A Data-Driven Method for Stochastic Shortest Path Problem*

Zhiguang Cao[1,2], Hongliang Guo[2], Jie Zhang[2], Dusit Niyato[2] and Ulrich Fastenrath[3]

***Abstract*— This paper aims at solving a stochastic shortest path problem. The objective is to determine an optimal path which maximizes the probability of arriving on time given a time constraint (i.e., a deadline). To solve this problem, we propose a data-driven approach. We first reformulate the original finding optimal path problem as a cardinality minimization problem. Then, we apply an L_1 norm minimization technique to solve the cardinality problem. The problem is transformed into a mix integer linear programming problem, which can be solved using standard solvers. This proposed approach has three advantages over the traditional methods: (1) the proposed approach can handle various or even unknown travel time distributions, while traditional stochastic routing algorithms can only work on specified distributions; (2) the proposed approach does not rely on the assumption that the travel time on different road segments is independent from each other; (3) unlike other existing approaches which require that the deadline must be larger than a certain value, the proposed approach can support more flexible deadline definition. Then we test our approach respectively on artificial and real-world road networks, the experimental results show that the proposed approach can achieve a comparatively high accuracy when the sampling size of travel time is large enough. Moreover, under some reasonable assumptions, the accuracy could be 100%.**

I. INTRODUCTION

Stochastic shortest path problems have been studied extensively in the fields of operations research and transportation engineering, especially in emergency response and disaster management [1]. The objective is usually to determine an optimal path with the shortest travel time. However, in real world, the traffic condition is often random because of various uncertainties, such as road work, bad weather, traffic accident, unexpected traffic lights. All those uncertain factors may prevent the vehicle from achieving an absolute shortest travel time. Therefore, it is necessary to redefine the optimal route in stochastic context.

A. Related Work

In stochastic shortest path problems, the least expected travel time (LET) is often used as the routing criterion. According to this criterion, the path is optimal if it guarantees the least expected travel time [2]. There have been many researchers studying on this: [3] provides a thorough review and discussion on the time dependent LET path problem without waiting policy; [4] solves the time-dependent LET path problem with waiting policy; [5] addresses the LET path problem considering the correlation between random link travel times. One reason why researchers employ LET is that under this criterion, the problem can be transformed into a deterministic routing problem and solved to obtain an optimal path. However, a path with minimum expected travel time may still have a high variance, which is undesirable for some drivers.

To address the risk issue in LET, [6] and [7] propose a Mean-Risk model. In this model, they seek the path that minimizes the weighted combination of expected travel time and the path's travel time standard deviation (minimize $E(x) + \gamma Var(x)$, where x is the path's travel time distribution). This is a convex combination problem, which can be converted into a deterministic shortest-path problem with respect to road link lengths equal to the linear combination of corresponding mean and variance. The model does help solve the risk problem to some extent, but it still has one obvious limitation: the drivers may not understand the physical meaning of this model and they also do not know what γ is suitable.

To overcome the above disadvantages, a probability tail model, which also incorporates both expected travel time and reliability, is proposed as an optimal criterion in [8]. It defines the optimal path to be the one that maximizes the probability of arriving on time. This criterion is reasonable in that it is consistent with people's travel planning behavior. For example, people would rather spend more traveling time but do not want to be late when they plan for important events. The key objective of routing in such a circumstance is to reduce the risk of arriving late rather than to minimize the expected travel time [9]. Unfortunately, the algorithm on how to compute such kind of optimal path is not laid out in [8]. Many subsequent researchers investigate this problem and many solutions have been achieved as in [1], [2], [9], [10] and [11].

Among the works to solve the probability tail model, three seminal works are [1], [10] and [11]. An adaptive method is developed in [1] to achieve maximal probability of arriving on time. This method provides an optimal policy for selecting the next road junction rather a prior path. It only computes the optimal road junction to visit next. A further road junction will only be determined when the vehicle arrives at the preceding one. Although it claims in [1] that the optimal path determined in this way is accurate, it is inconvenient and not applicable to implement this method at each road junction in a pre-planning scenario, as the

*This work is supported by BMW Group

[1]Zhiguang Cao is with the Energy Research Institute @ NTU (ERI@N), Interdisciplinary Graduate School, Nanyang Technological University, Singapore (caoz0005@e.ntu.edu.sg).

[2]Zhiguang Cao, Hongliang Guo, Jie Zhang and Dusit Niyato are with the School of Computer Engineering, Nanyang Technological University, Singapore ({GuoHL, ZhangJ, Dniyato}@ntu.edu.sg).

[3]Ulrich Fastenrath is head of traffic information management and routing optimization at BMW Group, Munich, Germany (Ulrich.Fastenrath@bmw.de).

driver needs a pre-set route when he/she drives toward the destination. By contrast, it aims to search for an optimal path in [10] and [11]. In [10], it shows that for a large range of deadlines, the problem requires the maximization of a quasi-convex function over the path polytope. Due to the particular form of its quasi-convex objective, the optimal path is obtained at an extreme point of the dominant of the projection (shadow) of the path polytope onto a two-dimensional plane. Based on the work in [10], it proposed a stochastic motion planning algorithm and applied it to traffic navigation in [11]. This algorithm searched for the optimal path efficiently by examining probe points which can efficiently help eliminate futile searching space. This algorithm improves the results in terms of computation complexity compared with [10].

B. Limitations of Existing Work

The methods in [10] and [11] seem more desirable since they can find exact solutions instead of approximated ones. However, they rely on three common assumptions: (1) the travel time for each road link must follow a normal distribution; (2) the travel time distribution on different road links is independent from each other; (3) the deadline should be large enough, i.e., at least larger than the expected travel time of the path with the least expected travel time.

The first two assumptions are made to simplify the computation. However, in real world, the travel time on road links dose not necessarily always follow a normal distribution, and they might not be independent either. There are literatures challenging these assumptions: travel times following a skewed distribution is identified in [12]; travel times best fitting a gamma distribution is concluded in [13]; a log-normal fit is derived in [14], [15], [16] and [17]; a bi-normal distribution is claimed in [18], one is for congestion situation and the other is for non-congestion; correlation for travel time on adjacent road links is also claimed in [15] and [16].

The third assumption is made to guarantee the quasi-convexity of the problem so as to facilitate the computation. However, in real traffic planning, people may not know what value for the deadline would be large enough especially when they are on an unfamiliar path.

C. Contribution

To solve these problems, we propose a data-driven approach, which does not need to assume one fixed distribution for travel time on road links. It can also work well when the travel time on some different road links are correlated with each other. Moreover, it can handle different deadlines. Although our approach only provides an approximated solution, the accuracy is satisfactorily high as it will be shown in the simulation results.

More specifically, to determine the optimal path, we formulate the problem of searching the path which guarantees the maximum probability of arriving on time into a cardinality minimization problem. The general model-driven optimization methods [19] cannot solve this problem efficiently since this optimization is not convex nor quasi-convex if we do not make the necessary assumptions on the travel time distribution, correlation and deadlines. However, the cardinality minimization, which is data-driven in this context, can avoid above assumptions, because it is directly applied on the detected or sampled travel time data set to approximate the real probability by frequency. To solve this cardinality minimization problem, we relax it by L_1 norm minimization, which in turn can be efficiently solved by mix integer linear programming.

The data-driven approach has been widely used in many areas [20], [21], [22], especially in machine learning, where a classifier is always built based on known data set and prediction is conducted upon coming unknown data. The nature of the data-driven approach in machine learning is usually to minimize the chance of mis-classification [20], while the data-driven approach in this paper is used to minimize the frequency of being later than the deadline on known data set, and to this end, they are similar with each other. However, the main part of the data-driven method in this paper is to minimize certain cardinality, and to our best knowledge, it is the first time that cardinality minimization is used to solve the stochastic shortest path problem where the objective is to find a path that maximizes the probability of arriving destination before one deadline.

The remainder of the paper is organized as follows: In Section II, we formulate the stochastic shortest path problem as finding a path that maximizes the probability of arriving on time and then we reformulate it as a cardinality minimization problem. In Section III, we use the L_1 norm minimization to relax and solve the cardinality problem, which is further formulated as a mix integer linear programming. In Section IV, we carry out various experiments to justify the advantages of our approach and provide analysis on the obtained results. Section V states the conclusions and our future work.

II. Problem Formulation for Stochastic Shortest Path

In this section, we first introduce the road network in terms of graph, then we formulate the stochastic shortest path problem based on it. In a general situation, e.g., without the normal distribution assumption, independent assumption or deadline assumption, this routing problem is usually not convex nor quasi-convex, and there is no efficient method to solve it. Considering that the probability of arriving on time usually can be approximated by the frequency of not being late, and this frequency is closely related with the number of times of being late based on travel time data, the routing problem can be directly formulated as cardinality minimization. The hope is that, when the travel time data size or the sampling data size is large enough, the frequency of being late will closely reflect the true probability of arriving on time. More importantly, this method with respect to cardinality minimization is not concerned with the travel time distribution type, correlation or deadline. Details about cardinality minimization problem is also introduced in this section.

A. Original Problem Formulation

We model the road network as a graph. Let $G = (V, A_r)$ be a directed graph, where $V = \{1, 2, \ldots, n\}$ represents the set of nodes and $A_r \subseteq \{(v, w) : v, w \in V, v \neq w\}$ represents the set of arcs, which also refer to the road links. More specifically, (v, w) means an arc from v to w. Then the stochastic shortest path problem which maximizes the probability of arriving the destination d from origin o not later than deadline T can be mathematically formulated as follows:

$$\max_x . \quad Prob(W^{'}x \leq T)$$

s.t.

$$\forall v \in V : \sum_{w \in V,(v,w) \in A_r} x(v,w) - \sum_{w \in V,(w,v) \in A_r} x(w,v) = \begin{cases} 1, if\ v = o \\ -1, if\ v = d \\ 0, otherwise \end{cases} \tag{1}$$

where the vector W denotes the real travel time for each arc; the vector $x \in \{0,1\}^{|A_r|}$, and each component of x refers to one arc on G, e.g., the arc $(w, v) \in A_r$ is on this optimal path if $x(v, w)$ is equal to 1, not on this optimal path if 0. Then this problem can be further compactly written as:

$$\begin{aligned} \max_x . \quad & Prob(W^{'}x \leq T) \\ \text{s.t.} \quad & Mx = b \\ & x \in \{0,1\}^{|A_r|} \end{aligned} \tag{2}$$

where $M \in \mathbb{R}^{n \times |A_r|}$ is the node-arc incidence matrix and $b \in \mathbb{R}^n$, where all elements are zeros except the $s - th$ and $t - th$ element, which are 1 and -1, and refers to origin o and destination d respectively [23].

In the general situation, the optimization problem in Eq.(2) is not convex or quasi-convex if we do not make further assumptions on the travel time distribution, correlation and deadlines, which also means there is no efficient method to solve this problem. So we seek to approximate the probability by frequency to avoid these difficulties, where cardinality minimization is needed. To this end, we first rewrite the "maximizing" problem in Eq.(2) as "minimizing". Considering that maximizing the probability of arriving not later than the deadline is equivalent to minimizing the probability of arriving later than the deadline, we can reformulate Eq.(2) as follows:

$$\begin{aligned} \min_x . \quad & Prob(W^{'}x > T) \\ \text{s.t.} \quad & Mx = b \\ & x \in \{0,1\}^{|A_r|}. \end{aligned} \tag{3}$$

B. Problem Reformulation as Cardinality Minimization

Definition 1: Cardinality is the number of non-zero elements in one vector or matrix. If $x = (x_1, x_2, x_3) = (0, 0, 4)$, then the cardinality of x is 1.

With respect to cardinality optimization, there are usually two typical problems: cardinality minimization problem and cardinality constrained problem [24], which are respectively described as follows:

$$\begin{aligned} \min_x . \quad & Card(x) \\ \text{s.t.} \quad & x \in \mathcal{F} \end{aligned} \tag{4}$$

$$\begin{aligned} \min_x . \quad & f(x) \\ \text{s.t.} \quad & Card(x) \leq \tau \\ & x \in \mathcal{F} \end{aligned} \tag{5}$$

where $f(x)$ is the objective function, x is a vector, τ is a constant, and $\mathcal{F}$ is the feasible set.

Regarding our routing problem, the objective is to minimize the probability of arriving later than the deadline (i.e., Eq.(3)), and statistically speaking, it is equal to minimizing the number of times of arriving later than the deadline if the sampling size for each path is large enough. If we travel 1000 times respectively on two paths from o to d, and there are 20 times being later than deadline for path 1, and 10 times for path 2, then path 2 should be optimal. The problem in this context can be formulated as a cardinality minimization problem, which is to minimize the cardinality of vector C defined as follows:

$$\begin{aligned} C(x) &= (c_1, c_2, \ldots, c_S) \\ &= ([W_1^{'}x - T]^+, [W_2^{'}x - T]^+, \ldots, [W_S^{'}x - T]^+) \end{aligned} \tag{6}$$

where $[\cdot]^+ = \max\{0, \cdot\}$, W_i is the travel time data for the whole graph at $i - th$ time, and S is the travel time data size.

Then our objective to minimize the probability of arriving later than deadline can be approximately formulated as:

$$\begin{aligned} \min_x . \quad & Card(C(x)) \\ \text{s.t.} \quad & Mx = b \\ & x \in \{0,1\}^{|A_r|} \end{aligned} \tag{7}$$

where x is the decision variable, which denotes the optimal path. Now, the original problem to find an optimal path that maximizes the probability of arriving not later than deadline becomes the cardinality minimization problem. In this problem, we could directly use the real detected or sampled travel time data on arcs instead of some distributions to arrive the optimal solution. The hope is that when the sampling size S is large enough, the frequency of not being late will closely reflect the probability of arriving on time, and the solution for this cardinality optimization problem is almost the same as the optimal path that we consider in Eq.(1).

III. Methodology

L_1 norm minimization and mix integer linear programming are respectively introduced in this section. The former is usually used to relax cardinality minimization problems while the latter always solves the former problem efficiently.

A. L_1 Norm Minimization to Solve Cardinality Minimization

Least-squares is usually used in optimization problems, especially the object of which is to minimize the squares of errors. However, for cardinality optimization problem, least-squares method does not perform well. A typical approach to solve cardinality minimization problems is to relax the problem by the L_1 norm, and the relaxed problem can be solved efficiently, where the L_1 norm is usually known as the convex envelop of the function $Card(x)$ [25].

L_1 norm of one vector is usually denoted by $\|\bullet\|_1$, which refers to the absolute sum of its elements. The mathematical form of L_1 norm for vector x is stated as:

$$\|x\|_1 = |x_1| + \cdots + |x_n| \tag{8}$$

where n is the length of x. Accordingly, minimization of L_1 norm for x can be formulated as:

$$\begin{aligned} \min_x . \quad & \|x\|_1 \\ \text{s.t.} \quad & x \in \mathcal{F} \end{aligned} \tag{9}$$

where $\mathcal{F}$ is the feasible set.

Incorporating L_1 norm, Eq.(7) can be reformulated as follows:

$$\begin{aligned} \min_x . \quad & \sum_{i=1}^{S} \xi_i \\ \text{s.t.} \quad & W_1^{'}x - T \leq \xi_1 \\ & \vdots \\ & W_S^{'}x - T \leq \xi_S \\ & Mx = b, \\ & \xi_i \geq 0, \\ & x \in \{0,1\}^{|A_r|}. \end{aligned} \tag{10}$$

where ξ_i is an intermediate variable which refers to the $[W_i^{'}x - T]^+$ in Eq.(6); x is the decision variable which refers to the optimal path.

By analyzing the optimization problem in Eq.(10), we find that the L_1 minimization problem can be easily transformed to a mix integer linear programming problem, for which there already exists mature solutions.

B. Mix Integer Linear Programming

Considering Eq.(10), we can further transform it into the standard form of Mixed Integer Linear Programming (MILP) which is written as follows:

$$\begin{aligned} \min_y . \quad & cy \\ \text{s.t.} \quad & Ay \leq B \\ & A_{eq}y = B_{eq} \\ & L_b \leq y \leq U_b \\ & y(1:|E|) \in \mathbb{Z} \end{aligned} \tag{11}$$

where

$$y = (x_1, \ldots, x_{|A_r|}, \xi_1, \ldots, \xi_S)_{(|A_r|+S)\times 1} \tag{12}$$

$$c = (0_1, \ldots, 0_{|A_r|}, 1_1, \ldots, 1_S)_{1\times(|A_r|+S)} \tag{13}$$

$$A = \begin{pmatrix} W_{11} & W_{12} & \cdots & W_{1|A_r|} & -1 & 0 & \cdots & 0 \\ W_{21} & W_{22} & \cdots & W_{2|A_r|} & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ W_{S1} & W_{S2} & \cdots & W_{S|A_r|} & 0 & 0 & \cdots & -1 \end{pmatrix} \tag{14}$$

$$B = (T, \ldots, T)_{S\times 1} \tag{15}$$

$$A_{eq} = (M, \quad \mathbf{0}_{|V|\times S}) \tag{16}$$

$$B_{eq} = b \tag{17}$$

$$L_b = \mathbf{0}_{(|A_r|+S)\times 1} \tag{18}$$

$$U_b = (\mathbf{1}_{1\times|A_r|}, \infty_{1\times S})_{(|A_r|+S)\times 1}. \tag{19}$$

In this problem, y is the decision variable, and x in Eq.(12) is the optimal path. To solve the MILP efficiently, we use the $intlinprog$ function in Matlab 2014a, which is mainly based on the branch and bound algorithm. The steps are stated as follows [26]:

- Initially reduce the problem size: in this step, linear program pre-processing is performed on the dual problem, the purpose of which is to eliminate the redundant variables and constraints;
- Solve the relaxed linear programming: the interior point method [27] is employed to determine the optimal solution for the relaxed problem, which is an efficient method for the linear programming and guarantees polynomial complexity;
- Tighten the LP relaxation of the mixed-integer problem: mixed-integer program pre-processing is conducted to analyze the linear inequalities and determine whether some bounds can be tightened;
- Further tighten the LP relaxation: cut generation is implemented in this step, where the cuts are some additional linear inequality constraints added to the problem, the function of which is restricting the feasible region of the LP relaxations to make sure the solution are closer to integers;
- Compute the integer-feasible solutions: heuristics are used to find feasible points for the branch and bound step below so that an upper bound on the objective functions can be determined [28]. Heuristics always refers to the methods which are used to speed up the process of finding a satisfactory solution where exhaustive search is impractical;
- Systematically search for the optimal solution: branch and bound method [29] is constructed as a sequence of sub-problems that attempt to converge to a solution of the MILP, where the sub-problems give a sequence of upper and lower bounds on the solution.

In this way, the function can solve the problem in any of the steps. If it solves the problem in one step, $intlinprog$ does not execute the later steps. More details about these steps can be found in [26].

IV. Simulation Results and Analysis

Our approach makes two approximations: (1) we use frequency of not being late on sampling data to approximate the probability of arriving on time; (2) we use L_1 norm to relax the cardinality minimization problem. Therefore, it is necessary to analyze the accuracy of our approach. To this end, we test our approach and compare the results with the exact solution which is obtained by enumerating all possible paths and computing corresponding probability, and then the accuracy is computed as the frequency that the two solutions are the same. Besides, we also compute the accuracy after a tolerance of 3%, which means that we accept the solution by our approach as optimal if the probability difference (the probability here refers to the chance that one path guarantees arriving the destination no later than the deadline) between the actual optimal path and the path provided by our approach is not larger than 3%.

Our approach has three advantages over other methods because it can address different distributions, correlation issue and different deadlines. To better justify these advantages, we classify the experiment into 3 cases with respect to travel time on arc: Case 1 involves single independent distribution; Case 2 involves blended independent distributions; Case 3 involves correlated distributions, where Case 1 and Case 2 help show that our approach is able to handle various or even unknown distributions; Case 3 helps show that our approach is able to address the correlation issue with respect to travel time; In all cases, we test with a large variety of deadlines, which shows that our approach works for different deadlines. Especially, when the data size is 500 (highlighted in 3 tables), most of the accuracy after tolerance for the 3 cases is already larger than 95%. When the data size becomes larger, most of the tolerated accuracy is close to 100%. Moreover, when tested in real-world road networking using real traffic data, our approach also offers satisfactory results.

A. Test Scenario I: Artificial Network with Artificial Data

We test our approach on an artificial network to validate the accuracy of the solution and provide useful insights into the nature of the results. Consider the 65-node, 123-arc network in Fig. 1, which is a directed graph. This is a fairly representative spatial network: (1) the graph contains cycles, and some arcs between two nodes are bi-directional; (2) there are some clusters in this graph and they are connected with each other. These clusters can represent a road network in a city. These two features make the graph consistent with actual traffic network.

Our approach is data-driven, and in real life, we will use the sensor-detected travel time data for each road link. For the testing on this graph, we employ some random distribution functions to generate data for each arc. In the following experiments, we consider Normal, Bi-Normal,

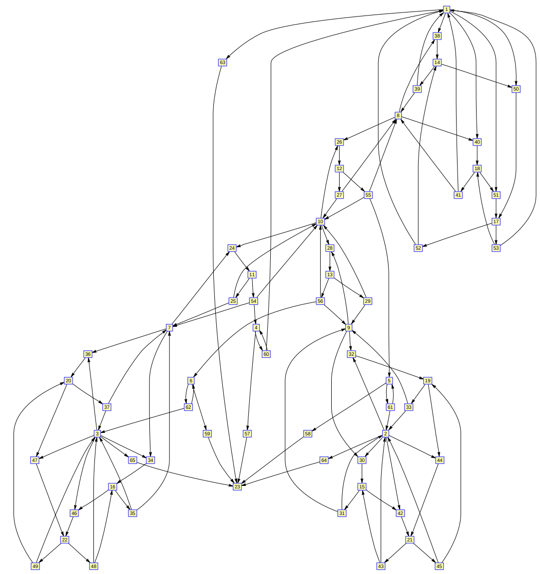

Fig. 1: A 65-node, 123-arc road network

Gamma and Log-normal distribution and their combinations. We randomly select one starting and ending point pair out of ten: {(1, 3), (3, 1), (3, 2), (2, 3), (10, 2), (2, 10), (1, 10), (10, 1), (3, 10), (10, 3)}.

B. Explanation of Symbols and Other Settings

1) S − the value of sampling data size. We take the values of 100, 500, 1000, and 1500.
2) α − the deadline coefficient with respect to T: $T = T_1 + \alpha * (T_2 - T_1)$, where T is the deadline, T_2 is the minimum longest travel time for all paths based on all the generated data, and T_1 is the shortest travel time with respect to the same path, and we take $\alpha = 0.2, 0.4, 0.6, 0.8, 1.0, 1.2$.
3) N_E − the number of times to run our approach to reach one accuracy, which is 1000 in this experiment.
4) N − denotes Normal distribution.
5) Bi − denotes Bi-normal distribution.
6) G − denotes Gamma distribution.
7) L − denotes Log-normal distribution.
8) $N + Bi$ − denotes Normal distribution combined with Bi-normal distribution.
9) $N + G$ − denotes Normal distribution combined with Gamma distribution.
10) $N + L$ − denotes Normal distribution combined with Log-normal distribution.

C. Case Study 1: Single Independent Distribution

For each arc, we generate S data respectively according to the four distributions. For each distribution, the data on different arcs are independent from each other. The results are shown in TABLE I, the structure of which is stated as:

1) the 1^{st} column stands for the types of distributions;
2) the 2^{nd} column stands for the value of deadline coefficient α;

TABLE I: Case 1: Accuracy for independent single distribution (%)

	α	100	500	1000	1500
	0.2	59.1, 90.1	56.3, **99.6**	56.9, 100	57.5, 100
	0.4	63.0, 84.5	62.4, **96.9**	61.1, 99.7	59.0, 99.9
N	0.6	67.5, 83.6	67.6, **96.6**	67.0, 98.9	68.6, 100
	0.8	71.4, 93.5	72.6, **100**	70.6, 100	71.4, 100
	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100
	0.2	57.9, 90.9	60.9, **99.8**	58.5, 100	58.8, 100
	0.4	64.4, 85.9	62.1, **95.6**	64.3, 99.1	62.9, 99.9
Bi	0.6	67.6, 86.6	66.8, **95.9**	69.0, 99.9	67.2, 99.9
	0.8	69.9, 93.1	69.3, **100**	71.5, 100	69.8, 100
	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100
	0.2	60.4, 89.2	57.4, **99.8**	57.6, 100	56.5, 100
	0.4	64.6, 83.1	59.8, **95.1**	63.1, 98.7	63.8, 99.9
G	0.6	65.9, 84.9	68.4, **97.4**	69.3, 99.6	68.7, 99.9
	0.8	68.6, 93.2	69.6, **100**	70.7, 100	73.4, 100
	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100
	0.2	60.4, 89.2	58.7, **99.8**	58.4, 100	59.7, 100
	0.4	62.7, 82.2	60.9, **94.4**	61.7, 99.1	60.0, 99.7
L	0.6	68.2, 84.5	68.8, **98.4**	68.1, 99.6	68.3, 99.9
	0.8	69.9, 94.8	71.6, **100**	71.0, 100	71.6, 100
	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100

3) In each of the $3^{rd} - 6^{th}$ column, there are two sub-columns. The first sub-column is for the accuracy and and the second sub-column is for the accuracy with 3%-tolerance.

Looking into each distribution in Table I, we can see that the accuracy always increases (until up to 100%) as the deadline becomes longer. When α is not less than 1, which corresponds to the situation that there exists at least one path guaranteeing arriving on time with 100% probability, our approach can always correctly find the actual optimal path. It is expected since the L_1 norm minimization in our approach is to minimize the total delay with respect to deadline T. If the deadline T is large enough, there always exists at least one path with zero total delay. When α is smaller than 1, i.e., 0.2, 0.4, 0.6, 0.8, which corresponds to the situation that there is no path that can guarantee arriving on time with 100%, the accuracy always falls between 50%-80%. It is not comparatively high because L_1 norm minimization is only an approximation of the cardinality minimization. However, the result by our approach is always close to the actual optimal solution because the 3%-tolerance accuracy is satisfactorily high, most of which are above 95% and close to 100%. Since in real life travel planning, people may not be concerned about a 3% gap regarding the actual optimal path, so our approach is quite acceptable in this sense.

Regarding the data size, we can see that the accuracy dose not necessarily increase with it, this is still because of the L_1 norm minimization. However, the results become closer to the actual optimal solutions since from the table, the accuracy with a 3%-tolerance always increases (until up to 100%) as the data size becomes larger. This happens because the L_1 norm minimization can better approximate the cardinality minimization although it is not the exact one. When the data size is not less than 1000, the accuracy with tolerance is almost 100%.

TABLE II: Case 2: Accuracy for blended independent distribution (%)

	α	100	500	1000	1500
	0.2	76.8, 96.0	79.1, **100**	69.7, 100	78.2, 100
	0.4	79.8, 92.3	82.7, **99.2**	72.8, 99.4	81.2, 100
N	0.6	80.4, 91.4	83.3, **99.2**	78.4, 99.6	83.3, 999
+	0.8	76.9, 95.0	80.8, **99.8**	78.4, 100	83.9, 100
Bi	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100
	0.2	61.4, 89.8	58.6, **100**	54.8, 100	55.6, 100
	0.4	65.1, 84.2	63.0, **96.7**	62.4, 98.9	62.6, 99.8
N	0.6	65.5, 83.7	66.9, **97.3**	69.5, 99.7	68.9, 100
+	0.8	68.2, 92.6	70.9, **100**	71.4, 100	73.1, 100
G	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100
	0.2	76.0, 96.1	76.1, **100**	77.2, 100	78.5, 100
	0.4	82.0, 92.5	82.0, **98.9**	81.6, 99.9	80.3, 100
N	0.6	83.6, 93.6	83.6, **99.0**	82.9, 100	84.4, 100
+	0.8	83.5, 96.0	83.5, **100**	84.1, 100	86.0, 100
L	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100

Another notable result is that the four different distributions share one similar pattern for the accuracy and 3%-tolerance accuracy under different deadlines and data sizes. The main reason is because the proposed approach is based on real data, and Eq.(10) only takes the collected data into account.

Based on above analysis, we can conclude that our approach is able to handle different independent distributions with different deadlines. Especially, when data size is large, our approach can achieve accuracy with a reasonable tolerance.

D. Case Study 2: Blended Distributions

For each arc, we adopt combinations of distributions to generate S data. We first use the sequence in the matrix M to order the arcs. Then, each time, the odd arcs use one distribution, and the even arcs will use a different one. The combinations are set as follows: Normal combined with Binormal (N+Bi), Normal combined with Gamma (N+G), and Normal combined with Log-normal (N+L). We also assume that the data on different arcs are independent from each other. Additionally, the accuracy and 3%-tolerance accuracy are shown in Table II.

Compared with the results for Case 1, the accuracy and 3%-tolerance accuracy for Case 2 share similar pattern under different deadlines and data size. From Table II, our approach can accommodate the blind distributions with different deadlines.

E. Case Study 3: Correlated Distributions

For each arc, we generate S data respectively according to the four distributions. Then we randomly choose some adjacent arc pairs, and the travel time data on which would be correlated with each other. Additionally, the accuracy and 3%-tolerance accuracy are shown in Table III.

Compared with the results for Case 1 and Case 2, the accuracy and 3%-tolerance accuracy for Case 3 also share a

TABLE III: Case 3: Accuracy for correlated single distribution (%)

	α	100	500	1000	1500
	0.2	57.1, 89.9	57.3, **99.6**	57.2, 100	56.3, 100
	0.4	64.1, 85.0	61.4, **96.8**	63.5, 99.2	60.2, 99.9
N	0.6	69.0, 85.4	69.0, **97.1**	69.3, 99.6	67.6, 99.8
	0.8	73.1, 92.9	72.1, **99.8**	69.3, 100	72.0, 100
	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100
	0.2	60.1, 89.4	59.4, **99.7**	58.9, 100	55.3, 100
	0.4	61.5, 82.1	61.9, **96.2**	62.6, 99.2	60.0, 100
Bi	0.6	68.7, 85.7	64.4, **97.0**	68.9, 99.3	66.6, 99.9
	0.8	73.6, 92.5	69.9, **100**	71.5, 100	74.8, 100
	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100
	0.2	59.0, 88.5	58.1, **99.6**	58.4, 100	57.8, 100
	0.4	65.5, 84.3	63.0, **96.5**	62.2, 99.2	63.9, 99.5
G	0.6	65.1, 81.8	69.2, **97.4**	66.5, 99.4	71.0, 99.9
	0.8	71.8, 93.9	71.8, **100**	70.8, 100	71.8, 100
	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100
	0.2	59.6, 88.5	57.9, **99.2**	58.1, 100	58.4, 100
	0.4	61.4, 81.3	66.4, **94.6**	65.5, 99.6	63.1, 100
L	0.6	67.1, 83.9	68.2, **99.1**	67.3, 99.7	68.7, 100
	0.8	70.6, 93.9	73.0, **99.9**	71.5, 100	73.9, 100
	1.0	100 , 100	100 , **100**	100, 100	100, 100
	1.2	100 , 100	100 , **100**	100, 100	100, 100

similar pattern under different deadlines and data size. From Tables III, our approach can address the correlation issue well.

F. *Time Complexity*

In all previous cases, to determine whether our approach can achieve the optimal path, we use the enumerating method to compute the real optimal one. To show that our approach is more efficient with respect to time complexity, we record all the running time for above experiments, which ran on a computer with Inter Core i7-3540M processor and 8.00 GB RAM. And the average running time regarding different travel time data sizes are shown in Table IV.

TABLE IV: Time complexity for the enumerating method and our approach (s)

	100	500	1000	1500
Enumerating method	0.6419	0.6389	0.5993	0.6108
Our approach	0.0546	0.2042	0.4052	0.4452

From Table IV we see that the average running time for our approach is always shorter than that of the enumerating method with respect to different travel time data sizes. The important reason is that the *MILP* can be solved more smartly compared with enumerating methods. Although the running time always increases with travel time data size in our approach, we do not necessarily need a very large set of travel time data to obtain the satisfactory solution based on the conclusions for the 3 cases we studied, which means that we can obtain the satisfactory path faster than the enumerating methods.

G. *Test Scenario II: Real Traffic Data on Munich City*

To better justify our method, we also use real traffic data to perform the test on one area of Munich city, which is shown

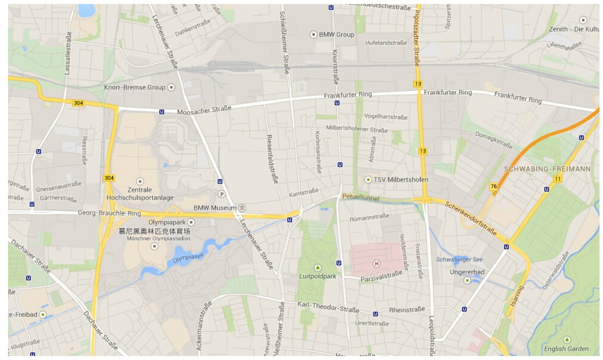

Fig. 2: One area of the Munich city, Germany

in the Fig. 2. The underlying graph of this area includes 270 nodes and 277 arcs. The experiment settings are similar with that in Scenario I, the major difference is that the travel time is the real traffic data, so there is no distribution or correlation assumptions in Scenario II. The results regarding the accuracy are provided in Table V.

TABLE V: Real test on Munich city using real traffic data(%)

α	100	500	1000	1500
0.2	86.0, 95.3	77.2, **99.8**	74.2, 100	75.4, 100
0.4	81.2, 96.8	70.0, **100**	68.7, 100	67.7, 100
0.6	74.8, 99.2	65.8, **100**	61.5, 100	63.8, 100
0.8	66.8, 99.5	61.0, **100**	61.7, 100	58.6, 100
1.0	100 , 100	100 , **100**	100, 100	100, 100
1.2	100 , 100	100 , **100**	100, 100	100, 100

In addition, the average time complexity regarding different travel time data sizes are displayed in Table VI.

TABLE VI: Time complexity for the enumerating method and our approach on real data and real map (s)

	100	500	1000	1500
Enumerating method	0.0703	0.0644	0.0607	0.0528
Our approach	0.0097	0.0248	0.0487	0.0684

Compared the Table V and Table VI with the results for Scenario I, it is evident to tell that Scenario II shares the similar pattern with Scenario I with respect to the accuracy and the time complexity. Then it is reasonable to evaluate the results in Scenario II by the analysis of Scenario I, which also means that our approach is applicable for the real world routing problem.

In summary, we can conclude that our approach works well for various typical situations. Three advantages are that it can handle the distribution, correlation and deadline issues efficiently. Additionally, the proposed approach can guarantee a comparatively high accuracy. More importantly, our approach can almost achieve 100% accuracy if we tolerate an error of 3.0% when the sampling data size is comparatively large, i.e., not less than 1000. This claim is reasonable since in real traffic planning, people may not be concerned much on the difference between actual optimal path and the path close to an optimal one. Besides, regarding the time complexity, our approach is more efficient than enumerating method within the same road network. Last but not least, the proposed approach is satisfactorily applicable to both the artificial and the real world routing service.

V. Conclusions and Future Work

This paper aims at solving a stochastic shortest path problem. The objective is to determine an optimal path that maximizes the probability of arriving on time. We have transformed the problem into a cardinality minimization problem, and further used an L_1 technique to solve the problem. The simulation results on artificial and real-world road networks have shown that the algorithm works well under a variety of distributions. The performance is not affected even when we consider the travel time dependencies. Moreover, it can solve the problem with different deadlines.

In the future, we will improve the algorithm in the following aspects. Firstly, the computation complexity in MILP is still high. We will improve the computation complexity, e.g., using approximate/heuristic algorithms. Secondly, we will theoretically study the proper amount of sampling data size that guarantees satisfactory results, based on for example the Chernoff-Hoeffding Bound [30]. Thirdly, extensive experiments on real large-scale road networks will be conducted.

References

[1] Y. Fan, R. Kalaba, and J. Moore II, "Arriving on time," *Journal of Optimization Theory and Applications*, vol. 127, no. 3, pp. 497–513, 2005.

[2] Y. Fan and Y. Nie, "Optimal routing for maximizing the travel time reliability," *Networks and Spatial Economics*, vol. 6, no. 3-4, pp. 333–344, 2006.

[3] E. D. Miller-Hooks and H. S. Mahmassani, "Least expected time paths in stochastic, time-varying transportation networks," *Transportation Science*, vol. 34, no. 2, pp. 198–215, 2000.

[4] B. C. Dean, "Algorithms for minimum-cost paths in time-dependent networks with waiting policies," *Networks*, vol. 44, no. 1, pp. 41–46, 2004.

[5] S. T. Waller and A. K. Ziliaskopoulos, "On the online shortest path problem with limited arc cost dependencies," *Networks*, vol. 40, no. 4, pp. 216–227, 2002.

[6] E. Nikolova, J. A. Kelner, M. Brand, and M. Mitzenmacher, "Stochastic shortest paths via quasi-convex maximization," in *Algorithms–ESA 2006*, pp. 552–563, Springer, 2006.

[7] E. Nikolova, "Approximation algorithms for reliable stochastic combinatorial optimization," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 338–351, Springer, 2010.

[8] H. Frank, "Shortest paths in probabilistic graphs," *Operations Research*, vol. 17, no. 4, pp. 583–599, 1969.

[9] Y. M. Nie and X. Wu, "Shortest path problem considering on-time arrival probability," *Transportation Research Part B: Methodological*, vol. 43, no. 6, pp. 597–613, 2009.

[10] E. Nikolova, J. A. Kelner, M. Brand, and M. Mitzenmacher, "Stochastic shortest paths via quasi-convex maximization," in *Algorithms–ESA 2006*, pp. 552–563, Springer, 2006.

[11] S. Lim, H. Balakrishnan, D. Gifford, S. Madden, and D. Rus, "Stochastic motion planning and applications to traffic," in *Algorithmic Foundation of Robotics VIII*, pp. 483–500, Springer, 2009.

[12] J. G. Wardrop, "Road paper. some theoretical aspects of road traffic research.," in *ICE Proceedings: Engineering Divisions*, vol. 1, pp. 325–362, Thomas Telford, 1952.

[13] A. Polus, "A study of travel time and reliability on arterial routes," *Transportation*, vol. 8, no. 2, pp. 141–151, 1979.

[14] M. Mogridge and S. Fry, "Variability of car journey times on a particular route in central london," *Traffic Engineering & Control*, vol. 25, no. HS-038 005, 1984.

[15] F. Montgromery and A. May, "Factors affecting travel times on urban radial routes," *Traffic engineering & control*, vol. 28, no. 9, 1987.

[16] H. Rakha, I. El-Shawarby, M. Arafeh, and F. Dion, "Estimating path travel-time reliability," in *Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE*, pp. 236–241, IEEE, 2006.

[17] K. Chen, L. Yu, J. Guo, and H. Wen, "Characteristics analysis of road network reliability in beijing based on the data logs from taxis," in *Transportation Research Board 86th Annual Meeting*, no. 07-1420, 2007.

[18] B. M. Fastenrath Ulrich, "process for selection of a route for a dynamic navigation on private transport," 2008.

[19] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[20] N. Kallus, "From predictions to data-driven decisions using machine learning," *arXiv preprint arXiv:1402.5481*, 2014.

[21] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[22] D. Solomatine, "Applications of data-driven modelling and machine learning in control of water resources," *Computational intelligence in control*, pp. 197–217, 2002.

[23] J. Cheng, S. Kosuch, and A. Lisser, "Stochastic shortest path problem with uncertain delays.," in *ICORES*, pp. 256–264, 2012.

[24] M. J. Abdi, *Cardinality optimization problems*. PhD thesis, University of Birmingham, 2013.

[25] Stanford University, *L1-norm Methods for Convex-Cardinality Problems*, 2008. http://www.youtube.com/watch?v=DsXzUU691ts.

[26] MathWorks, *Mixed-Integer Linear Programming Algorithms*, 2014. http://www.mathworks.com/help/optim/ug/mixed-integer-linear-programming-algorithms.html.

[27] Y. Zhang, "Solving large-scale linear programs by interior-point methods under the matlab environment," *Optimization Methods and Software*, vol. 10, no. 1, pp. 1–31, 1998.

[28] E. Danna, E. Rothberg, and C. Le Pape, "Exploring relaxation induced neighborhoods to improve mip solutions," *Mathematical Programming*, vol. 102, no. 1, pp. 71–90, 2005.

[29] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization*, vol. 18. Wiley New York, 1988.

[30] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American statistical association*, vol. 58, no. 301, pp. 13–30, 1963.