_A^{l_1}$  to  $t_A$  means alighting and the cost is 0. The edge from  $r_A^{l_1}$  to  $r_B^{l_1}$  shows the traveling of bus line  $l_1$  from bus stop  $A$  to  $B$ , and its associated cost is the expected travel time between the two stops.

Note that the cost can be multi-dimensional. More specifically, in our stochastic model, both bus waiting time and travel time have two dimensions, namely a mean (speediness) and a variance (reliability). In general, we define a  $k$ -dimension cost  $c$  as an array of  $k$  cost elements  $[c[0], c[1], \dots, c[k-1]]$ .

- The plus operation,  $c = c_1 + c_2$ , is defined as  $c[i] = c_1[i] + c_2[i] \forall i \in [0, k-1]$ .
- The compare operation,  $c_1 < c_2$ , is defined to be  $c_1[i] \leq c_2[i] \forall i \in [0, k-1]$  and  $\exists m$ , s.t.  $c_1[m] < c_2[m]$ .

When combining two log normal distributions of travel times, the aggregate mean can be calculated by the sum of individual means. However, the aggregate variance requires solving a convolutional integral [7], which is highly computationally expensive. Thus, we use the sum of variances as an approximate indication for the reliability of routes.

### B. Modified Multi-criteria Shortest Path Algorithm

As we consider two criteria (speediness and reliability) in our route planning, we mainly use multi-criteria shortest path algorithm [6] but adapt it to handle the lack of component aggregation property of bus travel time. The main algorithm is shown in Algorithm 2. We define a node label  $l_i$  as a pair structure  $(n_i, c_i)$  consisting of a node  $n_i$  and a multi-dimension cost  $c_i$ , which is the cost to reach node  $n_i$  from the source node. A priority queue  $pq$  is maintained to determine which node and label to explore next. It is sorted based on the mean travel time of the multi-dimensional cost. A predecessor map  $pm$  is used to keep track of the predecessors of the labels. For every node, there is a node cost list  $cl$  storing all non-dominated costs since a cost may not be strictly comparable to another. Lines 1 to 5 of the algorithm initialize all data structures and insert the source node into the priority queue with a cost 0. Line 6 is the main loop, which keeps running until there is no more unsettled labels in the priority queue.

In each iteration, the label with the lowest cost is retrieved and all of its outgoing edges are examined. Here, instead of simply adding the current cost and the edge cost as in the original algorithm, which causes the bus travel time's increasing error problem as discussed in Section I, we invoke a subroutine `getAccurateCost` to find the correct cost without any aggregation error in line 10. The details of this subroutine (Algorithm 3) is explained in the next section. Once we get the cost, it is compared with existing costs of the same node in line 12. If it is better than any, a new label is created and inserted into the priority queue, and dominated costs are removed from the node. The algorithm stops when all labels are processed and the queue becomes empty.

Recall that in Section II, the bus simulation model shows that passengers boarding and alighting time for the departure and arrival stops are not included in the total travel time. Hence, we face the lack of component aggregation property problem which is for any two bus stops that are  $k$  stops away ( $k > 1$ ), the expected travel time is longer than the sum of travel times of each pair of its consecutive stops, i.e.,  $\sum_{s=1}^k T_{i+(s-1),i+s}^j < T_{i,i+k}^j$ . In order to get the accurate travel time, we directly query the model to get  $T_{i,i+k}^j$  as discussed in the following subsection.

### C. Solution to Travel Time's Lack of Component Aggregation

Algorithm 3 shows the steps to get the accurate travel time cost between two nodes. The intuition is that aggregation error happens only when the query is between two route nodes, not for transfer nodes. Line 3 and 6 test if the query is made from a transfer node to a route node or the other way round. In both cases, taking the corresponding bus waiting time or cost 0 will do. However, if the query is made between two route nodes, as shown in line 8, a special handling is needed. To solve the lack of component aggregation property problem, we keep tracing

---

### Algorithm 2: Modified multi-criteria shortest path

---

```

Input: graph  $g$ , source node  $n_s$ , destination node  $n_d$ ,
         departure time  $t$ 
Output: shortest paths and their expected costs
/* initialization */
1 priorityQueue  $pq = \emptyset$ 
2 predecessorMap  $pm = \emptyset$ 
3 nodeCostsList  $cl = \emptyset$ 
4 label  $l_s = \text{createLabel}(n_s, c_s = 0)$ 
5  $pq.insert(l_s)$ 
6 while  $!pq = \emptyset$  do
7   label  $l_u = pq.pop()$ 
8   node  $n_u = l_u.getNode()$ 
9   foreach outgoing edge  $e_{u,v}$  do
10    cost  $c_v = \text{getAccurateCost}(l_u, e_{u,v}, t)$ 
11    List<cost>  $costs = cl.getCosts(n_v)$ 
12    if  $c_v$  is dominated by  $costs$  then
13      /* drop  $c_v$  since it is not
14       smaller than any of existing costs */
15      continue
16    end
17    /*  $c_v$  is not dominated */
18     $cl.put(n_v, c_v)$ 
19     $cl.removeDominated()$ 
20    label  $l_v = \text{createLabel}(n_v, c_v)$ 
21     $pm.put(l_v, l_u)$ 
22     $pq.insert(l_v)$ 
23  end
24 end

```

---

backwards for predecessors and find the first non-transfer node  $n_w$ . The path from this node to the current node includes all boarding and alighting information for the intermediate bus stops. Thus the accurate cost is calculated by cost of node  $n_w$  plus the travel time between  $n_w$  and current node.

A simple example that illustrates how to get accurate cost is shown in Figure 4. A bus serves three subsequent stops  $A, B$  and  $C$ . Starting from the transfer node  $t_A$ , the cost for route node  $r_A$  is simply a 7-minute waiting time. From  $r_A$  to  $r_B$ , a 2-minute travel time is added normally. However, when the algorithm reaches node  $r_C$ , instead of adding another 2-minute to existing cost ( $c_B + T_{B,C}$ ) and getting 11 minutes, the algorithm finds the first non-transfer node  $r_A$ , and the cost is calculated by  $c_A + T_{A,C}$ , which ends up to be 12 minutes.

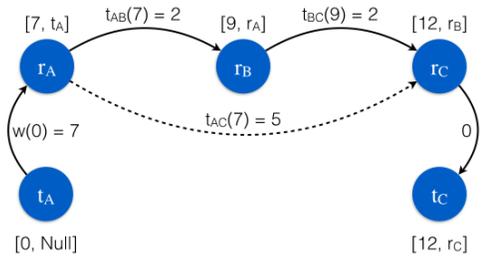


Fig. 4. Dealing with travel time's lack of component aggregation.

---

**Algorithm 3:** getAccurateCost

---

**Input:** label  $l_u$ , edge  $e_{u,v}$ , departure time  $t$  at source  
**Output:** cost  $c_v$

```
1 cost  $c_u = l_u.getCost()$ 
2 node  $n_u = l_u.getNode()$ 
3 if  $n_u$  is a transfer node and  $n_v$  is not then
    /* a boarding process */
4   cost  $c_w = e_{u,v}.getWaitingTime(t + c_u.mean)$ 
5    $c_v = c_u + c_w$ 
6 else if  $n_v$  is a transfer node and  $n_u$  is not then
    /* an alighting process */
7    $c_v = c_u$ 
8 else
9   label  $l_w$ 
    /* trace back to find the last
    non-transfer node  $n_w$  */
10  label  $l_i = l_u$ 
11  while  $n_i = l_i.getNode()$  is not a transfer node do
12    |  $l_w = l_i$ 
13    | label  $l_i = pm.getPredecessor(l_i)$ 
14  end
15  cost  $c_w = l_w.getCost()$ 
16   $c_v = c_w + e_{w,v}.getTravelTime(t + c_w.mean)$ 
17 end
18 return  $c_v$ 
```

---

## V. EXPERIMENTAL STUDY

In this section, we evaluate the quality and accuracy of the routes returned by our proposed dynamic route planning system in comparison with other popular route planners.

### A. Experimental Setup

The data used for building stochastic models of bus travel time and waiting time are collected from real smart card integrated bus network in Singapore and comprise of the recorded usage of all bus lines in a period of three months. A trip by a commuter recorded in the data set starts when the smart card is used to tap in the vehicle and ends when the passenger taps out.

**Data analysis.** We do a simple statistics analysis on the data to confirm the time-dependent characteristic of bus travel time. More specifically, we measure the impact on travel time when a journey by a commuter is performed during peak hours in weekdays compared to other times. Origin-destination (O-D) pairs are randomly selected and the travel times of the same pair during different times are compared: (1) during peak and off-peak hours in the same weekday, and (2) during the same peak hours of weekdays and weekends.

Table III shows the distribution of the ratio between the travel time during peak hours vs. off-peak hours in a weekday. O-D pairs are put into different bins based on the ratio, ranging from less than 1 (i.e., traveling during peak hours takes less time than non-peak hours, which are exception points in the data set) to greater than 1.4 (i.e., traveling during peak hours takes 1.4 times longer than non-peak hours). It can be seen that for more than 90% cases, the ratio is greater than 1 which means the travel time is typically longer during peak hours as expected. On average, 20% more time is spent due to traffic

congestions during peak hours. In addition, Table IV shows that peak hours have greater impact on weekdays rather than weekends. The same journey takes 20% more time to complete during peak hours on weekdays as compared to weekends.

TABLE III. WEEKDAYS AM (08:15 vs. 09:45), PM (18:15 vs. 16:15)

Peak/Offpeak	<1	1~1.1	1.1~1.2	1.2~1.3	1.3~1.4	>1.4
AM	9.7%	13.7%	20.3%	26.3%	16.6%	13.5%
PM	11.7%	13.3%	22.9%	20.5%	15.0%	16.7%

TABLE IV. WEEKDAYS VS. WEEKENDS (AM 08:15, PM 18:15)

Weekdays/ Weekends	<1	1~1.1	1.1~1.2	1.2~1.3	1.3~1.4	>1.4
AM	14.6%	19.7%	13.1%	16.4%	16.4%	19.7%
PM	8.9%	10.1%	16.6%	25.5%	19.4%	19.4%

**Baselines and metrics.** We compare our proposed dynamic route planning system with Google Maps<sup>2</sup> and Gothere.sg<sup>3</sup>, which are among the favorite route planners in Singapore. We are interested in the accuracy as well as the quality of the routes returned by these systems. More specifically, the following metrics are used to evaluate: (1) the accuracy of the expected total travel time of trips departing at various times including weekdays' peak and off-peak hours and weekends, (2) the ability to dynamically rank candidate routes dependent on the departure times, and last but not least (3) the ability to consider both the speediness and the reliability of routes.

### B. Experimental Results

**1) The accuracy of expected total travel time:** We first randomly select trip instances from the historical smart card data. Each selected trip includes the information of departure time, arrival time and an origin-destination (O-D) pair. Historical travel time  $T_h$  can be calculated from the arrival and departure times. Then, we query the route planners for the same O-D pair given the same departure time to get the travel time  $T_c$  of the optimal route returned by the comparing systems. Finally, we use root-mean-square error (RMSE) to measure the accuracy of the returned travel times. However, instead of using the absolute travel time difference ( $T_c - T_h$ ) which is hard to interpret, we use percentage values calculated as  $(\frac{T_c - T_h}{T_h})$ .

TABLE V. ACCURACY OF EXPECTED TOTAL TRAVEL TIME

	AM peak	PM peak	AM off-peak	PM off-peak	Weekend
DEPART	13.8%	10.8%	9.7%	9.3%	4.8%
Google Maps	22.2%	23.2%	13.7%	10.5%	14.2%
Gothere.sg	45.5%	46.7%	40.4%	29.0%	40.6%

The experiment results are shown in Table V. As can be seen, Gothere.sg suffers from the largest error. On average, its returned travel time is 40% different from the real value. Google Maps does relatively well for weekday off-peak hours and weekends when the error is just around 15%. However, its travel time estimation for peak hours drops sharply and the error is as high as 22%. In comparison, the accuracy of DEPART is the best of the three in all cases. The error is constantly below 14% and usually below 10%. Further,

---

<sup>2</sup><https://maps.google.com>

<sup>3</sup><http://gothere.sg/maps>

the travel time estimation for peak hours achieves similar accuracy as off-peak hours. The results confirm that DEPART can effectively deal with the uncertain and time-dependent characteristics of urban traffic.

2) *The ability to dynamically rank candidate routes dependent on departure times:* Existing route planners such as Google Maps and Gothere.sg are static in the sense that they return the same routes in spite of different departure times. In contrast, DEPART recommends routes with better quality due to its ability to calculate routes that are more adapted to traffic conditions, e.g., peak and off-peak hours. For example, there are two paths shown in Figure 5. Bus 700 goes by the right path which is the main road. Bus 167 goes by a side track path on the left. When traveling from bus stop 03223 to stop 08031 at 14:30 pm on a weekday, both routes take less than 15 minutes and bus 700 is slightly faster. Thus, DEPART recommends bus 700 traveling by the main road. Nevertheless, when the departure time is 18:30 pm, the expected travel times of bus 700 and bus 167 increase to 22 and 18 minutes respectively. This means that the previously faster route 700 on the main road turns out more likely to be congested and takes 20% longer travel time. In this case, DEPART dynamically recommends bus 167 to the user.

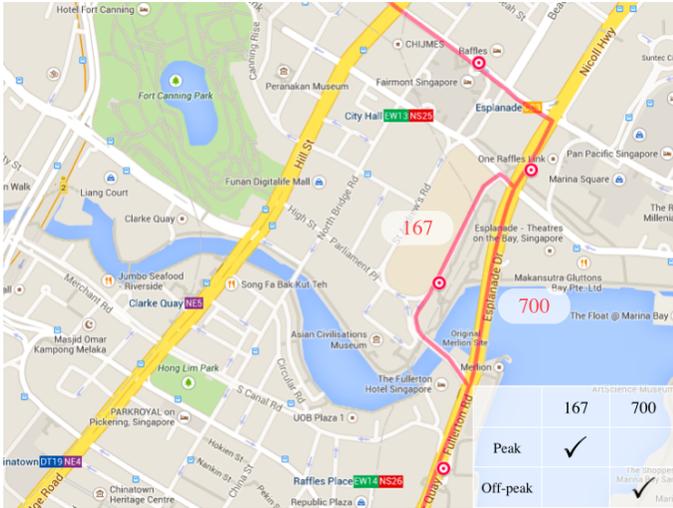


Fig. 5. Dynamic route recommendation.

3) *The ability to consider the reliability of paths:* Since the traffic network is stochastic in nature, the expected travel time is not always reliable. To optimize both the speediness and reliability of routes, our route planning system utilizes multi-criteria algorithm and finds both least expected travel time and most reliable routes.

For instance, there are two routes from downtown to a fencing club as shown in Figure 6. Historical data shows that during evening peak hours the upper route going through the main road takes 15% longer travel time than the lower route, but its variance is only 20% of the faster one. Both routes are returned by DEPART since they are better in either expected travel time or reliability. If a user has a fencing class to attend and does not want to miss it by any chance, he should take the slightly longer but more reliable route. Other users without strict deadlines may prefer the expected faster route.

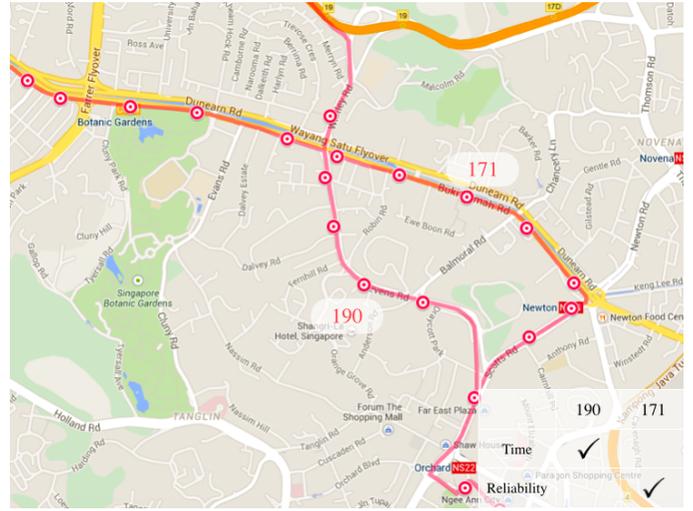


Fig. 6. Speediness vs. reliability of routes.

## VI. RELATED WORK

In this section, we review relevant works related to traffic modeling and stochastic route planning in time-dependent networks, and highlight our research contributions.

**Traffic modeling.** Research in modeling of bus movement originated with a analytic model [22] which allows for convenient mathematical analysis. This model and its variants assume a fairly simple transit network with a couple of vehicles which makes the models inapplicable in real world situations. With the increasingly popularity of smart card integrated public transportation, there exist efforts in utilizing travel smart card data for traffic modelling. In [13], linear interpolation of the bus positions is used in order to gain information about spatio-temporal movement of the buses. The work described in [23] uses smart card analysis to extract useful data about the traffic in the city. In [12], information from smart card data is used to model bus travel time and waiting time as stochastic time-dependent distributions. In this paper, we propose to linearize those stochastic step functions to ensure the FIFO property in a public transit network and incorporate that model in our dynamic route planning system.

**Stochastic route planning.** Loui [16] is the first to present the problem of finding optimal paths (specifically, minimizing a utility function) in generic stochastic networks. It is shown that the utility function has to be either affine linear or exponential; otherwise, optimal substructure of the problem is not guaranteed. Nikolova et al. [20] and Lim et al. [14], [15] further extend the work and focus on stochastic road networks. In [27], Wu et al. propose an approach to model risk-taking behavior based on the theory of stochastic dominance(SD), and use it to find optimal paths for different utility functions. Chen et al. [4] find reliable shortest paths by pre-specifying an on-time arrival probability. Overall, the above works are specially proposed for stochastic road networks while we target at stochastic time-dependent public transit networks.

**Route planning in time-dependent networks.** In stochastic time-dependent networks, the cost of an edge in the network is typically modeled as a discrete random variable and also has different values depending on the time intervals in a day. While

the works in [11], [18] study the least expected time paths in STD networks, Sun et al. [25] focus on finding the most robust paths instead. Overall, these works assume that the stochastic travel time in a generic network follows a discrete random distribution where only a limited number of values can be chosen. In our work, the travel time is modeled after practical smart card data set collected from a real bus network and follows a continuous distribution. Further, we optimize both the travel time and reliability of the returned paths.

Another research direction in STD networks is to find an optimal routing policy, which is a hierarchical decision that specifies which bus to take at the next transfer depending on the actual arrival times of buses [11]. Gao [8], [10] and Chabini [9] further consider link wise and time wise stochastic dependencies of travel times. Wu et al. [26] incorporate real-time information into routing policies in STD networks. Note that the above works only consider bus travel time in their route planning process. Nevertheless, in public transit networks, waiting time is an important contributing factor to the total journey time as well. Datar and Ranade [5] study bus networks with stochastic waiting time, where bus arrivals are assumed to follow a Poisson process. Boyan and Mitzenmacher [2] further improves the work by generalizing bus arrival distribution to any distribution as long as the bus waiting time has increasing failure rate. Our proposed dynamic route planning system advances these works by considering both waiting time and travel time in public transit networks.

## VII. CONCLUSION

In this paper, we have proposed DEPART – a practical route planning system that can effectively deal with uncertain and time-dependent characteristics of urban traffic. We extract the distributions of bus travel time and waiting time from real smart card data and build a FIFO stochastic model for the bus network of the entire city. We introduce a new algorithm that solves the lack of component aggregation property of bus travel time. Experimental results on real bus network confirm that by leveraging historical data and optimizing both the expected travel time and reliability of the routes, DEPART is able to recommend routes that are more adapted to traffic situations as compared to other popular route planners. Our future research in this direction includes devising an online route planning algorithm that leverages the real-time feed of bus arrival times.

## ACKNOWLEDGMENT

The authors would like to thank Chu Yao Lim, from TUM CREATE, for his helpful support in preparation and processing smart card data collected from the bus network in Singapore. The research is partially funded by the Economic Development Board and the National Research Foundation of Singapore.

## REFERENCES

- [1] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner, and R. Werneck. Route planning in transportation networks. In *Technical Report MSR-TR-2014-4*. 2014.
- [2] J. Boyan and M. Mitzenmacher. Improved results for route planning in stochastic transportation. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 895–902, 2001.
- [3] R. E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.

- [4] B. Y. Chen, W. H. Lam, A. Sumalee, Q. Li, H. Shao, and Z. Fang. Finding reliable shortest paths in road networks under uncertainty. *Networks and spatial economics*, 13(2):123–148, 2013.
- [5] M. Datar and A. G. Ranade. Commuting with delay prone buses. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 22–29, 2000.
- [6] Y. Disser, M. Müller-Hannemann, and M. Schnee. Multi-criteria shortest paths in time-dependent train networks. In *Experimental Algorithms*, pages 347–361. 2008.
- [7] D. Dufresne. Sums of lognormals. In *Actuarial Research Conference*, 2008.
- [8] S. Gao. *Optimal adaptive routing and traffic assignment in stochastic time-dependent networks*. PhD thesis, Massachusetts Institute of Technology, 2005.
- [9] S. Gao and I. Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B: Methodological*, 40(2):93–122, 2006.
- [10] S. Gao, E. Frejinger, and M. Ben-Akiva. Adaptive route choices in risky traffic networks: A prospect theory approach. *Transportation research part C: emerging technologies*, 18(5):727–740, 2010.
- [11] R. W. Hall. The fastest path through a network with random time-dependent travel times. *Transportation science*, 20(3):182–188, 1986.
- [12] J. Ivanchev, H. Ayt, and A. Knoll. Stochastic bus traffic modelling and validation using smart card fare collection data. In *Intelligent Transportation Systems (ITSC)*, pages 2054–2061, 2014.
- [13] D. Lee, L. Sun, and A. Erath. Study of bus service reliability in singapore using fare card data. In *12th Asia-Pacific Intelligent Transportation Forum*, 2012.
- [14] S. Lim, H. Balakrishnan, D. Gifford, S. Madden, and D. Rus. Stochastic motion planning and applications to traffic. In *Algorithmic Foundation of Robotics VIII*, pages 483–500. 2009.
- [15] S. Lim, C. Sommer, E. Nikolova, and D. Rus. Practical route planning under delay uncertainty: Stochastic shortest path queries. *Robotics*, page 249, 2013.
- [16] R. P. Loui. Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*, 26(9):670–676, 1983.
- [17] G. Menon and L. C. Kuang. Lessons from bus operations. *A report collected from web http://www.ptc.gov.sg\_files/Lessons\_from\_Bus\_Operations\_REV5.pdf*, 2006.
- [18] E. D. Miller-Hooks and H. S. Mahmassani. Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, 34(2):198–215, 2000.
- [19] E. Nikolova. Approximation algorithms for reliable stochastic combinatorial optimization. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 338–351. 2010.
- [20] E. Nikolova, M. Brand, and D. R. Karger. Optimal route planning under uncertainty. In *International Conference on Automated Planning and Scheduling (ICAPS)*, volume 6, pages 131–141, 2006.
- [21] E. Nikolova, J. A. Kelner, M. Brand, and M. Mitzenmacher. Stochastic shortest paths via quasi-convex maximization. In *European Symposium on Algorithms (ESA)*, pages 552–563. 2006.
- [22] E. Osuna and G. Newell. Control strategies for an idealized public transportation system. *Transportation Science*, 6(1):52–72, 1972.
- [23] J. Y. Park, D.-J. Kim, and Y. Lim. Use of smart card data to define public transit use in seoul, south korea. *Transportation Research Record: Journal of the Transportation Research Board*, 2063(1):3–9, 2008.
- [24] E. Pyrga, F. Schulz, D. Wagner, and C. Zaroliagis. Efficient models for timetable information in public transportation systems. *Journal of Experimental Algorithmics (JEA)*, 12:2–4, 2008.
- [25] S. Sun, Z. Duan, S. Sun, and D. Yang. How to find the optimal paths in stochastic time-dependent transportation networks? In *Intelligent Transportation Systems (ITSC)*, pages 2348–2353, 2014.
- [26] C. Wu, X. Zhang, and Y. Dong. Adaptive route guidance based on real-time information in stochastic time-dependent transportation networks. In *Intelligent Transportation Systems (ITSC)*, pages 2392–2397, 2014.
- [27] X. Wu and Y. M. Nie. Modeling heterogeneous risk-taking behavior in route choice: A stochastic dominance approach. *Transportation research part A: policy and practice*, 45(9):896–915, 2011.