2015 IEEE 18th International Conference on Intelligent Transportation Systems

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

Peng Ni[1,2], Hoang Tam Vo, Daniel Dahlmeier [1]SAP Innovation Center
Wentong Cai [2]Nanyang Technological University
Jordan Ivanchev, Heiko Aydt TUM CREATE

***Abstract*—While providing intelligent urban transportation services is one of the key enablers for realizing smart cities, existing transit 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`[1]– 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-practice route planners.**

I. Introduction

Intelligent urban transportation systems are becoming increasingly important for commuters in smart cities. Finding optimal routes, the fundamental problem 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, this does not really hold in a realistic transit network since bus arrival times are largely dependent on real traffic conditions which are 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 [15]. Popular route planners in Singapore, such as Google Maps and Gothere.sg, have two major drawbacks. Firstly, the query results are the same no matter whether the departure time falls in peak or off-peak hours. Secondly, the travel times estimated by these route planners are not that accurate. For example, Gothere.sg returns a travel time of 30 minutes for a journey consisting of 30 bus stops. In practice, based on the historical travel smart card data collected from the bus network this journey takes at least 50 minutes due to passengers boarding and alighting time at each stop.

Related Work. Loui [14] is the first to formulate the problem of finding optimal paths in generic stochastic networks by minimizing utility functions. Nikolova et al. [18] and Lim et al. [12], [13] further extend the work and focus on stochastic road networks. In [24], Wu et al. propose an approach to model risk-taking behavior based on the theory of stochastic dominance, and use it to find optimal paths for different utility functions. Overall, the above researches are specially proposed for stochastic road networks, while we target at stochastic time-dependent public transit networks.

The researches in [10], [16] study the least expected time paths in stochastic time-dependent (STD) networks, while Sun et al. [22] focus on finding the most robust paths instead. Overall, these proposals assume that the stochastic travel time follows a discrete random distribution. In our work, the travel time is modeled after practical smart card data and fit into a continuous distribution. Another research direction in STD networks is to find an optimal routing policy, which is a hierarchical decision scheme that specifies which bus to take next at each decision node. Gao [7], [9] and Chabini [8] consider link wise and time wise stochastic dependencies of travel times. Wu et al. [23] incorporate real-time information into routing policies in STD networks. However, the above researches only consider bus travel time and ignore waiting time. On the contrary, bus networks with stochastic waiting time are studied in [4], [2], while the travel time is assumed to be deterministic. Our proposed dynamic route planning system advances these proposals by considering both waiting time and travel time in public transit networks.

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 and waiting time have to be carefully devised so that transit networks ensure first-in-first-out (FIFO) property, i.e., buses do not overtake each other, which is unlike road networks where faster drivers can arrive early even if departing late. 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 STD network [10].

Secondly, it is challenging to properly incorporate the speediness of a route, i.e., the expected travel time, with its reliability, i.e., the standard deviation of the travel time. Most of researches done in this area use a mean-risk model [13], [19], [17] which combines the mean and variance of the travel time as a single linear objective function whose coefficients are usually determined in a heuristic manner. 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, the sum of the travel times from consecutive stops

[1]`DEPART`: Dynamic routE PlAnning tRansit neTworks

978-1-4673-6596-3/15 $31.00 © 2015 IEEE
DOI 10.1109/ITSC.2015.271

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 and alighting at the middle stop S2. 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 planing 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-practice 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 conclude the paper in Section VI.

II. System Overview

In this section, we present an overview of DEPART – our proposed solution to dynamic route planning in time-dependent public transit networks which recommends routes adapted to traffic situations. Figure 1 illustrates the overall architecture of the system.

The system leverages 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.

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).

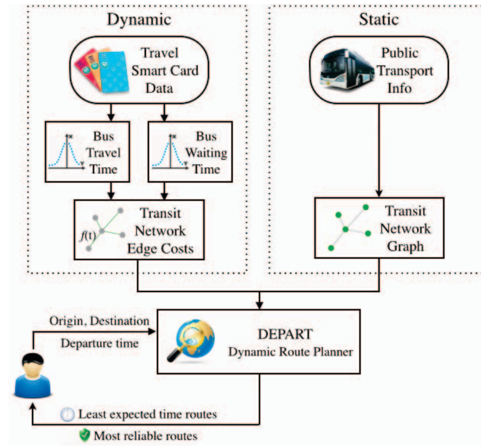

Fig. 1. Architecture of DEPART – a dynamic route planning system.

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 time and waiting time. 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 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 information needed is the estimated bus waiting time. However, the smart card records only points out when the passenger boarded the bus at a bus stop and not when the passenger actually arrived at that stop and waited for the bus, 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, and bus line number. To further place every trip in a unique time interval within a day, we use the midpoint of that trip, which is defined as the point in time that is in the middle between the start time and end time of the trip. In this way, we are able to classify the extreme case of a trip starting just at the end of a time interval, however, mainly taking place during the next interval. 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 time interval.

TABLE I. NOTATIONS FOR MODELING BUS TRAVEL AND WAITING TIME

$\mu^j_{i,k}$	mean of travel time between stop i and k in time interval j
$\sigma^j_{i,k}$	standard deviation of travel time between stop i and k in time interval j
$T^j_{i,k}$	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^j_i	expected waiting time at the i-th stop during the j-th time interval
B^m_i	the time when bus m is at stop i

C. Basic Models for Bus Travel Time and Waiting Time

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 modeled 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 fit into a log normal distribution as in Equation 1, which has the best *Anderson-Darling* goodness of fit tests on the data [11].

$$T^j_{i,k} \sim \log\mathcal{N}(\mu^j_{i,k}, \sigma^j_{i,k}) \tag{1}$$

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^m_1, 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:

$$B^m_k = B^m_i + \mu^j_{i,k} \tag{2}$$

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 within a day:

$$f(B^m_i, j) = 1 \quad if \quad B^m_i \in j \tag{3}$$

$$f(B^m_i, j) = 0 \quad if \quad B^m_i \notin j \tag{4}$$

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:

$$f^j_i = \frac{\sum_{k=1}^{M} f(B^k_i, j)}{R_l} \tag{5}$$

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

$$w^j_i = \frac{1}{2f^j_i} \tag{6}$$

D. Building a FIFO Stochastic Time-Dependent Network

As discussed above, time-dependent travel time and waiting time are step functions. More specifically, 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. Similar issues are also applied to the stochastic bus waiting time which is dependent on the time intervals within a day.

To solve the above issues, one way is to use moving time windows around every time point and build distributions for that point. However, this approach is either memory expensive if all distributions are to be stored or computationally expensive if the distributions are computed on the fly for each query. Instead, we use 30-minute intervals within a day and maintain the distributions for each time interval. Based on our analysis, this length of time intervals is sufficient to capture different traffic patterns such as peak and non-peak hours.

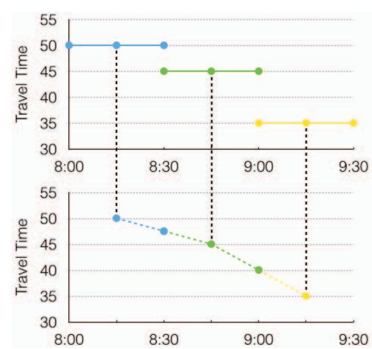

Fig. 2. Linearizing step-functions of time-dependent travel times.

Next, we propose a method to linearize the step functions of time-dependent travel time and waiting 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. 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. The similar linearization approach also applies to the waiting time.

After linearizing the step functions of travel time and waiting 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.

IV. Stochastic Route Planning in Time-dependent Networks

In this section, we first describe how to construct a time-dependent transportation graph [20], 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., $r_S^{l_1}$, ..., $r_S^{l_k}$ 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 bus travel time $T_{i,j}^l$ between stops i and j of bus line l. Both the bus waiting time w_i^l and travel time $T_{i,j}^l$ are stochastic and time-dependent as we modeled in section III.

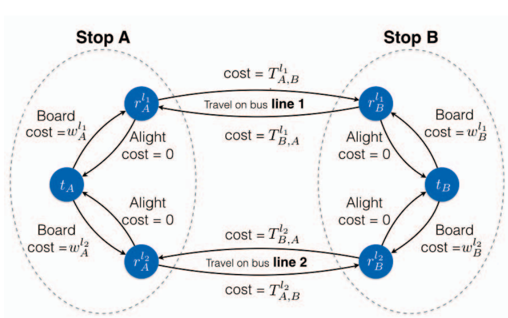

Fig. 3. Time-dependent transportation graph.

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 $[c[0], c[1], ..., c[k-1]]$. 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, the aggregate mean can be calculated by the sum of individual means. However, the aggregate variance requires solving a convolutional integral [6]. The research in [21] proposes another way to calculate the aggregated distribution of a complete path given distributions of each path segment. In our case, the complete path is unknown until the routing algorithm comes to its completion. With every path segment explored by the algorithm, a new distribution needs to be calculated, which is 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 system, we mainly use multi-criteria shortest paths algorithm [5] but adapt it to handle the lack of component aggregation property of bus travel time as discussed in Section I. Here, instead of simply adding the total cost at the current node and the cost of its out going edge as in the original algorithm, which causes the bus travel time's increasing error problem, we invoke a subroutine `getAccurateCost` to find the correct cost without any aggregation error.

Algorithm 1 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 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 the 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.

Algorithm 1: getAccurateCost

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

cost $c_u = l_u.getCost()$
node $n_u = l_u.getNode()$
if *n_u is a transfer node and n_v is not* **then**
  `/* a boarding process */`
  cost $c_w = e_{u,v}$.getWaitingTime($t + c_u.mean$)
  $c_v = c_u + c_w$
else if *n_v is a transfer node and n_u is not* **then**
  `/* an alighting process */`
  $c_v = c_u$
else
  label l_w
  `/* get last non-transfer node` n_w `*/`
  label $l_i = l_u$
  while *$n_i = l_i.getNode()$ is not a transfer node* **do**
    $l_w = l_i$
    label $l_i = pm$.getPredecessor(l_i)
  end
  cost $c_w = l_w.getCost()$
  $c_v = c_w + e_{w,v}.getTravelTime(t + c_w.mean)$
return c_v

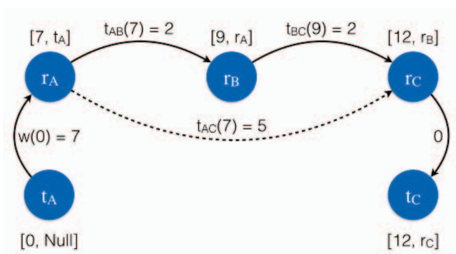

Fig. 4. Dealing with travel time's lack of component aggregation.

V. Experimental Study

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. We compare `DEPART` – our proposed dynamic route planning system with Google Maps[2] and Gothere.sg[3], 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.

B. Experimental Results

*1) **The accuracy of expected total travel time**:* We first select 30 trip instances that cover peak and off-peak hours in both weekdays and weekends from the historical smart card data. The total times taken by these trips range between 30 and 60 minutes. Each selected trip includes the information of departure time, arrival time and an origin-destination (O-D) pair. The historical total travel time can be calculated from the arrival and departure times, which is taken as the ground truth. Then, we query the comparing route planners for the same O-D pair given the same departure time to get the total travel time of the optimal route returned by these systems. Finally, we use root-mean-square error (RMSE) to measure the accuracy of the returned total travel times.

TABLE II. Error of expected total travel time in minutes

	AM peak	PM peak	AM off-peak	PM off-peak	Weekend
DEPART	8.0	6.5	3.9	4.5	2.3
Google Maps	15.4	20.0	5.5	4.5	7.5
Gothere.sg	28.8	32.7	17.4	14.2	20.9

The experimental results are shown in Table II. As can be seen, Gothere.sg suffers from the largest error. On average, its returned travel time is 20 minutes different from the real value, and it tends to underestimate the real travel time. Google Maps does relatively well for weekday off-peak hours and weekends when the errors are just around 6 minutes. However, its travel time estimation for peak hours drops sharply and the error is as high as 15 to 20 minutes. In comparison, `DEPART` provides the highest accuracy in all cases. The errors are constantly below 10 minutes. Further, 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.

Fig. 5. Dynamic route recommendation.

*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. 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 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

[2]https://maps.google.com
[3]http://gothere.sg/maps

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 as an alternative during peak hours.

*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 show 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 of the route. 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. 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 bus travel and waiting time distributions from smart card data and build a FIFO stochastic model for the bus network. We introduce a new algorithm that solves the lack of component aggregation property of bus travel time. Experimental results on bus network confirm that DEPART is able to recommend routes that are more adapted to traffic situations. Our future research 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 *SODA*, pages 895–902, 2001.

[3] R. E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:1–49, 1998.

[4] M. Datar and A. G. Ranade. Commuting with delay prone buses. In *SODA*, pages 22–29, 2000.

[5] Y. Disser, M. Müller-Hannemann, and M. Schnee. Multi-criteria shortest paths in time-dependent train networks. In *Experimental Algorithms*, pages 347–361. 2008.

[6] D. Dufresne. Sums of lognormals. In *Actuarial Research Conference*, 2008.

[7] S. Gao. *Optimal adaptive routing and traffic assignment in stochastic time-dependent networks*. PhD thesis, Massachusetts Institute of Technology, 2005.

[8] S. Gao and I. Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B: Methodological*, 40(2):93–122, 2006.

[9] 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.

[10] R. W. Hall. The fastest path through a network with random time-dependent travel times. *Transportation science*, 20(3):182–188, 1986.

[11] J. Ivanchev, H. Aydt, and A. Knoll. Stochastic bus traffic modelling and validation using smart card fare collection data. In *ITSC*, pages 2054–2061, 2014.

[12] 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.

[13] S. Lim, C. Sommer, E. Nikolova, and D. Rus. Practical route planning under delay uncertainty: Stochastic shortest path queries. *Robotics*, page 249, 2013.

[14] R. P. Loui. Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*, 26(9):670–676, 1983.

[15] 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.

[16] 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.

[17] E. Nikolova. Approximation algorithms for reliable stochastic combinatorial optimization. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 338–351. 2010.

[18] E. Nikolova, M. Brand, and D. R. Karger. Optimal route planning under uncertainty. In *ICAPS*, volume 6, pages 131–141, 2006.

[19] E. Nikolova, J. A. Kelner, M. Brand, and M. Mitzenmacher. Stochastic shortest paths via quasi-convex maximization. In *ESA*, pages 552–563. 2006.

[20] E. Pyrga, F. Schulz, D. Wagner, and C. Zaroliagis. Efficient models for timetable information in public transportation systems. *JEA*, 12:2–4, 2008.

[21] M. Ramezani and N. Geroliminis. On the estimation of arterial route travel time distribution with markov chains. *Transportation Research Part B: Methodological*, 46(10):1576–1590, 2012.

[22] S. Sun, Z. Duan, S. Sun, and D. Yang. How to find the optimal paths in stochastic time-dependent transportation networks? In *ITSC*, pages 2348–2353, 2014.

[23] C. Wu, X. Zhang, and Y. Dong. Adaptive route guidance based on real-time information in stochastic time-dependent transportation networks. In *ITSC*, pages 2392–2397, 2014.

[24] 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.