An Accurate Solution To The Cardinality-based Punctuality Problem

Zhiguang Cao[1], Yaoxin Wu[2], Akshay Rao[3], Felix Klanner[3], Stefan Erschen[3],
Wei Chen[1,*], Le Zhang[4], Hongliang Guo[5].

***Abstract*—This paper focuses on a specific stochastic shortest path (SSP) problem, namely the punctuality problem. It aims to determine a path that maximizes the probability of arriving at the destination before a specified deadline. The popular solution to this problem always formulates it as a cardinality minimization problem by considering its data-driven nature, which is approximately solved by the ℓ_1-norm relaxation. To address this problem accurately, we consider the special character in the cardinality-based punctuality problem and reformulate it by introducing additional variables and constraints, which guarantees an accurate solution. The reformulated punctuality problem can be further transformed into the standard form of integer linear programming (ILP), thus, can be efficiently solved by using the existing ILP solvers. To evaluate the performance of the proposed solution, we provide both theoretical proof of the accuracy, and experimental analysis against the baselines. Particularly, the experimental results show that in the following two scenarios, 1) artificial road network with simulated travel time, 2) real road network with real travel time, our accurate solution works better than others regarding the accuracy and computational efficiency. Furthermore, three ILP solvers, i.e., CBC, GLPK and CPLEX, are tested and compared for the proposed accurate solution. The result shows that CPLEX has obvious advantage over others.**

I. INTRODUCTION

Stochastic shortest path (SSP) problem has been researched broadly in the field of intelligent transportation systems, which is usually applied in route planning and navigation [1], [2]. Unlike traditional shortest path problem defined in a static environment, stochastic shortest path (SSP) problem takes the uncertainties of traffic conditions (e.g. caused by demand fluctuation, weather condition, or accidents) into account [3], [4]. Solving the SSP problem attracts extensive attention from both academy and industry, since it is directly related with people's daily life [5], [6].

In SSP problem, the criteria for optimal path are not unique due to the uncertain traffic condition and user demands [7]. Several optimal criteria are popular for the SSP problem, such as LET (least expected travel time) [8], [9], [10], [11], mean-risk model (minimal weighted combination of expected value and standard deviation for travel time) [12], [13], and probability tail model (maximal probability of punctuality) [14], [15], [16], [17]. Among the three criteria, probability tail model is promising in that, 1) the probability can better measure the stochastic nature of the random traffic [18], 2) it takes into account the specific user demand, i.e., a user-defined deadline. In addition, the punctuality criterion is more consistent with people's routing behavior, and extensively applied in many scenarios, such as attending an important meeting, catching flights, fire rescue and organ delivery. One common query could be that "I want to reach the airport in 45 minutes. Please recommend a path with maximum chance to arrive punctually". Thus, in this paper, we concentrate on the SSP problem, which considers punctuality as the optimal criterion.

Several solutions have been presented to solve the punctuality problem, which aims to find a path with maximum probability to arrive punctually. Fan et al. [14] solve an SSP problem by incorporating the probability tail model, which adaptively determines the road link to visit instead of a prior specified path. In their work, the maximum probability of punctual arrival is described by a class of *unknown* functions. The functions are estimated through Picard method of successive approximations. This way, the maximum probability of punctuality can be estimated, which avoids enumerating all candidate paths. To improve computational efficiency, they introduce Laplace transform to simplify the computation on the evaluation of convolution integrals, which is a main step in successive approximation. Christman et al. present a dynamic programming solution for the punctuality problem, considering cycling and waiting policy as well [19]. Although they claim satisfactory performance for the proposed solution, the computational efficiency and accuracy are not demonstrated accordingly in their work.

To better exploit the existing optimization technique, Nikolova et al. [12] consider an SSP problem aiming to find a path with maximum probability of not exceeding the given cost (i.e., the probability tail model based SSP problem applies if travel time is considered as the cost). Assuming that the cost distribution for each link is normally distributed and independent, Nikolova et al. propose an efficient algorithm based on the quasi-convex optimization. However, in their solution, the predefined cost needs to be strictly higher than the average cost of the paths with smallest-mean-cost. Specifically, if the departure time is close to the deadline, the predefined cost will be lower than the cost of all paths, then the problem formulation will lose the nature of quasi-convex. As a result, the computation for the optimal path would be inefficient.

*Dr. Wei Chen is the Corresponding Author to this article.

[1]Zhiguang Cao and Wei Chen are with the School of Automation, Guangdong University of Technology, China (zhiguangcao@gdut.edu.cn, chenwei@gdut.edu.cn).

[2]Yaoxin Wu is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore (Email: wyxacc@hotmail.com).

[3]Akshay Rao, Felix Klanner, and Stefan Erschen are with the BMW@NTU Future Mobility Research Lab, and the Energy Research Institute @ NTU(ERI@N), Nanyang Technological University, Singapore (Email: akshay@ntu.edu.sg, Felix.Klanner@bmw.de, stefan.erschen@gmx.net).

[4]Le Zhang is with the Advanced Digital Sciences Center (ADSC), the Singapore-based research center of the University of Illinois at Urbana-Champaign (UIUC) (zhang.le@adsc.com.sg).

[5]Hongliang Guo is with the School of Automation Engineering, University of Electronic Science and Technology of China, China (guohl1983@uestc.edu.cn).

Lim et al. [20] propose a stochastic route planning algorithm for traffic navigation. Given a pair of origin and destination, the algorithm can derive a path which minimizes the cost function expressed by a delay probability distribution. Thus, the equivalent problem is seeking a path with maximum probability of punctual arrival. Generally, that problem can not be solved by the classic shortest path algorithms, as the optimal substructure property does not hold for the criterion of probability tail model. Hence, they address the problem by developing a novel method also based on the quasi-convex optimization, which adopts a cutting technique to consider the mean and variance of travel time on all relevant paths. Lim et al. justify the proposed method by means of both simulation data and real traffic data, the latter of which is collected by a group of probe taxi with GPS sensors. The experimental result demonstrate that the proposed algorithm is more efficient than the method provided in [12].

However, the solution in [20] relies on several strong assumptions, such as Gaussian-like and time-independent travel time, and relatively loose deadline, which can not always hold in real-world traffic condition. To circumvent those strong assumptions, Cao et al. [16] propose a data-driven solution to express the punctuality problem in the form of cardinality minimization, which can be approximately solved via the ℓ_1-norm relaxation technique. Moreover, the relaxed problem can be further formulated as a mixed integer linear programming (MILP) problem, thus solved by the canonical solvers. Although this approach avoids the prior assumptions, in the meanwhile, it decreases the accuracy in finding the real optimal path.

As stated previously, the punctuality-based SSP problem is always applied in many critical cases, e.g., fire rescue or organ delivery, even 1% increment in accuracy may imply 1% increment in the chance of saving a life. Thus, an accurate solution in the punctuality problem is much more desirable than an approximate solution. Therefore, it is significant to develop new approach to achieve the accurate solution. Although the general approach to solve the cardinality minimization is using the ℓ_1-norm relaxation as demonstrated in [16], it is feasible to develop accurate approach by looking into the special character of this cardinality-based punctuality problem. In this paper, we propose a novel approach by analysing the subtle relation among the variables inside the cardinality-based punctuality problem. More specifically, we reformulate the punctuality problem as an integer linear programming (ILP) problem by introducing additional variables and constraints, which can be addressed by using the existing ILP solvers. Besides, we provide theoretical proof of the accuracy and experimental evaluation of the proposed approach. Additionally, we also test and compare several popular ILP solvers for the accurate solution.

The remainder of the paper is organized as follows. In Section II, we state the original punctuality problem and formulate it as cardinality minimization. In Section III, we first present the approximate solution, which utilizes the ℓ_1-norm relaxation to solve the cardinality minimization. Then we develop new approach to accurately solve the cardinality-based punctuality problem. In Section IV, we evaluate the performance of the proposed accurate solution by theoretical proof and experimental validation. In addition, three popular ILP solvers based on branch and bound are tested for the proposed approach. In Section V, we discuss some issues related to the accurate solution. In Section VI, we state the conclusion and future works.

II. The Data-Driven Punctuality Problem

In this section, we first describe a road network as a graph, based on which, we provide the formulation of a punctuality problem and present the typical data-driven interpretation. It is noteworthy that, the data-driven approach leverages samples of link travel times to seek the optimal path, where the sample value represents transit time on the corresponding road link. For instance, a vehicle took 200 seconds to traverse a road link, and another vehicle took 240 seconds to traverse the same road link. Those travel time samples are generally difficult to obtain directly. Nevertheless, they can be acquired by processing the corresponding GPS trajectory data which come into being when vehicles traverse the road links [21].

We start with a directed graph $G = (\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ is the set of nodes, representing road intersections, and $\mathcal{L}$ is the set of edges, representing road links. Besides, we have o,$d \in \mathcal{V}$, representing the origin and the destination, respectively. Thus, the punctuality problem can be formulated as below [19], [16]:

$$\max_{\vec{\mathbf{x}}} Prob(\vec{\mathbf{w}}^\top \vec{\mathbf{x}} \leq \tau) \left| \begin{array}{l} \mathbf{M}\vec{\mathbf{x}} = \vec{\mathbf{b}}; \\ \vec{\mathbf{x}} \in \{0,1\}^{|\mathcal{L}|}, \end{array} \right. \tag{1}$$

where $\vec{\mathbf{w}}$ is the random vector containing the travel times for the road links; $\mathbf{M}$ is the node-arc incidence matrix of G; τ is the user-defined time-to-deadline[1]; $\vec{\mathbf{b}}$ is the o-d (i.e., origin-destination) vector, all elements of which are zeros except for the origin o ("1") and the destination d ("-1"); $\vec{\mathbf{x}}$ is the decision variable, representing the set of road links where the element will be "1" if the corresponding road link is on the chosen connected path; and $|\mathcal{L}|$ refers to the quantity of links in the road network. The equality constraint guarantees that $\vec{\mathbf{x}}$ is a connected path from the origin o to the destination d [16].

With respect to Eq. (1), some similar research has been done. Nie et al. [22] address an SSP problem with on-time arrival reliability (SPOTAR), which aims to find the path that minimizes the time budget given a desired probability. Evolving from the dominance criteria that Miller-Hooks et al. proposed for the LET path in [8] and [23], Nie et al. adapt the label-correcting algorithm and its approximation method to solve SPOTAR and the corresponding time-dependent version. Further, Wu et al. explore the schemes of convolution and non-dominated path reduction for better computation and approximation [24], [25]. Although the method in [22] has achieved desirable results in the practical traffic networks [26], [27], some unsatisfactory assumptions on distribution and correlation of link travel times, still exist, which may prevent its wide application. Therefore, it is desirable to leverage a data-driven approach, which may help to get rid of the undesirable assumptions.

[1] It means the time budget with respect to the user-defined deadline.

Generally, the optimization problem in Eq. (1) is neither convex nor quasi-convex if there is no any further assumption on the distribution and correlation of link travel times, or the degree of deadline. Under the circumstances, there is no efficient way to settle this problem optimally. Alternatively, this problem can be re-interpreted from a data-driven perspective. Before presenting the data-driven solution, we first rewrite the "maximizing" problem in Eq. (1) as "minimizing", which can be formulated as follows [16]:

$$\min_{\vec{\mathbf{x}}} Prob(\vec{\mathbf{w}}^\top \vec{\mathbf{x}} > \tau) \left| \begin{array}{l} \mathbf{M}\vec{\mathbf{x}} = \vec{\mathbf{b}}; \\ \vec{\mathbf{x}} \in \{0,1\}^{|\mathcal{L}|}. \end{array} \right. \tag{2}$$

It is clear that the punctuality objective in Eq. (2) is minimizing the probability of reaching the destination beyond the deadline, which is equivalent to that of Eq. (1). At the same time, it is also equivalent to minimizing the number of times of arriving at the destination beyond the deadline if a driver travelled a lot of times with respect to the same o-d. For example, we assume a driver travelled 2000 times on two candidate paths, e.g., path A and path B, from o to d. On path A, the frequency of reaching d beyond the deadline is 50, and on path B it is 20. In this case, path B would be regarded as the optimal one in comparison with path A. Thus, the original problem can be reformulated as a cardinality[2] minimization problem, which is intended to minimize the cardinality of $C(\vec{\mathbf{x}})$, i.e., the vector represented as below:

$$\begin{aligned} C(\vec{\mathbf{x}}) &= (c_1, c_2, \ldots, c_N) \\ &= \Big([\vec{\mathbf{w}}_1^\top \vec{\mathbf{x}} - \tau]^+, [\vec{\mathbf{w}}_2^\top \vec{\mathbf{x}} - \tau]^+, \ldots, [\vec{\mathbf{w}}_N^\top \vec{\mathbf{x}} - \tau]^+\Big), \end{aligned} \tag{3}$$

where $[\cdot]^+ = \max\{0, \cdot\}^3$; each element in $\vec{\mathbf{w}}_i$ denotes the i_{th} travel time sample on the corresponding edge; N denotes the quantity of travel time samples on each edge. Each element of vector $C(\vec{\mathbf{x}})$ means a comparison between one historical path travel time and the deadline. Once travel time $\vec{\mathbf{w}}_i^\top \vec{\mathbf{x}}$ is less than deadline τ (i.e., punctual arrival), the corresponding element will be 0. Therefore, the cardinality minimization of $C(\vec{\mathbf{x}})$ means minimizing the frequency of being late (i.e., non-zero elements). Note that the cardinality minimization approach is data-driven naturally, which has no need for any specific distribution, correlation of link travel time or time-to-deadline. As a consequence, the data-driven punctuality problem is described as follows:

$$\min_{\vec{\mathbf{x}}} Card(C(\vec{\mathbf{x}})) \left| \begin{array}{l} \mathbf{M}\vec{\mathbf{x}} = \vec{\mathbf{b}}; \\ \vec{\mathbf{x}} \in \{0,1\}^{|\mathcal{L}|}, \end{array} \right. \tag{4}$$

where $Card(\cdot)$ is the cardinality function.

III. Solutions to the Data-Driven Punctuality Problem

In this section, we first present the typical approximate solution, which employs the ℓ_1-norm relaxation technique to convert the cardinality minimization problem into a mixed integer linear programming (MILP) problem. Then, by introducing additional variables and constraints, we reformulate the approximate MILP problem as an integer linear programming (ILP) problem, which leads to an accurate solution to the original punctuality problem.

A. Approximate Solution Expression

Previously, the data-driven punctuality problem is formulated as a cardinality minimization problem. In general, cardinality minimization is neither categorised as convex nor quasi-convex optimization, and it is often solved by the ℓ_1-norm relaxation technique [16]. The relaxed problem can be addressed effectively, where ℓ_1-norm is commonly known as convex envelop of the function $Card(\vec{\mathbf{x}})$.

The ℓ_1-norm of a vector is often denoted by $\|\cdot\|_1$, and it means absolute sum of all elements. For instance, the ℓ_1-norm of a vector $\vec{\mathbf{x}}$ can be represented as below:

$$\|\vec{\mathbf{x}}\|_1 = |x_1| + \cdots + |x_n|, \tag{5}$$

where n denotes the size of $\vec{\mathbf{x}}$. Accordingly, the minimization for ℓ_1-norm of $\vec{\mathbf{x}}$ can be expressed as follows:

$$\min_{\vec{\mathbf{x}}} \|\vec{\mathbf{x}}\|_1 \quad \Big| \quad \vec{\mathbf{x}} \in \mathcal{F}, \tag{6}$$

where $\mathcal{F}$ is a feasible set. Before presenting the approximation into the data-driven solution, we first reformulate the Eq. (4) as follows:

$$\min_{\vec{\mathbf{x}}} Card(\vec{\xi}) \left| \begin{array}{l} \vec{\mathbf{w}}_i^\top \vec{\mathbf{x}} - \tau \le \xi_i, i = 1, 2, ...N; \\ \mathbf{M}\vec{\mathbf{x}} = \vec{\mathbf{b}}, \xi_i \ge 0, \vec{\mathbf{x}} \in \{0,1\}^{|\mathcal{L}|}, \end{array} \right. \tag{7}$$

where ξ_i refers to the potential delay in Eq. (3), which is also a decision variable. After that, the approximate solution with the ℓ_1-norm relaxation can be expressed as follows:

$$\min_{\vec{\mathbf{x}}} \sum_{i=1}^{N} \xi_i \left| \begin{array}{l} \vec{\mathbf{w}}_i^\top \vec{\mathbf{x}} - \tau \le \xi_i, i = 1, 2, ...N; \\ \mathbf{M}\vec{\mathbf{x}} = \vec{\mathbf{b}}, \xi_i \ge 0, \vec{\mathbf{x}} \in \{0,1\}^{|\mathcal{L}|}. \end{array} \right. \tag{8}$$

In fact, Eq. (8) can be formulated as the form of MILP and then addressed by existing solvers. However, the solution has low accuracy because the potential delay (i.e., ξ_i) is random. For instance, assuming the potential delay vector $\vec{\xi}$ of path 1 and 2 are $\vec{\xi}_1$=[0,3,4,0] and $\vec{\xi}_2$=[1,2,2,1], respectively, Eq. (8) will take path 2 for the better one since the element sum of $\vec{\xi}_2$ is less. However, path 1 has a higher chance to guarantee punctual arrival because $\vec{\xi}_1$ has more zero elements, which indicates more punctual arrivals. Therefore, an accurate solution is desired for the cardinality minimization problem to maximize the chance of punctual arrival.

B. Accurate Solution Expression

An accurate solution is feasible by looking into the relationship of all variables inside the cardinality-based punctuality problem. Intuitively, if we turn the non-zero elements in potential delay vector $\vec{\xi}$ into "1", and calculate the ℓ_1-norm of the new binary vector, the obtained result will be equal to the cardinality of vector $\vec{\xi}$ (i.e., the sum of elements). For example, when the potential delay vector $\vec{\xi}$ of a candidate path

[2] Cardinality refers to the amount of non-zero elements in a vector, e.g., card([0,4,3,0])=2.

[3] $[\cdot]^+$ returns the larger component between 0 and the number in bracket, e.g.,$[3]^+$=max{0,3}, and $[-1]^+$=max{0,-1}=0.

is [0,3,4,0], we can convert it into $\vec{\theta}$=[0,1,1,0] so that the ℓ_1-norm of $\vec{\theta}$ will be 2, which is exactly the cardinality of $\vec{\xi}$, namely [0,3,4,0]. In view of this, we can turn the the potential delay vector $\vec{\xi}$ into binary vector $\vec{\theta}$ and replace them in the objective of Eq. (7) with ℓ_1-norm of $\vec{\theta}$, and then we can get the similar form of Eq. (8). To elaborate the transformation and arrive at the form of accurate solution, we first re-transform Eq. (8) as follows:

$$\min_{\vec{\mathbf{x}}} \sum_{i=1}^{N} \xi_i \left| \begin{array}{l} \vec{\mathbf{w}}_i^\top \vec{\mathbf{x}} - \xi_i \le \tau, i = 1, 2, ...N; \\ \mathbf{M}\vec{\mathbf{x}} = \vec{\mathbf{b}}; \xi_i \ge 0, \vec{\mathbf{x}} \in \{0,1\}^{|\mathcal{L}|}, \end{array} \right. \tag{9}$$

The logic is stated as follows: Suppose $\vec{\mathbf{w}}_i^\top \vec{\mathbf{x}}$ is larger than the deadline τ, which means that a delay happened, and ξ_i is larger than 0, say 3. In this case, we actually would like to minimize a "1" in the objective function. On the other hand, if $\vec{\mathbf{w}}_i^\top \vec{\mathbf{x}}$ is smaller than the deadline τ, then we want to minimize a "0" in the objective function. Therefore, we introduce the new variable $\vec{\theta} = \{0,1\}^N$. At the same time, we introduce a big value V, say, 10^{10}, then the accurate solution can be formulated by:

$$\min_{\vec{\mathbf{x}}} \sum_{i=1}^{N} \theta_i \left| \begin{array}{l} \vec{\mathbf{w}}_i^\top \vec{\mathbf{x}} - \theta_i \cdot V \le \tau, i = 1, 2, ...N; \\ \mathbf{M}\vec{\mathbf{x}} = \vec{\mathbf{b}}; \vec{\theta} \in \{0,1\}^N, \vec{\mathbf{x}} \in \{0,1\}^{|\mathcal{L}|}, \end{array} \right. \tag{10}$$

where the $\vec{\theta}$ and $\vec{\mathbf{x}}$ are the decision variables. According to Eq. (10), if there is a delay, it means that $\vec{\mathbf{w}}_i^\top \vec{\mathbf{x}} > \tau$, and θ_i has to be 1 to keep the first constraint feasible; if there is no delay, it means that $\vec{\mathbf{w}}_i^\top \vec{\mathbf{x}} \le \tau$, and θ_i will be forced to be 0 (although θ_i=1 also keeps the first constraint feasible, the objective function by minimization will force it to be 0). As a consequence, the MILP problem in Eq. (8) has been converted into an ILP problem, in which a real accurate solution can be achieved.

C. Accurate Solution Expressed as Integer Linear Programming

By analysing the structure of the cardinality-based punctuality problem, and introducing empirical variables and constraints into the approximate solution, the punctuality problem in Eq. (4) can be finally converted into an accurate solution expressed as an ILP problem in Eq. (10). To make it clear, we will transform Eq. (10) into a standard form of ILP, which can be expressed as follows:

$$\min_{\vec{\mathbf{z}}} \vec{\mathbf{f}}^\top \vec{\mathbf{z}} \left| \begin{array}{l} \mathbf{A}\vec{\mathbf{z}} \le \mathbf{B};\ \mathbf{A}_{\mathrm{e}}\vec{\mathbf{z}} = \mathbf{B}_{\mathrm{e}}; \\ \vec{\mathbf{l}} \le \vec{\mathbf{z}} \le \vec{\mathbf{u}};\ \vec{\mathbf{z}} \in \{0,1\}^{|\mathcal{L}|}, \end{array} \right. \tag{11}$$

where

$$\vec{\mathbf{z}} = \left[\begin{array}{cccccc} x_1 & \cdots & x_{|\mathcal{L}|}, & \theta_1 & \cdots & \theta_N \end{array} \right]^\top, \tag{12}$$

$$\vec{\mathbf{f}} = \left[\begin{array}{cccccc} 0_1 & \cdots & 0_{|\mathcal{L}|}, & 1_1 & \cdots & 1_N \end{array} \right]^\top, \tag{13}$$

$$A = \begin{bmatrix} W_{11} & W_{12} \cdots W_{1|\mathcal{L}|} & -V & 0 \cdots 0 \\ W_{21} & W_{22} \cdots W_{2|\mathcal{L}|} & 0 & -V \cdots 0 \\ \vdots & \vdots \ \ddots \ \vdots & \vdots & \vdots \ \ddots \ \vdots \\ W_{N1} & W_{N2} \cdots W_{N|\mathcal{L}|} & 0 & 0 \cdots -V \end{bmatrix}, \tag{14}$$

$$\mathbf{B} = \left[\begin{array}{ccc} \tau & \cdots & \tau \end{array} \right]^\top, \tag{15}$$

$$\mathbf{A}_{\mathrm{e}} = \left[\begin{array}{cc} \mathbf{M} & \mathbf{0}_{|\mathcal{V}| \times N}) \end{array} \right]^\top, \tag{16}$$

$$\mathbf{B}_{\mathrm{e}} = \vec{\mathbf{b}}, \tag{17}$$

$$\vec{\mathbf{l}} = \mathbf{0}_{(|\mathcal{L}|+N) \times 1}, \tag{18}$$

$$\vec{\mathbf{u}} = \mathbf{1}_{(|\mathcal{L}|+N) \times 1}. \tag{19}$$

Note that W_{ij} in Eq. (14) is the i_{th} travel time sample on arc j and $\mathbf{B}$ in Eq. (15) has the size of $N \times 1$.

In this standard form of ILP, $\vec{\mathbf{z}}$ is a decision variable, and the optimal value for $\vec{\mathbf{x}}$ in Eq. (12) corresponds to the optimal path, i.e., the path with the maximum probability of punctuality [28]. The proposed ILP problem can be solved by some mature algorithms, such as branch and bound and cutting planes. Those algorithms have been realized in many existing solvers, e.g., CBC, GLPK and CPLEX [29], [30].

IV. Performance Evaluation

In this section, we evaluate the proposed accurate solution from both the theoretical and experimental perspectives. Firstly, the accuracy of the proposed ILP is theoretically proved, by deriving the equivalency between the ILP and cardinality minimization problem. Then, we evaluate three methods with experiments, i.e., the Dijkstra (i.e., the shortest path algorithm) [31], MILP (i.e., the approximate solution based on the cardinality minimization) and ILP (i.e., our accurate solution based on the cardinality minimization), to compare their performances and verify the superiority of ILP. Finally, three solvers are tested for the ILP in order to recommend the best solver.

A. Theoretical Proof of Accuracy

In Eq. (10), the punctuality problem has been formulated as cardinality minimization, which is rewritten as below:

$$\min_{\vec{x}} \ Card(C(\vec{x})) \left| \begin{array}{l} \mathbf{M}\vec{x} = \vec{b}; \\ \vec{x} \in \{0,1\}^{|\mathcal{L}|}, \end{array} \right. \tag{20}$$

where $C(\vec{x}) = \left([\vec{w}_1^\top \vec{x} - \tau]^+, \cdots, [\vec{w}_N^\top \vec{x} - \tau]^+\right)$. As the MILP form of the punctuality problem resorts to the ℓ_1-norm relaxation, it probably leads to an approximate solution as we have illustrated in Section III-A. To maintain the exact solution, we directly derive an ILP problem from Eq. (20), which is described as follows:

$$\min_{\vec{x}, \vec{\theta}} \sum_{i=1}^{N} \theta_i \left| \begin{array}{l} \mathbf{M}\vec{x} = \vec{b}; \\ \vec{w}_i^\top \vec{x} - \tau \le V \cdot \theta_i; \\ \vec{x} \in \{0,1\}^{|\mathcal{L}|}, \theta_i \in \{0,1\}, \end{array} \right. \tag{21}$$

where $\vec{\theta} = (\theta_1, \cdots, \theta_N)$, and V denotes a huge positive number. As a consequence, the 0-1 variable θ_i will get 1 when $\vec{w}_i^\top \vec{x} - \tau$ is positive, otherwise θ_i equals 0. Obviously, $\sum_{i=1}^N \theta_i$, meaning the same thing as $Card(C(\vec{x}))$, indeed refers to the number of times of being late, and needs to be minimized. The proof of two formulations' equivalence is stated as follows:

1) Proof of Sufficiency: An auxiliary function $\varepsilon : R \rightarrow \{0,1\}$ is defined as below:

$$\varepsilon(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases} \tag{22}$$

by which the cardinality problem in Eq. (20) can be converted into:

$$\min_{\vec{x}} \sum_{i=1}^{N} \varepsilon(\vec{w}_i^{\mathsf{T}}\vec{x} - \tau) \ \left| \begin{array}{l} \mathbf{M}\vec{x} = \vec{b}; \\ \vec{x} \in \{0,1\}^{|\mathcal{L}|}, \end{array} \right. \tag{23}$$

Then, we replace $\varepsilon(\vec{w}_i^{\mathsf{T}}\vec{x}-\tau)$ with θ_i, i.e., $\theta_i = \varepsilon(\vec{w}_i^{\mathsf{T}}\vec{x}-\tau)$. Then, we can easily observe the following equivalence:

$$\min_{\vec{x}} \sum_{i=1}^{N} \theta_i \ \left| \begin{array}{l} x \in \mathcal{X}; \\ \theta_i = \varepsilon(\vec{w}_i^{\mathsf{T}}\vec{x} - \tau), \end{array} \right. \tag{24}$$

$$\Updownarrow$$

$$\min_{\vec{x},\vec{\theta}} \sum_{i=1}^{N} \theta_i \ \left| \begin{array}{l} x \in \mathcal{X}; \\ \theta_i = \varepsilon(\vec{w}_i^{\mathsf{T}}\vec{x} - \tau), \end{array} \right. \tag{25}$$

$$\Downarrow$$

$$\min_{\vec{x},\vec{\theta}} \sum_{i=1}^{N} \theta_i \ \left| \begin{array}{l} x \in \mathcal{X}; \theta_i \in \{0,1\}; \\ \vec{w}_i^{\mathsf{T}}\vec{x} - \tau \leq V \cdot \theta_i, \end{array} \right. \tag{26}$$

where V is a large value and $x \in \mathcal{X}$ represents feasible region of $\vec{x}$. The equivalence between Eq. (24) and Eq. (25) is obvious because θ_i is a binary value determined by $\vec{x}$, and increasing $\vec{\theta}$ will not change the value of objective function. On the contrary, the value of θ_i is not completely dependent on $\vec{x}$ in Eq. (26). For instance, when $\vec{w}_i^{\mathsf{T}}\vec{x} - \tau$ is negative, both $\theta_i = 0$ and 1 meet the inequality constraint. Nevertheless, we can easily derive Eq. (26) from Eq. (25). To be specific, whether the value of $\vec{w}_i^{\mathsf{T}}\vec{x} - \tau$ is positive or not, it always satisfys the relation as below:

$$\theta_i = \varepsilon(\vec{w}_i^{\mathsf{T}}\vec{x} - \tau), \tag{27}$$

$$\Downarrow$$

$$\vec{w}_i^{\mathsf{T}}\vec{x} - \tau \leq V \cdot \theta_i; \quad \theta_i \in \{0,1\}. \tag{28}$$

As a consequence, we have proved that Eq. (20) is the sufficient condition for Eq. (21), because Eq. (26) is definitely the same as Eq. (20).

2) Proof of Necessity: To prove the necessity, we start with Eq. (26) and manipulate it as follows:

$$\min_{\vec{x},\vec{\theta}} \sum_{i=1}^{N} \theta_i \ \left| \begin{array}{l} x \in \mathcal{X}; \theta_i \in \{0,1\}; \\ \vec{w}_i^{\mathsf{T}}\vec{x} - \tau \leq V \cdot \theta_i, \end{array} \right. \tag{29}$$

$$\Downarrow$$

$$\min_{\vec{x}}, \min_{\vec{\theta}} \sum_{i=1}^{N} \theta_i \ \left| \begin{array}{l} x \in \mathcal{X}; \theta_i \in \{0,1\}; \\ \vec{w}_i^{\mathsf{T}}\vec{x} - \tau \leq V \cdot \theta_i, \end{array} \right. \tag{30}$$

$$\Downarrow$$

$$\min_{\vec{x}} \sum_{i=1}^{N} \min_{\vec{\theta}} \theta_i \ \left| \begin{array}{l} x \in \mathcal{X}; \theta_i \in \{0,1\}; \\ \vec{w}_i^{\mathsf{T}}\vec{x} - \tau \leq V \cdot \theta_i, \end{array} \right. \tag{31}$$

To associate Eq. (31) with cardinality expression, we can resort to the auxiliary function defined in Eq. (22) and prove the relation as follows:

$$\min_{\vec{\theta}} \theta_i \ \left| \begin{array}{l} x \in \mathcal{X}; \theta_i \in \{0,1\}; \\ \vec{w}_i^{\mathsf{T}}\vec{x} - \tau \leq V \cdot \theta_i, \end{array} \right. \tag{32}$$

$$\Downarrow$$

$$\theta_i = \varepsilon(\vec{w}_i^{\mathsf{T}}\vec{x} - \tau) \ \left| x \in \mathcal{X}. \right. \tag{33}$$

Specifically, when $\vec{w}_i^{\mathsf{T}}\vec{x} - \tau$ is positive, we can derive the inequalities as below:

$$0 < \vec{w}_i^{\mathsf{T}}\vec{x} - \tau \leq V \cdot \theta_i; \tag{34}$$

$$\theta_i \in \{0,1\}, \tag{35}$$

where the value of θ_i can only be 1. Similarly, if $\vec{w}_i^{\mathsf{T}}\vec{x} - \tau$ is non-positive, θ_i must satisfy the inequality constraints as follows:

$$\vec{w}_i^{\mathsf{T}}\vec{x} - \tau \leq 0; \tag{36}$$

$$\vec{w}_i^{\mathsf{T}}\vec{x} - \tau \leq V \cdot \theta_i; \tag{37}$$

$$\theta_i \in \{0,1\}, \tag{38}$$

where both $\theta_i = 0$ and 1 meet the inequality constraints. However, to minimize the objective function value, the decision variable θ_i will only be 0. In view of the relation between Eq. (32) and Eq. (33), the formulation in Eq. (31) can be converted into:

$$\min_{\vec{x}} \sum_{i=1}^{N} \varepsilon(\vec{w}_i^{\mathsf{T}}\vec{x} - \tau) \ \left| x \in \mathcal{X}, \right. \tag{39}$$

$$\Downarrow$$

$$\min_{\vec{x}} \sum_{i=1}^{N} \varepsilon(\vec{w}_i^{\mathsf{T}}\vec{x} - \tau) \ \left| \begin{array}{l} \mathbf{M}\vec{x} = \vec{b}; \\ \vec{x} \in \{0,1\}^{|\mathcal{L}|}, \end{array} \right. \tag{40}$$

$$\Updownarrow$$

$$\min_{\vec{x}} Card(C(\vec{x})) \ \left| \begin{array}{l} \mathbf{M}\vec{x} = \vec{b}; \\ \vec{x} \in \{0,1\}^{|\mathcal{L}|}, \end{array} \right. \tag{41}$$

where $C(\vec{x}) = \big([\vec{w}_1^{\mathsf{T}}\vec{x} - \tau]^+, \cdots, [\vec{w}_N^{\mathsf{T}}\vec{x} - \tau]^+\big)$. Therefore, Eq. (20) is also the necessary condition for Eq. (21). In conclusion, Eq. (20) and Eq. (21) are necessary and sufficient conditions for each other, and the ILP formulation is an accurate solution to the cardinality-based punctuality problem.

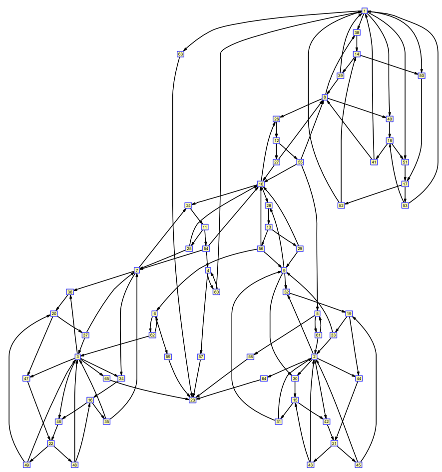


Fig. 1: An Artificial Road Network: 65 Nodes and 123 Arcs.

B. Experimental Results and Evaluation

To verify the superiority of the proposed accurate solution, i.e., the ILP method, we test it against two other methods, i.e., the Dijkstra method and MILP method. The Dijkstra method is the classical algorithm for shortest path problem. It manages to find a path with the least total weight in a graph [32], which can be employed to determine a path with the least travel time in the traffic network. The MILP method is proposed in [16], and it uses the ℓ_1-norm relaxation technique to calculate the approximate solution to the cardinality-based punctuality problem. The three methods are tested on two networks, i.e., an artificial road network and an area of real road network in Beijing, with simulation and processed GPS data respectively.

In each network, we firstly derive the samples of link travel times to compute the optimal paths for the three methods. Different origins, destinations and deadlines are subsequently specified to test the accuracy of three methods. Then, we calculate and compare the maximum probability of punctuality for the optimal paths from the three methods. Finally, we test and analyse the three methods' computational efficiency for the two networks. Note that the proposed method and MILP method are implemented by the solver CPLEX with Matlab R2017a, and the Dijkstra algorithm is implemented by calling the built-in function ***distances*** in Matlab.

1) Artificial Road Network with Simulation Data: To test the three methods, namely the Dijkstra, MILP, and our method, we first apply them to an artificial road network with simulation data, which is a directed graph with high connectivity degree. Particularly, this graph has 65 nodes and 123 links, which is shown in Fig. 1. Based on this artificial road network, our experiment is conducted by the following steps:

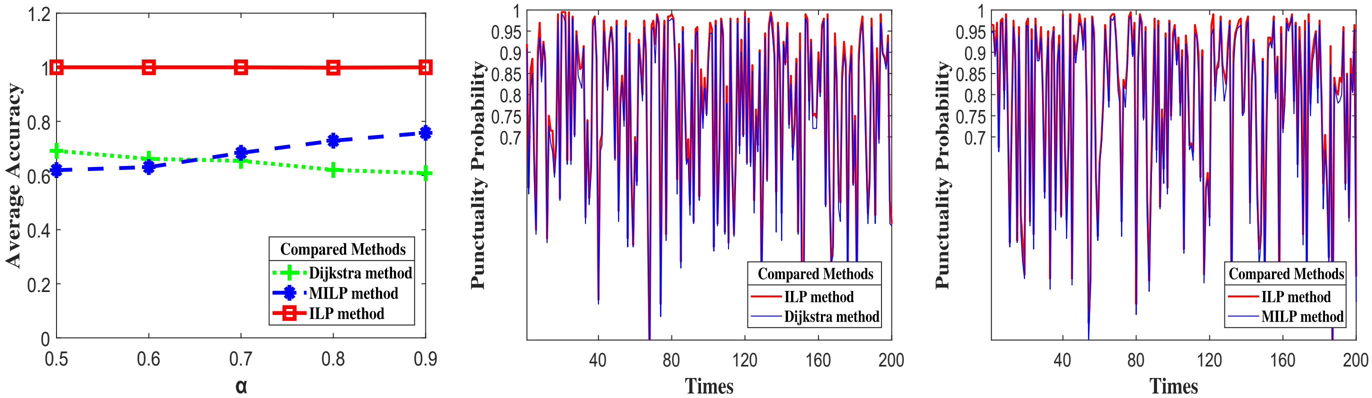
step 1: We randomly specify 20 origin-destination(O-D) pairs;
step 2: We generate 500 groups of simulation data for the whole road network, representing the samples of link travel times in the road network[4];
step 3: For each O-D pair, we use the samples of link travel times to calculate the optimal path, i.e., the path with maximum punctuality probability, given a specified deadline;
step 4: For each O-D pair and each deadline, step 2 and step 3 are repeated 100 times. To test the accuracy, the optimal path of ground-truth will be derived by an enumeration method every time[5];
step 5: We specify 5 types of deadlines for each of the 20 O-D pairs, and then repeat step 4 for the three methods, to calculate their accuracies in different scenarios (i.e., different O-D pairs, deadlines and varying link travel times);
step 6: Following the above steps, all the three methods are implemented to compare their average accuracies and the probabilities of punctuality for their optimal paths. Note that the groups of link travel times are directly incorporated into the MILP and our method, while the average travel time on each link is needed for the application of the Dijkstra method.

For each O-D pair, the deadline is determined by $\tau = \tau_1 + \alpha(\tau_2 - \tau_1)$, where τ_2 is the minimum longest travel time among all paths based on the generated link travel times; τ_1 is the shortest travel time on the same path with the minimum longest travel time; α is a coefficient. Thus, we can use α to represent deadlines of different levels. Generally, lower α indicates tight deadline, and vice versa. Moreover, it is obvious that when α equals to 1, τ will be equal to τ_2 and there will be a path with the 100% probability of punctuality. In our experiment, we use five different values of α, which corresponds to deadlines of five levels.

The corresponding experimental results are displayed in Fig. 2. More specifically, the average accuracies from each of the three methods with different deadlines are plotted in Fig. 2(a). We can easily observe that, for all specified deadlines, the proposed accurate solution by ILP method can always achieve accuracy of 100%. The Dijkstra and MILP methods obtain unsatisfactory average accuracy, both ranging from 60% to 80%. In fact, the average accuracy of MILP method is better than the Dijkstra method because highest accuracy of the Dijkstra method is only 70%, as shown in Fig. 2(a). In the experimentation, we calculate the average link travel times as weights and then use the Dijkstra method to obtain the optimal path. However, the Dijkstra method only consider the cost (i.e., travel time) on each road link, and it does not take into account the user-defined deadline. Consequently, the accuracy of seeking the optimal path is low. Regarding the MILP method, it employs the ℓ_1-norm relaxation technique to approximate the punctuality problem.

[4] In fact, all the three methods, namely the Dijkstra, MILP, and our method, exploit samples of link travel times to estimate the optimal path without any assumption of them. Here, the 500 groups of samples are generated at random, while for Beijing Road Network, the samples are part of real link travel times.

[5] If there are 200 instances of data on each link and we assume that 200 instances exist for each path between the O-D pair, then the ground-truth optimal path has the maximum number of times of not being late regarding the deadline. The ground-truth path can be found through enumerating all paths between the O-D pair.



(a) accuracy with different deadlines (b) punctual probability of Dijkstra and ILP (c) punctual probability of MILP and ILP

Fig. 2: Methods Evaluation in Artificial Road Network.

It is data-driven and has no requirements on probability distribution or correlation of the link travel times. Nevertheless, it is also influenced by data because the relaxation takes the delay duration as decision variables rather than the real cardinality of punctuality. Since it does consider the user-defined deadline, it achieves higher accuracy than that of Dijkstra method. The proposed accurate solution by ILP method is also data-driven and can be directly applied with the link travel times. Furthermore, its ILP formulation can always obtain an accurate solution to the cardinality-based punctuality problem.

To better demonstrate the advantage of our method, we also compare the punctuality probabilities of the optimal paths (i.e., the maximum probability of punctuality) from the three methods. Fig. 2(b) shows the the probability of punctuality for the Dijkstra method and the ILP method in 200 tests. It is obvious that our method can always derive the path with the higher punctuality probability than the path determined by the Dijkstra method. From Fig. 2(b), we can also see that the punctuality probabilities of the optimal paths from two methods differ slightly, which means the punctuality probabilities of some paths are very close. However, our method can still discern this distinction and achieve the highest probability of punctuality, which demonstrates satisfactory robustness of our method. Analogously, the comparison between MILP method and ILP method is shown in Fig. 2(c). According to Fig. 2(c), it can also be concluded that no matter how slight the difference of the probability is, our method can always achieve the highest one. We would like to note that, as mentioned previously, although the difference of punctuality probability is small, even 1% increment in the probability of arriving on time may imply 1% increment in the chance of saving a life, especially in the route planning for fire rescue or organ delivery.

2) An Area of Beijing Road Network with Processed GPS Data: To better verify the reliability and superiority of our method, we continue to test it on Beijing road network with real traffic data. The original dataset includes the trajectories of over 30000 taxicabs for one day in Beijing. Based on those GPS data, some researchers estimated the travel time for each road link [21], [33], and the effectiveness of the estimation was also verified [21]. We only extract an area of the Beijing road network to test the three methods, which is shown in Fig. 3 (i.e., 362 nodes and 528 arcs). The experiment is similar to that conducted on the artificial road network, except that the samples are from the real link travel times in Beijing road network. Accordingly, the corresponding results are shown in Fig. 4(a) - Fig. 4(c). Particularly, the accuracy comparison of three methods is displayed in Fig. 4(a).

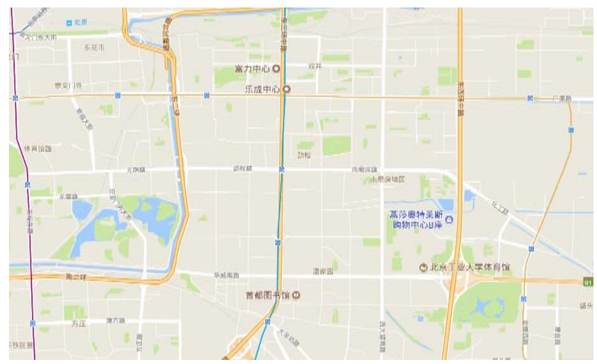
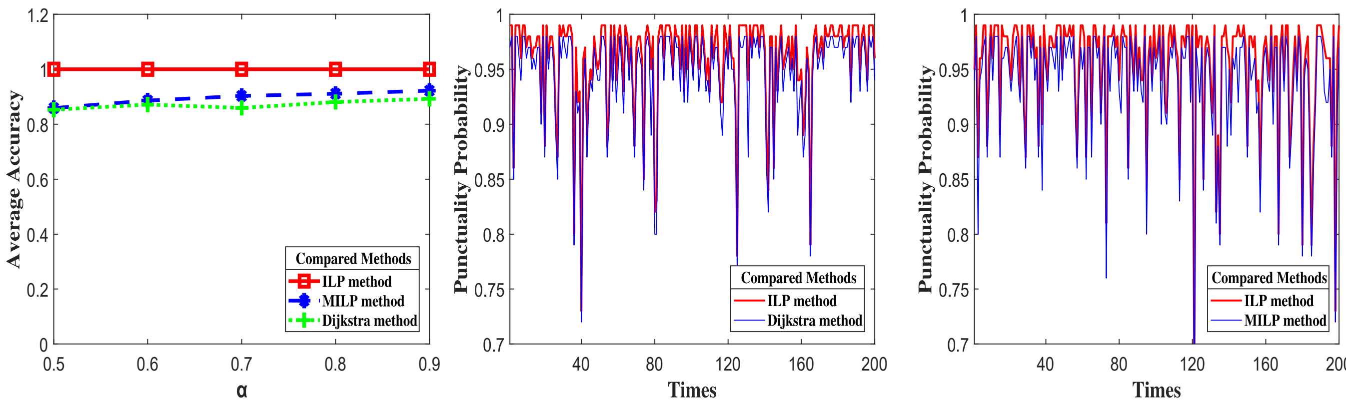


Fig. 3: An area of Beijing City

From Fig. 4(a), we can observe that the average accuracy of our method is higher than the other two methods, which always keeps 100% against all different deadlines. We can also see that, although the difference of accuracy for the Dijkstra method and MILP method is slight, the accuracy for the latter is always higher. One underlying rationale is that the MILP method takes into consideration both the randomness of travel time and the user-defined deadline, while the Dijkstra method only considers the expected travel time. Fig. 4(b) shows the comparison of punctuality probability between the Dijkstra method and our method for 200 tests. It is clear that our method can always choose a path with higher probability of punctuality. In fact, our method provides an accurate solution to calculate the optimal path with maximal probability of punctuality, so the probability of punctuality obtained by the Dijkstra method will not be larger than that of our method. Fig. 4(c) has the similar pattern as Fig. 4(b), and it also shares the similar rationale.



(a) accuracy with different deadlines (b) punctual probability of Dijkstra and ILP (c) punctual probability of MILP and ILP

Fig. 4: Methods Evaluation in Beijing Road Network.

3) Computational Efficiency Evaluation: To evaluate the computational efficiency, we also compare the average running time for the three methods in the two road networks. The average running time is calculated as the running time in all tests with different O-D pairs and deadlines. The result is recorded in Table I. As we can see, the computational efficiency of our method is almost tied with the MILP method for the two road networks. However, our method is more efficient than MILP method in the artificial road network. Although the Dijkstra method has the shortest average computation time in two road networks, its accuracy is much lower than our method and the MILP method. Moreover, lower accuracy is not in favour for many critical cases of routing planning, such as fire rescue, organ delivery and flight catching.

TABLE I: Computation Time for Different Methods (s)

Network	Dijkstra	MILP	Proposed method
Artificial Road Network	0.0028	0.3294	0.3215
Beijing Road Road Network	0.0036	0.9778	1.0452

In summary, no matter in the artificial road network with simulation data, or in real road network with real traffic data, the ILP method always guarantees highest accuracy and desirable computation time. Considering both the accuracy and computational efficiency, our method is more reliable and effective to solve the punctuality problem in comparison with the Dijkstra method and MILP method.

C. ILP Solvers Evaluation

The proposed accurate solution is expressed as the form of ILP. The ILP can be solved by some mature algorithms, which have been realized inside many solvers. We would like to mention that, although the proposed ILP method theoretically guarantees an accurate solution, some available ILP solvers may not always reach the real optimum due to various limitations. Therefore, in our work, we test the accuracy of three popular ILP solvers, including CBC (i.e., Coin-or Branch and Cut), GLPK (i.e., GNU Linear Programming Kit) and CPLEX (i.e., IBM ILOG CPLEX Optimization Studio). More specifically, CBC is an open-source MILP solver distributed under the COIN-OR (Computational Infrastructure for Operations Research) project, which solved ILP problem based on Branch and Cut algorithm [29]. GLPK is an open-source software package for solving large-scale LP, MILP and other related problems, which solved the ILP problem based on the branch and bound algorithm. CPLEX is a commercial high-performance mathematical programming solver, which solved the ILP mainly based on the cutting technique [30]. The three solvers are tested in both artificial road network and Beijing road network. In each network, the average accuracy under each deadline is calculated by taking the mean of accuracies with different O-D pairs and varying link travel times.

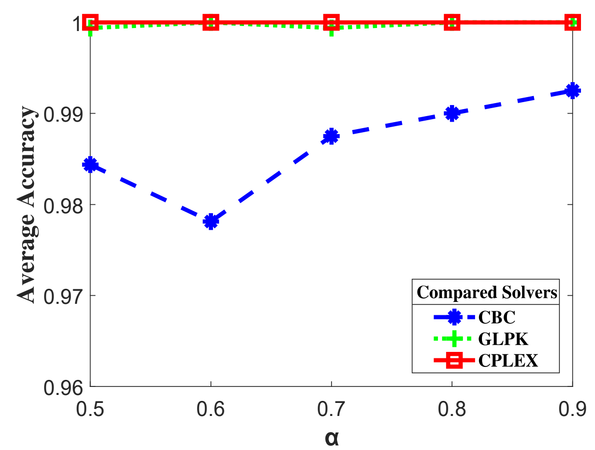


Fig. 5: Accuracy for Solvers in Artificial Road Network

The comparative result in artificial road network is shown in Fig. 5. It can be seen that no matter which solver is used, the average accuracy regarding different deadlines is always above 97%. The average accuracy of CBC is obviously lower than other two solvers. The average accuracies of GLPK and CPLEX are very close. However, when deadlines coefficient α is equal to 0.7, the accuracy for GLPK is slightly lower than 100%. Meanwhile, CPLEX can always achieve average accuracy of 100%. Therefore, CPLEX has better performance in comparison with the other two solvers. Similarly, the CPLEX's superiority is also verified by the comparative result

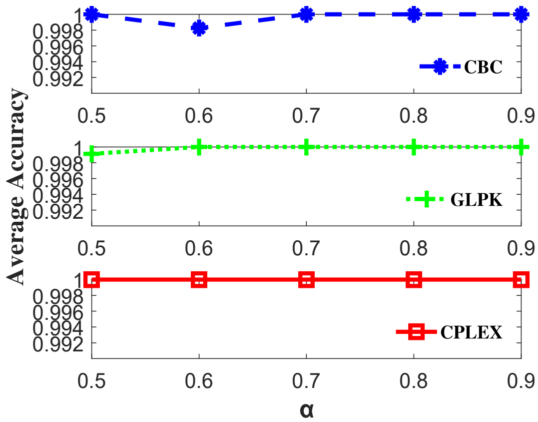


Fig. 6: Accuracy for Solvers in Beijing Road Network

in Beijing road network, as shown in Fig. 6. Particularly, although three solvers show quite high accuracy, only CPLEX exactly keeps 100% accuracy in all cases. When $\alpha = 0.6$ for CBC, and $\alpha = 0.5$ for GLPK, the accuracies are slightly lower than 100%. In addition, we further test the average computation time of the three solvers to compare their computational efficiency. The average computation time for both artificial road network and Beijing road network are recorded in Table II. The result in Table II shows that in both networks, CPLEX has shorter average running time in comparison with GLPK and CBC. Considering both the accuracy and efficiency, CPLEX is a perfect option to implement the proposed accurate solution in this paper.

TABLE II: Computation Time for Different Solvers (s)

Network	CBC	GLPK	CPLEX
Artificial Road Network	0.7992	0.8475	0.3215
Beijing Road Network	4.9942	1.7904	1.0452

V. Discussion

Our approach in this paper is data-driven, and it aims to minimize the numbers of times of being late, which can tell from Eq. (3) and (4). To this end, the original travel time on each road link for each time is more desirable than a random variable, because we need two metrics: how many times a vehicle runs on a path (consists of road links); how many times it arrives the destination before the deadline. On the other hand, as the number of times that a vehicle runs on a path becomes larger, it will more accurately represent the true random variable. Moreover, although the travel time samples for the artificial road network is randomly generated according to certain distribution, the travel time entities for the Beijing road network are real. The latter is obtained from the real individual cars, which can better represent the actual values.

In view of the fulfilling results of the model-based methods in [26] and [24], our data-driven approach has both advantages and disadvantages. The advantages come down to two aspects: 1) our approach does not need to construct any distribution function or probability density function; 2) our approach does not need to explicitly establish any correlation model because the real travel time data already naturally included the correlation characteristic. The two advantages are important because generally, it is difficult to construct an accurate function or model in 1) and 2). However, the advantages come at the price of computation efficiency. To achieve an accurate solution, the data-driven approach finally formulates the problem as an integer linear programming problem, the worst computation complexity of which is exponential with the problem scale. Therefore, the current solution is not suitable to apply on large road networks, which is inferior to the methods in [23] and [25] with respect to the computation efficiency. Consequently, improving the computation efficiency while reserving high accuracy would definitely be the most important future work.

We also would like to note that the time-dependent capability is an important metric for finding a stochastic shortest path. Since our approach is data-driven, it is easy and natural to develop the time-dependent capability by inputting the travel time data into the algorithm in a time-dependent manner [34]. For example, to find a shortest path between 7am and 9am, we may utilize the travel time data for that specific period as the input. Moreover, if the real time is unavailable, we may also adopt the historical data of that period as the input. On the other hand, the time- dependent performance for this kind of data-driven approach has been well investigated in the previous work [34]. Therefore, we did not detail the time-dependent capacity of our approach here, which is also not the focus of this paper.

VI. Conclusion and Future Work

In this paper, we aim at determining a path with maximum probability of punctuality in stochastic traffic. An accurate solution to the punctuality problem is proposed by introducing additional variables and constraints into the approximate solution, where the approximate solution is solved by the ℓ_1-norm relaxation. The proposed method is evaluated in three aspects. First, we theoretically prove the equivalence between the proposed solution and the accurate solution to the cardinality-based punctuality problem. Then, the proposed method is compared with the Dijkstra method and the ℓ_1-norm based approximation method in artificial and real road networks, in which the simulated and real GPS-based traffic data are adopted respectively. The result shows that our method has 100% accuracy and desirable computational efficiency in each scenario. Eventually, our method is implemented by CBC, GLPK and CPLEX solvers to determine the one with best performance. The result indicates that CPLEX is more reliable and effective than CBC and GLPK, which always guarantees the optimal solution.

In the future, we intend to develop our work in the following aspects: 1) more effort will be concentrated on improving the computational efficiency of the proposed method, which should be suitable for large road network; 2) in-filed driving will be conducted to verify the real probability of punctuality, rather than the theoretical calculation based on a dataset; 3) the time-dependent punctuality problem will be further investigated.

References

[1] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, "Understanding transportation modes based on gps data for web applications," *ACM Transactions on the Web (TWEB)*, vol. 4, no. 1, p. 1, 2010.

[2] J. Guo, Y. Wu, X. Zhang, L. Zhang, W. Chen, Z. Cao, L. Zhang, and H. Guo, "Finding the 'faster' path in vehicle routing," vol. 11, no. 10, pp. 685–694, 2017.

[3] D. P. Bertsekas and J. N. Tsitsiklis, "An analysis of stochastic shortest path problems," *Mathematics of Operations Research*, vol. 16, no. 3, pp. 580–595, 1991.

[4] Z. Sun, W. Gu, Y. Zhao, and C. Wang, "Optimal path finding method study based on stochastic travel time," *Journal of Transportation Technologies*, vol. 3, no. 4, pp. 260–265, 2013.

[5] J. C. Herrera, D. B. Work, R. Herring, X. J. Ban, Q. Jacobson, and A. M. Bayen, "Evaluation of traffic data obtained via gps-enabled mobile phones: The mobile century field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 4, pp. 568–583, 2010.

[6] Z. Cao, H. Guo, J. Zhang, and U. Fastenrath, "Multiagent-based route guidance for increasing the chance of arrival on time," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, pp. 3814–3820, 2016.

[7] C. E. Sigal, A. A. B. Pritsker, and J. J. Solberg, "The stochastic shortest route problem," *Operations Research*, vol. 28, no. 5, pp. 1122–1129, 1980.

[8] 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.

[9] H. Frank, "Shortest paths in probabilistic graphs," *operations research*, vol. 17, no. 4, pp. 583–599, 1969.

[10] 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.

[11] B. Yang and E. Miller-Hooks, "Adaptive routing considering delays due to signal operations," *Transportation Research Part B: Methodological*, vol. 38, no. 5, pp. 385–413, 2004.

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

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

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

[15] S. Samaranayake, S. Blandin, and A. Bayen, "A tractable class of algorithms for reliable routing in stochastic networks," *Transportation Research Part C: Emerging Technologies*, vol. 20, no. 1, pp. 199–217, 2012.

[16] Z. Cao, H. Guo, J. Zhang, D. Niyato, and U. Fastenrath, "Finding the shortest path in stochastic vehicle routing: A cardinality minimization approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 6, pp. 1688–1702, 2016.

[17] Z. Cao, H. Guo, J. Zhang, F. A. Oliehoek, and U. Fastenrath, "Maximizing the probability of arriving on time: A practical q-learning method.," in *Proceedings of the 31th AAAI Conference on Artificial Intelligence (AAAI)*, pp. 4481–4487, 2017.

[18] M. Hua and J. Pei, "Probabilistic path queries in road networks: traffic uncertainty aware path selection," in *Proceedings of the 13th International Conference on Extending Database Technology*, pp. 347–358, 2010.

[19] A. Christman and J. Cassamano, "Maximizing the probability of arriving on time," in *Proceedings of International Conference on Analytical and Stochastic Modeling Techniques and Applications*, pp. 142–157, 2013.

[20] 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, 2009.

[21] Y. Wang, Y. Zheng, and Y. Xue, "Travel time estimation of a path using sparse trajectories," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 25–34, 2014.

[22] 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.

[23] E. Miller-Hooks and H. Mahmassani, "Path comparisons for a priori and time-adaptive decisions in stochastic, time-varying networks," *European Journal of Operational Research*, vol. 146, no. 1, pp. 67–82, 2003.

[24] X. Wu and Y. Nie, "Application of discrete fourier transform in finding reliable shortest paths," *Journal of the Transportation Research Board*, 2010.

[25] X. Wu, "Finding reliable shortest paths in dynamic stochastic networks," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2333, pp. 80–90, 2013.

[26] Y. Nie, X. Wu, P. Nelson, and J. Dillenburg, "Providing reliable route guidance: a case study using chicago data," in *the 89th Transportation Research Board Annual Meeting, Washington DC, January*, 2010.

[27] X. Wu and H. Zhang, "Analysis of time-dependent travel time reliability for urban corridors: A cast study in houston," in *Proceedings of 19th IEEE International Conference on Intelligent Transportation Systems*, pp. 1939–1944, 2016.

[28] S. Suttinee, S. Jiang, N. Dusit, and P. Wang, "Optimal stochastic package delivery planning with deadline: A cardinality minimization in routing.," in *Proceedings of IEEE 86th Vehicular Technology Conference*, 2017.

[29] J. Forrest, "Cbc (coin-or branch and cut) open-source mixed integer programming solver, 2012," *URL https://projects. coin-or. org/Cbc*.

[30] I. I. CPLEX, "V12. 1: Users manual for cplex," *International Business Machines Corporation*, vol. 46, no. 53, p. 157, 2009.

[31] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

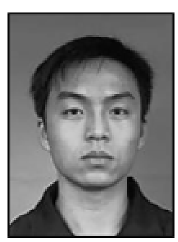
[32] R. K. Ahuja, K. Mehlhorn, J. Orlin, and R. E. Tarjan, "Faster algorithms for the shortest path problem," *Journal of the ACM (JACM)*, vol. 37, no. 2, pp. 213–223, 1990.

[33] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma, "Mining interesting locations and travel sequences from gps trajectories," in *Proceedings of the 18th international conference on World wide web*, pp. 791–800, 2009.

[34] Z. Cao, H. Guo, J. Zhang, D. Niyato, and U. Fastenrath, "Improving the efficiency of stochastic vehicle routing: A partial lagrange multiplier method," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3993–4005, 2016.

Zhiguang Cao received the Ph.D degree from Interdisciplinary Graduate School, Nanyang Technological University, Singapore, 2017. He received the B.Eng. degree in Automation from Guangdong University of Technology, Guangzhou, China, in 2009 and the M.Sc. degree in Signal Processing from Nanyang Technological University, Singapore, in 2012, respectively.

Zhiguang Cao hold a postdoc research fellow position with BMW@NTU Future Mobility Research Lab, and Energy Research Institute @ NTU (ERI@N), Singapore. He is currently an Distinguished Associate Professor with School of Automation, Guangdong University of Technology, China. His research interests focus on the applications of AI to traffic and transportation management.



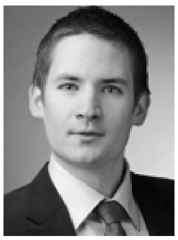
Yaoxin Wu received the B.Eng degree in traffic engineering from Wuyi University, Jiangmen, China, in 2015, and M.Eng degree in control theory and control engineering from Guangdong University of Technology, Guangzhou, China, in 2018. He is currently a Research Associate with the School of Computer Science and Engineering, NTU, Singapore.

His research interests include route planning, scheduling and optimization.

Akshay Rao received his Bachelor of Technology in Electrical Engineering from National Institute of Technology, Surat in 2007 and his PhD degree from Nanyang Technological University, Singapore in 2016. He is currently the Program Leader of the Autonomous Vehicle Research Group at the Future Mobility Research Lab, a joint initiative of BMW Group and Nanyang Technological University (NTU), Singapore. His main research interests lie in autonomous vehicles, sensor fusion and optimization algorithms for routing problems.

Felix Klanner received his diploma in mechanical engineering from Technical University Munich in 2005 and his doctoral degree from University of Technology in Darmstadt in 2008. He joined BMW in 2007 and was appointed the Director of Future Mobility Research, in 2014 as a part of a joint initiative of BMW Group and Nanyang Technological University (NTU), Singapore. His main research interest is in development of autonomous vehicles, applied machine learning and extraction of knowledge from Field Operational Tests (FOT) data.

Dr. Klanner is IEEE Member, reviewer for several IEEE Journals and Conferences, and Visiting Associate Professor at the School of Computer Science and Engineering at NTU. He published more than 80 patent applications and several peer-reviewed papers in books, journals and conference proceedings.



Stefan Erschen received his diploma in mechanical engineering from Technical University Munich in 2010. After his time as PhD fellow at Technical University Munich in collaboration with BMW Group in Munich he joined the BMW Future Mobility Research Lab, a joint initiative of BMW Group and Nanyang Technological University (NTU), Singapore in 2016. His main research interest is in development of autonomous vehicles, applied machine learning and optimization problems in various fields of mechanical engineering.

Wei Chen received the Ph.D. degree from South China University of Technology, China, in 2003. She received the B.Eng. degree in Precision Instruments & Mechanology from Tsinghua University, China, in 1986 and the M.Eng. degree in Robotics Control from South China University of Technology, China, in 1989, respectively.

She is currently a Full Professor with School of Automation, Guangdong University of Technology, China. Her research interests include intelligent control, robotics and pattern recognition.

Le Zhang is currently a researcher at the Advanced Digital Sciences Center (ADSC), the Singapore-based research center of the University of Illinois at Urbana-Champaign (UIUC). He received his Ph.D degree in school of EEE , Nanyang Technological University (NTU) in 2016. He also received his B.E. from University of Electronic Science and Technology Of China (UESTC) in 2011 and M.SC from NTU in 2012.

His current research interests include machine learning, computer vision and pattern recognition.

Hongliang Guo received his Bachelor of Engineering in Dynamic Engineering and a Master of Engineering in Dynamic Control at the Beijing Institute of Technology, China. He holds a PhD degree in Electrical and Computer Engineering from the Stevens Institute of Technology, USA. He later joined Almende in Rotterdam, the Netherlands as a postdoc research in 2011.

His research interests include self-organizing systems and agent-based technologies. In 2013, he joined NTU as a research fellow. From September 2016, he becomes an Associate professor in University of Electronics Science and Technology of China.