# Stochastic Skyline Route Planning Under Time-Varying Uncertainty

Bin Yang[1], Chenjuan Guo[1], Christian S. Jensen[2], Manohar Kaul[1], Shuo Shang[3]

[1] Department of Computer Science, Aarhus University, Denmark
{byang, cguo, mkaul}@cs.au.dk

[2] Department of Computer Science, Aalborg University, Denmark
csj@cs.aau.dk

[3] Department of Software Engineering, China University of Petroleum-Beijing, China
sshang@cs.aau.dk

***Abstract*—Different uses of a road network call for the consideration of different travel costs: in route planning, travel time and distance are typically considered, and green house gas (GHG) emissions are increasingly being considered. Further, travel costs such as travel time and GHG emissions are time-dependent and uncertain.**

**To support such uses, we propose techniques that enable the construction of a multi-cost, time-dependent, uncertain graph (MTUG) model of a road network based on GPS data from vehicles that traversed the road network. Based on the MTUG, we define stochastic skyline routes that consider multiple costs and time-dependent uncertainty, and we propose efficient algorithms to retrieve stochastic skyline routes for a given source-destination pair and a start time. Empirical studies with three road networks in Denmark and a substantial GPS data set offer insight into the design properties of the MTUG and the efficiency of the stochastic skyline routing algorithms.**

## I. Introduction

Reduction in green house gas (GHG) emissions from transportation is crucial in combating global warming. For example, in the EU, emissions from transportation account for nearly a quarter of all GHG emissions, and the EU has committed to reduce emissions to 20% below the 1990 levels by 2020.

To achieve politically agreed-upon reductions, and due to an increasing public awareness of environmental protection, fleet owners and individual drivers increasingly perform eco-routing [1], taking into account GHG emissions, in addition to travel time and distance, when planning routes. Eco-routing calls for solutions that contend with three challenging characteristics.

**Multiple Costs:** Multiple travel costs, e.g., travel times, distances, and GHG emissions, need to be considered. A recent study [2] suggests that neither the shortest nor the fastest routes generally have the lowest GHG emissions. GHG emissions are highly related to instantaneous velocities and accelerations [1] and are only loosely correlated with travel times and distances. Thus, eco-routing algorithms must be able to return routes that consider multiple, loosely-correlated costs.

**Time Dependence:** Travel costs such as travel times and GHG emissions are time-dependent. For example, traversing a road during peak hours may take much longer than that during off-peak hours. Further, different roads have different traffic behaviors, with some roads having clear peak and off-peak hours and some roads exhibiting nearly constant travel times. Thus, to support eco-routing, time dependence must be modeled appropriately and must be considered by routing algorithms.

**Uncertainty:** Some travel costs are inherently uncertain. For example, given the same road, aggressive driving may generate more GHG emissions (but shorter travel time) than does moderate driving. The resulting uncertainty may vary across time. For instance, during peak hours, the uncertainty of travel costs may be low because congestion forces drivers to drive similarly, while during off-peak hours, drivers have more freedom to drive fast or slow, thus increasing the travel cost uncertainty. Effective routing algorithms must take into account *time-varying uncertainty*.

We present techniques that enable the construction of a multi-cost, time-dependent and uncertain graph (MTUG) model of a road network that is capable of capturing multiple time-varying and uncertain travel costs. Specifically, each cost on a road segment is modeled as a vector of $(interval, random\ variable)$ pairs. The proposed techniques build an MTUG from a massive collection of GPS data collected from vehicles traveling in the road network.

Based on the MTUG, we define the cost of a route, a *dominance* relationship between routes based on their costs, and a natural notion of a *stochastic skyline route* for a given source-destination pair and a trip starting time. A stochastic skyline route is a pareto-optimal route with the property that no other route is better when considering all travel costs of interest. Finally, we propose efficient methods to retrieve stochastic skyline routes.

While existing routing services and navigation devices offer alternative routes, they generally return routes based on a single criterion (e.g., based on distance) or routes satisfying some road type constraints (e.g., avoiding toll roads). None of these provide a set of routes that takes into account multiple travel costs, and they consider neither GHG emissions nor the combination of time-dependence and uncertainty. We extend existing routing functionality to support stochastic skyline routes that consider multiple, time-varying and uncertain travel costs.

Stochastic skyline routes are of interest to both individual drivers and entities that control fleets of vehicles. For example, FlexDanmark[1], a large public fleet coordinator in Denmark,

[1] https://www.flexdanmark.dk/

978-1-4799-2555-1/14/$31.00 © 2014 IEEE

is interested in using the most eco-friendly routes while considering travel times and distances.

To the best of our knowledge, this paper is the first to propose a general framework that is able to effectively solve stochastic skyline route planning in a transportation network with multiple, time-varying and uncertain travel costs, a real problem for which there previously was no practical solution. Specifically, the paper makes four contributions. First, it proposes the MTUG model of a road network with multiple, time-varying, and uncertain travel costs. Second, it proposes techniques that enable instantiation of the MTUG based on a collection of GPS data that reflects real traffic behavior. Third, it presents stochastic skyline route queries on the MTUG along with efficient algorithms. Fourth, it reports on comprehensive experiments that involve three road networks in Denmark and a substantial GPS data set. These elicit design properties of the MTUG and characterize the efficiency of the stochastic skyline routing algorithms.

The remainder of the paper is organized as follows. Section II covers related work. Section III defines the MTUG, and Section V covers how to instantiate the MTUG. Section VI describe the routing algorithms. Section VII reports on the empirical evaluation. Finally, conclusions and research directions are offered in Section VIII.

## II. Related Work

Existing route planning algorithms can be classified with respect to the types of edge weights they support—see Table I. This classification considers weight type (i.e., deterministic or uncertain values), temporal variation (i.e., time-homogeneous or time-varying values), and cardinality of costs (i.e., single cost or multiple costs).

TABLE I. Classification of Routing Algorithms

| | *Deterministic* | *Uncertain* |
|---|---|---|
| *Time-Homogeneous* | **A Real Number**<br>Single Cost: [3], [4]<br>Multiple Costs: [8] | **A Random Variable**<br>Single Cost: [5]–[7]<br>Multiple Costs: [9] |
| *Time-Varying* | **A Piece-Wise Linear Function**<br>Single Cost: [10]–[13]<br>Multiple Costs: [17] | **(Interval, Random Variable) Pairs**<br>Single Cost: [14]–[16]<br>Multiple Costs: [18], [19] |

We consider the most general case: routing on a graph with multiple time-varying and uncertain costs. We proceed to compare with the two existing studies [18], [19] that also consider this case. First, none of these studies consider GHG emissions. Second, both studies use synthetic data to generate edge weights, which may not reflect real-world traffic behavior. For example, they use a single Gaussian distribution to model the travel time distribution for each edge [18], [19]. In contrast, we find that it is frequently impossible to fit the distributions of travel times and GHG emissions to a single Gaussian distribution, and we propose techniques that are able to learn appropriate time-dependent, uncertain weights based on real GPS data, thus reflecting real traffic behavior. Third, the existing studies rely on the assumption that edge weights follow Gaussian distributions, with one study [18] assuming that each edge has only one peak period and that travel times outside the peak period are constant. We do not make such assumptions, and we are able to support arbitrary distributions. Fourth, the existing studies only encompass small-scale experiments, e.g., using road networks with less than 300 vertices. In contrast, we report empirical studies on large road networks, one with more than 667K vertices.

The route skyline query [8] is also relevant to our work. Although it also considers multiple costs, the costs are time-homogeneous and deterministic. The proposed pruning strategies do not apply in our setting because time-varying uncertain weights can yield non-FIFO graphs.

We employ stochastic dominance to measure the dominance between two random variables. Although a stochastic skyline [20] also uses stochastic dominance, it focuses on vector spaces, not on the road network setting. Thus, existing techniques cannot be used directly in our setting.

## III. Road Network Modeling

We cover basic concepts of road networks and trajectories, a graphical model that is able to model time-dependent uncertainty, and the definition of a *multi-cost, time-varying, uncertain graph.*

### A. Road Networks, Trajectories, and Travel Costs

*Definition 1:* A **road network** is a directed graph $M = (V, E, F)$, 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 *starting vertex* $v_i$ to its *ending vertex* $v_j$. Function $F : V \cup E \rightarrow \mathit{Geometries}$ records geometrical information of the road network $M$. In particular, it maps a vertex and an edge to the point location of the corresponding road intersection and to a polyline representing the corresponding road segment. ■

Fig. 1 shows a road network with 4 vertices and 5 edges.

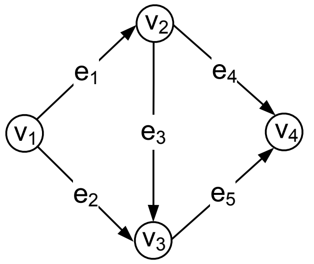


Fig. 1. A Road Network Example

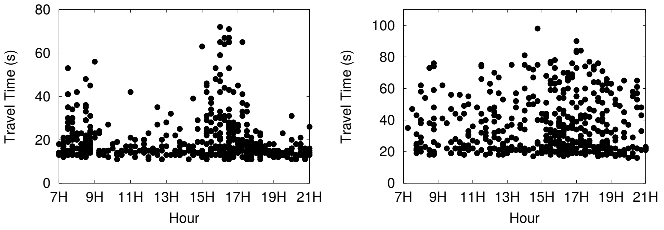
A **trajectory** $\mathcal{T} = \langle p_1, p_2, \ldots, p_A \rangle$ is a sequence of GPS records pertaining to a trip, where each record $p_i$ specifies a $(location, time)$ pair of a vehicle, where $p_i.time < p_j.time$ if $1 \leqslant i < j \leqslant A$. Map matching [21] is used to map a GPS record to a specific location on an edge in the underlying road network.

Map matching transforms a trajectory $\mathcal{T}$ into a sequence of **cost records** $\langle l_1, l_2, \ldots, l_B \rangle$. A record $l_j$ is of the form $(e, t, \mathbf{C})$, where $e$ is an edge traversed by trajectory $\mathcal{T}$, $t$ is the time when the traversal of edge $e$ starts, and cost vector $\mathbf{C} = \langle c_1, c_2, \ldots, c_N \rangle$ contains $N$ distinct travel costs associated with the traversal of edge $e$. For instance, to support eco-routing, $N = 3$ cost types, including travel distance, travel time, and GHG emissions, need to be considered.

The travel distances (i.e., lengths) of edges can be easily derived from the geometrical information recorded in function

$M.F$. Some travel costs, notably travel times, can be obtained directly from GPS records; while other travel costs, e.g., GHG emissions, can be derived from GPS records using vehicular environmental impact models [1].

Some costs, notably travel distance, are deterministic and time-homogeneous. Other costs, including travel time and GHG emissions, are uncertain, and the uncertainty can vary considerably across time. For example, in Fig. 2, we plot the travel times of trips w.r.t. the starting times of the trips. Fig. 2(a) shows data for an edge with clear morning and afternoon peaks, around 8:00 and 16:00, respectively; and the travel-time distribution differs across different intervals. In contrast, the edge covered in Fig. 2(b) has no clear peak periods, while its distribution of travel times also varies across time.



(a) Two peak periods (b) No clear peak periods

Fig. 2. Time-Dependent Uncertain Travel Times

### B. Modeling Time-Dependent Uncertainty

To model time-dependent uncertainty, we model the $n$-th travel cost of edge $e_i$ as a random variable $C_{e_i}^{(n)}$ that is dependent on a temporal context random variable $tc$ that in turn describes the distribution of possible starting time points of traversing the edge. The relationships among the cost random variables and the temporal context random variables are captured by the graphical model [22] shown in Fig. 3, where circles indicate random variables and lines with arrows indicate dependencies between random variables.

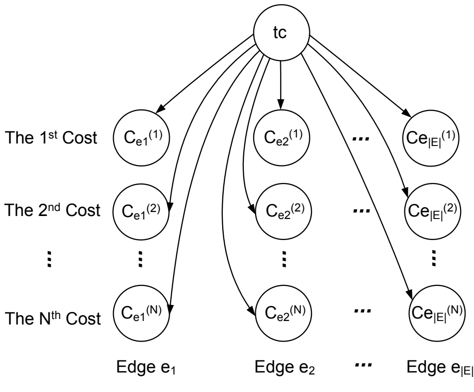


Fig. 3. Temporal Context and Travel Cost Random Variables

When modeling time-homogeneous travel costs, e.g., travel distance, the temporal context random variable is set to take only one constant value. Thus, time-homogeneous travel costs are unaffected by the temporal context. When modeling time-varying travel costs, e.g., travel times and GHG emissions, the temporal context random variable is set to take multiple values that indicate different starting times. The travel cost random variables are dependent on the temporal context random variable, meaning that the distribution of the travel costs on an edge generally vary across different starting times.

This model is fundamentally different from the independency models that are employed extensively in studies of routing with uncertain edge weights [6], [7], [9], [14], [15]. These models assume that the travel costs of different edges are independent. For example, the travel time of an edge is unrelated to the travel time of an adjacent edge.

In our model, the travel costs of different edges are dependent through the temporal context random variables. For example, consider a trip along edges $e_1$ and $e_4$ in Fig. 1. The travel time random variable of edge $e_1$ decides the probability when the trip starts on edge $e_4$ (captured by the temporal context random variable $tc$), thus affecting the travel time and GHG emissions random variables of edge $e_4$.

### C. Multi-Cost, Time-Dependent, Uncertain Graphs

To achieve the modeling shown in Fig. 3, Definition 2 proposes a multi-cost, time-dependent, uncertain graph.

*Definition 2:* A **Multi-cost, Time-dependent, Uncertain Graph (MTUG)** $G = (V, E, \mathbf{MM}, \mathbf{W})$ is a directed, weighted graph. $V$ and $E$ are the vertex and edge sets as stated in Definition 1.

First, $\mathbf{MM} = \langle MM^{(1)}, MM^{(2)}, \ldots, MM^{(N)} \rangle$ is a vector of functions, where function $MM^{(n)} : E \rightarrow \mathbb{R}^+ \times \mathbb{R}^+$ maintains the minimum and maximum values of the $n$-th cost type for all edges. For instance, $MM^{(2)}(e_i) = (12, 15)$ indicates that the minimum and maximum travel times of edge $e_i$ are 12 and 15 minutes, respectively.

Second, $\mathbf{W} = \langle W^{(1)}, W^{(2)}, \ldots, W^{(N)} \rangle$ is a vector of weight functions. Here, function $W^{(n)} : E \rightarrow 2^{T \times RV}$, where $T$ indicates the temporal domain of a day and $RV$ is a set of random variables, maintains the time-dependent distributions of the $n$-th cost type for all edges. ■

For each edge $e_k \in E$, $W^{(n)}$ maintains a set of tuples of the form $(I, X)$. Such a tuple indicates that the distribution of the $n$-th cost of traversing edge $e_k$ during interval $I \subseteq T$ is described by a random variable $X \in RV$. According to the model proposed in Section III-B, the probability function of random variable $X$ is $\mathbf{P}(C_{e_k}^{(n)} | tc \text{ is in interval } I)$.

To ease the presentation, we introduce important notation. Given an edge $e_k$, we let $S_{e_k}^{(n)}$ be the number of the intervals for the $n$-th cost; we let $I_{e_k,j}^{(n)}$ be the $j$-th interval for the $n$-th cost, where $1 \leqslant j \leqslant S_{e_k}^{(n)}$; and we let $X_{e_k,j}^{(n)}$ be a random variable that captures the distribution of the $n$-th cost values for edge $e_k$ during interval $I_{e_k,j}^{(n)}$.

For instance, if the travel time (when $n = 2$) on edge $e_k$ has a single peak interval $[8:00, 9:30)$ then $S_{e_k}^{(2)} = 3$ intervals need to be defined to cover a day: $I_{e_k,1}^{(2)} = [0:00, 8:00)$, $I_{e_k,2}^{(2)} = [8:00, 9:30)$, and $I_{e_k,3}^{(2)} = [9:30, 24:00)$. For each $j$ (where $1 \leqslant j \leqslant S_{e_k}^{(2)} = 3$), a random variable $X_{e_k,j}^{(2)}$ captures the distribution of the travel times observed during interval $I_{e_k,j}^{(2)}$.

Note that for different types of costs (i.e., for different $n$), the number of intervals $S_{e_k}^{(n)}$ and the intervals themselves may be different. For example, the travel time and GHG emissions on the same edge may have different peak and off-peak intervals.

To model time-homogeneous and deterministic cost types, e.g., travel distance (when $n = 1$), the cardinality of the intervals is set to 1. Thus, we have $S_{e_k}^{(1)} = 1$ for each $e_k \in E$. For example, assume that the length of edge $e_k$ is 2.5 km. Then $W^{(1)}(e_k) = \{(I_{e_k,1}^{(1)}, X_{e_k,1}^{(1)})\}$, where $I_{e_k,1}^{(1)} = [0:00, 24:00)$ indicates a whole day, and random variable $X_{e_k,1}^{(1)} = (2.5, 1.0)$ indicates that the cost being 2.5 km with probability 1.0.

In the following discussions, we mainly use eco-routing to illustrate route planning on an MTUG. Thus, we focus on the case $N = 3$, where $n = 1, 2, 3$ indicate travel distance, travel time, and GHG emissions, respectively. However, the techniques proposed in the paper also apply to cases with arbitrary $N$. Table II shows a set of example weights during [8:00, 10:00) for the MTUG representing the road network shown in Fig. 1, where the units of the costs are km, minutes, and ml. Section V describes how to instantiate an MTUG from a collection of GPS records.

## IV. STOCHASTIC SKYLINE ROUTES ON MTUGs

### A. Routes and Route Costs

*Definition 3:* A **route** $\mathcal{R} = \langle r_1, r_2, \ldots, r_p\rangle$, where $p \geqslant 1$, is a sequence of edges, where $r_i \in E$ and $r_i \neq r_j$ if $i \neq j$, and consecutive edges must share a vertex—the ending vertex of edge $r_i$ is the same as the starting vertex of edge $r_{i+1}$, where $1 \leqslant i < p$. The first $a$ edges in route $\mathcal{R}$ constitute a **pre-route** of route $\mathcal{R}$, denoted as $\mathcal{R}^{(a)} = \langle r_1, r_2, \ldots, r_a\rangle$, where $1 \leqslant a \leqslant p$. The cardinally of route $\mathcal{R}$, denoted as $|\mathcal{R}|$, is the number of edges in the route. ■

*Example 1:* Three different routes connect source $v_1$ and destination $v_4$ in the road network in Fig. 1: $\mathcal{R}_1 = \langle e_1, e_4\rangle$ (where $r_1 = e_1$, $r_2 = e_4$), $\mathcal{R}_2 = \langle e_1, e_3, e_5\rangle$, $\mathcal{R}_3 = \langle e_2, e_5\rangle$. Route $\mathcal{R}_1$ has 2 pre-routes: $\mathcal{R}_1^{(1)} = \langle e_1\rangle$ and $\mathcal{R}_1^{(2)} = \mathcal{R}_1 = \langle e_1, e_4\rangle$ with cardinalities $|\mathcal{R}_1^{(1)}| = 1$ and $|\mathcal{R}_1^{(2)}| = |\mathcal{R}_1| = 2$.

*Definition 4:* When starting travel on route $\mathcal{R}$ at time $t$, the **route cost** $RC(\mathcal{R}, t) = \langle DI, TT, GE\rangle$ is a vector of random variables, where random variable $DI$ ($TT$, $GE$) represents the distribution of route $\mathcal{R}$'s travel distance (travel time, GHG emissions). ■

We consider one cost type at a time. Intuitively, the travel cost random variable of a route is the sum of the travel cost random variables of all the edges in the route. Thus, once the travel cost random variable of each edge in the route is determined, the travel cost random variable of the route can also be determined.

Recall that the travel cost random variable of an edge is dependent on the edge's temporal context variable. Thus, the key to determining the travel cost random variable of an edge is to determine its corresponding temporal context. We distinguish between two cases. (i) For the first edge $r_1$ of a route, the temporal context is the fixed start time $t$. (ii) The remaining edges are more challenging. To determine the temporal context of the $k$-th edge ($k > 1$), we must consider all the travel time random variables of the proceeding $k-1$ edges in the route, as they influence the possible starting times of the edge. The detail of how to determine the temporal context are covered later in this section.

According to our graphical model (Fig. 3), any two travel cost random variables are conditionally independent if the temporal context is given. When determining the travel cost random variable of an edge in a route, the temporal context, i.e., the possible starting times on the edge, has been determined based on all the previous edges' travel time random variables. Thus, the obtained travel cost random variables of edges are independent of each other.

Let $X$ and $Y$ be two independent random variables with probability functions $X(z)$ and $Y(z)$, and let random variable $Q = X + Y$ indicate the sum of the two random variables. The probability function of $Q$ is the *convolution* of $X(z)$ and $Y(z)$ [23].

$$Q(z) = X \bigodot Y(z) = \int_{-\infty}^{+\infty} f_X(\tau) \cdot f_Y(z - \tau) d\tau,$$

where $\bigodot$ denotes the convolution operator. The intuition of convolution is that the probability of $Q$ having value $z$ is the sum of the probability of $X$ having value $\tau$ multiplied with the probability of $Y$ having value $z - \tau$ "summed" over all possible $\tau$.

Thus, the probability function of the travel cost random variable of route $\mathcal{R}$ is determined by the convolution of the probability functions of the random variables of the edges in the route. We next consider how to determine the probability functions for the travel distance, travel time, and GHG emissions random variables of a route.

*1) Determining DI:* As travel distances are time-homogeneous, temporal contexts are irrelevant and thus ignored. The distance random variable $RC(\mathcal{R}, t).DI$ is a deterministic value, i.e., the sum of edge lengths in route $\mathcal{R}$.

$$RC(\mathcal{R}, t).DI = \Big(\sum_{i=1}^{|\mathcal{R}|} \mathit{length}(r_i), 1.0\Big)$$

*2) Determining TT:* As travel times are time-varying, we determine the temporal context variable of each edge in route $\mathcal{R}$ as a precursor to determining the travel time random variable of each edge. Let $RC(r_i, t).TT$ denote the travel time random variable of edge $r_i$ in route $\mathcal{R}$:

$$RC(r_i, t).TT = \mathbf{P}(tc) \cdot \mathbf{P}(C_{r_i}^{(2)} | tc) = \sum_{j=1}^{S_{r_i}^{(2)}} x_{r_i,j}^{(2)} \cdot X_{r_i,j}^{(2)}$$

Here, $x_{r_i,j}^{(2)}$ is the probability that the temporal context $tc$ falls in edge $r_i$'s travel time interval $I_{r_i,j}^{(2)}$; and $X_{r_i,j}^{(2)}$ is edge $r_i$'s travel time random variable in $I_{r_i,j}^{(2)}$, as given in Definition 2.

Since the starting time on the first edge $r_1$ is time $t$, the temporal context is fixed and must be in a single travel time interval. Thus, we have the following for each $1 \leqslant j \leqslant S_{r_1}^{(2)}$.

$$x_{r_1,j}^{(2)} = \begin{cases} 1 & \text{if } t \text{ is in interval } I_{r_1,j}^{(2)} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

For edge $r_k$ ($k > 1$), the temporal context may be in more than one travel time interval. The possible starting time of edge

TABLE II. EXAMPLES FOR MULTI-COST, TIME-DEPENDENT, UNCERTAIN WEIGHTS DURING PERIOD $[8:00, 10:00)$

| Edges | $W^{(1)}$ | $W^{(2)}$ | $MM^{(2)}$ | $W^{(3)}$ | $MM^{(3)}$ |
|---|---|---|---|---|---|
| $e_1 =$ $(v_1, v_2)$ | $\{(I^{(1)}_{e_1,1} = [8:00, 10:00),$ $X^{(1)}_{e_1,1} = (4, 1.0))\}$ | $\{(I^{(2)}_{e_1,1} = [8:00, 8:30), X^{(2)}_{e_1,1}),$ $(I^{(2)}_{e_1,2} = [8:30, 10:00), X^{(2)}_{e_1,2})\}$ | (2, 17) | $\{(I^{(3)}_{e_1,1} = [8:00, 9:15), X^{(3)}_{e_1,1}),$ $(I^{(3)}_{e_1,2} = [9:15, 10:00), X^{(3)}_{e_1,2})\}$ | (150, 600) |
| $e_3 =$ $(v_2, v_3)$ | $\{(I^{(1)}_{e_3,1} = [8:00, 10:00),$ $X^{(1)}_{e_3,1} = (1.3, 1.0))\}$ | $\{(I^{(2)}_{e_3,1} = [8:00, 9:15), X^{(2)}_{e_3,1}),$ $(I^{(2)}_{e_3,2} = [9:15, 10:00), X^{(2)}_{e_3,2})\}$ | (1, 5) | $\{(I^{(3)}_{e_3,1} = [8:00, 10:00), X^{(3)}_{e_3,1})\}$ | (50, 230) |
| $e_5 =$ $(v_3, v_4)$ | $\{(I^{(1)}_{e_5,1} = [8:00, 10:00),$ $X^{(1)}_{e_5,1} = (5.5, 1.0)\}$ | $\{(I^{(2)}_{e_5,1} = [8:00, 9:00), X^{(2)}_{e_5,1}),$ $(I^{(2)}_{e_5,2} = [9:00, 9:30), X^{(2)}_{e_5,2}),$ $(I^{(2)}_{e_5,3} = [9:30, 10:00), X^{(2)}_{e_5,3})\}$ | (3, 18) | $\{(I^{(3)}_{e_5,1} = [8:00, 9:25), X^{(3)}_{e_5,1}),$ $(I^{(3)}_{e_5,2} = [9:25, 10:00), X^{(3)}_{e_5,2})\}$ | (480, 600) |

$r_k$ depends on the travel time of the pre-route $\mathcal{R}^{(k-1)}$, i.e., the $k-1$ edges before edge $r_k$. Thus, for each $1 \leqslant j \leqslant S^{(2)}_{r_k}$:

$$x^{(2)}_{r_k,j} = \int_{I^{(2)}_{r_k,j}} \left(RC(\mathcal{R}^{(k-1)}, t).TT(z) + t\right)\, dz \tag{2}$$

Based on the above, the travel time random variable $RC(\mathcal{R}, t).TT$ is the convolution of the travel time random variables of the edges in $\mathcal{R}$, as defined in Equation 3.

$$RC(\mathcal{R}, t).TT(z) = \bigodot_{i=1}^{|\mathcal{R}|} RC(r_i, t).TT(z). \tag{3}$$

*Example 2:* Consider a traversal of $\mathcal{R}_2 = \langle e_1, e_3, e_5\rangle$ at $t =$ 9:05. Since $t$ falls in edge $e_1$'s travel time interval $I^{(2)}_{e_1,2} =$ [8:30, 10:00) (see Table II), we have $RC(e_1, \text{9:05}).TT = RC(\mathcal{R}^{(1)}_2, \text{9:05}).TT = X^{(2)}_{e_1,2}$. According to the $MM^{(2)}$ column in Table II, the minimum and maximum travel times of traversing edge $e_1$ are 2 and 17 minutes, respectively, thus placing the possible starting times on the second edge $e_3$ in interval [9:07, 9:22], which overlaps with both of $e_3$'s travel time intervals $I^{(2)}_{e_3,1}$=[8:00, 9:15) and $I^{(2)}_{e_3,2}$=[9:15, 10:00). The probabilities that the starting time on edge $e_3$ falls in intervals $I^{(2)}_{e_3,1}$ and $I^{(2)}_{e_3,2}$ are computed as follows.

$$\begin{aligned} x^{(2)}_{e_3,j} &= \int_{I^{(2)}_{e_3,j}} \left(RC(\mathcal{R}^{(1)}_2, \text{9:05}).TT(z) + \text{9:05}\right)\; dz \\ &= \int_{I^{(2)}_{e_3,j}} \left(X^{(2)}_{e_1,2}(z) + \text{9:05}\right)\; dz, \quad \text{where } j = 1 \text{ or } 2; \end{aligned}$$

Fig. 4 illustrates the probability computation. Here, $x^{(2)}_{e_3,1}$ and $x^{(2)}_{e_3,2}$ are the areas of the regions between the curve representing random variable $X^{(2)}_{e_1,2}$ and the horizontal axis in intervals [2,10] and [10, 17], respectively.

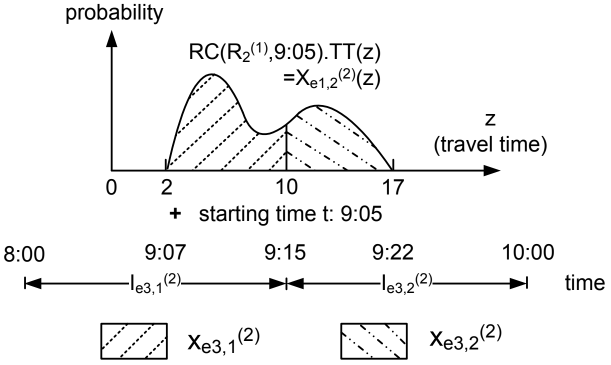


Fig. 4. Determining Probabilities on Different Travel Time Intervals

Next, we have

$$RC(\mathcal{R}^{(2)}_2, \text{9:05}).TT = X^{(2)}_{e_1,2} \odot \sum_{j=1}^{S^{(2)}_{e_3}=2} x^{(2)}_{e_3,j} \cdot X^{(2)}_{e_3,j}, \text{ and}$$

$$RC(\mathcal{R}_2, \text{9:05}).TT = X^{(2)}_{e_1,2} \odot \sum_{j=1}^{S^{(2)}_{e_3}=2} x^{(2)}_{e_3,j} \cdot X^{(2)}_{e_3,j} \odot \sum_{j=1}^{S^{(2)}_{e_5}=3} x^{(2)}_{e_5,j} \cdot X^{(2)}_{e_5,j}$$

*3) Determining GE:* As GHG emissions are also time-varying, the procedure for determining $GE$ is similar to that of determining $TT$. The GHG emission random variable of route $\mathcal{R}$ is defined as follows.

$$\begin{aligned} RC(\mathcal{R}, t).GE(z) &= \bigodot_{i=1}^{|\mathcal{R}|} RC(r_i, t).GE(z) \\ &= \bigodot_{i=1}^{|\mathcal{R}|} \left(\sum_{j=1}^{S^{(3)}_{r_i}} x^{(3)}_{r_i,j} \cdot X^{(3)}_{r_i,j}(z)\right), \end{aligned}$$

where the $x^{(3)}_{r_i,j}$ are defined as in Equations 1 and 2. The only difference is to use GHG emissions intervals $I^{(3)}_{r_1,j}$ and $I^{(3)}_{r_k,j}$ instead of travel time intervals $I^{(2)}_{r_1,j}$ and $I^{(2)}_{r_k,j}$.

*Example 3:* Consider the traversal of route $\mathcal{R}_2$ from Example 2. The GHG emission random variable of the route is defined as follows.

$$RC(\mathcal{R}_2, \text{9:05}).GE = X^{(3)}_{e_1,1} \odot \sum_{j=1}^{S^{(3)}_{e_3}=1} x^{(3)}_{e_3,j} \cdot X^{(3)}_{e_3,j} \odot \sum_{j=1}^{S^{(3)}_{e_5}=2} x^{(3)}_{e_5,j} \cdot X^{(3)}_{e_5,j}$$

### B. Stochastic Skyline Routes

Given a source, a destination, and a travel start time, it is of interest to identify a set of stochastic skyline routes from the source to the destination that are no worse than any other route according to the costs of interest. To define these stochastic skyline routes, we use a concept called stochastic dominance [24].

*Definition 5:* Let $X$ and $Y$ be random variables with cumulative distribution functions $F_X(z) = \mathbf{P}(X \leqslant z)$ and $F_Y(z) = \mathbf{P}(Y \leqslant z)$. If $F_X(z) \geqslant F_Y(z)$ for all $z \in \mathbb{R}^+$, $X$ **stochastically dominates** $Y$, denoted by $X \succ Y$. ■

*Definition 6:* Given two routes $\mathcal{R}_i$ and $\mathcal{R}_j$ and a start time $t$, $\mathcal{R}_i$ **dominates** $\mathcal{R}_j$, denoted as $\mathcal{R}_i \succ_t \mathcal{R}_j$, if for every $X \in \{DI, TT, GE\}$, $RC(\mathcal{R}_i, t).X$ is not dominated by $RC(\mathcal{R}_j, t).X$, and $RC(\mathcal{R}_i, t).X \succ RC(\mathcal{R}_j, t).X$ for at least one $X \in \{DI, TT, GE\}$. ■

*Definition 7:* Given a source $v_s$, a destination $v_d$, and a start time $t$, the set of **stochastic skyline routes** is defined as follows:

$$SKR(v_s, v_d, t) = \{\mathcal{R}_i \in RR \mid \neg\exists \mathcal{R}_j \in RR\ (\mathcal{R}_j \succ_t \mathcal{R}_i)\},$$

where $RR$ is the set containing all routes from $v_s$ to $v_d$. Thus, the stochastic skyline routes are the routes that are not dominated by any other routes, and they comply with pareto-optimality. Given $v_s$, $v_d$, and $t$, a **stochastic skyline route query** returns the set of stochastic skyline routes $SKR(v_s, v_d, t)$. ■

### C. Framework Overview

As shown in Fig. 5, the system consists of an off-line component and an on-line component. In the former, a preprocessing module takes as input a collection of trajectories and a road network (e.g., obtained from OpenStreetMap[2]), and it feeds a collection of cost records into the MTUG generation module that assigns time-varying, uncertain weights to the corresponding road network, thus producing an MTUG. In the on-line phase, a user provides a source, a destination, and a travel start time, in response to which a routing module returns the stochastic skyline routes.

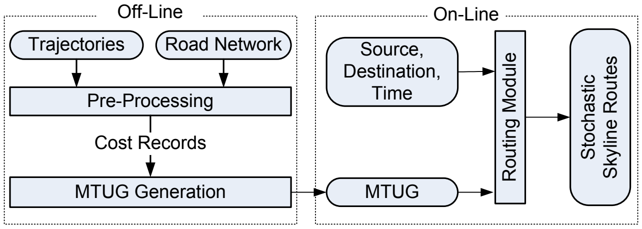


Fig. 5. Framework Overview

## V. MTUG Generation

The key to generating an MTUG is to assign a time-dependent uncertain weight to each edge in a road network based on a collection of trajectories from the road network.

### A. General Procedure

Recall that map matching transforms trajectories into cost records, as discussed in Section III-A. When assigning weights to edge $e_k$, we consider the cost records $L_{e_k} = \{l_i | l_i.e = e_k\}$. The minimum and maximum travel costs can be determined easily based on $L_{e_k}$. Thus, function vector MM in the MTUG is instantiated.

Next, we partition a day into $D = \lceil \frac{24 \cdot 60}{\alpha} \rceil$ intervals, where parameter $\alpha$ specifies the finest-granularity interval of interest in minutes. We may use $\alpha = 15$ minutes, which is typically the finest time granularity used in the transportation area [25], thus obtaining 96 intervals per day. Given an interval $I_j$, a subset of $L_{e_k}$ has times that fall into the interval, i.e., $L_{e_k}^{I_j} = \{l_i \in L_{e_k} \mid l_i.t \in I_j\}$. We then use the cost values in $L_{e_k}^{I_j}$ to learn a random variable (represented by its probability function) for each cost type in the interval. In particular, we use a continuous, parametric representation to denote a random variable, which is covered in Section V-B.

For each cost type, if the probability functions of the random variables in two consecutive intervals are similar (i.e., within a distance threshold $thDis$), the two intervals are combined into a longer interval. Using the cost records in the longer interval, a new random variable is learned. Any distance function that is able to quantify the distance between two distributions can be applied, e.g., the Kolmogorov-Smirnov test [26] or Kullback-Leibler divergence [22].

This process is applied iteratively until no random variables from consecutive intervals are similar enough to be combined. In each iteration, the two random variables with the highest similarity are combined. After combining similar random variables in consecutive intervals, the remaining random variables and their corresponding intervals are used as the time-dependent, uncertain weights. Fig. 6 exemplifies the process of combining distributions. Algorithm 1 describes the process of generating an MTUG.

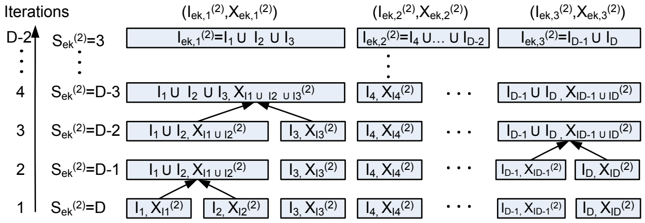

Iterations
$(I_{ek,1}^{(2)}, X_{ek,1}^{(2)})$
$(I_{ek,2}^{(2)}, X_{ek,2}^{(2)})$
$(I_{ek,3}^{(2)}, X_{ek,3}^{(2)})$
D-2
$S_{ek}^{(2)}=3$
$I_{ek,1}^{(2)}=I_1 \cup I_2 \cup I_3$
$I_{ek,2}^{(2)}=I_4 \cup \ldots \cup I_{D-2}$
$I_{ek,3}^{(2)}=I_{D-1} \cup I_D$
4
$S_{ek}^{(2)}=D-3$
$I_1 \cup I_2 \cup I_3, X_{I1 \cup I2 \cup I3}^{(2)}$
$I_4, X_{I4}^{(2)}$
$I_{D-1} \cup I_D, X_{ID-1 \cup ID}^{(2)}$
3
$S_{ek}^{(2)}=D-2$
$I_1 \cup I_2, X_{I1 \cup I2}^{(2)}$
$I_3, X_{I3}^{(2)}$
$I_4, X_{I4}^{(2)}$
$I_{D-1} \cup I_D, X_{ID-1 \cup ID}^{(2)}$
2
$S_{ek}^{(2)}=D-1$
$I_1 \cup I_2, X_{I1 \cup I2}^{(2)}$
$I_3, X_{I3}^{(2)}$
$I_4, X_{I4}^{(2)}$
$I_{D-1}, X_{ID-1}^{(2)}$
$I_D, X_{ID}^{(2)}$
1
$S_{ek}^{(2)}=D$
$I_1, X_{I1}^{(2)}$
$I_2, X_{I2}^{(2)}$
$I_3, X_{I3}^{(2)}$
$I_4, X_{I4}^{(2)}$
$I_{D-1}, X_{ID-1}^{(2)}$
$I_D, X_{ID}^{(2)}$

Fig. 6. An Example of Combining Travel Time Random Variable on $e_k$

**Algorithm 1: MTUG Generation**

```
Input : Cost Records: $\{l_i\}$; double: $\alpha$;
1 int $D = \lceil \frac{24 \cdot 60}{\alpha} \rceil$;
2 Split a day into $D$ intervals: $I = \{I_1, I_2, \ldots, I_D\}$;
3 for each edge $e_k \in G.E$ do
4     for each interval $I_j \in I$ do
5         for each cost type $n \in \{1 \ldots N\}$ do
6             $CostMultiSet \leftarrow \bigcup_{l_i \in L_{e_k}^{I_j}} l_i.c_n$;
7             Learn a random variable $X_{I_j}^{(n)}$, i.e., the
              probability function of the $n$-th cost in
              interval $I_j$, based on $CostMultiSet$;
8     for each cost type $n \in [1 \ldots N]$ do
9         Combine random variable $X_{I_1}^{(n)}, \ldots, X_{I_A}^{(n)}$ as
          shown in Fig. 6;
```

### B. Representing A Random Variable

A random variable is represented by its probability function. A naive way to represent a random variable is to treat it as a discrete random variable and then represent its probability mass function by a histogram. For instance, assuming that we have $CostMultiSet = \{\{15, 7, 15, 12, 12, 12, 15, 15, 7, 15\}\}$ (in line 6 in Algorithm 1), the histogram representation of the random variable learned based on $CostMultiSet$ is $H = \langle(7, 0.2), (12, 0.3), (15, 0.5)\rangle$.

Although this representation is simple, it has two weaknesses. First, it may miss some possible cost values. For example, according to $H$, the probability of cost 10 is zero because the sampled data set $CostMultiSet$ does not contain 10. However, in reality, it is likely to have cost 10 or any other value in range $[7, 15]$. Second, the convolution of histogram-based probability functions becomes inefficient when histograms contain large numbers of $(cost, probability)$ pairs, which occurs when computing travel costs for long routes.

To overcome these drawbacks, we use a **continuous, parametric** approach to represent a random variable. Since the approach is continuous, it is able to return a probability for a value that is not observed in $CostMultiSet$. Further, since the approach is parametric, a probability function can be described by a small number of parameters, allowing the convolution operation to be performed efficiently using these parameters.

[2] http://www.openstreetmap.org/

The key challenge of using the continuous, parametric approach is to choose an appropriate parametric probability function, e.g., Gaussian, exponential, or Gamma distribution, that best fits the data, i.e., the cost values in $CostMultiSet$. Once the parametric probability function is chosen, learning a random variable is equivalent to learning the parameters that govern the probability function.

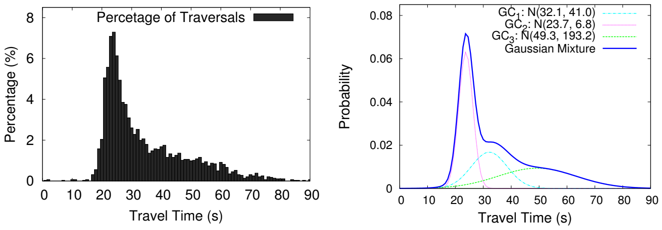
A careful analysis of the travel times and GHG emissions obtained from more than 180 million GPS records leads us to the conclusion that the distributions of both are complex and that they can generally not be modeled well using any single parametric probability function. Next, we observe that by varying the number of Gaussian components and the mixing coefficients, a Gaussian Mixture Model (GMM) is able to approximate any complex probability functions [22]. Thus, we choose to use a GMM to represent travel time or GHG emissions distributions. In particular, the parameters of a GMM are defined in Equation 4.

$$GMM(x) = \sum_{k=1}^{K} m_k \cdot \mathcal{N}(x|\mu_k, \delta_k^2) \quad (4)$$

A GMM is a weighted sum of $K$ Gaussian distributions, each of which is called a Gaussian component and is associated with a mixing coefficient $m_k$. These satisfy $\sum_{k=1}^{K} m_k = 1$. A Gaussian component is governed by a mean $\mu_k$ and a variance $\delta_k^2$.

Given $CostMultiSet$, if $K$, the number of Gaussian components, is also given, basic clustering algorithms, e.g., Expectation-Maximization based K-Means [22], can be applied directly to identify a GMM that best describes the distribution of cost values in $CostMultiSet$. However, deciding an appropriate $K$ in advance is difficult because different intervals have different, arbitrary cost values. An overly small $K$ may not fully capture all the representative travel costs on the edge, thus resulting in under-fitting; and an overly large $K$ may capture the travel costs over-specifically, yielding over-fitting, and also reduces the efficiency when convoluting two distributions. We thus apply a procedure that is able to select an appropriate $K$. The procedure starts with $K = 1$ and increments $K$ by 1 until the benefit (e.g., likelihood) of using $K$ is smaller than that of using $K - 1$. Due to the space limitation, we omit the detailed algorithm, which is covered elsewhere [27].

Consider the travel times of the edge shown in Fig. 2(b). We plot the percentage of traversals (on the y-axis) of the edge with each cost value (on the x-axis) in Fig. 7(a). After applying the GMM learning procedure, the GMM with 3 Gaussian components in Fig. 7(b) is found to best describe the travel-time distribution.



(a) Percentage of Traversals

(b) Learned GMM

Fig. 7. Fitting Travel Times Using a GMM

Next, we consider the convolution of two GMMs. The convolution of Gaussian distributions $\mathcal{N}(\mu_1, \delta_1^2)$ and $\mathcal{N}(\mu_2, \delta_2^2)$ is a Gaussian distribution: $\mathcal{N}(\mu_1 + \mu_2, \delta_1^2 + \delta_2^2)$ [28]. Based on this, it is straightforward to prove that the convolution of two GMMs is also a GMM, which is a weighted sum of the convolutions of Gaussian components from each of the two GMMs. The proof is omitted due to the space limitation. Algorithm 2 describes the procedure of convoluting two GMMs.

**Algorithm 2: ConvolutionGMMs**

```
   Input  : GMMs: $gmm_1, gmm_2$;
   Output: GMM: $gmm_1 \odot gmm_2$;
 1 GMM $gmm \leftarrow \emptyset$;
 2 int $x \leftarrow 1$;
 3 for each $i$ in $[1...gmm_1.K]$ do
 4     for each $j$ in $[1...gmm_2.K]$ do
 5         $m_x \leftarrow gmm_1.m_i \cdot gmm_2.m_j$;
 6         $\mu_x \leftarrow gmm_1.\mu_i + gmm_2.\mu_j$;
 7         $\delta_x^2 \leftarrow gmm_1.\delta_i^2 + gmm_2.\delta_j^2$;
 8         Create a new Gaussian distribution
           $m_x \cdot \mathcal{N}(\mu_x, \delta_x^2)$ and add it as the $x$-th
           component in $gmm$;
 9         $x$++;
10 if $x > thGMM$ then
11     Re-estimate a new GMM $gmm$ with fewer
       Gaussian components;
12 return $gmm$;
```

The convolution of two GMMs that have $K_i$ and $K_j$ Gaussian components typically produces a GMM with $K_i \cdot K_j$ Gaussian components. As we need to continue convoluting many GMMs when computing route costs, especially for long routes, it may produce GMMs with a huge number of Gaussian components, which significantly reduces the efficiency of the convolution operation. However, in such cases, many Gaussian components only have negligibly small mixing coefficients, meaning that they are insignificant in describing the overall distribution. Thus, when the convolution of two GMMs have more than $thGMM$ Gaussian components, we draw a set of points using the convoluted GMM, and re-estimate a new GMM with fewer Gaussian components using the points (line 11 in Algorithm 2). The number of Gaussian components of the new GMM depends on the number of Gaussian components in the original GMM whose mixing coefficients are not negligibly small (e.g., exceed 0.1).

## VI. Stochastic Skyline Route Planning

A pruning strategy and an efficient method for checking stochastic dominance are proposed to support efficient stochastic skyline route planning.

### A. Routing Algorithms

A brute force approach is to enumerate all possible routes from the source and the destination, compute their costs, and check whether one route dominates another. This approach is very inefficient and works only for very small road networks. Instead, we propose a method that is able to return stochastic skyline routes much more efficiently.

In the following, we call a route that is from the source but has not yet reached the destination a partially explored route, and we call a route that has reached the destination a complete route. The proposed method is able to estimate the best possible travel costs to the destination for partially explored routes. If a partially explored route with its estimated travel cost is dominated by a complete route, the partially explored route can be disregarded, which considerably reduce the search space. The proposed stochastic skyline route planning algorithm is described in Algorithm 3.

**Algorithm 3: StochasticSkylineRouteQueries**

```
   Input : Source: $v_s$; Destination: $v_d$; StartingTime: $t$;
   Output: StochasticSkylineRoutes: $SKR(v_s, v_d, t)$;
   /* Facilitating route cost estimation. */
1  $OneToAllSP(v_d, DI)$; $OneToAllSP(v_d, minTT)$;
   $OneToAllSP(v_d, minGE)$;
2  $SKR \leftarrow$ UpdateSKR($\mathcal{R}_{DI}$);
   $SKR \leftarrow$ UpdateSKR($\mathcal{R}_{TT}$);
   $SKR \leftarrow$ UpdateSKR($\mathcal{R}_{GE}$)
3  Define a priority key and initialize a priority queue $Q$;
4  $Q.enqueue\big(0, (\langle v_s, v_s\rangle, (0, 1.0))\big)$;
5  repeat
6      $\mathcal{R}_{next} \leftarrow Q.dequeue()$;
7      $v \leftarrow$ the last vertex of $\mathcal{R}_{next}$;
8      for each $e_k \in E$ if $e_k$'s starting vertex is $v$ and $e_k$
       is not in $\mathcal{R}_{next}$ do
9          $\mathcal{R}_{next} \leftarrow$ append $e_k$ to $\mathcal{R}_{next}$;
10         if $\mathcal{R}_{next}$ reaches destination $v_d$ then
11             UpdateSKR($\mathcal{R}_{next}$);
           /* Applying the pruning strategy. */
12         else
13             $furtherExplore \leftarrow$ true;
14             for each $\mathcal{R}^*$ in $SKR$ do
15                 if $\mathcal{R}^*$ dominates $\mathcal{R}_{next}$ with its
                   estimated route cost $\widehat{RC}(\mathcal{R}_{next}, t)$ then
16                     $furtherExplore \leftarrow$ false;
17                     break;
18             if $furtherExplore$ then
19                 $Q.enqueue\big(key, (\mathcal{R}_{next}, RC(\mathcal{R}_{next}, t))\big)$;
20 until $Q$ is empty;
21 return $SKR$;
```

**Estimating the best possible travel costs for partially explored routes:** The algorithm starts by calling three one-to-all shortest path queries (e.g., using Dijkstra's algorithm) from the destination on three graphs using distances, minimum travel times, and minimum GHG emissions as their edge weights, respectively (line 1). Thus, each vertex $v$ can be associated with the shortest distance ($v.di$), the fastest time ($v.tt$), and the lowest GHG emissions ($v.ge$) to the destination. This information is used when estimating the best possible travel costs to the destination for a partially explored route.

Let vertex $v$ be the last vertex of a partially explored route $\mathcal{R}$. $\mathcal{R}$'s estimated route cost to the destination, denoted as $\widehat{RC}(\mathcal{R}, t)$, is defined in Equation 5.

$$\widehat{RC}(\mathcal{R}, t).X = RC(\mathcal{R}, t).X \bigodot (v.x, 1.0), \tag{5}$$

where $X \in \{DI, TT, GE\}$ and random variable $(v.x, 1.0)$ corresponds to the shortest distance, the fastest time, and the lowest GHG emissions from vertex $v$ to the destination. For instance, when $X = TT$, random variable $(v.tt, 1.0)$ is used.

**Pruning Strategy:** Since we use the minimum cost values to estimate route costs, it is clear that the estimated travel cost random variable $\widehat{RC}(\mathcal{R}, t).X$ is the "best" (i.e., having the shortest distance, the fastest travel time, and the least GHG emissions) possible travel cost random variable for any complete route $\mathcal{R}'$ that has $\mathcal{R}$ as its pre-route. Equivalently, we have $\widehat{RC}(\mathcal{R}, t).X$ stochastically dominates $\widehat{RC}(\mathcal{R}', t).X$.

Thus, if a partially explored route $\mathcal{R}$ with its estimated route cost is dominated by a complete route $\mathcal{R}^*$, the partially explored route $\mathcal{R}$ can be disregarded because any complete route that extends it will be dominated by the complete route $\mathcal{R}^*$ and thus can not become a stochastic skyline route. The pruning strategy is described in lines 13–19 in Algorithm 3

The pruning strategy is facilitated by two fundamental data structures. First, a set $SKR = \{(\mathcal{R}, RC(\mathcal{R}, t))\}$ is maintained, where each element represents a candidate stochastic skyline route $\mathcal{R}$ along with its cost $RC(\mathcal{R}, t)$. A candidate stochastic skyline route must be a complete route. As we keep identifying new complete routes, set $SKR$ is updated according to Algorithm 4. Only complete routes that are not dominated by other complete routes are kept.

**Algorithm 4: UpdateSKR**

```
Input : A Complete Route: $\mathcal{R}$;
Output: A Updated $SKR$;
if $SKR$ is empty then
    $SKR \leftarrow SKR \cup (\mathcal{R}, RC(\mathcal{R}, t))$;
else
    for each route $\mathcal{R}_{skr} \in SKR$ do
        if $R_{skr}$ dominates $\mathcal{R}$ then
            return $SKR$;
    for each route $\mathcal{R}_{skr} \in SKR$ do
        if $\mathcal{R}$ dominates $R_{skr}$ then
            Remove $\mathcal{R}_{skr}$ from $SKR$;
    $SKR \leftarrow SKR \cup (\mathcal{R}, RC(\mathcal{R}, t))$;
return $SKR$;
```

Second, we maintain a queue $Q = (key, value)$ prioritized by the real valued $key$ to manage partially explored routes that may become stochastic skyline routes. In particular, $value = (\mathcal{R}, RC(\mathcal{R}, t))$ represents a partially explored route $\mathcal{R}$ and its route cost. The corresponding key is derived from the route cost $RC(\mathcal{R}, t)$, e.g., the distance, the expected travel time, or the expected GHG emissions. Since the routing algorithm continues to explore routes based on the priority queue, different instantiations of $key$ yield different strategies for exploring the search space.

Because the pruning strategy works only when there is at least one complete route, Algorithm 3 initially inserts three routes in $SKR$ (line 2). Routes $\mathcal{R}_{DI}$, $\mathcal{R}_{TT}$, and $\mathcal{R}_{GE}$ are identified while running the three one-to-all shortest path queries and are the routes with the shortest distance, the possible shortest time, and the possible lowest GHG emissions, respectively.

### B. Efficient Stochastic Dominance Checking

The most time consuming part of the stochastic skyline route planning algorithm is to check the stochastic dominance between two routes (line 15 in Algorithm 3 and lines 5 and 8 in Algorithm 4). Thus, we proceed to propose an efficient stochastic dominance checking method. We note that the proposed method is generally applicable to comparing stochastic dominance between any two continuous random variables, not only GMM random variables.

According to Definition 6, to check whether route $\mathcal{R}_1$ dominates route $\mathcal{R}_2$, we need to check the dominance relationship between the cumulative distribution functions of the routes for each travel cost random variable of interest. For a GMM random variable whose probability density function (pdf) is defined by Equation 4, its corresponding cumulative density function (cdf) is defined as follows.

$$CDF_{GMM}(x) = \sum_{k=1}^{K} \frac{m_k}{2} \cdot \left(1 + erf(\frac{x - \mu_k}{\sqrt{2 \cdot \delta_k^2}})\right),$$

where $erf$ is the error function. Given two such cdfs, a naive way to evaluate the stochastic dominance between them is to compare the cumulative probabilities of all possible costs in $\mathbb{R}^+$. This is inefficient, as it involves a huge number of comparisons. We instead propose a method that is able to avoid considerable cumulative probability comparisons.

We consider one travel cost type at a time and denote the cdf of the cost's random variable on route $\mathcal{R}$ by $F_{\mathcal{R}}$. Recall that we maintain the minimum and maximum cost values for each edge using the function vector **MM** in the MTUG. It is then possible to obtain the minimum and maximum cost values, $F_{\mathcal{R}}^{min}$ and $F_{\mathcal{R}}^{max}$, for a route $\mathcal{R}$ by simply summing up the minimum and maximum cost values of all edges. Although it is improbable for a route to be associated with these minimum or maximum cost values, to ensure the stochastic dominance checking is correct, we cannot use larger minimum-cost or smaller maximum-cost values.

Next, we distinguish three cases, and we assume that $F_{\mathcal{R}_1}^{min} \leqslant F_{\mathcal{R}_2}^{min}$.

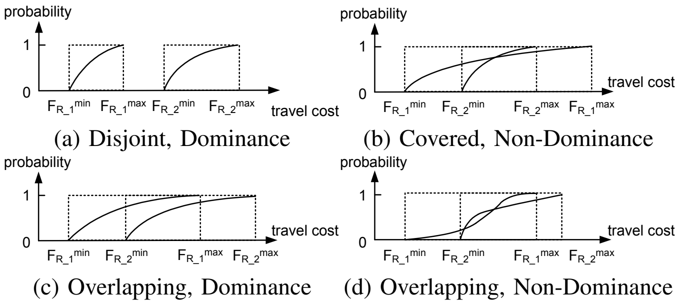
**Disjoint case:** If $F_{\mathcal{R}_1}^{max} \leqslant F_{\mathcal{R}_2}^{min}$, $F_{\mathcal{R}_1}$ stochastically dominates $F_{\mathcal{R}_2}$. As shown in Fig. 8(a), for any possible cost in $[F_{\mathcal{R}_1}^{min}, F_{\mathcal{R}_1}^{max}]$, its cumulative probability on $\mathcal{R}_1$ always exceeds that on $\mathcal{R}_2$ (which is actually 0).

**Covered case:** If $F_{\mathcal{R}_1}^{min} < F_{\mathcal{R}_2}^{min}$ and $F_{\mathcal{R}_2}^{max} < F_{\mathcal{R}_1}^{max}$, the one cdf cannot stochastically dominate the other. As shown in Fig. 8(b), when the cost value is $F_{\mathcal{R}_2}^{min}$, route $\mathcal{R}_1$ has a higher cumulative probability; however, when the cost value is $F_{\mathcal{R}_2}^{max}$, it is route $\mathcal{R}_2$ that has a higher cumulative probability.

**Overlapping case:** Two cdfs that do not satisfy the disjoint or covered conditions belong to the overlapped case. Dominance may occur (e.g., $F_{\mathcal{R}_1} \succ F_{\mathcal{R}_2}$ in Fig. 8(c)) or may not occur (e.g., no dominance in Fig. 8(d)).

To determine whether dominance occurs in the overlapping case, we partition the possible travel costs into three intervals.

**First interval** $[F_{\mathcal{R}_1}^{min}, F_{\mathcal{R}_2}^{min})$**:** A cost $c$ in the first interval has a larger cumulative probability on route $\mathcal{R}_1$ than on route $\mathcal{R}_2$ because $F_{\mathcal{R}_1}(c) > 0$ and $F_{\mathcal{R}_2}(c) = 0$.



(a) Disjoint, Dominance (b) Covered, Non-Dominance

(c) Overlapping, Dominance (d) Overlapping, Non-Dominance

Fig. 8. Three Cases for Determining Stochastic Dominance

**Second interval** $(F_{\mathcal{R}_1}^{max}, F_{\mathcal{R}_2}^{max}]$**:** The same conclusion holds because $F_{\mathcal{R}_1}(c) = 1$ and $F_{\mathcal{R}_2}(c) < 1$. Thus, $F_{\mathcal{R}_1}$ stochastically dominates $F_{\mathcal{R}_2}$ in the first two intervals.

**Third interval** $[F_{\mathcal{R}_2}^{min}, F_{\mathcal{R}_1}^{max}]$**:** For every possible cost in the interval, if its cumulative probability on $\mathcal{R}_1$ always exceeds that on $\mathcal{R}_2$ then $F_{\mathcal{R}_1} \succ F_{\mathcal{R}_2}$ (e.g., Fig. 8(c)); otherwise, no dominance between the two cdf is found (e.g., Fig. 8(d)). Instead of comparing cumulative probabilities for each possible cost, we propose an efficient algorithm that is able to check sub-intervals of possible costs, as described in Algorithm 5.

**Algorithm 5: DominaceCheck**

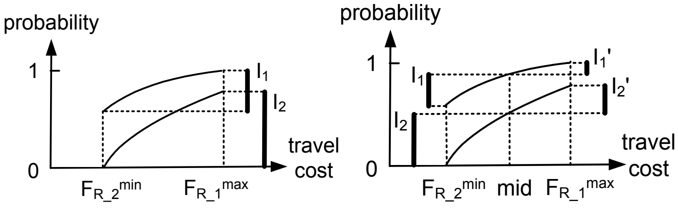
```
Input : CDFs: $F_{\mathcal{R}_1}, F_{\mathcal{R}_2}$; Double(Integer):
        $F_{\mathcal{R}_2}^{min}, F_{\mathcal{R}_1}^{max}$;
Output: DominanceRelationship $dom$;
$noDom \leftarrow$ false;
$R1DomR2 \leftarrow$ false; $R2DomR1 \leftarrow$ false;
Initialize a queue $Q$ where an element in the queue
indicates a range $[l, u]$;
$Q.enqueue([F_{\mathcal{R}_2}^{min}, F_{\mathcal{R}_1}^{max}])$;
repeat
    $lb \leftarrow Q.dequeue().l$; $ub \leftarrow Q.dequeue().u$;
    $I_1 \leftarrow [F_{\mathcal{R}_1}(lb), F_{\mathcal{R}_1}(ub)]$;
    $I_2 \leftarrow [F_{\mathcal{R}_2}(lb), F_{\mathcal{R}_2}(ub)]$;
    if $I_1$ intersects $I_2$ then
        $mid \leftarrow \frac{lb+ub}{2}$;
        if $[lb, mid] > \epsilon$ then
            $Q.enqueue([lb, mid])$;
            $Q.enqueue([mid, ub])$;
    else
        if $I_1.l > I_2.u$ then
            $R1DomR2 \leftarrow$ true;
        if $I_2.l > I_1.u$ then
            $R2DomR1 \leftarrow$ true;
    if $R1DomR2 \wedge R2DomR1$ then
        $noDom \leftarrow$ true;
        break;
until $Q$ is empty;
return $DomTest(noDom, R1DomR2, R2DomR1)$;
```

Since a cdf $F_{\mathcal{R}}$ is monotonically non-decreasing, given a cost interval $[l, u]$, its corresponding cumulative probability interval is $[F_{\mathcal{R}}(l), F_{\mathcal{R}}(u)]$. The cumulative probability intervals of the two cdfs may be disjoint. For example, this occurs when $F_{\mathcal{R}_1}(l) \geqslant F_{\mathcal{R}_2}(u)$, in which case $F_{\mathcal{R}_1}$ dominates $F_{\mathcal{R}_2}$ in the interval, because the smallest possible cumulative probability of $F_{\mathcal{R}_1}$ is greater than the largest possible cumulative probability of $F_{\mathcal{R}_2}$. Otherwise, the interval are split into sub-intervals, and the sub-intervals are checked.

Based on the above, Algorithm 5 uses a queue $Q$ to maintain all the cost intervals that need to be checked (lines 3–4). When checking, if two cumulative probability intervals overlap, the cost interval is split into two sub-intervals that are inserted into the queue $Q$ if they are longer than a threshold $\epsilon$ that controls the finest granularity of interest for cost intervals. For instance, $\epsilon$ can be set to one second or one minute for travel times (lines 9–11).

If two cumulative probability intervals are disjoint, dominance can be determined (lines 12–16). If $F_{\mathcal{R}_1} \succ F_{\mathcal{R}_2}$ for some cost sub-intervals and $F_{\mathcal{R}_2} \succ F_{\mathcal{R}_1}$ for some other cost sub-intervals, there is no dominance relation between the two cdfs (lines 17–19). Finally, the dominance relationship between $F_{\mathcal{R}_1}$ and $F_{\mathcal{R}_2}$ is returned (line 21).

Consider an example for the case shown in Fig. 8(c). For cost interval $[F_{\mathcal{R}_2}^{min}, F_{\mathcal{R}_1}^{max}]$, the corresponding cumulative probability intervals $I_1$ and $I_2$ overlap, as shown in Fig. 9(a). Then the cost interval is split into two. For each sub-interval, the corresponding cumulative probability intervals are disjoint ($I_1$ and $I_2$ for $[F_{\mathcal{R}_2}^{min}, mid]$; $I_1'$ and $I_2'$ for $[mid, F_{\mathcal{R}_1}^{max}]$), which means that $F_{\mathcal{R}_1}$ dominates $F_{\mathcal{R}_2}$. See Fig. 9(b).



(a) Iteration 1

(b) Iterations 2, 3

Fig. 9. A Running Example of Algorithm 5

## VII. Empirical Studies

We report on empirical studies of the effectiveness and efficiency of the paper's proposals.

### A. Experimental Setup

**GPS Records:** We use more than 180 million GPS records collected at 1 Hz (i.e., one GPS record per second) in Denmark during week days in 2007 and 2008. The data is from an experiment where young drivers start out with a rebate on their car insurance and then are warned if they speed and are penalized financially if they continue to speed.

**Road Networks:** We obtain the road networks of Aalborg (*AA*), North Jutland (*NJ*), and Jutland (*JU*) from the OpenStreetMap. *AA* has 4,981 vertices and 12,614 edges, *NJ* has 68,318 vertices and 163,546 edges, and *JU* has 667,876 vertices and 1,622,974 edges. Since Aalborg is the largest city in North Jutland, and North Jutland is one of the regions in Jutland, *AA* is part of *NJ*, and *NJ* is part of *JU*. The majority of the GPS records are collected from *NJ*.

**Travel Costs:** We consider three commonly used travel costs for eco-routing: travel distance (*DI*), travel time (*TT*), and GHG emissions (*GE*). The travel distances of edges are computed based on the coordinates of the corresponding vertices that are recorded in OpenStreetMap. Travel times are obtained as the difference between the times of the last and first GPS records of trajectories on an edge. We use the VT-micro model [29] to estimate the GHG emissions based on instantaneous velocities and accelerations, which are derived from the available GPS records. A recent benchmark [1] indicates that VT-micro is appropriate for this purpose.

**Queries:** We consider stochastic skyline route queries with three different combinations of travel costs: (i) *DI+TT*, (ii) *DI+GE*, and (iii) *DI+TT+GE*.

The sizes of the three road networks are different. The shortest distances between the two furthest apart vertices in networks *AA*, *NJ*, and *JU* are 8 km, 113 km, and 313 km, respectively. We generate different groups of source-destination pairs for the different road networks. In each group, the shortest distances between the source-destination pairs are chosen according to a pre-defined distance range, as shown in Table III. For instance, the shortest travel distances of source-destination pairs in the third group in *AA* are between 2 km and 3 km. We randomly choose 50 source-destination pairs for each group, and each pair is associated with a randomly chosen trip starting time.

TABLE III. Distance Ranges for Source-Destination Pairs

| | Distance Range (km) |
|---|---|
| *AA* | (0, 1], (1, 2], (2, 3], (3, 4], (4, 5], (5, 6] |
| *NJ* | (0, 2], (2, 5], (5, 10], (10, 15], (15, 20], (20, 50], (50, 100] |
| *JU* | (0, 50], (50, 100], (100, 150], (150, 200], (200, 250], (250, 300] |

**Implementation Details:** Threshold $thGMM$ used in Algorithm 2 is set to 10,000. Threshold $\epsilon$ used in Algorithm 5 is set to 30 s and 100 ml for *TT* and *GE*, respectively. The priority queue used in Algorithm 3 is prioritized by distance.

All algorithms are implemented in Java using JDK 1.7. To ease the management of Gaussian mixture models, the jEMF package[3] is applied. A computer with Windows 7 Enterprise, a 3.40GHz Intel Core i7-2600 CPU, and 16 GB main memory is used for all experiments.

### B. MTUG Generation

We distinguish between "hot" and "cold" edges when generating an MTUG. Hot edges are covered by GPS records, and we run Algorithm 1 to obtain the time-dependent, uncertain weights for these edges. Specifically, in Algorithm 1, we set $\alpha = 15$ minutes thus yielding 96 intervals per day and we apply KL divergence to measure the similarity between two distributions. When the KL divergence between two distributions of travel costs in adjacent intervals is below 0.1 (i.e., setting the threshold *thDis* used in Section V-A to 0.1), the two intervals are combined.

For all three road networks, the largest numbers of *TT* and *GE* intervals for an edge are 65 and 72, respectively, while the smallest numbers are 1 for both *TT* and *GE*. Fig. 10 shows the average number of *TT* and *GE* intervals on the three road networks, respectively. The figure indicates that the travel times and the GHG emissions of the edges in *AA* and *NJ* have more substantial temporal variations than for *JU*. The reason is two-fold. First, *AA* is the largest city in *NJ*, and the traffic in *AA* is much more dynamic than the traffic in other parts of *NJ*. Second, the majority of the GPS records are collected from *NJ*,

[3] http://www.lix.polytechnique.fr/~nielsen/MEF/

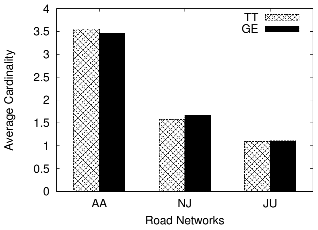


Fig. 10. # of *TT* and *GE* Intervals

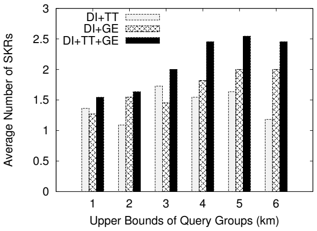


Fig. 11. # SKR, AA

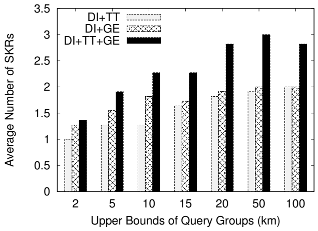


Fig. 12. # SKR, NJ

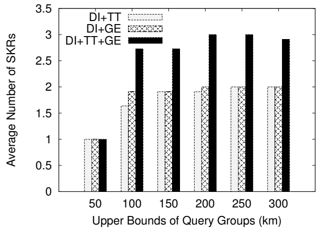


Fig. 13. # SKR, JU

and the traffic in other parts of *JU* is captured more sparsely by the GPS records.

Cold edges are not covered by any GPS records. Such edges are less likely to have much traffic and traffic variation, or the vehicles in the GPS data set do not cover the edges. For a cold edge, a travel time value is derived by dividing the length of the edge by the speed limit of the edge; and a GHG emission value is estimated using the SIDRA-Running model [1] based on the length and speed limit of the edge. Next, as the edge's *TT* and *GE* weights, we generate a Gaussian random variable with the derived cost value as the mean and the square of one fifth of the mean as the variance. Advanced methods for dealing with cold edges are beyond the scope of the paper, but can be found elsewhere [30].

The MTUG generation for the largest network JU took almost 5 hours (specifically, 17,977 seconds). MTUG generation occurs off-line and is not time-critical.

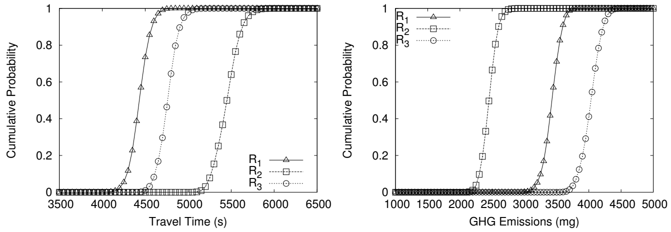
### *C. Stochastic Skyline Route Queries*

**Number of stochastic skyline routes:** We first study the average number of the stochastic skyline routes ($\#$ *SKR*) returned by different queries. The results for *AA*, *NJ*, and *JU* are reported in Fig. 11, Fig. 12, and Fig. 13, respectively. As the number of travel costs considered increases, the number of the stochastic skyline routes also increases. The number of stochastic skyline routes returned by the queries with *DI+TT+GE* generally exceed those returned by the queries with *DI+TT* and *DI+GE*.

Specifically, the number of stochastic skyline routes with *DI+TT* are smaller than those with *DI+GE*. This is because the travel times are quite correlated with the travel distances in our settings. In many cases, the shortest routes are also the fastest. In contrast, the correlation between distances and GHG emissions are weaker, thus yielding larger numbers of stochastic skyline routes for the queries with *DI+GE*. The queries with *DI+TT+GE* have the largest number of stochastic skyline routes because neither the shortest path nor the fastest path has the least GHG emissions in many cases.

To further illustrate stochastic skyline routes, we show a concrete example of a query with *DI+TT+GE* in the *NJ* network. Three routes, $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$, are identified as stochastic skyline routes for the query, where the travel distances are 94,849, 106,216, and 91,382 meters, respectively. Obviously, $\mathcal{R}_3$ is the shortest route. According to the cdfs of *TT* and *GE*, as shown in Fig. 14 (a) and Fig. 14 (b), respectively, $\mathcal{R}_1$ dominates $\mathcal{R}_3$ and $\mathcal{R}_2$ in terms of *TT*, and $\mathcal{R}_2$ dominates $\mathcal{R}_1$ and $\mathcal{R}_3$ in terms of *GE*. None of the routes is able to dominate the others in all three travel costs, so are all stochastic skyline routes for the query.

Fig. 14 also suggests that the correlation between *DI* and *TT* is more obvious than that between *DI* and *GE*. Specifically, since $\mathcal{R}_1$ and $\mathcal{R}_3$ have similar distances, which are much shorter than that of $\mathcal{R}_2$, both have similar travel times, which are also much faster than that of $\mathcal{R}_2$. In contrast, the longest route $\mathcal{R}_3$ takes the longest travel time but generates the lowest *GE*. This is consistent with the finding that the numbers of stochastic skyline routes returned by queries with *DI+TT* are generally smaller than those returned by queries with *DI+GE*, as shown in Figs.11, 12, and 13.



(a) cdf of *TT*

(b) cdf of *GE*

Fig. 14. Results on *AA*

**Runtime:** The average run time per stochastic skyline route query is reported in Figs. 15, 16, and 17. As the number of travel costs considered in stochastic skyline route queries increases, the runtime also increases. Next, the longer the distance between source-destination pairs, the longer the runtime. The longest runtime per query (i.e., *DI+TT+GE* queries for source-destination pairs that are located more than 250 km in the biggest network *JU*) is below 5 seconds, and most queries can be returned in less than 2 seconds, which is acceptable for real-time use. The experiments on runtime do not include the existing methods [18], [19] because (1) one method [18] took more than 2 minutes for a query in the smallest AA network, which is unacceptable for on-line use, and (2) the other method [19] does not work in our setting because it relies on Gaussian distributions.

Next, we show the effect of the proposed pruning strategy. We compare the proposed method with the brute force method described in Section VI-A. The brute force method cannot return stochastic skyline routes in less than 30 minutes, even for the queries in the shortest group $(0, 2]$ on the smallest network (*AA*), regardless of whether naive or advanced dominance checking is applied. This is because the naive method has to compute route costs for all possible routes connecting the source and the destination. This suggests that the pruning strategy is quite effective.

Finally, we show the effect of the proposed advanced dominance checking (Algorithm 5) by comparing it with the naive method when using the routing algorithm with the pruning strategy. Fig. 18 reports the average runtime for all

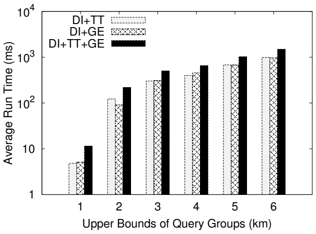


Fig. 15. Runtime, AA

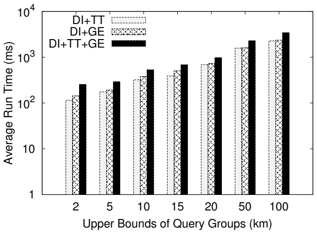


Fig. 16. Runtime, NJ

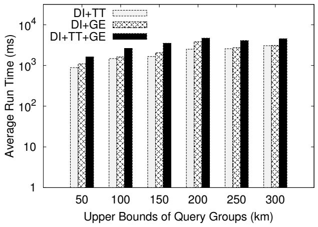


Fig. 17. Runtime, JU

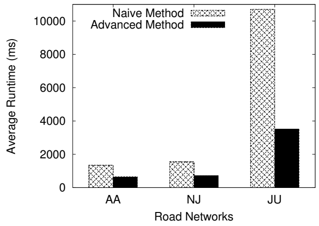


Fig. 18. Runtime, Naive

queries on the three networks. The advanced method always outperforms the naive one. On large networks, the benefits of using the advanced method become more substantial, since long routes require more dominance check—see the last two bars of *JU* in Fig. 18.

## VIII. Conclusions and Outlook

We propose and study stochastic skyline route queries in road networks with multiple travel costs that are time dependent and uncertain. We provide techniques for generating a multi-cost, time-dependent, uncertain graph based on a collection of GPS records. We also propose an effective pruning strategy and an efficient stochastic dominance checking method to support the efficient computation of stochastic skyline route queries. Empirical studies with three real road networks and a large collection of GPS records suggest that the proposed methods are effective and efficient.

In future work, it is of interest to incorporate drivers' preference profiles to further reduce the travel cost uncertainties for different types of drivers, and it is of interest to rank the skyline routes for individual drivers.

## Acknowledgments

This work was supported by the Reduction project that is funded by the European Commission as FP7-ICT-2011-7 STREP project number 288254. Shuo Shang is supported by Science Foundation of China University of Petroleum, Beijing (NO. 2462013YJRC031).

## References

[1] C. Guo, Y. Ma, B. Yang, C. S. Jensen, and M. Kaul, "EcoMark: Evaluating models of vehicular environmental impact," in *GIS*, pp. 269–278, 2012.

[2] O. Andersen, K. Torp, C. S. Jensen, and B. Yang, "EcoTour: Reducing the environmental footprint of vehicles using eco-routes," in *MDM*, 338–340, 2013.

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

[4] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

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

[6] E. Nikolova, M. Brand, and D. R. Karger, "Optimal route planning under uncertainty," in *ICAPS*, 2006, pp. 131–141.

[7] A. Chen and Z. Ji, "Path finding under uncertainty," *Journal of Advanced Transportation*, vol. 39, no. 1, pp. 19–37, 2005.

[8] H.-P. Kriegel, M. Renz, and M. Schubert, "Route skyline queries: A multi-preference path planning approach," in *ICDE*, 2010, pp. 261–272.

[9] A. B. Wijeratne, M. A. Turnquist, and P. B. Mirchandani, "Multiobjective routing of hazardous materials in stochastic networks," *European Journal of Operational Research*, vol. 65, no. 1, pp. 33–43, 1993.

[10] A. Orda and R. Rom, "Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length," *JACM*, vol. 37, no. 3, pp. 607–625, 1990.

[11] D. E. Kaufman and R. L. Smith, "Fastest paths in time-dependent networks for intelligent vehicle-highway systems application?" *Journal of ITS*, vol. 1, no. 1, pp. 1–11, 1993.

[12] B. Ding, J. X. Yu, and L. Qin, "Finding time-dependent shortest paths over large graphs," in *EDBT*, 2008, pp. 205–216.

[13] E. Kanoulas, Y. Du, T. Xia, and D. Zhang, "Finding fastest paths on a road network with speed patterns," in *ICDE*, 2006, p. 10.

[14] M. P. Wellman, M. Ford, and K. Larson, "Path planning under time-dependent uncertainty," in *UAI*, 1995, pp. 532–539.

[15] S. Lim, C. Sommer, E. Nikolova, and D. Rus, "Practical route planning under delay uncertainty: Stochastic shortest path queries," in *Robotics: Science and Systems*, 2012.

[16] 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.

[17] H. W. Hamacher, S. Ruzika, and S. A. Tjandra, "Algorithms for time-dependent bicriteria shortest path problems," *Discrete Optimization*, vol. 3, no. 3, pp. 238–254, 2006.

[18] E. Miller-Hooks and H. S. Mahmassani, "Optimal routing of hazardous materials in stochastic, time-varying transportation networks," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1645, no. 1, pp. 143–151, 1998.

[19] T.-S. Chang, L. K. Nozick, and M. A. Turnquist, "Multiobjective path finding in stochastic dynamic networks, with application to routing hazardous materials shipments," *Transportation Science*, vol. 39, no. 3, pp. 383–399, 2005.

[20] X. Lin, Y. Zhang, W. Zhang, and M. A. Cheema, "Stochastic skyline operator," in *ICDE*, 2011, pp. 721–732.

[21] F. Pereira, H. Costa, and N. Pereira, "An off-line map-matching algorithm for incomplete map databases," *European Transport Research Review*, vol. 1, no. 3, pp. 107–124, 2009.

[22] C. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[23] C. M. Grinstead and J. L. Snell, *Introduction to probability*. American Mathematical Soc., 1998.

[24] M. Shaked and J. G. Shanthikumar, *Stochastic orders and their applications*. Academic Press, 1994.

[25] W. Zheng, D. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," *Journal of Transportation Engineering*, vol. 132, no. 2, pp. 114–121, 2006.

[26] F. J. Massey Jr, "The Kolmogorov-Smirnov test for goodness of fit," *Journal of the American Statistical Association*, vol. 46, no. 253, pp. 68–78, 1951.

[27] B. Yang, C. Guo, and C. S. Jensen, "Travel cost inference from sparse, spatio-temporally correlated time series using Markov models," *PVLDB*, vol. 6, no. 9, 769-780, 2013.

[28] P. Bromiley, "Products and convolutions of Gaussian distributions," *Medical School, Univ. Manchester, Manchester, UK, Tech. Rep*, vol. 3, p. 2003, 2003.

[29] K. Ahn, H. Rakha, A. Trani, and M. Van Aerde, "Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels," *Journal of Transportation Engineering*, vol. 128, no. 2, pp. 182–190, 2002.

[30] B. Yang, M. Kaul, and C. S. Jensen, "Using incomplete information for complete weight annotation of road networks," *TKDE*, to appear.