Path Cost Distribution Estimation Using Trajectory Data

Jian Dai[1] Bin Yang[2*] Chenjuan Guo[2] Christian S. Jensen[2] Jilin Hu[2]
[1]School of Computing, National University of Singapore
[2]Department of Computer Science, Aalborg University, Denmark
daij@comp.nus.edu.sg {byang, cguo, csj, jilin}@cs.aau.dk

ABSTRACT

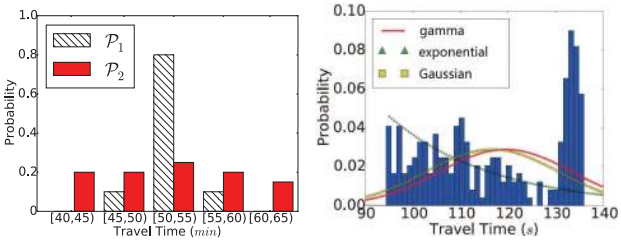
With the growing volumes of vehicle trajectory data, it becomes increasingly possible to capture time-varying and uncertain travel costs in a road network, including travel time and fuel consumption. The current paradigm represents a road network as a weighted graph; it blasts trajectories into small fragments that fit the underlying edges to assign weights to edges; and it then applies a routing algorithm to the resulting graph. We propose a new paradigm, the *hybrid graph*, that targets more accurate and more efficient path cost distribution estimation. The new paradigm avoids blasting trajectories into small fragments and instead assigns weights to paths rather than simply to the edges.

We show how to compute path weights using trajectory data while taking into account the travel cost dependencies among the edges in the paths. Given a departure time and a query path, we show how to select an optimal set of weights with associated paths that cover the query path and such that the weights enable the most accurate joint cost distribution estimation for the query path. The cost distribution of the query path is then computed accurately using the joint distribution. Finally, we show how the resulting method for computing cost distributions of paths can be integrated into existing routing algorithms. Empirical studies with substantial trajectory data from two different cities offer insight into the design properties of the proposed method and confirm that the method is effective in real-world settings.

1. INTRODUCTION

Increasing volumes of vehicle trajectories are becoming available that contain detailed traffic information. It is of interest to exploit this data source as well as possible to understand the state of a road network [1, 4]. For instance, it is of interest to know the *distributions* of travel costs (e.g., travel times or greenhouse gas (GHG) emissions [2, 3]) of paths at a given departure time in order to plan travel or to calculate payments for transportation, e.g., in settings where such services are outsourced.

Consider a scenario where a person wants to reach the airport within 60 min to catch a flight. Figure 1(a) shows the travel time distributions for two alternative paths for a given departure time. If only considering averages, $\mathcal{P}_2$ (with mean 51.5 min) is better than $\mathcal{P}_1$ (with mean 52 min). However, $\mathcal{P}_1$ is preferable because the probability of arriving at the airport within 60 min is 1, while with $\mathcal{P}_2$, the probability is 0.9. The example illustrates why being able to compute a distribution rather than a mean is important.



(a) $\mathcal{P}_1$ vs. $\mathcal{P}_2$ (b) Complex Distribution

Figure 1: Motivating Examples

A natural question is then how to best utilize trajectories to accurately and efficiently derive cost distribution for any path at a given departure time. This is the fundamental problem addressed in this paper. Solving this problem is important in its own right, and it also represents a significant step towards more accurate and efficient stochastic routing. To solve the problem, three challenges must be addressed.

Complex travel cost distributions: Travel time and GHG emissions vary over time and even vary across vehicles traversing the same path at the same time. To exemplify the latter, the bars in Figure 1(b) represent the travel time, derived from GPS trajectories, of a path during the time interval [8:00, 8:30). Next, distributions do not follow standard distributions. Figure 1(b) shows the Gaussian [17], gamma [20], and exponential [20] distributions obtained using maximum likelihood estimation, illustrating that the cost distribution does not follow any of these standard distributions.

Sparseness: With enough trajectories that contain a path during a particular time interval, we could derive a distribution during the interval for the path using those trajectories. However, we report on analyses showing that even with large volumes of trajectory data, it is practically impossible to cover all paths in a road network with sufficient numbers of trajectories during all time intervals—a road network has a very large number of meaningful paths. We must thus contend with data sparseness.

Dependency: The cost distribution of a path can be estimated by summing up, or convoluting, the cost distributions of its edges [6, 13, 22]. However, the results are only accurate if the edge distributions are independent. We offer evidence that this is generally

*Corresponding author

Proceedings of the VLDB Endowment, Vol. 10, No. 3

not so in our setting. To derive accurate distributions for paths, the dependencies must be considered.

The conventional paradigm for path cost estimation fragments trajectories into pieces that fit the individual edges to assign uncertain weights to the edges. Convolution is applied to the uncertain edge weights to compute the cost distribution of the path [6, 11, 13, 22, 26, 28, 29]. This approach falls short in addressing the above challenges and suffers in terms of both accuracy and efficiency. The accuracy suffers because dependencies among different edges are not accounted for. The efficiency suffers because many expensive convolutions must be performed.

We propose part of the foundation for a new paradigm, the *hybrid graph*, that aims to achieve better accuracy and efficiency by addressing the three challenges. In the hybrid graph, weights are assigned to paths; and path weights are joint distributions that fully capture the cost *dependencies* among the edges in the paths. Further, multi-dimensional histograms are used to approximate *complex* distributions accurately and compactly. Given a departure time and a query path, we are then able to select an optimal set of path weights such that the selected paths together cover the query path at the departure time and such that the path weights produce the most accurate joint distribution of the edges in the query path. The joint distribution can then be transferred into the cost distribution of the query path. The capability of deriving the distributions of long paths with insufficient trajectories from the distributions of carefully selected sub-paths with sufficient trajectories addresses the *sparseness* problem.

The paper's proposal is compatible with existing stochastic routing algorithms and we show how the proposal can be seamlessly integrated into existing stochastic routing algorithms while boosting the efficiency and accuracy of these algorithms.

To the best of our knowledge, this is the first study to enable accurate and efficient path travel cost distribution estimation using trajectories while contending with sparseness, dependency, and distribution complexity. In particular, we make four contributions. (i) We propose the hybrid graph and a multi-dimensional histogram based method to instantiate path weights using trajectories. (ii) We propose an algorithm that identifies an optimal set of path weights, enabling accurate estimation of the joint distribution of a query path. (iii) We propose a method that derives the cost distribution of the query path, represented as a one-dimensional histogram, from a joint distribution, represented as a multi-dimensional histogram. (iv) We report on empirical studies that demonstrate that the paper's proposal significantly outperforms existing proposals in terms of both accuracy and efficiency.

Paper outline: Section 2 covers basic concepts and baselines. Section 3 introduces the hybrid graph and the method for instantiating path weights. Section 4 gives the algorithms for estimating the travel cost of a path. Section 5 reports on the empirical study. Related work is covered in Section 6, and Section 7 concludes.

2. PRELIMINARIES

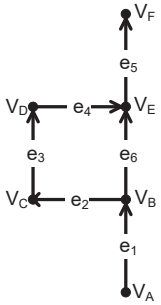
2.1 Basic Concepts

A *road network* is modeled as a directed graph $G = (V, E)$, where V is a vertex set and $E \subseteq V \times V$ is an edge set. A vertex $v_i \in V$ represents a road intersection or an end of a road. An edge $e_k = (v_i, v_j) \in E$ models a directed road segment, indicating that travel is possible from its *start vertex* v_i to its *end vertex* v_j. We use $e_k.s$ and $e_k.d$ to denote the start and end vertices of edge e_k. Two edges are *adjacent* if one edge's end vertex is the same as the other edge's start vertex. Figure 2(a) shows an example road network.

A *path* $\mathcal{P} = \langle e_1, e_2, \ldots, e_A \rangle$, $A \geqslant 1$, is a sequence of adjacent edges that connect distinct vertices in the graph, where $e_i \in E$, $e_i.d = e_{i+1}.s$ for $1 \leqslant i < A$, and the vertices $e_1.s$, $e_2.s$, ..., $e_A.s$, and $e_A.d$ are distinct. The cardinality of path $\mathcal{P}$, denoted as $|\mathcal{P}|$, is the number of edges in the path. Path $\mathcal{P}' = \langle g_1, g_2, \ldots, g_x \rangle$ is a *sub-path* of $\mathcal{P} = \langle e_1, e_2, \ldots, e_a \rangle$ if $|\mathcal{P}'| \leqslant |\mathcal{P}|$ and there exists an edge sequence in $\mathcal{P}$ such that $g_1 = e_i$, $g_2 = e_{i+1}$, ..., and $g_x = e_{i+x-1}$.

Given two paths $\mathcal{P}_i$ and $\mathcal{P}_j$, we use $\mathcal{P}_i \cap \mathcal{P}_j$ to denote the path that is shared by both paths, and we use $\mathcal{P}_i \setminus \mathcal{P}_j$ to denote the sub-path of $\mathcal{P}_i$ that exclude edges in $\mathcal{P}_j$. For instance, we have $\langle e_1, e_2, e_3 \rangle \cap \langle e_2, e_3, e_4 \rangle = \langle e_2, e_3 \rangle$ and $\langle e_1, e_2, e_3 \rangle \setminus \langle e_2, e_3, e_4 \rangle = \langle e_1 \rangle$.

A *trajectory* $\mathcal{T} = \langle p_1, p_2, \ldots, p_C \rangle$ is a sequence of GPS records pertaining to a trip, where each p_i is a $(location, time)$ pair of a vehicle, where $p_i.time < p_j.time$ if $1 \leqslant i < j \leqslant C$. Map matching [16] is able to map GPS records in a trajectory $\mathcal{T}$ to specific locations on different edge and thus it aligns trajectory $\mathcal{T}$ with a path. We call this path as the *path of trajectory* $\mathcal{T}$, denoted as $\mathcal{P}_\mathcal{T}$. A trajectory $\mathcal{T}$ *occurred on* path $\mathcal{P}$ *at* time t if and only if path $\mathcal{P}$ is a sub-path of the path of trajectory $\mathcal{P}_\mathcal{T}$ and the first GPS record in the first edge in path $\mathcal{P}$ is obtained at t. Figure 2(b) shows 10 trajectories. For example, trajectory $\mathcal{T}_1$ occurred on path $\langle e_1, e_2, e_3, e_4 \rangle$ at 8:01.



(a) Road Network

Trajectory	Path of Trajectory	Time
$\mathcal{T}_1$	$\langle e_1, e_2, e_3, e_4 \rangle$	8:01
$\mathcal{T}_2$	$\langle e_1, e_2, e_3, e_4 \rangle$	8:02
$\mathcal{T}_3$	$\langle e_1, e_2, e_3 \rangle$	8:10
$\mathcal{T}_4$	$\langle e_1, e_2, e_3 \rangle$	8:07
$\mathcal{T}_5$	$\langle e_2, e_3, e_4 \rangle$	8:01
$\mathcal{T}_6$	$\langle e_2, e_3, e_4 \rangle$	8:10
$\mathcal{T}_7$	$\langle e_2, e_3, e_4 \rangle$	15:21
$\mathcal{T}_8$	$\langle e_4, e_5 \rangle$	8:07
$\mathcal{T}_9$	$\langle e_4, e_5 \rangle$	8:07
$\mathcal{T}_{10}$	$\langle e_6, e_5 \rangle$	8:08

(b) Trajectories

Figure 2: A Road Network and Trajectories

The travel cost (e.g., travel time or GHG emissions) of using a path $\mathcal{P}$ can be obtained from the trajectories that occurred on $\mathcal{P}$. Given a trajectory $\mathcal{T}$ that occurred on path $\mathcal{P}$ at t, the travel time of using $\mathcal{P}$ at t is the difference between the time of the last GPS record and the time of the first GPS record on path $\mathcal{P}$; and the GHG emissions of using $\mathcal{P}$ at t can be computed from the speeds and accelerations when traversing $\mathcal{P}$, which can be derived from the GPS records on path $\mathcal{P}$, and road grades that are available in 3D road networks [24], using vehicular environmental impact models [8, 9].

Problem Definition: Given a large collection of trajectories $\mathbb{T}$ that occurred in a road network G, *travel cost distribution estimation* takes as input a path $\mathcal{P}$ in G and a departure time t, and *accurately* and *efficiently* estimates the travel cost distribution of traversing path $\mathcal{P}$ at t. The output is a univariate random variable that describes the distribution of the cost of traversing path $\mathcal{P}$ at t.

2.2 Accuracy-Optimal Cost Estimation

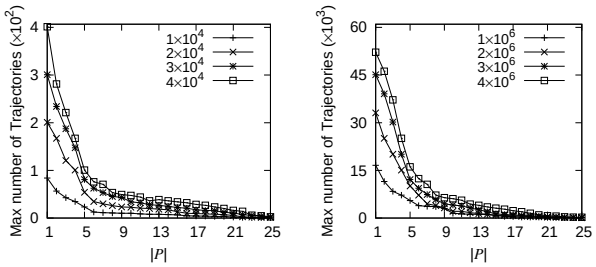
The most accurate way of estimating the travel cost distribution of path $\mathcal{P}$ at time t is to employ a sizable set of qualified trajectories. A trajectory $\mathcal{T}$ is qualified if $\mathcal{T}$ occurred on $\mathcal{P}$ at t' and the difference between t' and t is less than a threshold, e.g., 30 minutes. For instance, if we want to estimate the travel cost distribution

of path $\langle e_2, e_3, e_4 \rangle$ at 8:05, $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_5$, and $\mathcal{T}_6$ are qualified trajectories, but not $\mathcal{T}_7$ (cf. Figure 2).

A qualified trajectory captures traffic conditions (e.g., the time it takes to pass intersections, wait at traffic lights, and make turns at intersections) of the entire path $\mathcal{P}$ during the interval of interest. Thus, no explicitly modeling of complex traffic conditions at intersections is needed. To ensure an accurate estimation for a path $\mathcal{P}$ at t and to not overfit to the cost values of a few trajectories, we require the use of more than β qualified trajectories. The effect of parameter β is studied empirically in Section 5.

We regard this method as an accuracy-optimal baseline, and we let the resulting distribution $\mathbf{D}_{GT}(\mathcal{P}, t)$ serve the role of a *ground truth* distribution in our proposal since $\mathbf{D}_{GT}(\mathcal{P}, t)$ is the most accurate cost distribution computed from available trajectories.

However, this baseline is not always a practical approach, as it is very often inapplicable due to data sparseness. Figure 3 shows that the maximum number of trajectories that occurred on a path decreases rapidly as the cardinality of the path increases, based on two large trajectory collections from Aalborg and Beijing, with no time constraint applied.



(a) Aalborg, Denmark (b) Beijing, China

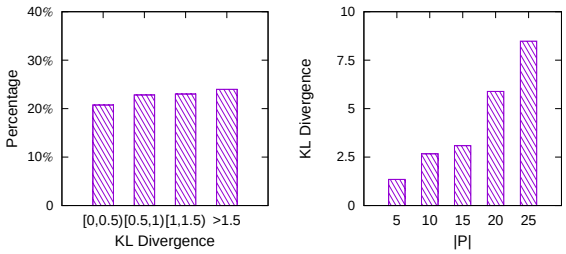
Figure 3: Data Sparseness Problem

2.3 Legacy Graph Model

To contend with the aforementioned data sparseness, existing stochastic route planning use a graph model that operates at *edge granularity* [6, 13, 17, 22]. The model is a weighted graph $G = (V, E, W_E)$ with an *edge weight function* $W_E : E \times T \rightarrow RV$, where T is the time domain of a day and RV denotes a set of random variables. The weight function takes as input an edge $e \in E$ and a time $t \in T$ and returns a random variable that represents the travel cost distribution of traversing e at t.

A recent study [22] instantiates the edge weight function W_E using the accuracy-optimal baseline on each individual edge, where sparseness is not likely to be a significant problem. Next, independence is assumed so that a path's travel cost distribution is the *convolution* of the travel costs distributions of the edges in the path. We denote the resulting distribution by $\mathbf{D}_{LB}(\mathcal{P}, t) = \odot_{e_i \in \mathcal{P}} W_E(e_i, t_{e_i})$, where $\odot$ denotes the convolution of two distributions and $W_E(e_i, t_{e_i})$ denotes the travel cost distribution of edge e_i at t_{e_i}. t_{e_i} is the arrival time on edge e_i, which may be different from the departure time t and needs to be progressively updated according to the travel times of e_i's predecessor edges [22].

To examine the effect of dependence on the accuracy of the result of convolution, we consider 500 paths that each consists of two adjacent edges (i.e., with path cardinality being 2) and on which at least 100 trajectories occurred during [7:30, 8:00). For each path $\mathcal{P}$, we compute the distribution D_{GT} using the accuracy-optimal baseline and distribution D_{LB} using the legacy baseline. If the distributions of the two edges in a path are independent, D_{GT} and D_{LB} should be identical. To see if this holds, we compute the KL-divergence of D_{LB} from D_{GT}, denoted as $KL(D_{GT}, D_{LB})$. The larger the KL-divergence, the more different the two distributions are, meaning that the convoluted distribution is less accurate. Figure 4(a) suggests that most of the adjacent edges are not independent.



(a) $KL(D_{GT}, D_{LB})$ (b) Varying $|\mathcal{P}|$

Figure 4: Examining the Independence Assumption

Next, we conduct an experiment on 100 paths with different cardinalities and where each path has at least 30 qualifying trajectories during an interval. We compute D_{GT} and D_{LB} for each path and then compute the average KL-divergence values between the two distributions for paths with varying cardinality. Figure 4(b) suggests that the more edges a path has, the more different the convoluted distribution D_{LB} is from the ground truth distribution D_{GT}. Hence, the legacy graph model is likely to yield inaccurate travel cost distributions, especially for long paths.

3. HYBRID GRAPH

The analyses of the legacy graph model suggest that the independency assumption does not always hold and that explicitly modeling of the dependency among the travel costs of different edges in a path is needed to achieve accurate results. This motivates us that when computing the cost distribution for a path, we should try to use trajectories that occurred on long sub-paths of the path because they capture the cost dependencies among different edges.

To this end, we propose a novel model—the **hybrid graph** model $G = (V, E, W_P)$. Instead of having an edge weight function W_E in the legacy graph model, the hybrid graph maintains a *path weight function* $W_P : Paths \times T \rightarrow RV$, where $Paths$ is a set of paths. Specifically, the path weight function W_P takes as input a path $\mathcal{P}$ and a time t and returns a multi-variate random variable that represents the joint distribution of path $\mathcal{P}$'s edges' travel costs. The joint distribution fully captures the dependency among the travel costs of different edges in path $\mathcal{P}$. We proceed to describe how to instantiate W_P using trajectories.

3.1 Instantiating W_P for Unit Paths

A *unit path* consists of a single edge. We partition a day into a few intervals, where parameter α specifies the finest-granularity interval of interest in minutes, e.g., 30 minutes. We let $V^{I_j}_{\langle e_i \rangle} = p(c_{e_i})$ denote a random variable that describes the travel cost distribution on unit path $\langle e_i \rangle$ during interval I_j.

To derive the distribution of $V^{I_j}_{\langle e_i \rangle}$, a set of qualified trajectories that occurred on $\langle e_i \rangle$ at t where $t \in I_j$ is obtained. If the trajectory set cardinality exceeds threshold β, the same parameter used in accuracy-optimal baseline in Section 2.2, the travel cost values obtained from the qualified trajectories are employed to instantiate the distribution of $V^{I_j}_{\langle e_i \rangle}$, which is the ground truth distribution.

If the set cardinality does not exceed β, the distribution of $V^{I_j}_{\langle e_i \rangle}$ is derived from the speed limit of edge e_i to avoid overfitting to the limited number of travel costs. We also regard this as the ground truth distribution, since based on the available trajectories, we cannot get a more accurate distribution for the unit path during the interval. Thus, in both cases, $V^{I_j}_{\langle e_i \rangle}$ represents the ground-truth distribution of unit path $\langle e_i \rangle$ during interval I_j.

Representing Univariate Distributions We proceed to discuss how to represent a distribution when having more than β trajectories. We represent distributions by histograms because they enable compact approximation of arbitrary, complex distributions. Gaussian mixture models are also able to represent non-standard travel cost distributions [21, 22], but are less compact. In particular, a one-dimensional histogram is employed to represent a univariate distribution.

From the qualified trajectories, we are able to obtain a multiset of cost values of the form $\langle cost, perc \rangle$, representing that $perc$ percentage of the qualified trajectories took cost $cost$. We call this a *raw cost distribution*. A histogram then approximates the raw cost distribution as a set of pairs: $\{\langle bu_i, pr_i \rangle\}$. A bucket $bu_i = [l, u)$ is a range of travel costs, and pr_i is the probability that the travel cost is in the range, and it holds that $\sum_i pr_i = 1$.

Given the number of buckets b, existing techniques, e.g., V-Optimal [12], are able to optimally derive a histogram based on a raw cost distribution such that the sum of errors between the derived histogram and the raw cost distribution is minimized. However, selecting a global value for b is non-trivial because the traffic on different edges, and even the traffic on the same edge during different intervals, may differ significantly. A self-tuning method is desired so that more buckets are used for edges or intervals with more complex traffic conditions.

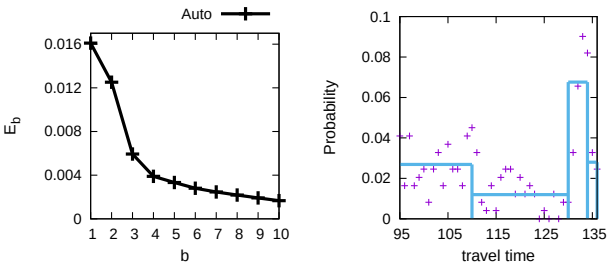
To this end, we propose a simple yet effective approach to automatically identify the number of buckets. The procedure starts with $b = 1$, i.e., using only one bucket, and computes an error value E_b. Next, it incrementally increases b by 1 and computes a new error value E_b. Obviously, as the number of buckets increases, the error value keeps decreasing. However, the error values often initially drop quickly, but then subsequently drop only slowly. Based on this, the process stops when the error value of using b does not lead to a significant decrease compared to the error value of using $b-1$. Then, $b-1$ is chosen as the bucket number. This yields a compact and accurate representation of the raw data distribution.

The error value E_b of using b buckets is computed using f-fold cross validation [18]. First, the multi-set of cost values is randomly split into f equal-sized partitions. Each time, we reserve the cost values in one partition, say the k-th partition, and use the cost values in the remaining f-1 partitions to generate a histogram with b buckets using V-Optimal, denoted as $H^k_b = \{\langle bu_i, pr_i \rangle\}$. Next, we compute the raw data distribution of the cost values in the reserved partition, denoted as $D^k = \{\langle cost_i, perc_i \rangle\}$. After that, we compute the squared error between H^k_b and D^k: $SE(H^k_b, D^k) = \sum_{c \in costs}(H^k_b[c], D^k[c])^2$. We repeat the procedure f times—once for each partition. The error value of using b buckets, i.e., E_b, is the average of the f squared errors.

Take the data in Figure 1(b) as an example. Figure 5(a) shows how the error E_b decreases as the number of buckets b increases. First, E_b decreases sharply and then slowly (i.e., when $b > 4$). Figure 5(b) shows the histogram using $b = 4$ buckets and the original raw data distribution.

3.2 Instantiating W_P for Non-Unit Paths

Based on the distributions of unit paths, we employ a bottom-up procedure to derive joint distributions of non-unit paths, i.e., paths



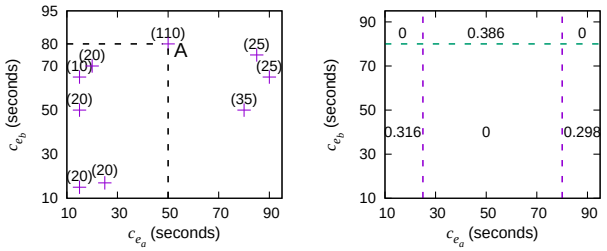
(a) E_b vs. b (b) Raw Dist. vs. Histogram

Figure 5: Identifying the Number of Buckets

with cardinalities more than one. In particular, the joint distributions of paths with cardinalities k, $k \geqslant 2$, are computed based on the joint distributions of paths with cardinalities $k-1$.

Given two paths $\mathcal{P}_i$ and $\mathcal{P}_j$ with cardinalities $k-1$, if they share $k-2$ edges and can be combined into a valid path $\mathcal{P} = \langle e_1, e_2, \ldots, e_k \rangle$ with cardinality k, we check if a time interval I_j exists during which more than β qualified trajectories occurred on path $\mathcal{P}$. If so, a random variable $V^{I_j}_{\mathcal{P}} = p(c_{e_1}, \ldots, c_{e_k})$ is instantiated based on the qualified trajectories. The travel cost distribution $p(c_{e_1}, \ldots, c_{e_k})$ is a ground-truth joint distribution on the k variables $c_{e_1}, \ldots, c_{e_k}$. This procedure continues until longer paths cannot be obtained.

For example, consider two unit paths $\mathcal{P}_a = \langle e_a \rangle$ and $\mathcal{P}_b = \langle e_b \rangle$, which can be combined into a valid path $\mathcal{P} = \langle e_a, e_b \rangle$. Figure 6(a) shows a raw joint distribution. Point A indicates that 110 trajectories passed e_a with cost 50 s and then e_b with cost 80 s.



(a) Actual Distribution (b) 2D Histogram

Figure 6: An Example of Multiple Dimensional Histogram

We use multi-dimensional histograms to describe joint distributions, where a dimension corresponds to the cost of an edge. A multi-dimensional histogram is a set of hyper-bucket and probability pairs: $\{\langle hb_i, pr_i \rangle\}$. A hyper-bucket $hb_i = \langle bu^1_i, \ldots, bu^n_i \rangle$ consists of n buckets that each corresponds to one dimension. Value pr_i equals the probability that the travel costs on multiple edges are in hyper-bucket hb_i, and it holds that $\sum_i pr_i = 1$.

To derive a multi-dimensional histogram, we automatically identify the optimal number of buckets for each dimension using the method from Section 3.1. Next, we employ V-optimal to identify the optimal bucket boundaries on each dimension and thus obtain a set of hyper-buckets. Finally, we compute the probability for each hyper-bucket. For example, Figure 6(b) shows a 2-dimensional histogram that corresponds to the joint distribution shown in Figure 6(a). The dimension for c_{e_a} is partitioned into 3 buckets and the dimension for c_{e_b} is partitioned into 2 buckets, yielding 6 hyper-buckets in the 2-dimensional histogram.

3.3 Path Weight Function W_P

We let $\mathbf{C}_{\mathcal{P}}$ be a vector of variables $\langle c_{e_1}, c_{e_2}, \ldots, c_{e_k} \rangle$ that corresponds to path $\mathcal{P} = \langle e_1, e_2, \ldots, e_k \rangle$. We use $p(\mathbf{C}_{\mathcal{P}}) = p(c_{e_1}, c_{e_2}, \ldots, c_{e_k})$ to denote the ground-truth joint distribution of using path $\mathcal{P}$ at time t. Following the same idea in Section 2.2, we regard the joint distribution obtained from at least β qualified trajectories as the ground-truth joint distribution.

Given a set of trajectories, we let $Paths_\beta$ be a set of non-unit paths where each path has at least β qualified trajectories. Note that given different sets of trajectories that occurred on a same road network, $Paths_\beta$ may contain different non-unit paths.

So far, we are able to instantiate path weight function W_P for non-unit paths in $Paths_\beta$ and all unit paths. Specifically, given a path $\mathcal{P}$ and a time $t \in I_j$, the path weight function $W_P(\mathcal{P}, t)$ returns random variable $V_{\mathcal{P}}^{I_j}$ that represents the travel cost distribution of traversing path $\mathcal{P}$ at t. Since each random variable $V_{\mathcal{P}}^{I_j}$ is obtained by at least β qualified trajectories (or speed limits for some unit paths), they also correspond to ground-truth distributions. Thus, we have

$$W_P(\mathcal{P}, t) = V_{\mathcal{P}}^{I_j} = p(\mathbf{C}_{\mathcal{P}}) = \begin{cases} p(c_{e_i}), & \mathcal{P} = \langle e_i \rangle \text{ is a unit path;} \\ p(c_{e_1}, c_{e_2}, \ldots, c_{e_k}), & \mathcal{P} = \langle e_1, e_2, \ldots, e_k \rangle \in Paths_\beta; \end{cases} \quad (1)$$

where (i) if $\mathcal{P}$ is a unit path, the path weight function returns the ground truth distribution $p(\mathbf{C}_{\mathcal{P}}) = p(c_{e_i})$, represented as a one-dimensional histogram; (ii) if $\mathcal{P}$ is a non-unit path, the path weight function returns the ground truth joint distribution $p(\mathbf{C}_{\mathcal{P}}) = p(c_{e_1}, c_{e_2}, \ldots, c_{e_k})$, represented as a multi-dimensional histogram.

The random variables $\{V_{\mathcal{P}}^{I_j}\}$ that are maintained in the path weight function W_P are called *instantiated random variables*. The *rank* of a variable $V_{\mathcal{P}}^{I_j}$ is the cardinality of its path $|\mathcal{P}|$. In the legacy graph model, only random variables with rank one are considered, and they cannot capture distribution dependencies among edges, which are found in trajectories. In contrast, in the proposed hybrid graph model, the random variables with rank larger than one fully capture the distribution dependencies among the edges in a path.

4. GETTING PATH COST DISTRIBUTION

Given any path $\mathcal{P}$ and a departure time t, we perform travel cost distribution estimation using the instantiated hybrid graph, in two steps. First, the joint distribution of path $\mathcal{P}$, which models the travel cost dependency among edges in $\mathcal{P}$, is computed. Second, the cost distribution of path $\mathcal{P}$, which captures the cost distribution of traversing the whole path $\mathcal{P}$, is derived based on the path's joint distribution.

4.1 The Joint Distribution of a Path

The ground truth joint distribution of path $\mathcal{P} = \langle e_1, e_2, \ldots, e_n \rangle$ at t is denoted as $p(\mathbf{C}_{\mathcal{P}}) = p(c_{e_1}, c_{e_2}, \ldots, c_{e_n})$, where c_{e_i} ($1 \leqslant i \leqslant n$) is a random variable representing the travel cost distribution of path $\langle e_i \rangle$.

Given a path $\mathcal{P}$ and a departure time t, we aim at estimating its most accurate joint distribution. If we are lucky, the path weight function $W_P(\mathcal{P}, t)$ returns $V_{\mathcal{P}}^{I}$, where $t \in I$. Since $V_{\mathcal{P}}^{I}$ corresponds to the ground truth join distribution $p(\mathbf{C}_{\mathcal{P}})$, we are done.

In contrast, if $W_P(\mathcal{P}, t)$ returns an empty result, this means that there does not exist at least β qualified trajectories on path $\mathcal{P}$ around t, so it is impossible to obtain its ground truth distribution. This case occurs often due to the data sparseness problem, as shown in Figure 3, especially for long paths.

To handle this unlucky but common case, we proceed to propose a method that is able to derive an accurate, estimated joint distribution $\hat{p}(c_{e_1}, c_{e_2}, \cdots, c_{e_n})$ based on the ground-truth distributions of path $\mathcal{P}$'s sub-paths, which can be obtained from the instantiated path weight function W_P. While we may be able to obtain multiple estimations of joint distributions using different combinations of sub-paths' distributions, we aim at identifying and deriving the most accurate one. In the following, we first prove that the combination with the coarsest sub-paths gives the most accurate estimation, and then we propose an efficient way to identify the coarsest combination.

4.1.1 Path Decompositions

To facilitate the following discussions, we first introduce the concept of *path decomposition*. The decomposition of a path $\mathcal{P}$ is a sequence of paths $DE = (\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k)$, $k > 1$, that satisfies the following *spatial conditions*:
(1) Each path $\mathcal{P}_i \in DE$ is a sub-path of $\mathcal{P}$;
(2) All paths in DE together cover $\mathcal{P}$, i.e., $\mathcal{P}_1 \cup \mathcal{P}_2 \cup \ldots \cup \mathcal{P}_k = \mathcal{P}$;
(3) A path $\mathcal{P}_i$ is not a sub-path of another path $\mathcal{P}_j$, where $1 \leqslant i, j \leqslant k$ and $i \neq j$;
(4) A path $\mathcal{P}_i$ appears before another path $\mathcal{P}_j$ where $1 \leqslant i < j \leqslant k$ if and only if the first edge of $\mathcal{P}_i$ appears earlier than the first edge of $\mathcal{P}_j$ in path $\mathcal{P}$.

A path $\mathcal{P}$ may have more than one decomposition. For instance, consider path $\mathcal{P} = \langle e_1, e_2, e_3, e_4, e_5 \rangle$. The following path sequences are possible decompositions for the path.
$DE_1 = (\langle e_1 \rangle, \langle e_2 \rangle, \langle e_3 \rangle, \langle e_4 \rangle, \langle e_5 \rangle)$,
$DE_2 = (\langle e_1, e_2, e_3 \rangle, \langle e_2, e_3, e_4 \rangle, \langle e_5 \rangle)$,
$DE_3 = (\langle e_1, e_2, e_3 \rangle, \langle e_3, e_4 \rangle, \langle e_5 \rangle)$.

Next, we introduce a *coarser* relationship between two path decompositions DE_i and DE_j. We define DE_i to be coarser than DE_j if for each path $\mathcal{P}_b \in DE_j$, there is a path $\mathcal{P}_a \in DE_i$ such that $\mathcal{P}_b$ is a sub-path of $\mathcal{P}_a$ and at least one $\mathcal{P}_b \neq \mathcal{P}_a$.

To illustrate, DE_2 is coarser than DE_3 because $\langle e_1, e_2, e_3 \rangle$, $\langle e_3, e_4 \rangle$, $\langle e_5 \rangle$ are sub-paths of paths $\langle e_1, e_2, e_3 \rangle$, $\langle e_2, e_3, e_4 \rangle$, $\langle e_5 \rangle$, respectively; and $\langle e_2, e_3, e_4 \rangle$ from DE_2 is different from $\langle e_3, e_4 \rangle$ from DE_3. Similarly, DE_2 is coarser than DE_1 as well.

4.1.2 Distribution Estimation using Decompositions

Following the principles of decomposable models [5, 7, 15], a decomposition of path $\mathcal{P}$ corresponds to a set of independence assumptions among the cost variables in $\mathbf{C}_{\mathcal{P}}$. Specifically, given a decomposition $DE = (\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k)$, for any two paths $\mathcal{P}_i$ and $\mathcal{P}_j$ in the decomposition, where $1 \leqslant i, j \leqslant k$, we have the following cases.
(i) If $\mathcal{P}_i \cap \mathcal{P}_j = \emptyset$, this indicates that the cost variables in $\mathbf{C}_{\mathcal{P}_i}$ are independent of the cost variables in $\mathbf{C}_{\mathcal{P}_j}$;
(ii) If $\mathcal{P}_i \cap \mathcal{P}_j \neq \emptyset$, this indicates that the cost variables in $\mathbf{C}_{\mathcal{P}_i \setminus \mathcal{P}_j}$ are conditionally independent of the cost variables in $\mathbf{C}_{\mathcal{P}_j \setminus \mathcal{P}_i}$ given the cost variables in $\mathbf{C}_{\mathcal{P}_i \cap \mathcal{P}_j}$.

Based on the above, given a decomposition DE, we have a corresponding independence assumption. Based on the independence assumption, we are able to estimate path P's joint distributions using the distributions of the paths in DE based on Bayes' theorem.

Formally, given a decomposition $DE = (\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k)$, the joint distribution of path $\mathcal{P}$ is estimated according to Equation 2.

$$\hat{p}_{DE}(\mathbf{C}_{\mathcal{P}}) = \frac{\prod_{\mathcal{P}_i \in P_X} p(\mathbf{C}_{\mathcal{P}_i})}{\prod_{\mathcal{P}_j \in P_Y} p(\mathbf{C}_{\mathcal{P}_j})}, \quad (2)$$

where path set $P_X = \bigcup_{1 \leqslant i \leqslant k} \mathcal{P}_i$ consists of all paths in DE and $P_Y = \bigcup_{2 \leqslant i \leqslant k} \mathcal{P}_i \cap \mathcal{P}_{i-1}$ consists of the shared paths between all adjacent paths in DE.

In the running example, DE_1 assumes all cost variables c_{e_1}, c_{e_2}, c_{e_3}, c_{e_4}, and c_{e_5} are independent because no paths in DE_1 intersect. According to Equation 2, we have $\hat{p}_{DE_1}(\mathbf{C}_\mathcal{P}) = p(c_{e_1}) \cdot p(c_{e_2}) \cdot p(c_{e_3}) \cdot p(c_{e_4}) \cdot p(c_{e_5})$. This corresponds to the legacy graph model where all cost variables are independent. Next, DE_2 assumes that c_{e_1} is conditionally independent of c_{e_4} given c_{e_2} and c_{e_3} because $\langle e_1, e_2, e_3\rangle \cap \langle e_2, e_3, e_4\rangle = \langle e_2, e_3\rangle$; and c_{e_5} is independent of all the other cost variables because $\langle e_5\rangle$ does not intersect with other paths in DE. Thus, Equation 2 gives us $\hat{p}_{DE_2}(\mathbf{C}_\mathcal{P}) = \frac{p(c_{e_1},c_{e_2},c_{e_3})\cdot p(c_{e_2},c_{e_3},c_{e_4})\cdot p(c_{e_5})}{p(c_{e_2},c_{e_3})}$.

Given a path $\mathcal{P}$, we have more than one path decomposition. For each decomposition, we are able to derive an estimated joint distribution according to Equation 2. The challenge is to identify the decomposition that gives the most accurate estimation.

To measure the accuracy of an estimated distribution with respect to the true distribution, we employ Kullback-Leibler divergence of the estimated joint distribution $\hat{p}_{DE}(\mathbf{C}_\mathcal{P})$ and the true joint distribution $p(\mathbf{C}_\mathcal{P})$, denoted as $KL(p(\mathbf{C}_\mathcal{P}), \hat{p}_{DE}(\mathbf{C}_\mathcal{P}))$. The smaller the divergence, the better. We proceed to prove that the coarser a decomposition is, the more accurate the estimated joint distribution is, i.e., the more smaller the divergence is.

THEOREM 1. *If path $\mathcal{P}'$ is a sub-path of path $\mathcal{P}$, we have $\sum_{\mathbf{C}_\mathcal{P}} p(\mathbf{C}_\mathcal{P}) \log p(\mathbf{C}_{\mathcal{P}'}) = -H(\mathbf{C}_{\mathcal{P}'})$, where $H(\cdot)$ is entropy function.*

Proof: Since path $\mathcal{P}'$ is a sub-path of path $\mathcal{P}$, path $\mathcal{P}$ can be represented as $\mathcal{P} = \mathcal{P}_s \circ \mathcal{P}' \circ \mathcal{P}_e$, where $\mathcal{P}_s$ and $\mathcal{P}_e$ are the possibly empty paths before and after path $\mathcal{P}'$, and $\circ$ denotes concatenation. Thus, the cost variables in $\mathbf{C}_{\mathcal{P}_s}$, $\mathbf{C}_{\mathcal{P}'}$, or $\mathbf{C}_{\mathcal{P}_e}$ must be a subset of the cost variables in $\mathbf{C}_\mathcal{P}$. In addition, we have $\mathbf{C}_{\mathcal{P}_s} \cup \mathbf{C}_{\mathcal{P}'} \cup \mathbf{C}_{\mathcal{P}_e} = \mathbf{C}_\mathcal{P}$. Based on the above, we get

$$
\begin{aligned}
&\sum_{\mathbf{C}_\mathcal{P}} p(\mathbf{C}_\mathcal{P}) \log p(\mathbf{C}_{\mathcal{P}'}) \\
&= \sum_{\mathbf{C}_{\mathcal{P}_s}, \mathbf{C}_{\mathcal{P}'}, \mathbf{C}_{\mathcal{P}_e}} p(\mathbf{C}_{\mathcal{P}_s}, \mathbf{C}_{\mathcal{P}'}, \mathbf{C}_{\mathcal{P}_e}) \log p(\mathbf{C}_{\mathcal{P}'}) \\
&= \sum_{\mathbf{C}_{\mathcal{P}'}} \big(\log p(\mathbf{C}_{\mathcal{P}'}) \sum_{\mathbf{C}_{\mathcal{P}_s}, \mathbf{C}_{\mathcal{P}_e}} p(\mathbf{C}_{\mathcal{P}_s}, \mathbf{C}_{\mathcal{P}'}, \mathbf{C}_{\mathcal{P}_e}) \big) \\
&= \sum_{\mathbf{C}_{\mathcal{P}'}} p(\mathbf{C}_{\mathcal{P}'}) \log p(\mathbf{C}_{\mathcal{P}'}) = -H(\mathbf{C}_{\mathcal{P}'}) \qquad \blacksquare
\end{aligned}
$$

THEOREM 2. *Given an estimated joint distribution $\hat{p}_{DE}(\mathbf{C}_\mathcal{P})$, we have $KL(p(\mathbf{C}_\mathcal{P}), \hat{p}_{DE}(\mathbf{C}_\mathcal{P})) = H_{DE}(\mathbf{C}_\mathcal{P}) - H(\mathbf{C}_\mathcal{P})$, where $H(\mathbf{C}_\mathcal{P})$ and $H_{DE}(\mathbf{C}_\mathcal{P})$ are the entropies of random variables $\mathbf{C}_\mathcal{P}$ under distributions $p(\mathbf{C}_\mathcal{P})$ and $\hat{p}_{DE}(\mathbf{C}_\mathcal{P})$, respectively.*

Proof: $KL(p(\mathbf{C}_\mathcal{P}), \hat{p}_{DE}(\mathbf{C}_\mathcal{P}))$

$$
\begin{aligned}
&= \sum_{\mathbf{C}_\mathcal{P}} p(\mathbf{C}_\mathcal{P}) \log \big(\frac{p(\mathbf{C}_\mathcal{P})}{\hat{p}_{DE}(\mathbf{C}_\mathcal{P})} \big) \\
&= -H(\mathbf{C}_\mathcal{P}) - \sum_{\mathbf{C}_\mathcal{P}} p(\mathbf{C}_\mathcal{P}) \log\, \hat{p}_{DE}(\mathbf{C}_\mathcal{P}) \\
&= -H(\mathbf{C}_\mathcal{P}) - \sum_{\mathbf{C}_\mathcal{P}} p(\mathbf{C}_\mathcal{P}) \big(\sum_{\mathcal{P}_i \in P_X} \log\, p(\mathbf{C}_{\mathcal{P}_i}) \\
&\quad - \sum_{\mathcal{P}_j \in P_Y} \log\, p(\mathbf{C}_{\mathcal{P}_j}) \big) \quad \text{(due to Eq. 2)} \\
&= -H(\mathbf{C}_\mathcal{P}) - \sum_{\mathcal{P}_i \in P_X} \big(\sum_{\mathbf{C}_\mathcal{P}} p(\mathbf{C}_\mathcal{P}) \log\, p(\mathbf{C}_{\mathcal{P}_i}) \big) \\
&\quad + \sum_{\mathcal{P}_j \in P_Y} \big(\sum_{\mathbf{C}_\mathcal{P}} p(\mathbf{C}_\mathcal{P}) \log\, p(\mathbf{C}_{\mathcal{P}_j}) \big) \\
&= -H(\mathbf{C}_\mathcal{P}) + \sum_{\mathcal{P}_i \in P_X} H(\mathbf{C}_{\mathcal{P}_i}) - \sum_{\mathcal{P}_j \in P_Y} H(\mathbf{C}_{\mathcal{P}_j}) \quad \text{(due to Th. 1)} \\
&= H_{DE}(\mathbf{C}_\mathcal{P}) - H(\mathbf{C}_\mathcal{P}) \qquad \blacksquare
\end{aligned}
$$

THEOREM 3. *Given two decompositions DE and DE', where DE is coarser than DE', DE is able to provide a more accurate joint distribution estimation than is DE'.*

Proof: Due to the space limitation, the detailed proof is included in the supplementary document [31]. ■

4.1.3 Identifying the Coarsest Decomposition

According to Theorem 3, the estimated joint distribution from the coarsest decomposition is the most accurate. Given a path $\mathcal{P}$ and a departure time t, we proceed to identify the coarsest decomposition for $\mathcal{P}$ based on the instantiated random variables that are maintained in the path weight function W_P. Specifically, we first identify the random variables that are *spatially* relevant to the path $\mathcal{P}$ and *temporally* relevant to the departure time t; we then identify the coarsest decomposition from the relevant variables.

Recall that a random variable maintained in the path weight function W_P is in the form of $V_{\mathcal{P}_i}^{I_j}$, which represents the joint distribution of path $\mathcal{P}_i$ during interval I_j. A random variable $V_{\mathcal{P}_i}^{I_j}$ is spatially relevant to the query path $\mathcal{P}$ if the variable's path $\mathcal{P}_i$ is a sub-path of $\mathcal{P}$. Next, we test if a spatially relevant random variable $V_{\mathcal{P}_i}^{I_j}$ is also temporally relevant to t.

We distinguish two cases. We first consider the case where the first edge in the variable's path $\mathcal{P}_i$ is the same as the first edge in the query path $\mathcal{P}$. This case is simple because the departure time on $\mathcal{P}$ is also the departure time on $\mathcal{P}_i$, as both paths start from the same edge. Thus, we just need to test if the departure time t is during I_j, i.e., $t \in I_j$. The second case is complicated because the departure time on path $\mathcal{P}_i$ is no longer the original departure time t. We propose a *shift-and-enlarge* procedure to progressively update the departure time to test the variable's temporal relevance.

We use an example path $\langle e_1, e_2\rangle$ and a departure time t to illustrate the procedure. Since the travel time on e_1 is uncertain, the departure time on e_2 belongs to the interval $[t + V_{\langle e_1\rangle}^{I_j}.min, t + V_{\langle e_1\rangle}^{I_j}.max]$, where $V_{\langle e_1\rangle}^{I_j}.min$ and $V_{\langle e_1\rangle}^{I_i}.max$ denote the minimum and maximum travel times of traversing e_1, which are recorded in the random variable $V_{\langle e_1\rangle}^{I_j}$.

Formally, given a time interval $[t_s, t_e]$ and a random variable $V_{\langle e_k\rangle}^{I_j}$, we define the *shifted-and-enlarged* interval as $SAE([t_s, t_e], V_{\langle e_k\rangle}^{I_j}) = [t_s + V_{\langle e_k\rangle}^{I_j}.min, t_e + V_{\langle e_k\rangle}^{I_j}.max]$.

Given a sub-path $\mathcal{P}_i$ of path $\mathcal{P}$, assume that the first edge in $\mathcal{P}_i$ is the k-th edge in $\mathcal{P}$, where $2 \leqslant k \leqslant |\mathcal{P}|$. The updated departure time on $\mathcal{P}_i$ based on the original departure time t is UI_k, as defined in Equation 3.

$$
UI_k = \begin{cases} SAE([t,t], V_{\langle e_1\rangle}^{I_j}), & \text{if } k\text{=2;} \\ SAE(UI_{k-1}, V_{\langle e_{k-1}\rangle}^{I_j}), & \text{otherwise;} \end{cases} \tag{3}
$$

Given a spatially relevant random variable $V_{\mathcal{P}_i}^{I_j}$, if I_j intersects $\mathcal{P}_i$'s updated departure time interval UI_k according to Equation 3 then it is temporally relevant; otherwise, it is not temporally relevant. For a given sub-path $\mathcal{P}_i$, if multiple variables $V_{\mathcal{P}_i}^{I_j}$, $(j = 1, 2, \cdots, m)$ are temporally relevant, the one with the largest overlap is selected, i.e., the one where $I_j = \arg\max_{j\in[1,m]} \frac{|I_j \cap UI_k|}{|UI_k|}$.

Having identified all spatially and temporally relevant variables, we organize them into a two-dimensional candidate array. Each row corresponds to an edge e_k in query path $\mathcal{P}$ and contains the instantiated random variables whose corresponding paths start with

edge e_k. If more than one variable exist, the variables are ordered by their rank. An example candidate array is shown in Table 1.

	$rank = 1$	$rank = 2$	$rank = 3$	$rank = 4$
e_1	$V_{\langle e_1 \rangle}$	$V_{\langle e_1, e_2 \rangle}$	$V_{\langle e_1, e_2, e_3 \rangle}$	$V_{\langle e_1, e_2, e_3, e_4 \rangle}$
e_2	$V_{\langle e_2 \rangle}$	$V_{\langle e_2, e_3 \rangle}$	$V_{\langle e_2, e_3, e_4 \rangle}$	
e_3	$V_{\langle e_3 \rangle}$	$V_{\langle e_3, e_4 \rangle}$		
e_4	$V_{\langle e_4 \rangle}$	$V_{\langle e_4, e_5 \rangle}$		
e_5	$V_{\langle e_5 \rangle}$			

Table 1: Example Candidate Array

To identify the coarsest decomposition, we consider the path of the random variable with the highest rank for each edge (i.e., the rightmost variable for each row in Table 1). If one random variable's path is a sub-path of another random variable's path, the former random variable's path should be omitted because it otherwise violates spatial condition (3) (cf. Section 4.1.2). The remaining paths constitute the coarsest decomposition. The procedure is summarized in Algorithm 1.

To illustrate the procedure, consider the example shown in Table 1. In the beginning, we consider path $\langle e_1, e_2, e_3, e_4 \rangle$ and add it to DE_{coa}. Next, since $\langle e_2, e_3, e_4 \rangle$ and $\langle e_3, e_4 \rangle$ are sub-paths of $\langle e_1, e_2, e_3, e_4 \rangle$, both are omitted. Then, $\langle e_4, e_5 \rangle$ is added to DE_{coa}. Since $\langle e_5 \rangle$ is a sub-path of $\langle e_4, e_5 \rangle$, it is omitted. Finally, the coarsest decomposition is $DE_{coa} = (\langle e_1, e_2, e_3, e_4 \rangle, \langle e_4, e_5 \rangle)$, which is highlighted in Table 1.

Algorithm 1: Identify the Coarsest Decomposition

Input : Query Path $\mathcal{P}$, Departure Time t
Output: The Coarsest Decomposition DE_{coa}

1 Identify a set of random variables $STRV$ that are spatially relevant to path $\mathcal{P}$ from all the instantiated random variables;
2 Eliminate the random variables that are not temporally relevant to departure time t from $STRV$;
3 Organize the spatio-temporally relevant random variables in $STRV$ in an two dimensional array;
4 $DE_{coa} \leftarrow null$;
5 **for** $k = 1; k \leqslant |\mathcal{P}|; k++$ **do**
6 Identify the random variable $V_{\mathcal{P}_k}^{I_j}$ with the highest rank from the k-th row;
7 **if** $\mathcal{P}_k$ is not a sub-path of any path in DE_{coa} **then**
8 Append $\mathcal{P}_k$ to DE_{coa};
9 **return** DE_{coa};

THEOREM 4. *The unique decomposition DE_{coa} returned by Algorithm 1 is the coarsest.*

Proof: We prove the theorem by contradiction. Suppose that we cannot identify the coarsest decomposition by Algorithm 1. This happens only if the coarsest decomposition, say DE'_{coa}, contains a sub-path P'_k that starts with edge e_k but is not the longest sub-path that starts with edge e_k. Otherwise, it must be identified by Algorithm 1 since it considers the longest sub-path for each edge in path $\mathcal{P}$.

Following the above assumption, we assume that the longest sub-path starting from e_k is P_k. By replacing P'_k by P_k, we are able to get a new decomposition DE_{coa} that is coarser than DE'_{coa}. This contradicts the assumption that DE'_{coa} is the coarsest. ■

4.2 The Cost Distribution of a Path

The coarsest decomposition DE_{coa} enables accurate estimation of the joint distribution of a path which fully captures the dependencies among edges in the path. Recall that we are interested in knowing the cost distribution of a path $p(V_{\mathcal{P}})$, where $V_{\mathcal{P}}$ is a univariate random variable indicating the travel cost of path $\mathcal{P}$. We proceed to derive $p(V_{\mathcal{P}})$ based on the joint distribution of a path $\hat{p}_{DE_{coa}}(\mathbf{C}_{\mathcal{P}})$ using

$$p(V_{\mathcal{P}} = x) = \sum_{c_1 + \ldots + c_n = x} \hat{p}_{DE_{coa}}(c_{e_1} = c_1, \ldots, c_{e_n} = c_n).$$

Since the estimated joint distribution of a path $\hat{p}_{DE_{coa}}(\mathbf{C}_{\mathcal{P}})$ is represented as a multi-dimensional histogram, we need to transform it to a one-dimensional histogram that represents the cost distribution of $\mathcal{P}$.

Recall that a multi-dimensional histogram is of the form $\{\langle hb_i, pr_i \rangle\}$, where hyper-bucket $hb_i = \langle bu_i^1, \ldots bu_i^n \rangle$ consists of n buckets, each corresponding to one dimension. For each hyper-bucket $hb_i = \langle bu_i^1, \ldots bu_i^n \rangle$, we derive a bucket bu_i whose upper (lower) bound is the sum of the upper (lower) bounds of the buckets in the hyper-bucket, i.e., $bu_i = [\sum_{j=1}^{n} bu_i^j.l, \sum_{j=1}^{n} bu_i^j.u)$. Thus, we get a one-dimensional histogram $\{\langle bu_i, pr_i \rangle\}$.

The buckets in the obtained one-dimensional histogram may overlap. We need to rearrange the buckets such that they are disjoint and update their corresponding probabilities. We check each pair of buckets as follows. If two buckets bu_i and bu_j are disjoint, keep both buckets. If buckets bu_i and bu_j overlap, range $[\min(bu_i.l, bu_j.l), \max(bu_i.u, bu_j.u))$ is split into three buckets according to the increasing order of $bu_i.l$, $bu_j.l$, $bu_i.u$, and $bu_j.u$, and each bucket is assigned an adjusted probability. The one-dimensional histogram with the rearranged buckets and the adjusted probabilities represents the final cost distribution.

Figure 7 shows a running example on the aforementioned procedure on path $\mathcal{P}_1 = \langle e_1, e_2 \rangle$. The first table in Figure 7 shows the joint distribution of the path. The upper, left hyper-bucket $\langle [20, 30), [20, 40) \rangle$ has value 0.3, which means that when going through path $\mathcal{P}_1$, the probability that the travel time on e_1 is between 20 s and 30 s and the travel time on e_2 is between 20 s and 40 s is 0.3. Next, the second table in Figure 7 shows the corresponding cost distribution after transferring each hyper-bucket to a bucket. For example, hyper-bucket $\langle [20, 30), [20, 40) \rangle$ becomes bucket $[40, 70)$.

	$c_{e_1} \in [20, 30)$	$c_{e_1} \in [30, 50)$
$c_{e_2} \in [20, 40)$	0.30	0.25
$c_{e_2} \in [40, 60)$	0.20	0.25

$[40, 70)$	$[50, 90)$	$[60, 90)$	$[70, 110)$
0.30	0.25	0.20	0.25

$[40, 50)$	$[50, 60)$	$[60, 70)$	$[70, 90)$	$[90, 110)$
0.1000	0.1625	0.2292	0.3833	0.1250

Figure 7: A Joint Distribution and Its Marginal Distribution

Consider the first two (bucket, probability) pairs shown in the second table, i.e., $\langle [40, 70), 0.30 \rangle$ and $\langle [50, 90), 0.25 \rangle$. Since the two buckets overlap, range $[40, 90)$ is split into $[40, 50)$, $[50, 70)$, and $[70, 90)$. In a histogram, the probability in each bucket is uniformly distributed, so each bucket is assigned an adjusted probability as follows. Bucket $[40, 50)$ is given probability $\frac{|[40,50)|}{|[40,70)|} \cdot 0.3 = 0.1$, bucket $[50, 70)$ is given probability $\frac{|[50,70)|}{|[40,70)|} \cdot 0.3 + \frac{|[50,70)|}{|[50,90)|} \cdot$

$0.25 = 0.325$, and bucket $[70, 90)$ is associated with probability $\frac{|[70,90)|}{|[50,90)|} \cdot 0.25 = 0.125$. Bucket $[40, 50)$ does not overlap with other buckets, its adjusted probability is the final probability. Since buckets $[50, 70)$ and $[70, 90)$ still overlap with the next bucket $[60, 70)$, their buckets should be further rearranged and their probabilities should be adjusted. The final cost distribution is shown in the third table in Figure 7.

4.3 Using the Proposed Method

Recall the example in Figure 1(a) with the question "Which path has a higher probability of arriving the airport within 60 mins?" To illustrate how the proposed method is used for accurate estimation of the cost distributions of paths in path planning in order to answer such questions, we consider two typical scenarios.

In the first scenario, multiple candidate paths, e.g., using or not using highways or toll roads, are given. Here, the proposed method can be applied to compute the cost distributions for the given paths, identifying the most appropriate path based on the requirements, e.g., the one with highest probability of arriving within 60 mins.

The second scenario concerns stochastic routing algorithms, e.g., stochastic fastest routing [13], probabilistic top-k routing [10], stochastic skyline routing [22], that aim to identify a path or set of paths that satisfy some conditions, e.g., the path with highest probability of arriving within 60 mins. Such algorithms need to explore many candidate paths, and pruning of uncompetitive candidate paths early is attractive. When making pruning decisions, the algorithms compare the cost distributions of candidate paths. The proposed method can be easily incorporated into existing stochastic routing algorithms by simply using it when the algorithms compute cost distributions of paths.

Existing stochastic routing algorithms apply a "path + another edge" pattern to explore candidate paths. Therefore, a path cost distribution estimation method must satisfy the so-called "incremental property" that enables reuse of the cost distribution of the existing path when computing the cost distribution for a new path extended from this existing path. Our proposal satisfied this property and can be applied to "path + another edge" path finding algorithms. The details are covered elsewhere [31].

The run-time of a stochastic routing algorithm is dominated by $\sum_{\mathcal{P} \in CP} RT(\mathcal{P}, method)$, where CP contains the candidate paths whose cost distributions need to be evaluated. Thus, CP differs among stochastic routing algorithms with different strategies for selecting candidate paths. Function $RT(\mathcal{P}, method)$ refers to the run-time of computing the cost distribution of path $\mathcal{P}$ using a specific method, e.g., our method or a legacy method. In Section 5, we empirically demonstrate that computing the cost distribution of a path using our method is more efficient than using the state-of-the-art legacy baselines, i.e., $RT(\mathcal{P}, hybrid\_graph) < RT(\mathcal{P}, legacy\_baseline)$. We also show that the accuracy of the cost distributions estimated by our method is higher than that of the legacy baseline. Thus, integration of our method is able to improves both the efficiency and accuracy of existing stochastic routing algorithms.

5. EMPIRICAL STUDY

5.1 Experimental Setup

Road networks: Two road networks are used. The Aalborg road network N_1 has 20,195 vertices and 41,276 edges, and the Beijing road network N_2 has 28,342 vertices and 38,577 edges. Road network N_1 is obtained from OpenStreetMap and contain all roads, while road network N_2 is obtained from the Beijing traffic management bureau and contains only highways and main roads.

Trajectories: Two GPS data sets are used. The first, D_1, contains 37 million GPS records that occurred in Aalborg from January 2007 to December 2008. The sampling rate is 1 Hz (i.e., one record per second). The second, D_2, contains more than 50 billion GPS records that occurred in Beijing from September 2012 to November 2012. The sampling rate is at least 0.2 Hz. We apply a well-known method [16] to map match the GPS records.

Travel Costs: We consider two time-varying, uncertain travel costs—travel time and GHG emissions. Due to the space limitation, the results on GHG emissions are included elsewhere [30].

Parameters: We vary the finest time interval α, the qualified trajectory count threshold β, the cardinality of a query path $|\mathcal{P}_{query}|$; see Table 2, where default values are shown in bold.

Parameters	Values
α (min)	15, **30**, 45, 60, 120
β	15, **30**, 45, 60
$\lvert\mathcal{P}_{query}\rvert$	5, 10, 15, **20**, 40, 60, 80, 100

Table 2: Parameter Settings

Implementation Details: All algorithms are implemented in Python 2.7 under Linux Ubuntu 14.04. All experiments are conducted on a modern server with 512 GB main memory and 64 2.3 GHz 8-core AMD Opteron(tm) 6376 CPUs.

5.2 Experimental Results

5.2.1 Instantiated Random Variables

We conduct experiments to obtain insight into different aspects of the instantiated random variables that are maintained in the hybrid graph's weight function W_P and also describe how to tune parameters α and β. To highlight the random variables that are instantiated from trajectories, random variables derived from speed limits are excluded unless stated otherwise.

Tuning α: We vary α from 15 to 120 minutes. A large α suggests that more trajectories may become qualified trajectories, which instantiates more random variables. We use E' to denote the set of edges that are covered by the instantiated random variables and E'' to denote the set of edges that has at least one GPS record. Coverage is defined as the ratio between $|E'|$ and $|E''|$. Figure 8(a) shows that as α increases, the coverage increases as well on both data sets. However, the coverage ratio remains below 85% for $\alpha = 120$. This is because the GPS data is skewed—some edges have only few GPS records.

Although a large α enables more instantiated random variables, they may be inaccurate since traffic may change significantly during a long interval, e.g., two hours. We report the average entropies of the instantiated random variables when varying α; see Figure 8(b). We only show the results on D_2 as the results on D_1 exhibit similar trends. According to Theorem 2, variables with smaller entropy lead to more accurate joint distribution estimates. Figure 8(b) shows that using $\alpha = 30$ does not significantly increase the entropy compared to using $\alpha = 15$. Figure 8(a) shows that $\alpha = 30$ gives a clear increase in the number of instantiated random variables compared to $\alpha = 15$. This suggests that $\alpha = 30$ provides a good trade-off between the accuracy of the random variables and the numbers of random variables. Thus, we use $\alpha = 30$ as the default value.

Tuning β: Intuitively, we prefer a large β since having more qualified trajectories enables instantiated random variables that are more accurate. However, Figure 9 shows that as β increases, the

number of instantiated random variables drops. This occurs because a large β requires more qualified trajectories in order to be able to instantiate W_P. We choose $\beta = 30$ as the default because the number of instantiated random variables is only slightly less than than for $\beta = 15$, while achieving more accurate variables.

Varying Dataset Sizes: We use 25%, 50%, 75%, and 100% of the trajectories in D_1 and D_2, respectively. Figure 10 shows that the number of instantiated random variables increases as the number of trajectories increases. The number of instantiated random variables with large rank also increases steadily. This occurs because the more trajectories, the more likely it is to find long paths with more than β qualified trajectories, thus resulting in random variables with large rank. It also shows that the instantiated random variables are typically insufficient to enable the *accuracy-optimal* baseline for arbitrary paths—the sizes of variables with high rank (e.g.,$|V| \geqslant 4$) are small.

Histogram Approximation: We evaluate the accuracy and space savings of the histogram representations of the instantiated random variables. Recall that our method is able to automatically identify the number of buckets per dimension (cf. Section 3.1). We call this method *Auto*. We first compare *Auto* with methods using standard distributions including Gaussian, Gamma, and exponential distributions. Figure 11(a) shows the KL divergences between the raw data distribution and the distributions represented by different methods. The results of using exponential distributions are not shown since their KL divergences exceed 1.0 and are significantly worse than the other ones. The results clearly suggest that *Auto* provides the most accurate estimation and that travel-time distributions typically do not follow standard distributions.

Auto adaptively determines the bucket count for each dimension and then optimally selects the bucket boundaries, thus being able to represent arbitrary distributions. We compare *Auto* with a static histogram approach that uses a fixed number of buckets per dimension. The method that uses b buckets per dimension is called *Sta-b*. Figure 11(b) shows the KL divergences between the raw distribution that is obtained from the trajectories' travel costs and the different histograms. As the number of buckets increases, *Sta-b* produces a smaller KL divergence value because the more buckets a histogram has, the better accuracy it can achieve. *Auto* is able to achieve as good accuracy as *Sta*-4. This suggests that the *Auto* method is effective.

We evaluate the space-savings achieved by the histogram representation. Intuitively, the more buckets a histogram has, the more storage it needs. We report the space saving ratio $1 - \frac{S_H}{S_R}$, where S_H and S_R represent the storage required by the histograms and the underlying raw data distribution, respectively. The raw data distribution is of the form $(cost, frequency)$. The higher the ratio, the better space-savings are achieved by the histograms. Figure 11(c) shows that *Auto* has a better compression ratio than *Sta*-4 has. This suggests that *Auto* achieves a good trade-off between accuracy and space-saving.

Total Memory Usage: As the size of the trajectory data set grows, the memory use of recording the instantiated random variables also grows, as shown in Figure 12. Since we use histograms to represent the distributions of instantiated random variables, the memory use is small such that the hybrid graph's weight function W_P can be accommodated in main memory. In particular, the instantiated random variables, including the ones that are derived from speed limits, for Aalborg and Beijiing occupy around 1.8 GB and 4.2 GB, respectively.

Run-time: Since deriving the instantiated random variables is an off-line task, the run-time is not critical. The procedure can be parallelized in a straightforward manner. Using the default parameter setting, it takes less than 2 minutes with 48 threads to learn the random variables from D_1, and takes around 45 minutes with 48 threads to learn random variables from D_2. This also suggests that when receiving new trajectories regularly, the procedure can be conducted periodically to efficiently instantiate random variables.

5.2.2 Path Cost Distribution Computation

We consider four methods. (a) *OD*: the proposed method using the optimal (i.e., coarsest) decomposition. (b) *LB* [22]: the legacy baseline as described in Section 2.3. *LB* is regarded as one of the state-of-the-art approaches used in the conventional paradigm. In our setting, *LB* only considers the random variables with ranks one. (c) *HP* [10] assumes that the joint distributions for every pair of edges in a path are known and then computes the joint probability distribution of the path taking these into account. In our setting, this means that *HP* only considers random variables with ranks two. (d) *RD* computes an estimated distribution using a randomly chosen decomposition rather than the coarsest decomposition.

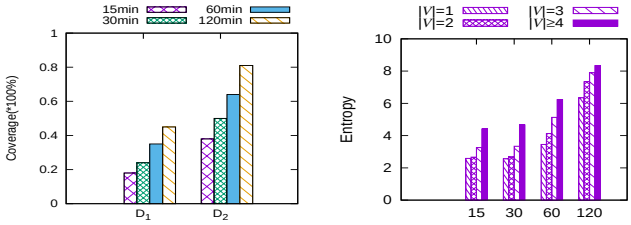
Accuracy Evaluation with Ground Truth: We select 100 paths where each path has more than $\beta = 30$ trajectories during an interval of $\alpha = 30$ minutes. We use these trajectories to compute the ground-truth distribution using the accuracy-optimal baseline. Next, we exclude these trajectories from the trajectory data set. Thus, we have the data sparseness problem, and the accuracy-optimal baseline does not work.

First, we consider a concrete example shown in Figure 1(b). The distributions estimated using *OD*, *LB*, *HP*, and *RD* are shown in Figures 13(a)-(d). It is clear that *OD* captures the main characteristics of the ground-truth distribution. The convolution-based estimation *LB* seems to approach a Gaussian distribution (cf. the Central Limit Theorem). However, it is clear that a Gaussian distribution is unable to capture the ground-truth distribution, and *LB* is inaccurate. The distribution computed by *HP* is inaccurate either, which suggests that the dependencies among the edges in a path cannot be fully captured by only considering the dependencies between adjacent edges. Method *RD* suggests that a randomly chosen decomposition provides a less accurate estimation compared to the estimation based on the optimal decomposition.

Next, we report results when using paths with different cardinalities. Specifically, we vary $|P_{query}|$ from 5 to 20. Figure 14 shows the KL-divergence values $KL(p, \hat{p})$, where p is the ground-truth distribution derived by the accuracy-optimal baseline and $\hat{p}$ is the estimated distribution using *OD*, *LB*, *RD*, and *HP*. As the number of edges in a path increases, the benefits of using the proposed *OD* becomes more significant. In particular, the KL-divergence values of *OD* grow slowly while the KL-divergence values of *LB* grow quickly. This is not surprising because *LB* assumes independencies, and the longer a path is, the more likely it is that the edges in the path are not independent. Next, *OD* is also better than *RD*, which suggests that the optimal decomposition produces the most accurate estimation. Further, *HP* is better than *LB* because *HP* considers the correlation between adjacent edges. However, *HP* always has larger KL-divergence values than do *RD* and *OD*. This is because coarser random variable sets have smaller KL-divergence (cf. Theorem 3).

In summary, Figure 14 suggests that the proposed *OD* is able to accurately estimate travel cost distributions and that it outperforms the other methods, especially for long paths.

Accuracy Evaluation without Ground Truth: We consider long paths which do not have corresponding ground truth distributions. We randomly choose $1,000$ paths for each cardinality with an arbitrary departure time and report average values; and vary the path cardinality from 20 to 100. Figure 15 shows that *OD* produces



(a) Coverage (b) Entropy

Figure 8: Effect of α

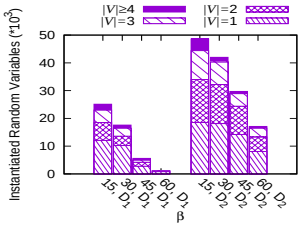


Figure 9: Effect of β

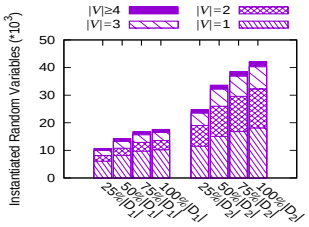
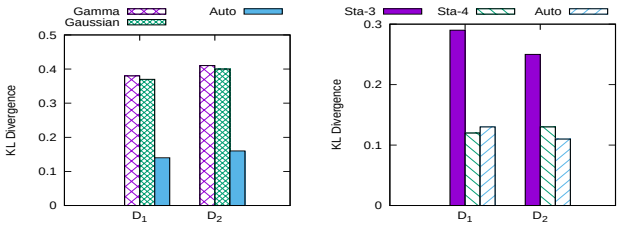


Figure 10: Varying Dataset Sizes



(a) Varying Methods (b) Varying Histograms (c) Space Saving Ratio

Figure 11: Performance of Multi-Dimensional Histograms

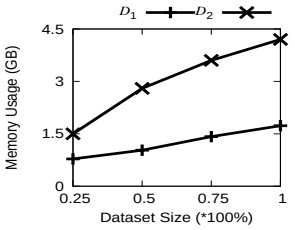
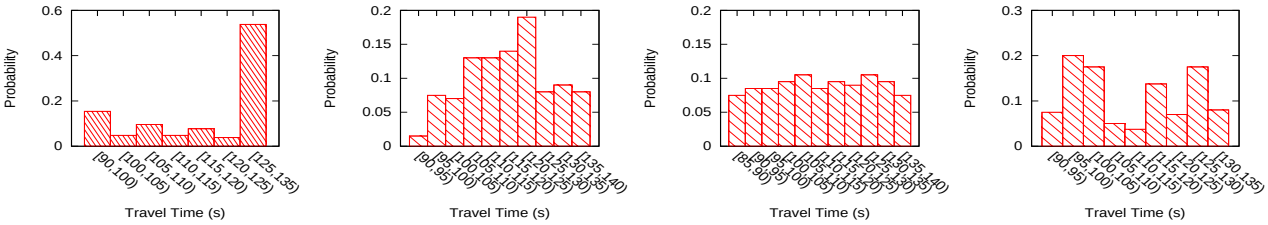
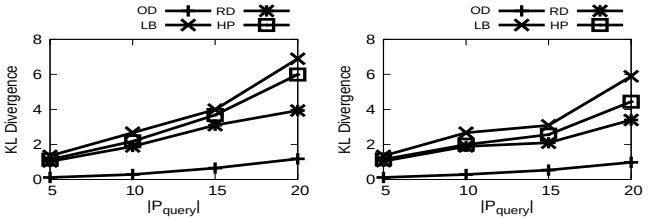


Figure 12: Memory Usage



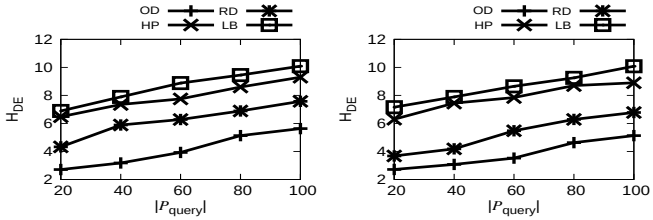
(a) *OD* (b) *LB* (c) *HP* (d) *RD*

Figure 13: Accuracy Comparison on a Particular Path



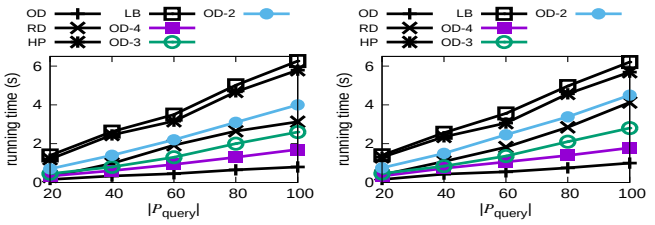
(a) D_1 (b) D_2

Figure 14: Accuracy Comparison With The Ground Truth



(a) D_1 (b) D_2

Figure 15: Entropy Comparison



(a) D_1 (b) D_2

Figure 16: Efficiency

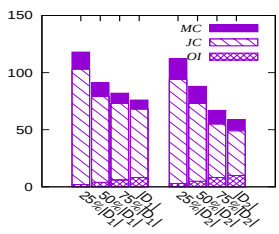


Figure 17: Run-Time Analysis

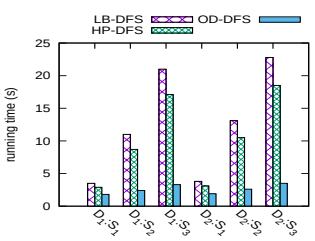


Figure 18: Route Planning Time

the least entropy, which is consistent with the design of identifying the optimal decomposition. This suggests that the proposed method is able to accurately estimate the distribution of a path.

Efficiency, Estimating the Cost Distribution of a Path: Figure 16 reports the run-times of the different methods. We also consider the methods that use instantiated random variables with ranks at most 2, 3, and 4, denoted as *OD-2*, *OD-3*, and *OD-4*, respectively. As the cardinality of a query path increases, the run-time also grows. *HP* and *LB* have to consider at least $|\mathcal{P}_{query}|$ instantiated random variables to compute the joint distribution, yielding higher running times than for the remaining methods. In contrast, *OD*, *OD-x*, and *RD* exploit instantiated random variables with higher ranks. Thus, they use significantly fewer instantiated random variables and are faster than *HP* and *LB*. *OD* uses coarser random variables than does *RD*, and it is able to use fewer instantiated random variables, making it faster than *RD*. Following the same reasoning, *OD-x* is faster than *OD-y* if $x > y$. Figure 16 clearly shows that *OD* is the most efficient.

To further investigate the run-time of *OD*, a detailed analysis of the three major steps in *OD* is reported in Figure 17 for paths with cardinality 20 and using differently sized subsets of trajectories.

Three steps are involved in *OD*. First, the optimal decomposition is identified, denoted by *OI*. Thanks to Theorems 4, this part is very efficient. Second, the joint distribution is computed, denoted by *JC*. This is the most time-consuming part as it goes through many hyper-buckets of the histograms to compute the joint distributions according to Equation 2. However, with more trajectories, there are more instantiated random variables with higher ranks, which improves the run-time of *JC*. Thus, as data volumes increase, the performance improves. Third, deriving the cost distribution (denoted by *MC*) is also very efficient.

Efficiency, Stochastic Routing: We consider the efficiency of using the hybrid graph in existing stochastic routing algorithms following the discussions offered in Section 4.3. In particular, we use an existing depth first search (DFS) based stochastic routing algorithm [10] for answering the question posed in Figure 1(a). We implement the algorithm in versions that use *LB*, *HP*, or *OD* to compute cost distributions, respectively.

We randomly choose 100 source-destination pairs, and we set the travel time budgets to $S_1 = 10$ min, $S_2 = 20$ min, and $S_3 = 30$ min, respectively. The average running times are reported in Figure 18, which shows that *OD-DFS* always outperforms the other two. We see that replacing the legacy baselines by the paper's proposal is able to accelerating an existing stochastic routing algorithm. Due to the space limitation, experimental results on the effects on quality of using the paper's proposal for stochastic routing are covered elsewhere [31].

5.2.3 Summary

The empirical study shows that: (1) In realistic settings with sparse data, the proposed *OD* method is the most accurate and efficient method and is scalable w.r.t. the path cardinality, meaning that it is able to support long paths. (2) *OD* is able to approximate arbitrary raw cost distributions well using limited space, making it possible to fit the instantiated random variables into main memory. (3) *OD* is scalable w.r.t. the number of trajectories. First, as random variable instantiation can be parallelized easily, it is possible to periodically re-instantiate random variables when new trajectories are received. Second, more trajectories yield more random variables with higher ranks, which improves the efficiency and accuracy of the approach. (4) *OD* can be incorporated into existing stochastic routing algorithms with the effect of improving their efficiency and accuracy.

We conclude that the proposed *hybrid-graph* and *OD* method successfully address the challenges caused by *data sparseness*, *complex distributions*, and *dependencies*; and they are able to efficiently provide accurate travel cost distribution estimation, thus enabling efficient and accurate stochastic routing.

6. RELATED WORK

We review recent studies on estimating *deterministic* and *uncertain* path costs, respectively.

Estimating Deterministic Path Costs: Most such studies focus on accurate estimation of travel costs of individual edges using trajectory data and loop detector data, based on which the travel cost of a path is then computed as the sum of the travel costs of its edges.

In many cases, the available trajectory data is unable to cover all edges in a road network. To address data sparseness, some methods [11,19,26,29] transfer the travel costs of edges that are covered by trajectories to edges that are not covered by trajectories. However, these methods do not support travel cost distributions, and they do not model dependencies among edges. Therefore, they do not apply to the problem we consider.

When all edges have travel costs, the travel cost of any path can be estimated by summing up the travel costs of the edges in the path [11, 26, 29]. However, using the sum of travel costs of edges as the travel cost of a path can be inaccurate because it ignores hard-to-formalize aspects of travel, such as turn costs. Thus, a method [19] is proposed to identify an optimal set of sub-paths that can be concatenated into a path. The path's travel cost is then the sum of the travel costs of the sub-paths. This method does not support travel cost distributions, and it assumes independence among sub-paths.

Another study explicitly models turn costs [23], and the path cost is the sum of costs of edges and the costs of turns. However, the study models turn costs based on many assumptions, e.g., maximum turning speeds, but not on real-world traffic data, and the accuracy of the modeling is unknown. In contrast, we do not explicitly model turn costs but use the joint distributions of paths to implicitly model such hard-to-formalize factors. Further, the study [23] does not consider time-dependent and uncertain costs.

Estimating Path Cost Distributions: Studies exist that model the travel cost uncertainty of a path. However, they make assumptions that do not apply in our setting.

First, some studies assume that travel cost distributions follow a standard distribution, e.g., a Gaussian distribution. However, the travel cost distribution of a road segment often follows an arbitrary distribution, as shown in recent studies [21, 22, 25] and in Figure 1(b) in Section 1 and Figure 11(a) in Section 5. We use multi-dimensional histograms to represent arbitrary distributions.

Second, some studies assume that the distributions on different edges are independent of each other [6, 13] or conditionally independent given the arrival times at different edges [22], thus mirroring the LB approach covered in Section 5. The independence assumption often does not hold (cf. Section 2), and our approach outperforms LB, as shown in Section 5. Further, with the exception of one study [22], all the above studies use generated distributions in empirical evaluations; we use large, real trajectory data.

The most advanced method, the HP [10] approach covered in Section 5, does not make the independence assumption. Rather, it assumes that the travel costs of pairs of adjacent edges are dependent, but it does not consider dependencies among multiple edges in a path. We propose a more generic model that employs joint distributions to fully capture the dependencies among all the edges in a path. In addition, we identify distributions from real-world trajectory data and support time-varying distributions, while the HP

approach employs synthetically generated distributions and does not support time-varying distributions.

Although two recent studies [14,27] employ histograms to represent travel cost distributions, they consider travel cost distributions on individual edges and assume that distributions are independent.

7. CONCLUSION AND FUTURE WORK

Accurate estimation of the travel cost distribution of a path is fundamental functionality in spatial-network related applications. We propose techniques that are able to model joint distributions that capture the travel cost dependencies among sub-paths that form longer paths, which in turn enables accurate travel cost estimation of any path using sparse historical trajectory data. Empirical studies in realistic settings offer insight into the design properties of the proposed solution and suggest that it is effective and efficient.

This study provides part of the foundation for a new and promising paradigm where travel costs are associated not only with road-network edges, but with sub-paths. The arguably most pertinent next step is to develop novel stochastic routing algorithms that fully consider the distinct characteristics of the novel paradigm.

Acknowledgments

This research was supported in part by the National Research Foundation, Prime Minister's Office, Singapore, under its Competitive Research Programme (CRP Award No. NRF CRP8-2011-08), by a grant from the Obel Family Foundation, and by the DiCyPS project. We thank Prof. Beng Chin Ooi and the anonymous reviewers for their insightful comments.

8. REFERENCES

[1] C. Guo, C. S. Jensen, and B. Yang. Towards Total Traffic Awareness. *SIGMOD Record*, 43(3):18–23, 2014.

[2] C. Guo, B. Yang, O. Andersen, C. S. Jensen, and K. Torp. EcoSky: Reducing vehicular environmental impact through eco-routing. In *ICDE*, pages 1412–1415, 2015.

[3] O. Andersen, C. S. Jensen, K. Torp, and B. Yang. EcoTour: Reducing the Environmental Footprint of Vehicles Using Eco-routes. In *MDM*, pages 338–340, 2013.

[4] J. Dai, B. Yang, C. Guo and Z. Ding. Personalized route recommendation using big trajectory data. In *ICDE*, pages 543–554, 2015.

[5] Y. M. Bishop, S. E. Fienberg, and P. W. Holland. Discrete multivariate analysis: theory and practice. *Springer*, 2007.

[6] A. Chen and Z. Ji. Path finding under uncertainty. *Journal of Advanced Transportation*, 39(1):19–37, 2005.

[7] J. Darroch and T. Speed. Additive and multiplicative models and interactions. *The Annals of Statistics*, 11(3):724–738, 1983.

[8] C. Guo, Y. Ma, B. Yang, C. S. Jensen, and M. Kaul. Ecomark: evaluating models of vehicular environmental impact. In *SIGSPATIAL*, pages 269–278, 2012.

[9] C. Guo, B. Yang, O. Andersen, C. S. Jensen, and K. Torp. Ecomark 2.0: empowering eco-routing with vehicular environmental models and actual vehicle fuel consumption data. *GeoInformatica*, 19(3):567–599, 2015.

[10] M. Hua and J. Pei. Probabilistic path queries in road networks: traffic uncertainty aware path selection. In *EDBT*, pages 347–358, 2010.

[11] T. Idé and M. Sugiyama. Trajectory regression on road networks. In *AAAI*, pages 203–208, 2011.

[12] H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. C. Sevcik, and T. Suel. Optimal histograms with quality guarantees. In *VLDB*, pages 275–286, 1998.

[13] S. Lim, C. Sommer, E. Nikolova, and D. Rus. Practical route planning under delay uncertainty: Stochastic shortest path queries. *Robotics: Science and Systems*, 8(32):249–256, 2013.

[14] Y. Ma, B. Yang, and C. S. Jensen. Enabling time-dependent uncertain eco-weights for road networks. In *GeoRich*, Article 1, 2014.

[15] F. M. Malvestuto. Approximating discrete probability distributions with decomposable models. *IEEE Transactions on SMC*, 21(5):1287–1294, 1991.

[16] P. Newson and J. Krumm. Hidden Markov map matching through noise and sparseness. In *SIGSPATIAL*, pages 336–343, 2009.

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

[18] P. Smyth. Model selection for probabilistic clustering using cross-validated likelihood. *Statistics and Computing*, 10(1):63–72, 2000.

[19] Y. Wang, Y. Zheng, and Y. Xue. Travel time estimation of a path using sparse trajectories. In *SIGKDD*, pages 25–34, 2014.

[20] M. P. Wellman, M. Ford, and K. Larson. Path planning under time-dependent uncertainty. In *UAI*, pages 532–539, 1995.

[21] B. Yang, C. Guo, and C. S. Jensen. Travel cost inference from sparse, spatio-temporally correlated time series using Markov models. *PVLDB*, 6(9):769–780, 2013.

[22] B. Yang, C. Guo, C. S. Jensen, M. Kaul, and S. Shang. Stochastic skyline route planning under time-varying uncertainty. In *ICDE*, pages 136–147, 2014.

[23] R. Geisberger, and C. Vetter. Efficient routing in road networks with turn costs. In *ESA*, pages 100–111, 2011.

[24] M. Kaul, B. Yang, and C. S. Jensen. Building Accurate 3D Spatial Networks to Enable Next Generation Intelligent Transportation Systems. In *MDM*, pages 137–146, 2013.

[25] M. Asghari, T. Emrich, U. Demiryurek, and C. Shahabi Probabilistic estimation of link travel times in dynamic road networks. In *SIGSPATIAL*, Article 47, 2015.

[26] B. Yang, M. Kaul, and C. S. Jensen. Using incomplete information for complete weight annotation of road networks. *TKDE*, 26(5):1267–1279, 2014.

[27] J. Yuan, Y. Zheng, X. Xie, and G. Sun. T-drive: Enhancing driving directions with taxi drivers' intelligence. *TKDE*, 25(1):220–232, 2013.

[28] B. Yang, C. Guo, Y. Ma, and C. S. Jensen. Toward personalized, context-aware routing. *VLDB Journal*, 24(2):297–318, 2015.

[29] J. Zheng and L. M. Ni. Time-dependent trajectory regression on road networks via multi-task learning. In *AAAI*, pages 1048–1055, 2013.

[30] J. Dai, B. Yang, C. Guo, and C. S. Jensen. Efficient and Accurate Path Cost Estimation Using Trajectory Data. *CoRR*, abs/1510.02886, 2015.

[31] Supplementary document. `http://people.cs.aau.dk/%7Ebyang/218-appendix.pdf`.